КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Управління проєктами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Дослідження технологій управління проєктом розробки веб-платформи для роботи страхових посередників»

Студентки 2-го курсу групи УПз-21

Олександри ПРИЩЕПИ
(ім'я, прізвище)

(підпис студента)

Науковий керівник:

к.т.н.
(науковий ступінь, вчене звання)

Любов КУБЯВКА
(ім'я, прізвище)

(дата) *(підпис)*

Попередній захист:

(Висновок: "До захисту в Екзаменаційній комісії")

Завідувач кафедри технологій управління

Віктор МОРОЗОВ

(підпис) *(ім'я, прізвище)* *(дата)*

Київ – 2025

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій

Кафедра технологій управління
Освітній рівень Магістр
Спеціальність 122 Комп'ютерні науки
Освітньо-наукова програма Управління проєктами

ЗАТВЕРДЖУЮ
Завідувач кафедри
професор Віктор МОРОЗОВ

_____
«29» вересня 2025 року

З А В Д А Н Н Я

НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Студентка: Олександра ПРИЩЕПА

Група: УПз-21

1. Тема кваліфікаційної роботи

«Дослідження технологій управління проєктом розробки веб-платформи для роботи страхових посередників»

Затверджена Протоколом № 15 від «16» червня 2025 р.

2. Строк подання студентом готової роботи – «10» грудня 2025 р.

3. Цільова установка та вихідні дані до роботи: дослідження методів і інструментів управління проєктами під час розробки веб-платформи для страхових посередників Insurance Broker Assistant System IBAS; обґрунтування вибору підходу до управління проєктом на основі гібридної моделі з використанням практик Scrum на етапі розробки; формування паспорта проєкту, визначення зацікавлених сторін, цілей; отримані результати аналізу сучасного стану українського страхового ринку та нормативно-правових вимог до діяльності страхових посередників; виявлені проблеми та потреби брокерів і агентів, що потребують автоматизації; сформовані функціональні та нефункціональні вимоги до системи з акцентом на безпеку, надійність і продуктивність; визначені принципи архітектури та технологічного стеку,

зокрема мікросервісний підхід, веб-інтерфейс, база даних PostgreSQL та хмарна інфраструктура; розроблені WBS та OBS і базові підходи до планування термінів і ресурсів; ідентифіковані ризики проєкту, виконано їх оцінку, визначено тригери та заходи реагування; розрахований орієнтовний бюджет проєкту.

4. Зміст роботи: обґрунтування актуальності створення веб-платформи IBAS для автоматизації роботи страхових посередників; аналіз предметної області, ринку та регуляторних вимог, визначення ключових стейкхолдерів; виконання PEST- та SWOT-аналізу, побудова дерева проблем і дерева цілей; формування технічного завдання у вигляді паспорта проєкту та опис концепції майбутньої системи. Визначення функціональних і нефункціональних вимог, опис основних сценаріїв використання та підготовка діаграм/моделей, що відображають поведінку системи. Обґрунтування технології управління проєктом, формування WBS, OBS, структури команди, підготовка переліку робіт і принципів ітераційної реалізації через беклог і спринти для адаптивної частини. Проведення управління ризиками: ідентифікація, якісна/кількісна оцінка, визначення пріоритетів, розробка протиризикових заходів і підхід до моніторингу. Розрахунок ресурсного забезпечення та бюджету. Опис архітектури рішення, визначення технологічного стеку та інфраструктури розгортання; проєктування концептуальної, логічної та фізичної моделей бази даних.

5. Перелік графічного матеріалу (слайдів): PEST; SWOT; конкуренти; стейкхолдери; дерево проблем і цілей; паспорт; життєвий цикл; концептуальна модель; WBS; OBS; вимоги; беклог/User Stories; ризики; бюджет; архітектура; бази даних.

6. Календарний план виконання роботи:

№ з/п	Назва частин роботи	Виконання роботи
1	Вивчення літературних джерел з предмету дослідження	01.10.25-10.10.25
2	Збір і вивчення матеріалів	13.10.25-17.10.25

3	Складання розгорнутого плану кваліфікаційної роботи	20.10.25-22.10.25
4	Ознайомлення наукового керівника з планом кваліфікаційної роботи	23.10.25-24.10.25
5	Підготовка розділу 1	27.10.25-31.10.25
6	Підготовка розділу 2	03.11.25-07.11.25
7	Підготовка розділу 3	10.11.25-14.11.25
8	Підготовка розділу 4	17.11.25-21.11.25
9	Оформлення кваліфікаційної роботи	24.11.25-26.11.25
10	Передача кваліфікаційної роботи науковому керівникові	27.11.25-28.11.25
11	Передача кваліфікаційної роботи рецензенту для рецензування	10.12.25-14.12.25
12	Захист кваліфікаційної роботи	22.12.25-22.12.25

Дата видачі завдання «29» вересня 2025 р.

Кандидат технічних наук, Любов КУБЯВКА

_____
(підпис)

Завдання прийняла на виконання

студентка групи УПз-21 Олександра ПРИЩЕПА

_____
(підпис)

ЗМІСТ

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему

«Дослідження технологій управління проєктом розробки веб-платформи для роботи страхових посередників»

Студентка: Прищепа Олександра Віталіївна

Науковий керівник: Кубявка Любов Богданівна

Рік захисту: 2025.

Метою кваліфікаційної роботи магіста є розробка проєкту створення інформаційної системи, яка забезпечить спрощення процесу укладання договорів страхування страховими посередниками «IBAS (Insurance Broker Assistant System)».

Ціль проєкту є створення та впровадження веб-платформи для страхових посередників, що підтримує ведення клієнтської бази, шаблонне формування договорів, розрахунок страхової премії та комісій, контроль строків дії договорів, автоматизацію пролонгацій і нагадувань, формування звітності, а також забезпечує надійність, продуктивність і захист даних.

Практична цінність полягає у розробці інформаційної системи «IBAS: Insurance Broker Assistant System», яка забезпечує централізоване управління процесами укладання договорів страхування через страхових посередників. Запропоноване рішення дозволяє брокерам та агентам автоматизувати підготовку документів, а також підвищити ефективність обробки запитів клієнтів.

Кваліфікаційна робота складається з анотації, вступу, основної частини, яка включає чотири розділи, висновків, переліку використаних інформаційних джерел та додатків.

У першому розділі проведено дослідження предметної області та проблематики процесу укладання договорів страхування страховими посередниками. Виконано аналіз страхового ринку та наявних програмних рішень, здійснено PEST-аналіз зовнішнього середовища і аналіз потенційних

конкурентів. Проведено SWOT-аналіз проєкту, визначено зацікавлені сторони, побудовано дерево проблем і дерево цілей. Сформульовано технічне завдання на розробку у вигляді паспорта проєкту та обґрунтовано вибір життєвого циклу проєкту.

У другому розділі розроблено концептуальну модель інформаційної системи IBAS та виконано постановку задачі в математичному вигляді. Визначено основні множини та позначення, формалізовано підходи до розрахунку страхової премії, розподілу комісій між агентами, а також наведено приклад розрахунку премії для конкретного страхового продукту.

У третьому розділі обґрунтовано вибір технології управління проєктом та доведено доцільність застосування гібридного підходу Water Scrum Fall для реалізації IBAS. Розроблено структуру декомпозиції робіт, сформовано організаційну структуру управління проєктом та розподіл ролей у межах Scrum. Визначено функціональні й нефункціональні вимоги до продукту, сформовано беклог, виконано планування першого спринту. Проведено планування ресурсного забезпечення та бюджету, здійснено ідентифікацію й оцінювання ризиків, визначено їх тригери та розроблено протиризикові заходи.

У четвертому розділі описано технічні рішення реалізації інформаційної системи як кінцевого продукту проєкту. Подано архітектурний підхід із використанням мікросервісів, охарактеризовано клієнтську частину у вигляді Angular та серверну частину, а також інструменти CI/CD і DevOps-процеси, підходи до моніторингу, логування й забезпечення безпеки. Розроблено моделі бази даних, зокрема концептуальну, логічну та фізичну, описано алгоритм роботи додатку та представлено інтерфейс веб-додатку.

Кваліфікаційна робота складається з 101 сторінки основного тексту, містить 22 рисунки, 20 таблиць, 8 формул та 8 додатків.

Ключові слова: страхові посередники, інформаційна система, веб-платформа, управління проєктом, Water-Scrum-Fall, WBS, ризики, бюджет, безпека, страхова премія, комісія.

ВСТУП

Актуальність теми дослідження. На сучасному етапі розвитку українського страхового ринку спостерігається чітка тенденція до його консолідації та очищення. Це проявляється, зокрема, у суттєвому скороченні кількості учасників ринку протягом останніх років. Станом на 30 травня 2023 року на ринку діяло 115 страхових компаній, на 30 червня 2024 року їх кількість зменшилася до 78, у травні 2025 року в реєстрі налічувалося 51 страховика, а вже станом на листопад 2025 року – лише 49 компаній, що здійснюють non-life страхування (ризикові види) [1]. Таким чином, у 2023–2025 роках кількість страхових компаній в Україні скоротилася на 66 учасників ринку. Зменшення кількості гравців на ринку зумовлено жорсткішою політикою Національного банку України, який 3 1 липня 2020 року виконує функцію регулятора небанківського фінансового сектору [2]. Зі зміною регулятора змінилися і підходи, тож НБУ запровадив нову модель регулювання ринку страхування, яка враховує положення директив Європейського Союзу, міжнародні принципи IAIS, світові практики [3; 4]. Водночас на фоні скорочення кількості страховиків спостерігається активний розвиток сегмента страхового посередництва. За даними відкритого реєстру страхових посередників НБУ станом на листопад 2025 року в ньому зареєстровано близько 23 тисячі страхових посередників, тоді як станом на червень 2025 року в Україні було зареєстровано понад 19 тисяч страхових посередників [5]. Це свідчить про суттєву концентрацію професіоналів ринку саме в посередницькій ланці, а отже зростає потреба в цифровізації та автоматизації процесів у сфері страхових послуг, зокрема – укладання договорів через страхових посередників. Саме тому розробка інформаційної системи , яка допоможе страховим посередникам працювати з документами та клієнтами є актуальною та своєчасною.

Метою кваліфікаційної роботи магіста є розробка проєкту створення інформаційної системи, яка забезпечить спрощення процесу укладання

договорів страхування страховими посередниками «IBAS (Insurance Broker Assistant System)».

Для досягнення цієї мети поставлено такі ***задачі***:

- дослідити сучасний стан українського страхового ринку, проаналізувати нормативно-правові вимоги до діяльності страхових посередників та визначити ключові проблеми, які потребують автоматизації;
- здійснити PEST-аналіз та SWOT-аналіз проєкту, визначити внутрішніх та зовнішніх зацікавлених сторін;
- сформувати технічне завдання у вигляді паспорта проєкту, визначити функціональні та нефункціональні вимоги, а також побудувати концептуальну модель майбутньої системи;
- розробити обґрунтування вибору технології управління проєктом, визначити структуру команди, провести декомпозицію робіт;
- провести ідентифікацію, якісну та кількісну оцінку ризиків, визначити їхні тригери, можливі наслідки та розробити карту управління ризиками;
- побудувати алгоритм роботи основних бізнес-процесів, розробити концептуальну та даталогічну моделі бази даних, визначити особливості використання технологічного стеку та спроектувати архітектуру вебплатформи;

Об'єктом дослідження є процеси розробки вебплатформи для спрощення укладання договорів страхування страховими посередниками «IBAS (Insurance Broker Assistant System)».

Предметом дослідження є методи та процеси управління проєктом, щодо розробки вебплатформи для спрощення укладання договорів страхування страховими посередниками «IBAS (Insurance Broker Assistant System)».

Методи дослідження. У роботі використовуються такі методи:

- аналіз нормативно-правової бази;
- методи системного аналізу для побудови концептуальних моделей;
- методи декомпозиції;
- моделювання бізнес-процесів;

- побудова математичних моделей і алгоритмів;
- методи ризик-менеджменту.

Практичне значення отриманих результатів полягатиме у розробці інформаційної системи «IBAS: Insurance Broker Assistant System», яка забезпечує централізоване управління процесами укладання договорів страхування через страхових посередників. Запропоноване рішення дозволяє брокерам та агентам автоматизувати підготовку документів, а також підвищити ефективність обробки запитів клієнтів.

Система включає модуль авторизації, базу даних договорів, інтерфейс користувача та інтелектуальні алгоритми, які допомагають швидко формувати типові договори, перевіряти надійність клієнта та рекомендувати страхові продукти. Впровадження системи сприятиме підвищенню точності й швидкості обслуговування, мінімізації людських помилок, забезпеченню відповідності нормативним вимогам і цифровій трансформації ринку страхових послуг України.

Апробація результатів роботи

Kubiavka L., Pryshchepa O. Insurance broker assistant system (IBAS): From idea to implementation. Information Technology and Implementation (Satellite): Conference Proceedings, November 21, 2025, Kyiv, Ukraine / Taras Shevchenko National University of Kyiv.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ

Перший розділ роботи присвячений дослідженню передумов створення інформаційної системи для страхових посередників та обґрунтуванню доцільності й життєздатності проєкту Insurance Broker Assistant System (IBAS). У межах цього розділу послідовно аналізуються проблеми чинного процесу укладання договорів страхування, стан і тенденції розвитку страхового ринку України, а також наявні інформаційні рішення, що уже використовуються учасниками ринку. Такий підхід дає змогу перейти від загального опису ситуації до чіткого формулювання, яку саме потребу має задовольнити запропонована система, та які обмеження необхідно врахувати під час її розробки. Визначаються ключові зацікавлені сторони проєкту. На основі дерева проблем і дерева цілей формулюється технічне завдання у вигляді паспорта проєкту та обґрунтовується вибір життєвого циклу його реалізації. Таким чином, розділ задає концептуальну й аналітичну основу для подальшого проєктування та розробки інформаційної системи IBAS.

1.1 Дослідження проблематики процесу укладання договорів страховими посередниками

На сьогоднішній день процес укладання страхових договорів страховими посередниками супроводжується значною кількістю проблем, які суттєво знижують ефективність роботи. Однією з головних проблем є необхідність багаторазового дублювання інформації: після занесення даних про договір та страхувальника в базу даних агенти змушені повторно вносити ці ж дані вручну до шаблонів у текстових редакторах, таких як MS Word. Це додатково ускладняється тим, що деякі компанії не надають агентам доступу до централізованих баз даних, що робить неможливим відслідковувати оплати та пролонгацію договорів.

Відсутність єдиної інформаційної системи також спричиняє значні труднощі у веденні та контролі за актуальністю інформації. Страхові посередники часто співпрацюють із декількома страховими компаніями одночасно, кожна з яких надсилає свої окремі звіти та акти з інформацією про укладені договори. Ці звіти часто містять численні помилки, неточності щодо стану оплати та інші недоліки, що змушує агентів неодноразово перевіряти та переписувати дані. Через це значна частина робочого часу витрачається на рутинні завдання замість розвитку клієнтської бази та залучення нових клієнтів.

Додатковою суттєвою проблемою є постійні зміни в законодавстві, особливо після того, як у 2024 році вступила в дію оновлена редакція Закону України «Про страхування» [6]. Внаслідок регулярних змін законодавчих вимог страхові компанії змушені постійно змінювати шаблони договорів, що призводить до повторних завантажень і ручних налаштувань шаблонів. Часті зміни стосуються як структури документів (наприклад, коректність оформлення території дії, оферт, QR-кодів), так і безпосередньо даних страхувальника, що створює додаткові складнощі у процесі оформлення договорів.

Окремо слід зазначити проблему створення супровідних документів, які обов'язково формуються разом із договором. Це такі документи, як інформація про страховий продукт, інформація про страхового посередника, різні згоди, заяви та додаткові додатки. Всі ці документи також створюються вручну, і в кожному з них необхідно індивідуально вносити персональні дані страхувальника, включаючи номер договору. Навіть для укладання одного незначного договору, наприклад, страхування від нещасних випадків на суму 100 гривень на короткий період, агент змушений витрачати близько 30 хвилин через необхідність створення та оформлення великої кількості супутніх файлів.

Таким чином, наявність великої кількості ручної праці, відсутність єдиної інформаційної платформи та необхідність оперативного реагування на постійні законодавчі зміни роблять очевидною потребу в розробці спеціалізованої інформаційної системи, яка автоматизує процес укладання та ведення страхових

договорів, мінімізує кількість помилок, скоротить витрати часу та підвищить ефективність роботи страхових посередників.

1.2 Дослідження страхового ринку та аналіз існуючих рішень

Перед тим як проєктувати рішення для автоматизації роботи страхового посередника, важливо ретельно оцінити зовнішнє середовище та чітко зрозуміти, які рішення на ринку уже розроблені. Це дозволяє не лише коректно визначити вимоги до майбутньої системи, а й показати, яку саме нішу вона має зайняти в екосистемі страхового ринку.

1.2.1 PEST – аналіз зовнішнього середовища

Для оцінювання впливу зовнішніх факторів на розвиток страхового ринку України та реалізацію проєкту веб-платформи для страхових посередників IBAS у роботі проведено PEST-аналіз [7]. Узагальнені результати дослідження подано в табл. 1.1, яка систематизує ключові тенденції та дає змогу оцінити їх позитивний чи негативний вплив на подальший розвиток ринку.

Як свідчать дані табл. 1.1, політичне середовище нині загалом сприятливе: впроваджено нове страхове законодавство, посилено нагляд з боку НБУ, відбувається поступова імплементація європейських директив Solvency та Insurance Distribution, що підвищує прозорість і стійкість ринку. Впровадження обов’язкового страхування, наприклад цьогоріч стартувала вимога про обов’язкове страхування страхових посередників, які мають спеціальний рахунок, тобто задіяні в обігу грошових коштів. Це зумовило здорову конкуренцію на ринку, збільшення страхових премій, а також захищає клієнтів від можливих помилок страхових посередників.

Проте вагомим дестабілізуючим чинником залишаються воєнні ризики, особливо для роботи в прифронтових регіонах, а страхування від воєнних ризиків є дороговартісним для споживача або взагалі недоступним, наприклад якщо квартира знаходиться біля воєнного об’єкту. Страхові компанії максимально уникають високих ризиків та відмовляють у страхуванні.

Економічні фактори характеризуються поєднанням високих темпів зростання премій, активів і прибутковості страховиків та ще недостатньою глибиною проникнення страхування в економіку, що свідчить про значний резерв подальшого зростання, але й про чутливість ринку до макроекономічної ситуації.

Таблиця 1.1

PEST – аналіз страхового ринку

Фактор	Опис	Вплив
1	**2**	**3**
Політичні	Після переходу страховиків під нагляд НБУ запроваджено нові закони про фінансові послуги та страхування (з 2024 р.), новий закон про «автоцивілку» та розділ закону про страхових посередників, створюється реєстр посередників. Ринок стає більш прозорим і керованим [6].	Позитивний
Політичні	Курс на імплементацію директив ЄС (Solvency I та Solvency II, Insurance Distribution Directive, Motor Insurance Directive) формує довгострокове зближення з європейськими стандартами нагляду та захисту споживачів [8; 9; 10].	Стратегічно позитивний
Політичні	Посилення фінансового моніторингу та очищення ринку від схемних компаній: внутрішнє фіктивне перестрахування скоротилося більш ніж у 15 разів, ринок перестав бути частиною тіньового сектору [11].	Позитивний
Політичні	Впровадження обо’язкового страхування [6].	Позитивний
Економічні	Страховий ринок демонструє високе зростання: премії ризикових страховиків +39% р/р, виплати +27%, активи +31%, чистий прибуток ризикових страховиків за 9 місяців зріс удвічі до історичного максимуму [1].	Позитивний
Економічні	Попри скорочення кількості компаній до 59, загальні активи ринку за 5 років зросли на 42%, а понад 90% активів страховиків – високоліквідні інструменти, що підвищує стійкість ринку [4].	Позитивний
Економічні	Частка страхування у ВВП становить 0,84% проти приблизно 2% у Польщі, що свідчить про значний нереалізований потенціал зростання українського страхового ринку [4; 11].	Потенційно позитивний
Економічні	Висока вартість страхування воєнних ризиків (2–4% страхової суми) обмежує попит і потребує державної компенсації частини премії, запуск якої заплановано з січня 2026 року [11].	Змішаний (для клієнтів –негативний, для ринку – позитивно)
Соціальні	Завдяки реформі та новій моделі нагляду ринок став відповідальнішим щодо споживачів: частка порушень законодавства за результатами звернень клієнтів до НБУ скоротилася більш ніж удвічі [11].	Позитивний

Закінчення табл. 1.1

1	2	3
Соціальні	Об'єднання страхових асоціацій у Федерацію страхових об'єднань України та відкриття її офісу в Брюсселі посилює саморегуляцію ринку, його суб'єктність і діалог із регулятором та європейськими інституціями [12].	Позитивний
Технологічні	Перехід до наступного етапу розвитку (корпоративне управління, системи ризик-менеджменту, прозора щомісячна звітність, посилений фінмоніторинг) створює сталий попит на сучасні IT-рішення, аналітичні платформи та автоматизовані системи управління [6].	Позитивний
Технологічні	Збільшення кількості страхових посередників стимулює до автоматизації процесів не тільки всередині страхових компаній, а також і в посередників, що створює можливості для систем на кшталт IBAS.	Позитивний

Соціальні чинники відображають формування нової культури страхування, підвищення відповідальності компаній перед споживачами та консолідацію професійної спільноти, що посилює діалог ринку із регулятором.

Технологічне середовище характеризується переходом страховиків до більш зрілих моделей корпоративного управління, розвитку систем ризик-менеджменту та зростанням попиту на сучасні IT-рішення й автоматизацію процесів, що створює додаткові можливості для впровадження спеціалізованих інформаційних систем на кшталт IBAS.

Отже, у сукупності результати PEST-аналізу дозволяють зробити висновок, що зовнішнє середовище є переважно сприятливим для цифрової трансформації страхової діяльності, хоча потребує врахування воєнних та цінових ризиків при плануванні проєкту.

1.2.2 Аналіз аналогів інформаційних систем для страхових посередників

У даному підрозділі представлено фрагмент аналізу потенційних конкурентів (див. табл. 1.2), в якій наведено ключові ринкові рішення. Повний аналіз потенційних конкурентів представлено в табл. А.1 Додатку А. Оцінювання здійснювалося за такими параметрами, як основні послуги,

особливості, цільова аудиторія, доступні платформи, а також сильні та слабкі сторони компаній. Першим розглянуто рішення компанії ProffITsoft, яка спеціалізується на розробці комплексних програмних продуктів для страхового та фінансового сектору [13]. Другим конкурентом є платформа EWA, яка виступає у форматі маркетплейсу страхових продуктів [14].

Таблиця 1.2

Фрагмент аналізу потенційних конкурентів

Параметр	Опис компанії
1	**2**
Назва	ProfITsoft [13]
Основні послуги	ProfITsoft пропонує повний цикл розробки програмного забезпечення й тестування: від підготовки й організації процесу розробки до підтримки успішно реалізованих проєктів і оптимізації продуктивності існуючих рішень.
Особливості	ProfITsoft має понад 20 років досвіду у розробці високотехнологічних рішень, зокрема глибоку експертизу в Java та веб-технологіях. Компанія позиціонує себе як провідний розробник рішень для страхової галузі, включно з власною платформою “Complex System of Automation of the Insurance Company” (Integrated Insurance Company Automation System). Наголошується на високому рівні компетенцій, корпоративній культурі, довгострокових відносинах із клієнтами та бізнес-гнучкості. Використовуються сучасні фронтенд-технології (ReactJS, Angular, TypeScript тощо), бекенд на Java зі Spring, REST/SOAP, бази даних SQL/NoSQL, хмари (AWS, Azure, Google Cloud), інструменти тестування й CI/CD-практики
Цільова аудиторія	Основні клієнти ProfITsoft – середні й великі компанії, зокрема страховики та фінансові установи, які потребують комплексних рішень автоматизації бізнес-процесів. Крім того, компанія орієнтується на європейський ринок і працює з міжнародними замовниками, а також на проєкти в галузі Travel & Hospitality та FinTech. Для страхового сегмента це, насамперед, страхові компанії та, можливо, великі брокерські мережі, які потребують інтегрованих систем для автоматизації договорів, калькуляцій і звітності
Доступні платформи	Крос платформний доступ через браузер, з охопленням хмарної інфраструктури і можливістю інтеграції з мобільними або десктопними клієнтами через API.

На основі проведеного аналізу, бачимо, що прямих конкурентів нашому застосунку IBAS фактично немає: існують лише схожі платформи, які без фактичного залучення страхових компаній не можуть функціонувати автономно для страхового посередника. Тому доцільність розробки власного модульного рішення під специфіку брокерської діяльності очевидна: воно забезпечить

більшу гнучкість у налаштуванні шаблонів, калькуляцій і швидку адаптацію до змін регуляторних вимог.

1.3 SWOT-аналіз проєкту

Для оцінювання стратегічного положення інформаційної системи IBAS було проведено SWOT-аналіз [15]. Він дає змогу структуровано проаналізувати внутрішні характеристики проєкту і зовнішні фактори середовища, що впливають на перспективи його розвитку. Результати SWOT-аналізу подано в табл. 1.3-1.6.

У табл. 1.3 наведені сильні сторони проєкту IBAS. Вони відображають ключові внутрішні переваги системи порівняно з альтернативними рішеннями на ринку.

Таблиця 1.3

Сильні сторони (Strengths) проєкту

Код	Характеристика сильних сторін
S01	Спеціалізація системи під потреби страхових посередників українського ринку з урахуванням вимог регулятора.
S02	Автоматизоване формування договорів, додатків, пролонгацій та супровідних документів за гнучкими шаблонами.
S03	Вбудований калькулятор страхових премій з підтримкою різних продуктів і страхових компаній.
S04	Єдина централізована база клієнтів, посередників і договорів із розширеними можливостями пошуку та фільтрації.
S05	Контроль строків дії договорів, нагадування, задачі та моніторинг статусів, що знижує операційні ризики.
S06	Можливість формування управлінської та регуляторної звітності в напівавтоматичному режимі.
S07	Сучасна архітектура (мікросервіси, контейнеризація, хмарне розгортання), що забезпечує масштабованість і відмовостійкість.
S08	Розвинена система ролей і прав доступу, аудит дій користувачів, що підвищує рівень інформаційної безпеки.
S09	Висока гнучкість налаштувань (тарифи, продукти, шаблони, ролі), що дає змогу адаптувати систему під конкретного брокера чи мережу.

Сильні сторони IBAS пов’язані насамперед зі спеціалізацією на потребах страхових посередників українського ринку, високим рівнем автоматизації

операцій, наявністю вбудованого калькулятора премій та єдиної централізованої бази клієнтів і договорів.

У табл. 1.4 наведені слабкі сторони проєкту IBAS. Слабкі сторони відображають внутрішні обмеження та недоліки проєкту, які можуть ускладнити впровадження системи або знизити її привабливість для частини цільової аудиторії. Їх своєчасне виявлення дозволяє закласти заходи щодо мінімізації ризиків.

Таблиця 1.4

Слабкі сторони (Weaknesses) проєкту

Код	Характеристика слабких сторін
W01	Висока початкова вартість впровадження для невеликих брокерів та індивідуальних агентів.
W02	Залежність від якості й стабільності зовнішніх API
W03	Обмежені ресурси команди розробки, що підвищує ризик перевантаження та затримок у реалізації функціоналу.
W04	Складність міграції історичних даних із файлів Excel та застарілих облікових систем.
W05	Необхідність навчання користувачів, зважаючи на різний рівень цифрової грамотності страхових посередників.
W06	Підвищені вимоги до інформаційної безпеки та відповідності регуляторним нормам, що потребує додаткових витрат.
W07	Висока функціональна складність системи може бути надлишковою для дрібних агентів.
W08	Залежність від хмарної інфраструктури (ризик збоїв провайдера та коливань вартості хмарних сервісів).

До ключових слабких сторін належать висока початкова вартість впровадження для невеликих брокерів, залежність від якості й стабільності зовнішніх API та хмарної інфраструктури, обмежені ресурси команди розробки, а також складність міграції історичних даних і потреба в навчанні користувачів. Додатковим викликом є підвищені вимоги до інформаційної безпеки та регуляторної відповідності, що потребують постійних інвестицій.

Можливості характеризують сприятливі зовнішні фактори ринку та регуляторного середовища, які проєкт може використати для свого розвитку.

Вони створюють потенціал для розширення функціоналу, клієнтської бази та джерел доходу (див. табл. 1.5).

Таблиця 1.5

Можливості (Opportunities) проєкту

Код	Характеристика можливостей
O01	Зростання кількості страхових посередників та потреби в автоматизації після скорочення кількості страхових компаній.
O02	Перехід страхових продажів в онлайн та гібридний формат, попит на дистанційне обслуговування клієнтів.
O03	Посилення регуляторних вимог до звітності та прозорості, що стимулює попит на спеціалізовані IT-рішення.
O04	Використання аналітики даних як доданої цінності для посередників.
O05	Розширення функціоналу на суміжні сегменти (фінансові консультанти, кредитні брокери, банки-партнери).
O06	Можливість надання системи за моделлю SaaS із підпискою, що формує стабільний регулярний дохід [16].
O07	Використання штучного інтелекту для генерації документів, рекомендацій страхових продуктів, виявлення підозрілих операцій

Загрози описують зовнішні чинники, що можуть негативно вплинути на розвиток проєкту: від ринкової конкуренції до регуляторних та макроекономічних ризиків.

У табл. 1.6 наведено основні загрози для проєкту пов’язані з конкуренцією з боку існуючих рішень і міжнародних платформ для брокерів, економічною та воєнною нестабільністю, частими змінами законодавства, кіберризиками, залежністю від ключових страхових партнерів та зростанням вартості хмарних сервісів.

Таблиця 1.6

Загрози (Threats) для проєкту

Код	Характеристика загроз
1	**2**
T01	Конкуренція з існуючими CRM/ERP-рішеннями для страхового ринку та міжнародними платформами для брокерів.
T02	Економічна та воєнна нестабільність, що знижує платоспроможність клієнтів і інвестиційну активність.
T03	Часті зміни страхового законодавства й вимог регулятора, що потребують постійних доопрацювань системи.

Закінчення табл. 1.6

1	2
T04	Кіберзагрози та ризик витоку персональних даних, що можуть негативно вплинути на репутацію продукту.
T05	Залежність від ключових партнерських страхових компаній та можливість зміни умов співпраці.
T06	Зростання вартості хмарних сервісів, що впливає на собівартість і ціну для клієнтів.
T07	Імовірність копіювання концепції великими гравцями з більшими ресурсами та власними каналами збуту.

Додатковим ризиком є потенційне копіювання концепції більшими гравцями ринку, які мають ширші фінансові та маркетингові ресурси.

Узагальнюючи результати SWOT-аналізу, можна зробити висновок, що проєкт IBAS має виражені конкурентні переваги, пов’язані зі спеціалізацією, високим рівнем автоматизації та сучасною технологічною основою. Водночас для успішної реалізації необхідно цілеспрямовано працювати над подоланням виявлених слабких сторін і проактивно керувати зовнішніми загрозами. Отримані у межах SWOT-аналізу результати будуть використані під час подальшого планування розвитку системи та формування заходів з управління ризиками проєкту.

1.4 Визначення зацікавлених сторін проєкту

Ефективне управління проєктом розробки веб-платформи для страхових посередників потребує чіткого визначення кола зацікавлених сторін та аналізу їхнього впливу на проєкт і очікуваних вигод. Зацікавленими сторонами вважають фізичних осіб, групи осіб або організації, які прямо чи опосередковано впливають на хід реалізації та результати проєкту або зазнають впливу від впровадження його продукту [17].

До внутрішніх зацікавлених сторін у контексті даного дослідження віднесено замовника проєкту, яким виступає керівництво брокерської компанії, страхових посередників (фахівців зі страхування) як основних користувачів системи, команду розробників, а також керівника проєкту та потенційних інвесторів і партнерів, що забезпечують фінансування ініціативи (див. табл. 1.7).

Замовник та керівник проєкту визначають стратегічні цілі, затверджують бюджет і пріоритети розвитку платформи. Фахівці зі страхування щоденно працюватимуть із системою для оформлення договорів, розрахунку премій та ведення клієнтської бази, тому саме вони формують ключові вимоги до функціональності й зручності інтерфейсу .

Таблиця 1.7

Аналіз зацікавлених сторін проєкту

Зацікавлені сторони проєкту	**Вплив на проєкт**	**Вигода для зацікавленої сторони**
1	**2**	**3**
Замовник проєкту (керівництво брокерської компанії)	Визначає цілі проєкту, затверджує бюджет і строки, приймає ключові рішення щодо функціоналу та пріоритетів	Отримує інструмент для підвищення ефективності роботи посередників, зростання продажів та контролю бізнес-процесів
Страхові посередники - фахівці зі страхування (кінцеві користувачі системи)	Формують основні вимоги до функціоналу та зручності системи, впливають на рівень прийняття рішення щодо впровадження та подальшого використання платформи	Скорочення часу на оформлення договорів, зменшення кількості помилок, зручний доступ до клієнтської бази та комісій, підвищення продуктивності роботи
Керівник проєкту	Координує виконання робіт, визначає пріоритети, відповідає за досягнення цілей проєкту	Отримує керований, прогнозований проєкт та продукт, що відповідає очікуванням стейкхолдерів
Команда розробників	Відповідають за технічну реалізацію, інтеграції, супровід системи; впливають на якість, надійність та безпеку рішення	Можливість впроваджувати сучасні технології, стандартизувати процеси розробки, отримати кероване навантаження та зрозумілі вимоги
Потенційні інвестори та партнери	Впливають на доступний бюджет та рішення щодо подальшого розвитку й масштабування проєкту	Потенційний дохід від успішного комерційного впровадження платформи та її масштабування на нові ринки
Страхові компанії-партнери	Формують вимоги до форматів обміну даними та шаблонів договорів; можуть впливати на спектр страхових продуктів у системі	Збільшення обсягів продажів страхових продуктів, автоматизація обміну інформацією, покращення якості та швидкості звітності

Закінчення табл. 1.7

1	2	3
Постачальники ІТ-інфраструктури (хмарні сервіси, ЕЦП, платіжні сервіси)	Визначають технічні обмеження й можливості системи (доступність, масштабованість, безпека)	Довгострокова співпраця, стабільне навантаження на сервіси, розширення клієнтської бази
Регуляторні органи (НБУ та ін.)	Встановлюють нормативні вимоги, яких має дотримуватися система; можуть ініціювати зміни в підходах до зберігання та обробки даних	Забезпечення прозорості діяльності страхових посередників, коректна та своєчасна звітність, зниження рівня порушень законодавства
Страхувальники (клієнти посередників)	Опосередковано впливають на вимоги до сервісу через рівень задоволеності роботою посередників	Швидке та зрозуміле оформлення договорів, менше помилок у документах, підвищення якості обслуговування

До зовнішніх зацікавлених сторін віднесено страхові компанії-партнери, постачальників ІТ-інфраструктури, регуляторні органи та страхувальників – клієнтів посередників. Страхові компанії-партнери надають шаблони договорів і тарифні дані, формують вимоги до форматів обміну інформацією й зацікавлені в збільшенні обсягів продажів власних продуктів.

Для кожної із визначених груп проаналізовано ступінь впливу на проєкт та очікувані вигоди від впровадження системи IBAS. Узагальнені результати такого аналізу наведено в табл. 1.7, де відображено роль кожної сторони, характер її впливу на проєкт та основні переваги, які вона отримує.

Проведене групування і аналіз (див. табл. 1.7) формують основу для подальшого планування комунікацій, уточнення вимог до веб-платформи та управління ризиками, пов’язаними з очікуваннями й впливом різних учасників проєкту.

1.5 Побудова дерева проблем та дерева цілей

У цьому підрозділі здійснюється побудова дерева проблем та відповідного дерева цілей для проєкту створення інформаційної системи IBAS. На основі

результатів аналізу зовнішнього середовища та внутрішнього стану галузі узагальнюються ключові суперечності в роботі страхових посередників і структуруються у вигляді дерева проблем.

1.5.1 Дерево проблем

Для глибокого розуміння головної проблеми проєкту та визначення її корінних причин і наслідків було побудовано дерево проблем [18] (див. рис. 1.1). Такий підхід дозволяє системно проаналізувати, чому саме виникають негативні ефекти, і як вони впливають на бізнес, а також слугує основою для формулювання пріоритетних завдань рішення.

Далі подано опис етапів побудови дерева проблем та короткий огляд його вмісту.

1. Визначення центральної проблеми

Першим кроком було чітко сформульовано центральну проблему, яка відображає основний біль бізнесу: неефективне управління процесом укладання та обробки страхових договорів страховими посередниками. Ця проблема охоплює надмірну тривалість підготовки договорів, високу ймовірність помилок, відсутність прозорості та складнощі з масштабуванням бізнесу.

2. Збір інформації про поточні процеси

Щоб розібратися в джерелах проблеми, проведено аналіз існуючих робочих процесів:

- Інтерв’ю з агентами та адміністраторами для з’ясування болючих місць (час на підготовку, оновлення шаблонів, калькуляції, нагадування).
- Огляд наявних інструментів: Word/Excel-шаблони, CRM, таблиці та інші локальні рішення.
- Вивчення вимог регулятора щодо змін у документах і полісах.
- Оцінка IT-ресурсів і навичок команди: чи є готові внутрішні рішення, чи доводиться користуватися сторонніми CRM/ERP із доопрацюваннями.

3. Декомпозиція на гілки: причини та наслідки

Дерево проблем поділене на дві основні частини. Ліва гілка (Причини) це глибинні фактори, через які процеси залишаються неефективними. Права гілка (Наслідки) це негативні результати, що виникають, якщо проблему не вирішувати. Центральний вузол позначає формулювання проблеми, з якого стрілками ведуться гілки причин і наслідків.

Рис. 1.1. Дерево проблем

Отже, побудова дерева проблем дала змогу систематизувати результати аналізу поточних «болючих точок», виділити пріоритетні області для автоматизації та обґрунтувати формування ключових вимог до системи IBAS.

1.5.2 Дерево цілей

Для подальшої конкретизації бачення проєкту та узгодження його зі стейкхолдерами було розроблено дерево цілей (рис. 1.2). Воно дозволяє чітко структурувати загальну мету проєкту, розкласти її на основні напрями та конкретні завдання, а також наочно показує, яким чином досягнення окремих підцілей забезпечує досягнення стратегічної цілі. Це, своєю чергою, полегшує планування робіт і комунікацію в команді та із зацікавленими сторонами.

Рис. 1.2. Дерево цілей

У центрі нашого дерева цілей стоїть головна мета проєкту:

Підвищити ефективність процесу укладання та обробки страхових договорів страховими посередниками шляхом розробки та впровадження інформаційної системи IBAS.

Від неї виходять п’ять основних гілок (конкретних цілей), які відображають ключові напрями роботи над системою. Кожна конкретна ціль деталізується низкою підцілей, що у сукупності забезпечують досягнення загальної мети.

1. Оптимізувати операційні процеси

Ціль спрямована на усунення ручних та трудомістких етапів підготовки договорів і пов’язаних операцій.

У схемі до цього напряму відносяться наступні підцілі:

- Автоматизувати укладання договорів завдяки шаблонам: заміна ручного формування документів на механізм завантаження шаблонів із плейсхолдерами та автоматичну підстановку даних, що значно скоротить час і зменшить помилки.
- Спростити процес пролонгації договорів: створити інструменти для відстеження термінів і автоматичного запуску процедури пролонгації, щоб агенти могли працювати проактивно і клієнти отримували своєчасні пропозиції.
- Реалізувати автоматизований калькулятор премій і комісій: забезпечити гнучкий механізм розрахунку страхових премій за правилами конкретного страховика та точний облік комісій агентів/субагентів без ручних обчислень.
- Впровадити систему нагадувань: автоматичні сповіщення агентам і клієнтам про ключові події (оплата, закінчення строку, нові продукти), що підвищує якість обслуговування й утримання клієнтів.

Таким чином, цей блок охоплює ті процеси, які до цього виконувалися локально або вручну, і перетворює їх на автоматизовані сервісні компоненти системи.

2. Забезпечити фінансовий контроль та аналітику

Цей напрям фокусується на можливостях керівництва і агентів отримувати прозору інформацію про доходи, ефективність та перспективи розвитку.

У схемі виділені підцілі:

- Розробити модуль звітності: формування стандартних і користувацьких звітів про обсяги укладених договорів, суми премій і комісій, доходи субагентів, що дає змогу контролювати фінансові показники.
- Спростити облік доходів субагентів: відокремлений облік комісій і виплат субагентам із можливістю швидкого доступу до історії виплат, що допомагає уникати помилок та підвищує прозорість.

- Розробити модуль аналітики: аналітичні інструменти для глибшого розуміння тенденцій портфеля продуктів, прогнозування пролонгацій, оцінки продуктивності агентів і прийняття стратегічних рішень.

Завдяки цьому блоку IBAS перетвориться на джерело актуальної інформації, необхідної для планування, оцінки ризиків і розвитку бізнесу.

3. Спростити пошук даних клієнтів і договорів

Ціль покликана надати користувачам швидкий і зручний доступ до всієї наявної інформації:

- Зробити централізовану базу даних: створити єдине сховище для записів про клієнтів та договори замість розпорошених локальних файлів.
- Зробити міграцію існуючих записів (Excel/Word) у єдину базу: забезпечити перенесення історичних даних у нову систему, мінімізувати втрату інформації й гарантувати цілісність даних.

Цей напрям є фундаментальним: без централізованої БД неможливо реалізувати інші автоматизовані функції й аналітику.

4. Забезпечити безпеку та стійкість системи

Надійність і безпека – критичні вимоги до будь-якого рішення, що працює з персональними та фінансовими даними:

- Хмарне розгортання і масштабування: забезпечити можливість гнучкого збільшення ресурсів у хмарі за потреби, щоб система працювала стабільно при зростанні навантаження.
- Забезпечити шифрування даних: зберігання та передача чутливої інформації у зашифрованому вигляді, захист від несанкціонованого доступу.
- Впровадити механізм з автоматичним резервним копіюванням даних: регулярні бекапи та можливість швидкого відновлення після збоїв, що гарантує безперервну роботу і мінімальні втрати даних.

Цей блок гарантує відповідність вимогам інформаційної безпеки, стійкість до інцидентів та масштабованість.

5. Забезпечити відповідність системи Закону України

Окремий напрям присвячено відповідності нормативно-правовим вимогам:

- Відповідність Закону України про стимулювання розвитку цифрової економіки: застосунок має дотримуватися стандартів електронного документообігу та захисту цифрових сервісів згідно з чинним законодавством.
- Відповідність Закону України про страхування: система повинна підтримувати структуру та валідацію договорів, необхідні за останніми редакціями, з можливістю швидкого оновлення бізнес-правил під нові регуляторні вимоги.

Цей блок забезпечує легальність роботи IBAS і мінімізує ризики штрафів чи інших санкцій.

1.6 Формулювання технічного завдання на розробку у вигляді паспорту проєкту

Формування паспорта проєкту є важливим етапом, що передбачає чітке визначення основних параметрів, таких як мета, завдання, часові рамки, ресурси та очікувані результати. Паспорт проєкту дає змогу структуровано описати ключові ознаки, які характеризують проєкт, визначити його межі й закласти основу для ефективного управління на всіх етапах реалізації [19].

Для наочного представлення основних параметрів інформаційної системи IBAS було розроблено паспорт проєкту, який наведений у табличній формі (табл. 1.8). У таблиці визначено назву проєкту, сформульовано мету та описано основні ознаки, які є критично важливими для його успішного виконання.

Таблиця 1.8

Паспорт проєкту

Параметр	Опис
1	**2**
Назва	IBAS (Insurance Broker Assistant System) - інформаційна система для укладання договорів страховими посередниками
Ознаки (елементи проєкту)	*1. Чітко визначена мета та результат:* основною метою є підвищити ефективність процесу укладання та обробки страхових договорів страховими посередниками шляхом розробки інформаційної системи IBAS.

Закінчення табл. 1.8

1	2
Ознаки (елементи проєкту)	Результат — готовий до використання програмний продукт. *2. Тимчасові рамки:* проєкт має обмежений строк виконання — 1 рік. *Тривалість проєкту з 1 вересня 2025 року по 30 серпня 2026 року* *3. Унікальність:* проєкт спрямований на створення нової системи з конкретними функціональними вимогами. Ця система раніше не існувала і розробляється для специфічних потреб страхових посередників. *4. Визначений обсяг робіт:* описані конкретні етапи, які включають ініціацію, планування, розробку, впровадження та завершення. Проєкт охоплює різні сфери діяльності, такі як програмування, дизайн, тестування, безпека, страхування. *5. Обмежені ресурси:* є обмеження бюджету 4 200 000,00 грн. Для досягнення цілей потрібно оптимізувати використання даного ресурсу *6. Командна робота:* для реалізації проєкту залучається команда спеціалістів (розробники, дизайнер, аналітик, тестувальник, проєктний менедже, девопс). Кожен учасник виконує свою роль у рамках проєкту. *7. Управління ризиками:* проєкт враховує ризики: перевищення бюджету, затримки розробки, форс-мажорні обставини. Передбачені заходи для їхнього мінімізації (тестування, резервні ресурси). *8. Орієнтованість на клієнта:* Результат проєкту створюється з урахуванням потреб конкретної аудиторії (страхові агенти та брокери). Унікальність полягає в адаптації функціоналу до реальних бізнес-процесів, які на даний момент затребувані на ринку страхування. *9. Орієнтованість на якість:* випуск альфа-версії та усунення можливих помилок, етап впровадження містить обов'язковий супровід системи. *10. Орієнтовність на досягнення стійкого та довготривалого результату.* Сервіс має самостійно функціонувати.
Цілі проєкту	1. Оптимізувати операційні процеси. 2. Забезпечити фінансовий контроль та аналітику. 3. Спростити пошук даних клієнтів і договорів. 4. Забезпечити безпеку та стійкість системи. 5. Забезпечити відповідність системи законодавству України.
Зацікавлені сторони проєкту	*Основні внутрішні:* Замовник проєкту (керівництво брокерської компанії) Страхові посередники - фахівці зі страхування (кінцеві користувачі системи) Керівник проєкту Команда розробників Інвестори, які вкладають гроші в потенційно вигідний проєкт.
Зацікавлені сторони проєкту	*Зовнішні:* Страхові компанії-партнери, які надають шаблони договорів і тарифні дані. Постачальники IT-інфраструктури Регуляторні органи (послугами системи повинно відповідати законодавству). Страхувальники (клієнти посередників) опосередковано — через якість обслуговування агента.

Таким чином, сформований паспорт проєкту узагальнює ключові характеристики інформаційної системи IBAS, визначає її мету, коло

зацікавлених сторін, основні обмеження та припущення. Він слугує базовим документом для постановки ключових завдань проєкту, надає важливу інформацію щодо строків реалізації та обсягу бюджету, а також є відправною точкою для подальшого планування робіт, вибору життєвого циклу, розроблення структури управління проєктом і деталізації вимог до програмного продукту.

1.7 Вибір та обґрунтування життєвого циклу проєкту

Життєвий цикл проєкту створення інформаційної системи Insurance Broker Assistant System (IBAS) визначає послідовність етапів, через які проходить проєкт від моменту зародження ідеї до формального завершення та передачі результатів у роботу замовнику. Опис життєвого циклу дає змогу структурувати роботи, узгодити очікування зацікавлених сторін, визначити логіку переходів між етапами та забезпечити керованість проєкту на всіх стадіях його реалізації [20]. У межах даної роботи життєвий цикл проєкту IBAS представлено у вигляді п’яти послідовних етапів: ініціації, планування, розробки, впровадження та завершення, кожен з яких має власні цілі та основні завдання (див. рис. 1.3).

1. Ініціація проєкту

Цілі етапу полягають у формуванні загального бачення, обґрунтуванні доцільності реалізації та визначенні ключових очікувань від проєкту IBAS. На цьому етапі створюються підстави для подальшого планування, здійснюється попередній аналіз проблем, які потребують вирішення, а також окреслюється коло зацікавлених сторін. Ініціація забезпечує офіційний старт проєкту та його визнання в межах організації.

Основні завдання етапу:

- виконати дослідження страхового ринку та визначити актуальні проблеми діяльності страхових посередників;
- провести PEST- та SWOT-аналізи для оцінки внутрішніх і зовнішніх факторів впливу на проєкт;
- побудувати дерево проблем і дерево цілей як логічну основу для формування мети проєкту;

- визначити основних зацікавлених сторін та їхні очікування від результатів упровадження системи IBAS;
- підготувати паспорт проєкту з описом мети, завдань, обмежень, орієнтовного бюджету та строків;
- попередньо визначити життєвий цикл проєкту та обрати гібридну модель управління.

2. Планування

Цілі етапу полягають у перетворенні стратегічної ідеї IBAS у структурований і реалістичний план її реалізації. На цьому етапі визначається зміст робіт, ресурси, строки та підхід до виконання, що забезпечить ефективну координацію дій команди і контроль за ходом реалізації. Планування створює основу для управління обсягом, часом, вартістю та ризиками.

Основні завдання етапу:

- побудувати концептуальну модель інформаційної системи IBAS із визначенням її ключових компонентів і взаємозв'язків;
- сформулювати та структуризувати функціональні і нефункціональні вимоги до системи;
- розробити постановку задач у математичному вигляді для подальшої формалізації алгоритмів і розрахунків;
- створити беклог продукту з урахуванням пріоритетів реалізації функціональних модулів;
- здійснити декомпозицію робіт і побудувати WBS для відображення структури робіт проєкту;
- визначити організаційну структуру управління проєктом, ролі та зони відповідальності учасників;
- спланувати ресурсне забезпечення, бюджет і строки виконання робіт з урахуванням тривалості проєкту;
- врахувати результати SWOT- та PEST-аналізів при розробці планів управління ризиками, комунікаціями та якістю.

3. Розробка

Цілі етапу полягають у створенні робочих інкрементів системи IBAS з поступовим розширенням функціональності та перевіркою її працездатності на практиці. Етап реалізується за адаптивною моделлю, що забезпечує можливість уточнення вимог і пріоритетів у процесі роботи. Основна увага приділяється гнучкому управлінню змінами, швидкому отриманню зворотного зв'язку та поступовому наближенню продукту до очікувань користувачів.

Основні завдання етапу:

- деталізувати вимоги у вигляді беклогу продукту та перетворити їх на конкретні завдання розробки і тестування;
- виконати проєктування UX/UI-дизайну та технічної архітектури системи (frontend, backend, база даних, інтеграції);
- реалізувати frontend-частину системи (користувацькі інтерфейси для страхових посередників та адміністраторів);
- реалізувати backend-частину системи (модулі бізнес-логіки, роботу з даними та інтеграційні сервіси);
- організувати виконання робіт у вигляді послідовних інкрементів (спринтів), у межах яких реалізується та перевіряється окрема частина функціональності;
- забезпечити паралельне проведення тестування розроблених компонентів;
- за результатами кожного спринта аналізувати отриманий зворотний зв'язок і за потреби коригувати подальші завдання.

4. Впровадження

Цілі етапу полягають у забезпеченні переходу системи IBAS з етапу розробки до стадії реальної експлуатації. Основна мета – досягнення стабільної роботи системи в продуктивному середовищі та забезпечення готовності користувачів до її застосування. Цей етап завершує технічну реалізацію і фокусується на практичному впровадженні результатів у діяльність організації.

Основні завдання етапу:

- підготувати інфраструктуру для розгортання системи, включаючи налаштування серверів, середовищ та засобів захисту інформації;
- здійснити розгортання системи IBAS у продуктивному середовищі;
- провести приймально-здавальні випробування із залученням замовника та ключових користувачів;
- забезпечити навчання користувачів роботі з системою та надати необхідну супровідну документацію;
- організувати період дослідної експлуатації з фіксацією зауважень, пропозицій і потенційних покращень;
- підготувати систему до переходу в режим повноцінної промислової експлуатації.

5. Завершення проєкту

Цілі етапу полягають у формальному завершенні робіт, оцінюванні досягнутих результатів та документальному закритті проєкту IBAS. Основна увага приділяється аналізу ефективності, підведенню підсумків та передачі системи в подальшу експлуатацію і супровід. Етап забезпечує завершення усіх адміністративних, організаційних і фінансових процедур.

Основні завдання етапу:

- провести підсумкову оцінку досягнення цілей проєкту, визначених під час ініціації;
- виконати аналіз результатів реалізації за показниками часу, вартості, якості та рівня ризиків;
- підготувати підсумковий звіт про реалізацію проєкту та узагальнити ключові результати;
- формально передати систему IBAS замовнику для промислової експлуатації та подальшого супроводу;
- здійснити закриття контрактів, фінансових зобов’язань та внутрішніх адміністративних процедур;

- зафіксувати отримані уроки, висновки та рекомендації для майбутніх проєктів аналогічного типу.

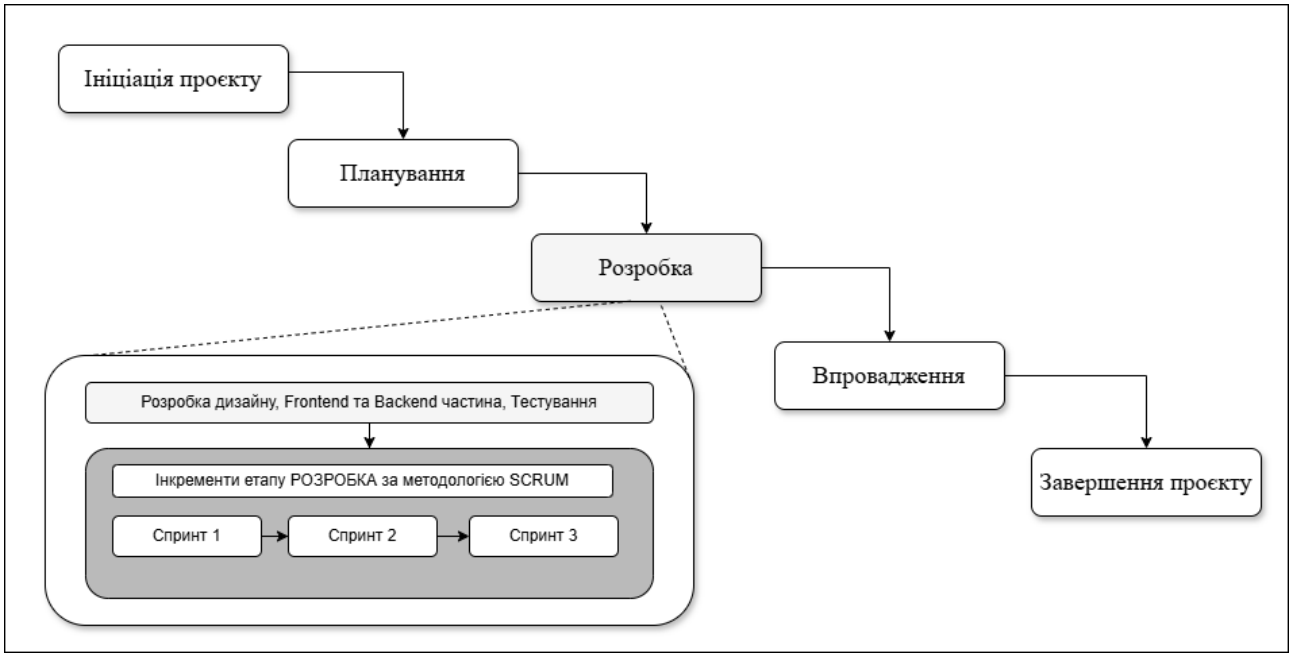


Рис. 1.3. Життєвий цикл проєкту IBAS

Таким чином, запропонований життєвий цикл проєкту IBAS, забезпечує логічну та послідовну організацію робіт на всіх стадіях реалізації проєкту. Чітке формулювання цілей і завдань кожного етапу, а також розмежування сфер застосування традиційної та адаптивної моделей управління дозволяє підвищити прозорість процесу, своєчасно виявляти та враховувати ризики, ефективно використовувати ресурси та орієнтуватися на досягнення запланованих результатів. Описаний життєвий цикл слугує методологічною основою для подальшого деталізованого планування, побудови структур декомпозиції робіт і ресурсів, формування календарних планів, бюджетів та системи контролю перебігу проєкту.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

У цьому розділі виконується детальне проектування інформаційної системи: спочатку формується концептуальна модель доменної області, потім формалізуються ключові задачі у вигляді математичних постановок.

2.1. Розробка концептуальної моделі інформаційної системи

У цьому підрозділі описано підхід до створення концептуальної моделі: визначення контексту системи, зовнішніх факторів впливу та підсистем, їхніх взаємодій і ключових бізнес-процесів [21]. Концептуальна модель дає загальне уявлення про головні об’єкти системи та зв’язки між ними ще до початку детального проєктування бази даних чи архітектури. Далі буде поетапно показано, як формується контекстна діаграма, описуються надсистеми й підсистеми, визначаються сутності та основні бізнес-процеси.

1. Визначимо назву та призначення системи.

IBAS (Insurance Broker Assistant System) - інформаційна система для укладання договорів страховими посередниками

Забезпечення автоматизації роботи страхових посередників, агентів і брокерів у частині створення, управління, збереження договорів і внутрішньої фінансової звітності.

2. Проаналізуємо систему з фізичної точки зору.

Надсистема: страхові компанії, державні органи (НБУ-регулятор ринку страхування), клієнти, постачальники послуг, Закони України (Закон про страхування, Закони про захист даних, Захист прав споживачів)

3. Визначимо зовнішні зв’язки із надсистемою.

Зовнішні фактори впливу:

Клієнти замовляють страхові послуги.

Страхові компанії надають шаблони договорів, вимоги до звітності, рахунок для оплати, здійснюють виплати в разі настання страхового випадку, тобто надають продукт для продажу, який продають агенти.

Державні органи (НБУ-регулятор ринку страхування) контролюють відповідність законодавству страхові компанії, контролюють агентів, програмне забезпечення, відповідність до вимог закону договорів і так далі.

Постачальники послуг забезпечують хостинг і зберігання даних.

Закони України (Закон про страхування, Закони про захист даних, Захист прав споживачів) забезпечують чіткі вимоги до договору, до програмного забезпечення.

4. Визначимо перелік підсистем.

Основні підсистеми системи:

- веб-платформа: інтерфейс для страхових агентів і брокерів;
- офіс;
- співробітники.

5. Визначимо перелік елементів для кожної підсистеми.

5.1 Веб-платформа:

- авторизація;
- модуль управління страховими компаніями;
- модуль управління користувачами;
- модуль адміністратора: створення, налаштування шаблону для укладання певного виду договору;
- модуль укладання договорів: редагування, друк, збереження в пдф-скану, шляхом автоматичного накладання фото печаті та підписку;
- модуль звітності: фінансові звіти (дохід агента, витрати), контроль пролонгації, операційні звіти (кількість укладених договорів), графіки та аналітика;
- інтеграційний модуль: API для зв’язку з державними реєстрами, цифровий підпис;
- база даних: дані про клієнтів (ПІБ, контакти), договори (умови, дати), фінансова інформація (дохід, витрати).

5.2 Співробітники:

- директор;

- бухгалтер;
- фахівці зі страхування;
- фахівець по зв’язкам з громадськістю;
- фахівець з врегулювання;
- юрист.

5.3 Офіс:

- кабінети;
- туалет;
- кабінет для клієнта.

6. Спроектуємо вигляд системи.

На рис. 2.1 зображено концептуальну модель системи, яка дає загальне уявлення про головні об’єкти системи та зв’язки між ними ще до початку детального проєктування бази даних чи архітектури.

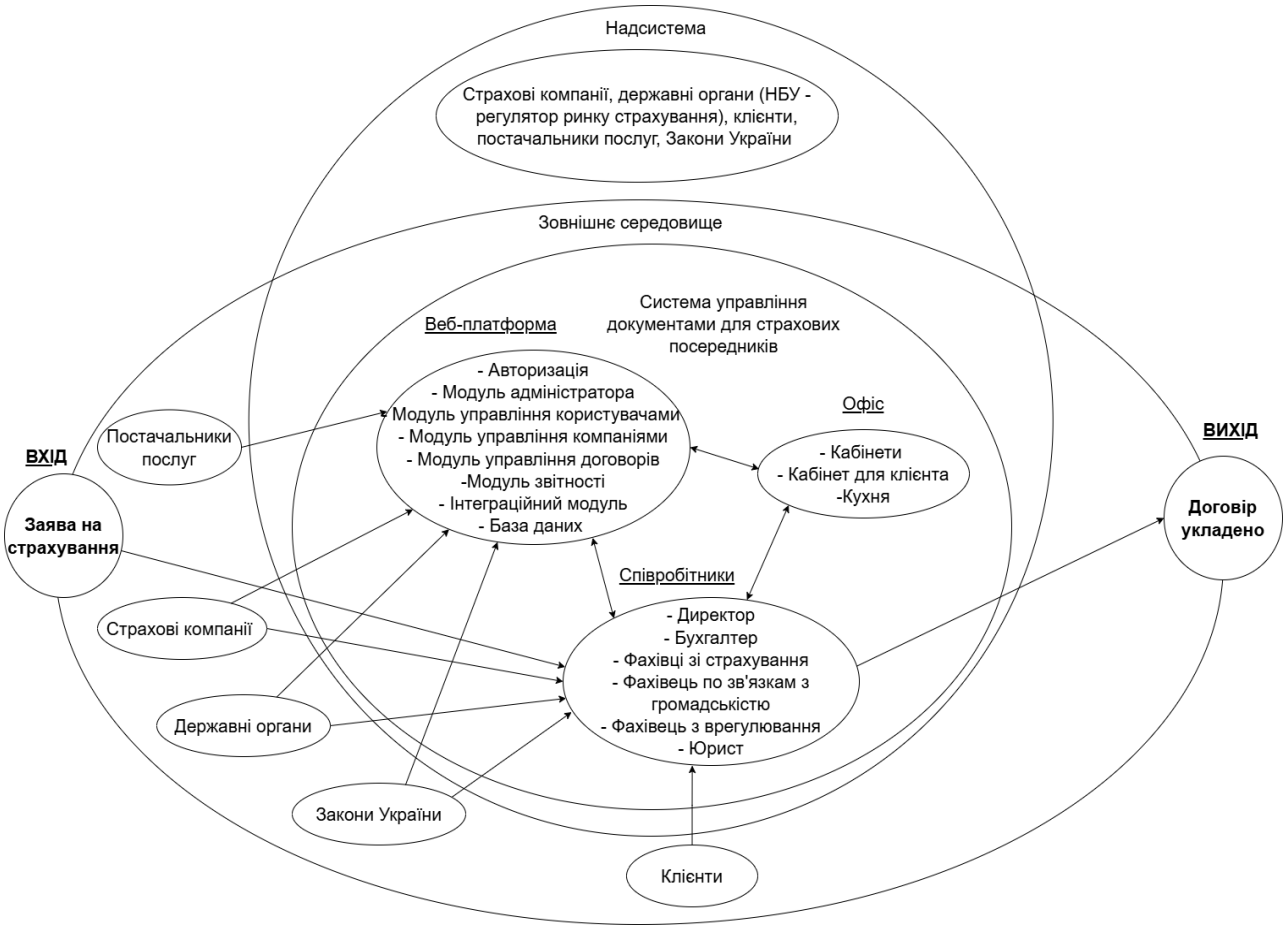


Рис. 2.1. Концептуальна модель інформаційної системи

7. Визначимо підпроцеси, як елементи системи.

Замовлення договору:

1. Клієнт замовляє договір страхування.
2. Менеджер приймає замовлення та обирає найкращу пропозицію порівнюючи пропозиції компаній з, якими співпрацює.
3. Клієнт обирає пропозицію надсилає свої документи.
4. Менеджер готує шаблон договору в веб-застосунку, автоматично перевіряючи клієнта по базам фінансового моніторингу.
5. Дані передаються у БД.
6. Клієнт оплачує договір.
7. Менеджер надсилає підписаний договір клієнту.
8. Зворотній зв’язок із клієнтом для покращення обслуговування.
9. Визначимо надсистему для процесу і зовнішні зв'язки з нею.

Результатом було укладено договір страхування. Після цього відбувається зовнішній зв’язок від клієнта до менеджера, можливий відгук залишається на сайті або у соціальних мережах.

2.2. Постановка задачі в математичному вигляді

Постановка задач у математичному вигляді допомагає чітко визначити вхідні дані, вихідні результати, відношення та обмеження, а також сформулювати алгоритмічні кроки для реалізації бізнес-логіки системи. Далі опишемо математичні моделі для основних операцій: розрахунок страхової премії, розподіл комісій. Наведемо приклад розрахунку для продукту «Медичні витрати + Нещасний випадок». Такий підхід допомагає чітко визначити, які дані потрібні на вході, як вони обробляються і які перевірки виконуються перед підрахунком.

2.2.1 Основні множини і позначення

Спочатку введемо базові множини, що використовуються у формулюваннях:

C – множина усіх клієнтів (страхувальників).

I – множина усіх партнерських страхових компаній.

P – множина видів страхових продуктів.

A – множина страхових агентів.

K – множина укладених договорів.

Кожен договір $k \in K$ описується як кортеж:

$$k = (c, i, p, s, e, S, \dots) \tag{2.1}$$

де

$c \in C$ – клієнт;

$i \in I$ – страхова компанія;

$p \in P$ – продукт;

s – дата початку (початковий момент) ;

e – дата закінчення договору;

S – страхова сума (сума покриття).

Для кожного страховика i продукту p існує набір параметрів та коефіцієнтів, що визначають розрахунок премії. Позначимо ці правила як функцію:

$$R_{i,p} = \text{параметри тарифу для } (i, p) \tag{2.2}$$

де

i – страхова компанія;

p – продукт.

Для кожного агента $a \in A$ і компанії $i \in I$ існує ставка комісії або схема $\mathrm{C}_{a,i}$. Якщо субагент вкладається по роботі, як і головний агент, тоді головний агент a отримує частину, субагент a' — іншу частину винагороди.

2.2.2 Розрахунок страхової премії

Для кожного нового договору необхідно обчислити розмір страхової премії $П$ на підставі параметрів клієнта та тарифних правил страховика.

Нехай для договору $k = (c, i, p, \dots)$ задана страхова сума S та інші параметри $X = (x_1, x_2, \dots, x_n)$ — наприклад, вік клієнта, характеристики об'єкта страхування, строк страхування, франшиза. Тоді треба визначити функцію

$$f_{i,p}: R^m \to R_+ \tag{2.3}$$

яка за параметрами X та сумою S повертає премію $П$. Зазвичай це виглядає як:

$$П = fi, p(S, X) = S \times \rho_{i,p}(X) \tag{2.4}$$

де $\rho_{i,p}(X)$ - комбінований тарифний коефіцієнт, що обчислюється за правилами $R_{i,\text{P}}$.

Формально:

$$\rho_{i,p}(X) = g_{i,p}(x_1, x_2, \dots, x_n) \tag{2.5}$$

де $g_{i,p}$ визначається набором умов і коефіцієнтів з $R_{i,p}$.

Функція $f_{i,p}$ реалізована в коді як сервіс калькулятора. Вхід: параметри договору. Вихід: числова премія $П$.

2.2.3 Розподіл комісій між агентами

Після обчислення премії $П$ необхідно розподілити комісійні виплати між учасниками: агентом, субагентами, якщо такий присутній.

Нехай для договору k є набір учасників $A_k = \{a_1, a_2, \dots, a_r\}$ із тарифними ставками $\alpha_{a_j,i}$ (комісійні ставки для агента a_j у страховика i). Тоді виплата комісії M_{aj} кожному агенту обчислюється як:

$$M_{aj} = П \times \alpha_{a_j,i} \tag{2.6}$$

Якщо ставки вкладені (наприклад, головний агент отримує α_{main} субагенту лишається α_{sub}), слід врахувати, що сума часток не перевищує певного максимуму. Можна задати обмеження:

$$\sum_{j=1}^{r} \alpha_{a_j,i} \leq a_{max}^{i} \tag{2.7}$$

де a_{max}^{i} - максимально допустима сума комісій для страховика i.

Якщо сума ставок більша, треба переглянути схему або автоматично скоригувати.

Виходить в програмі буде реалізована функція g_{comm} або сервіс, що за вхідними даними (премія $П$, список агентів та їх ставки) повертає вектор виплат $(M_{a1}, \dots, M_{ar})$.

2.2.4 Приклад розрахунку премії для певного страхового продукту

Нижче наведено приклад розрахунку на основі тарифікатора для програми комплексного страхування подорожуючих по Україні. Страхові суми за медичне покриття можуть обиратися серед значень 3000 грн, 5000 грн або 10000 грн, при цьому страхові суми для покриття «Нещасний випадок» відповідають тим самим значенням. Базові тарифи визначаються згідно з таблицею: наприклад для медичних витрат за першу добу та тривалість до 5 днів при страхових сумах 3000 грн базовий тариф становить 0,155 %; аналогічно для нещасного випадку: при 3000 грн БТ = 0,0117. Якщо тривалість перевищує 5 днів і досягає до 21 дня, базовий тариф коригується згідно з тарифними знижками або надбавками для довших періодів, отже в практиці розрахунку слід підставляти БТ із відповідного рядка таблиці для конкретної тривалості D у межах від 1 до 21 дня.

Коефіцієнт за віком застрахованої особи К1 встановлюється згідно таблиці також наприклад, від 18 до 54 років – 1,0. Коефіцієнт за розміром групи К2 складає 1,0 для груп до 20 осіб. Коефіцієнт виду ризику або активності К3, наприклад, для туризму встановлюється на рівні 1,0, для активного дозвілля з підвищеним ризиком – 3,0. Коефіцієнт винагороди агента К4 обчислюється за формулою:

$$\text{К4} = \frac{1-0{,}03-0{,}45}{1-0{,}03-\text{V}} \qquad (2.8)$$

де V – ставка винагороди агента у форматі десяткового дробу.

Наприклад, V = 0,45 (45 %).

Розрахунок виконується для тривалості договору D, наприклад D = 7 днів, при цьому перевіряється умова $D \leq 21$. Для кожної застрахованої особи премія обчислюється окремо за медичне покриття та за нещасний випадок.

Таким чином, процес розрахунку премії для полягає в підстановці заданих страхових сум і тривалості договору у базові тарифи, застосуванні вікових, групових, ризикових та агентських коефіцієнтів, обчисленні окремих частин премії за кожне покриття для кожної особи та підсумовуванні результатів з урахуванням мінімальних обмежень тарифікатора.

Цей підхід у формалізованому вигляді гарантує коректність розрахунків і дозволяє автоматизувати модуль калькуляції в системі IBAS.

РОЗДІЛ 3. ВИБІР ТЕХНОЛОГІЇ УПРАВЛІННЯ ПРОЄКТОМ ТА РОЗРОБКА ЙОГО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Розробка системи IBAS неможлива без підходу до управління проєктом та чітко вибудуваного інформаційного підґрунтя. У цьому розділі послідовно визначаються теоретичні засади вибору технології управління, аналізуються можливі підходи та обґрунтовується застосування гібридної моделі Water-Scrum-Fall для проєкту IBAS.

На основі обраної технології формується система артефактів управління: декомпозиція робіт і WBS, організаційна структура управління проєктом, функціональні та нефункціональні вимоги до продукту, беклог продукту та Спринту №1. Окрему увагу приділено плануванню ресурсного забезпечення та бюджету, а також ідентифікації ризиків і побудові карти управління ними.

Сукупність цих елементів забезпечує керованість, прозорість і прогнозованість реалізації проєкту IBAS.

3.1 Вибір та обгрунтування технології управління проєктом

Ефективна реалізація проєкту створення інформаційної системи Insurance Broker Assistant System потребує свідомого вибору технології управління, яка враховує як особливості предметної області (страховий ринок, регуляторні вимоги, робота з персональними даними), так і специфіку розробки веборієнтованих інформаційних систем. У цьому підрозділі розглянуто теоретичні підходи до вибору технології управління відповідно до стандарту PMBOK, проаналізовано їх придатність для проєкту IBAS і обґрунтовано доцільність використання гібридної моделі.

3.1.1 Теоретичні засади вибору підходу

Стандарт PMBOK Guide виділяє три основні підходи до організації виконання робіт: предиктивний, адаптивний та гібридний [17]. Предиктивні або каскадні підходи передбачають детальне визначення вимог, обсягу, графіка,

вартості та ресурсів на ранніх фазах життєвого циклу. Подальші зміни розглядаються як відхилення від узгодженого плану, а сам план виступає основним інструментом контролю. Такі підходи доцільно застосовувати у проєктах зі стабільними вимогами, високою передбачуваністю та жорсткими регуляторними й безпековими обмеженнями.

Адаптивні підходи, до яких належать agile-методи, спираються на ітеративну та інкрементну розробку, короткі цикли виконання робіт, регулярний зворотний зв'язок зі стейкхолдерами та постійне уточнення вимог [22]. Вони ефективні в ситуаціях значної невизначеності, коли очікується часта зміна вимог, а також у випадках, коли важливо швидко постачати користувачам проміжні результати з відчутною бізнес-цінністю. Гібридні підходи поєднують елементи предиктивних і адаптивних моделей: для окремих компонентів продукту чи фаз життєвого циклу застосовується каскадна логіка, тоді як для інших — гнучкі методи на кшталт Scrum або Kanban.

PMBOK наголошує, що вибір підходу має враховувати не лише загальну філософію управління, а й характеристики продукту, проєкту та організації [17]. До таких характеристик належать ступінь інноваційності продукту, рівень визначеності та стабільності вимог, очікувана інтенсивність змін, вимоги щодо безпеки й нормативної відповідності, наявні обмеження за часом і бюджетом, рівень залученості стейкхолдерів, організаційна структура та культура, а також досвід і готовність команди до використання гнучких методів.

Сучасні емпіричні дослідження практики управління IT-проєктами показують, що на підприємствах рідко використовуються «чисті» моделі [23]. Значно поширенішими є гібридні комбінації, у яких стандартизований каскадний або регламентований процес задає рамки управління та відповідності нормативним вимогам, а на рівні команд розробки застосовуються Scrum, Kanban, XP та інші agile-підходи [24]. Подібну конфігурацію часто описують терміном Water-Scrum-Fall, підкреслюючи водоспадний характер початкових і завершальних фаз життєвого циклу та ітеративно-інкрементну розробку в його середині [25].

3.1.2 Аналіз можливих підходів для проєкту IBAS

Проєкт IBAS має низку особливостей, які безпосередньо впливають на вибір технології управління. Тривалість проєкту становить один рік, що вимагає завчасного планування бюджету, ресурсів та основних контрольних віх. На початку проєкту частина функціональних і нефункціональних вимог визначена, однак очікується їх подальше уточнення й деталізація на основі зворотного зв'язку від страхових посередників та результатів проміжних релізів. Система працюватиме в регульованій сфері надання страхових послуг, тому до неї висуваються підвищені вимоги щодо інформаційної безпеки, надійності та відповідності законодавству України й нормативам Національного банку України. Крім того, IBAS має інтегруватися з зовнішніми системами страхових компаній та платіжних сервісів, що зумовлює додаткові технічні ризики.

За таких умов застосування суто предиктивного підходу, наприклад класичної каскадної моделі, виглядає недоцільним. Для каскаду критичною є наявність повного та стабільного набору вимог на початку проєкту. У випадку IBAS це нереалістично, оскільки частина вимог щодо інтерфейсів, звітності, автоматизації бізнес-процесів брокерів та сценаріїв взаємодії з клієнтами неминуче еволюціонуватиме. Жорстка фіксація обсягу робіт на старті призвела б до значних переробок, затримок та втрати гнучкості.

Разом з тим суто адаптивний підхід, коли весь життєвий цикл організовано в рамках лише Scrum або іншої agile-методології без чітких предиктивних рамок, також не є достатнім. Проєкт має фіксований горизонт планування, бюджетні обмеження, а також повинен передбачати документально оформлені рішення щодо архітектури, безпеки й відповідності регуляторним вимогам. Ці аспекти складно повноцінно забезпечити в умовах повністю «легкого» agile-процесу без елементів традиційного управління.

Отже, з огляду на наведені фактори та рекомендації PMBOK, для проєкту IBAS доцільно застосовувати гібридну технологію управління, яка поєднує предиктивні й адаптивні елементи та дозволяє балансувати між передбачуваністю та гнучкістю.

3.1.3 Обрана технологія управління: гібридний підхід Water-Scrum-Fall

У проєкті IBAS обрано гібридну технологію управління, побудовану за моделлю Water-Scrum-Fall із використанням процесних груп та областей знань PMBOK як методологічної основи. На рівні загального управління проєктом застосовуються процеси планування інтеграції, обсягу, строків, вартості, якості, ризиків, ресурсів, комунікацій та зацікавлених сторін. Виконується моніторинг і контроль виконання, управління змінами та формальне завершення фаз і всього проєкту. Це забезпечує прозорість для керівництва організації та зовнішніх стейкхолдерів, а також дозволяє оцінювати прогрес за обсягом, строками й бюджетом.

Життєвий цикл проєкту має гібридну структуру. На етапі ініціації формується бізнес-кейс, визначаються цілі та критерії успіху, ідентифікуються ключові зацікавлені сторони й затверджується статут проєкту. Далі відбувається детальне планування та архітектурне проєктування, у межах якого будується декомпозиція робіт (WBS), оцінюється трудомісткість, розробляється цільова архітектура IBAS, формується початковий беклог продукту та релізний план. Ці фази мають переважно предиктивний характер.

Основна частина робіт припадає на фазу ітеративно-інкрементної розробки, яка реалізується за методологією Scrum. Команда працює в серії спринтів фіксованої тривалості. На початку кожного спринта уточнюються та пріоритезуються елементи беклогу, які підлягають реалізації; упродовж спринта виконуються проєктування, розробка та тестування; наприкінці спринта демонструється інкремент продукту зацікавленим сторонам, збирається зворотний зв’язок, а беклог коригується. У такий спосіб адаптивна частина життєвого циклу набуває ітеративно-інкрементного характеру: з одного боку, відбувається уточнення вимог та рішень, з іншого — послідовне нарощування функціональності системи.

Після завершення ключових інкрементів передбачено фазу переходу та впровадження. На цьому етапі завершується інтеграція модулів, проводиться

системне та приймальне тестування за участі страхових посередників, виконується перевірка відповідності вимогам безпеки й нормативним актам, здійснюється розгортання системи в промислове середовище та супроводжується період пілотної експлуатації. Завершальна фаза закриття проєкту передбачає оцінювання досягнення цілей, формалізацію висновків і уроків, а також передачу IBAS в операційний супровід. Таким чином, початок і кінець життєвого циклу мають переважно предиктивний характер, а середня фаза розробки — адаптивний, що відповідає суті моделі Water-Scrum-Fall.

3.1.4 Переваги обраної технології для проєкту IBAS

Гібридний підхід, обраний для управління проєктом IBAS, забезпечує низку важливих переваг.

По-перше, він дозволяє поєднати передбачуваність і керованість, необхідні для регульованої сфери страхування, із гнучкістю, потрібною для уточнення вимог та оптимізації користувацького досвіду. Предиктивні фази ініціації та планування задають чіткі межі щодо бюджету, строків та цільових показників, тоді як адаптивна фаза розробки дає змогу реагувати на зміну ринкових умов і запитів стейкхолдерів без втрати контролю над проєктом.

По-друге, використання Scrum у фазі розробки забезпечує орієнтацію на користувача і бізнес-цінність: інкременти системи регулярно демонструються страховим посередникам, а їхній зворотний зв’язок безпосередньо впливає на пріоритезацію подальших робіт. Це дозволяє знизити ризики створення функціональності, яка не матиме практичної цінності для користувачів, і зосередитися на тих можливостях, які реально підвищують ефективність роботи брокерів.

Нарешті, гібридний підхід сприяє системному управлінню ризиками. На стратегічному рівні ризики ідентифікуються, аналізуються та плануються заходи реагування відповідно до процесів PMBOK, тоді як на операційному рівні невизначеність додатково зменшується завдяки раннім інкрементам, швидкому виявленню проблем та регулярному перегляду пріоритетів у межах спринтів. Усе

це відповідає сучасній практиці управління IT-проєктами, де традиційні стандарти поєднуються з agile-підходами, і створює достатні передумови для успішної реалізації проєкту IBAS у визначені строки з очікуваним рівнем якості.

3.2 Декомпозиція робіт і побудова WBS

Для підвищення керованості проєкту створення інформаційної системи IBAS здійснено декомпозицію робіт та побудовано ієрархічну структуру WBS за етапами життєвого циклу [26].

На рисунку 3.1 бачимо на першому рівні WBS виділено проєкт у цілому – розробку інформаційної системи IBAS. Другий рівень відповідає основним фазам життєвого циклу: ініціація, проєктування, розробка, впровадження та завершення проєкту. Така побудова дає змогу безпосередньо пов’язати структуру робіт із прийнятим життєвим циклом та вибраною гібридною моделлю управління.

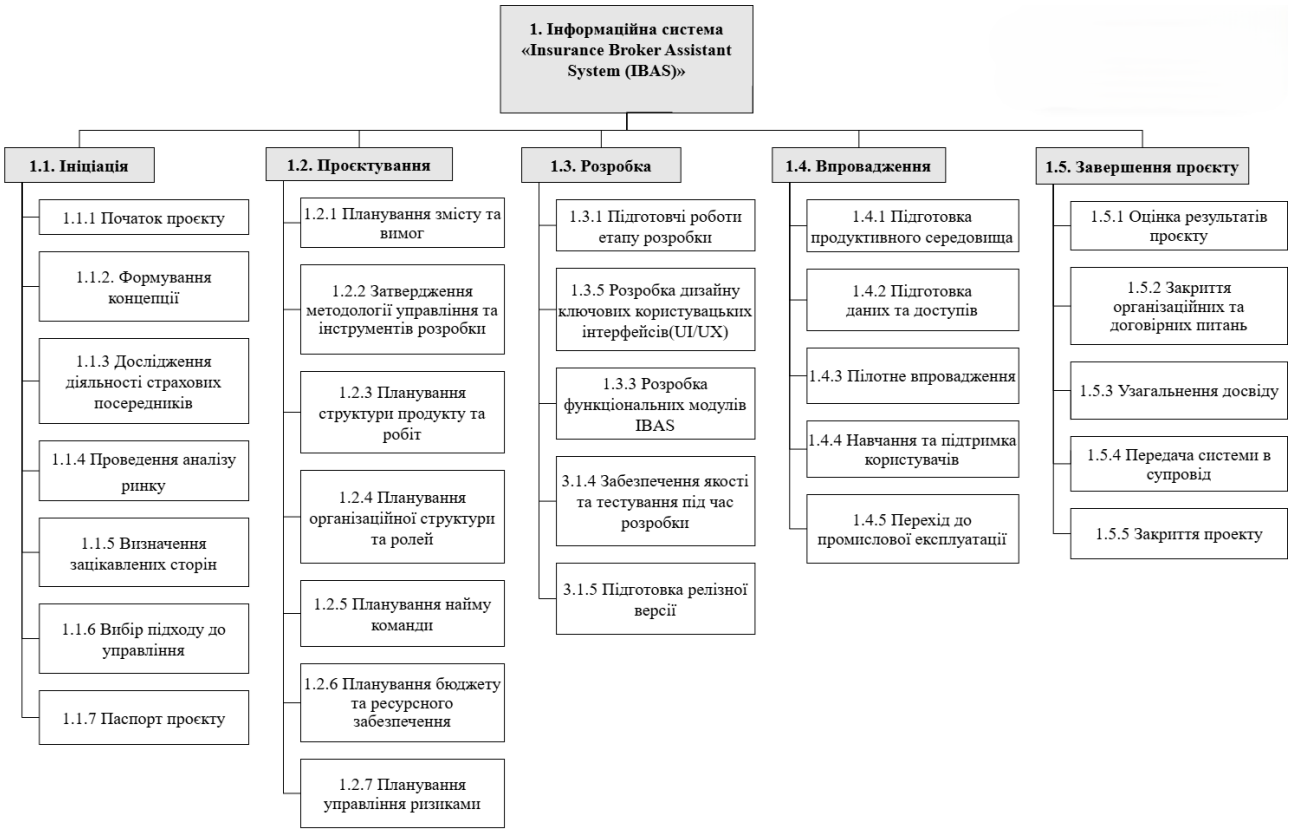


Рис. 3.1. WBS за етапами життєвого циклу

Особливістю проєкту є те, що для різних етапів застосовуються різні підходи до управління. Етапи ініціації та проєктування керуються за предиктивною моделлю, оскільки на цих стадіях доцільно послідовно виконати дослідження, аналіз, формування концепції, опис вимог, планування структури продукту, ресурсів, бюджету та ризиків. Важливо підкреслити, що система IBAS розробляється під конкретного замовника, для якого важлива заздалегідь визначеність термінів реалізації та фіксований бюджет, тому саме предиктивний підхід на етапах планування дозволяє узгодити обсяг робіт, оцінити їхню вартість і мінімізувати невизначеність. Відповідно WBS для етапів 1.1 та 1.2 деталізує ці роботи у вигляді логічних підпроцесів, що завершуються чітко визначеними результатами: паспортом проєкту, погодженими вимогами, структурою системи, організаційною структурою та комплектом планів управління проєктом.

Етап 1.3 «Розробка» реалізується за гнучкою методологією Scrum, однак у WBS він побудований за результатним принципом: виділено підготовчі роботи етапу (архітектура, база даних, інфраструктура, початковий Product Backlog), розробка дизайну інтерфейсів, розробку функціональних модулів системи IBAS, забезпечення якості та тестування під час розробки, а також підготовку релізної версії. Основною причиною такого підходу є те, що в межах Scrum неможливо наперед формально зафіксувати ані точну кількість спринтів, ані детальний перелік задач кожного спринту, оскільки планування відбувається поступово й орієнтується лише на найближчу ітерацію, інакше б це суперечило б самій методології. Натомість на рівні WBS уже відомі ключові результати, яких необхідно досягти на етапі розробки, тому роботи згруповано не за спринтами, а за логічними блоками діяльності та модулями системи. Такий підхід дозволяє одночасно зберегти гнучкість Scrum під час реалізації та забезпечити прозоре календарне й ресурсне планування, узгоджене з каскадною структурою інших етапів життєвого циклу проєкту.

Подальші етапи – впровадження (1.4) та завершення проєкту (1.5) – знову керуються за каскадною моделлю, оскільки включають переважно послідовні

процеси: підготовку продуктивного середовища, підготовку даних та доступів, пілотне впровадження, навчання користувачів, перехід до промислової експлуатації, оцінку результатів, закриття організаційних і договірних питань, узагальнення досвіду та передачу системи в супровід. Відповідна WBS відображає ці роботи як послідовні логічні блоки з чітко визначеними виходами, що дозволяє формально зафіксувати момент завершення проєкту.

Декомпонована WBS з деталізацією пакетів робіт для кожного етапу подана у Додатку Б кваліфікаційної роботи.

3.3 Розробка організаційної структури управління проєктом

Організаційна структура управління проєктом відображає склад учасників проєкту, розподіл управлінських та виконавчих ролей, а також підпорядкованість між ними. На відміну від WBS, яка показує структуру робіт і продуктів, OBS відповідає на запитання «хто за що відповідає у проєкті». Формування OBS є необхідною передумовою для подальшого розподілу ролей за елементами WBS, побудови RACI-матриці та планування завантаженості ресурсів.

Для проєкту створення інформаційної системи IBAS організаційну структуру проєктної команди побудовано у вигляді ієрархічної схеми (рис. 3.2). На верхньому рівні знаходиться керівник проєкту (Project Manager), який здійснює загальне операційне управління проєктом, а на нижніх рівнях – ключові фахівці, залучені до розробки, тестування та забезпечення інфраструктури системи.

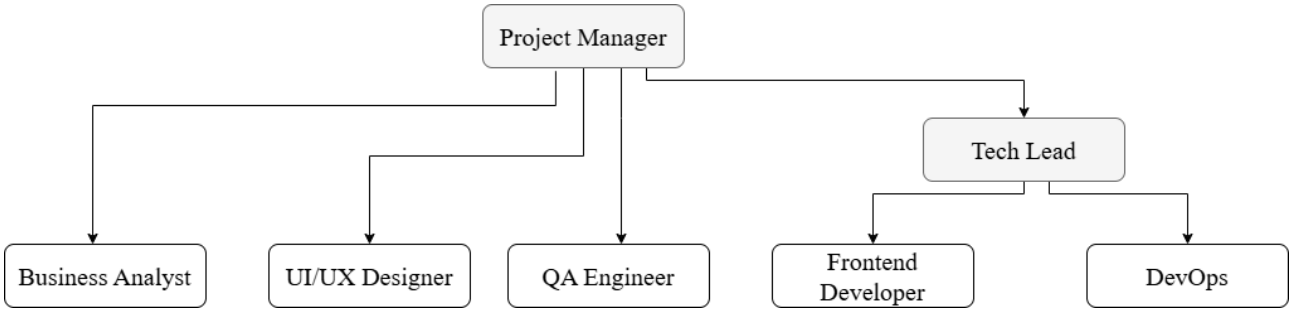


Рис. 3.2. Організаційна структура управління

У межах OBS виділено такі основні ролі:

Керівник проєкту (Project Manager) - відповідає за досягнення цілей проєкту IBAS у визначених межах строків, бюджету та обсягу робіт. До його завдань належать планування етапів і робіт, координація членів команди, управління ризиками й змінами, організація комунікацій із замовником та зацікавленими сторонами, підготовка звітності про стан проєкту. Саме керівник проєкту приймає оперативні управлінські рішення та делегує виконання завдань відповідним спеціалістам. Також він комунікує із Замовником.

Технічний лідер (Tech Lead). Підпорядковується керівнику проєкту й виконує роль ключової технічної фігури. Tech Lead відповідає за вибір технологічного стеку, архітектурні рішення, узгодження технічних підходів між усіма учасниками розробки та розробку Backend частини вебплатформи. Він контролює якість реалізації, надає технічні консультації членам команди, проводить код-рев'ю та забезпечує відповідність реалізованого рішення вимогам щодо продуктивності, безпеки та масштабованості. Через Tech Lead організовано взаємодію розробників, DevOps-фахівця та QA-інженера з точки зору технічної цілісності продукту.

Бізнес-аналітик (Business Analyst). Працює під безпосереднім керівництвом Project Manager та взаємодіє з представниками замовника. Основними функціями бізнес-аналітика є збір і формалізація вимог до системи IBAS, аналіз бізнес-процесів страхових посередників, узгодження змін із зацікавленими сторонами, підготовка специфікацій вимог і описів користувацьких сценаріїв. Аналітик забезпечує прозорий місток між бізнес-очікуваннями та технічною реалізацією, зменшуючи ризики непорозуміння між замовником і командою розробки.

UI/UX-дизайнер (UI/UX Designer). Відповідає за проєктування інтерфейсу користувача системи IBAS, розробку макетів екранів та прототипів, забезпечення зручності та інтуїтивності взаємодії користувачів із системою. Дизайнер співпрацює з бізнес-аналітиком (щодо сценаріїв використання) та

frontend-розробником (щодо реалізації дизайну), а також узгоджує ключові рішення з керівником проєкту. Наявність окремої ролі дизайнера є важливою для підвищення прийнятності системи кінцевими користувачами – страховими агентами та брокерами.

Інженер із забезпечення якості (QA Engineer). Підпорядковується як керівнику проєкту, так і технічному лідеру в частині дотримання технічних стандартів. QA-інженер розробляє стратегію тестування, готує тест-кейси, проводить функціональне, інтеграційне та регресійне тестування, фіксує й відслідковує дефекти. Він бере участь у приймальних перевірках інкрементів системи перед їх демонстрацією замовнику. Таким чином, роль QA є ключовою для забезпечення необхідного рівня якості програмного продукту.

Frontend-розробник (Frontend Developer). Реалізує клієнтську частину веб-платформи IBAS, відповідає за відтворення користувацьких інтерфейсів відповідно до вимог UI/UX-дизайнера й бізнес-аналітика. Frontend-розробник забезпечує коректну взаємодію користувацького інтерфейсу з бекенд-сервісами, бере участь в оцінюванні трудомісткості задач, усуненні виявлених дефектів та оптимізації продуктивності клієнтської частини.

DevOps-інженер (DevOps Engineer). Займається налаштуванням і підтримкою середовищ розробки, тестування та дослідної експлуатації, автоматизацією процесів розгортання (CI/CD), моніторингом роботи компонентів системи й базових показників продуктивності. DevOps-інженер забезпечує стабільність інфраструктури, а також бере участь в організації резервного копіювання й базового моніторингу безпеки. У проєкті IBAS ця роль є критичною з огляду на використання хмарних сервісів і мікросервісної архітектури.

Також важливо врахувати, що оскільки життєвий цикл проєкту IBAS побудований за гібридною моделлю, організаційна структура управління набуває додаткової специфіки на етапі розробки. Для етапів ініціалізації, проєктування, впровадження та завершення використовується передиктивний

підхід, для якого характерна чітка ієрархія підпорядкування, відображена на рис. 3.2. Натомість етап розробки реалізується за гнучкою методологією Scrum [27].

З огляду на це, для адаптивної фази розробки доцільно представити структуру команди в термінах ролей (див. рис. 3.3). У теорії Scrum всі учасники кросфункціональної команди вважаються спільно відповідальними за створення цінності для замовника, а формальні лінії підпорядкування відходять на другий план. В проєкті IBAS така логіка збережена, хоча базова ієрархія, наведена на рис. 3.2.

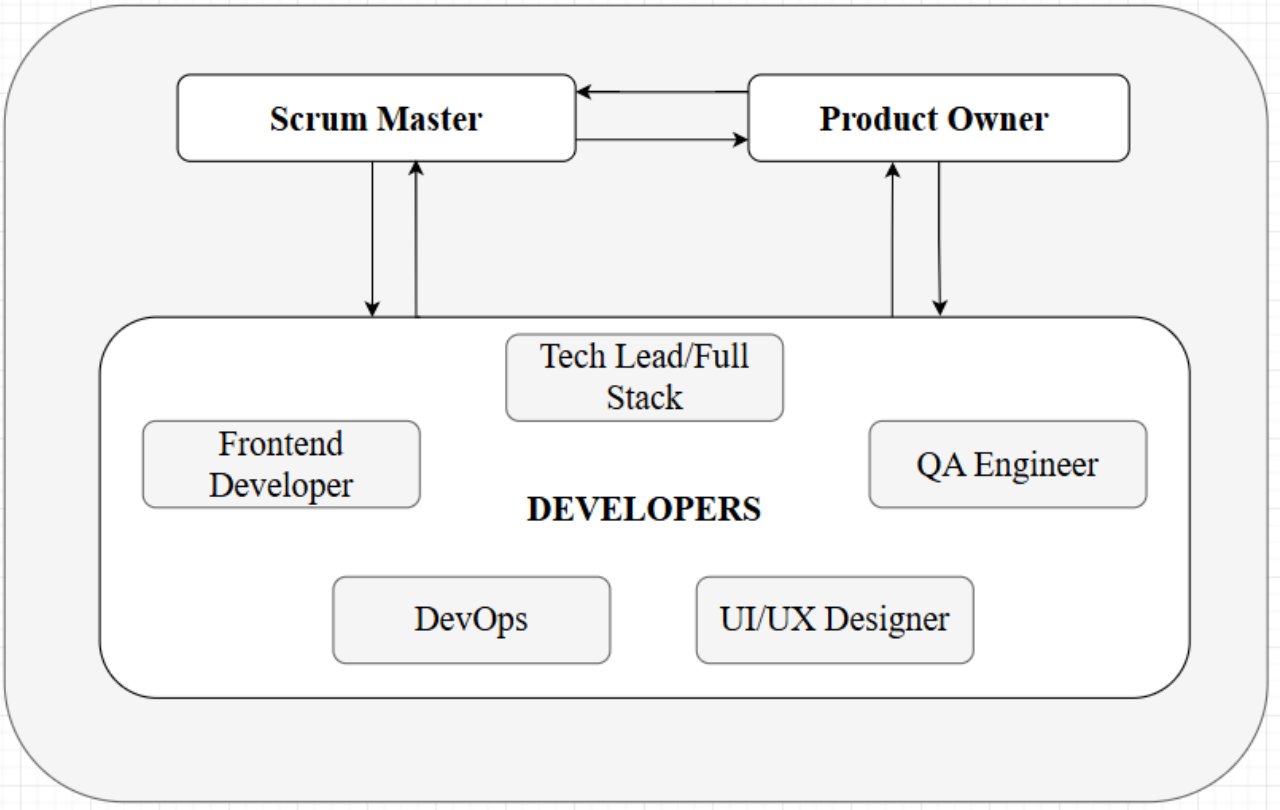


Рис. 3.3. Організаційна структура управління за методологією SCRUM

У ролі Scrum Master фактично виступає керівник проєкту (Project Manager), який відповідає за дотримання принципів і подій Scrum, усунення перешкод у роботі команди та фасилітацію комунікацій. Роль Product Owner виконує бізнес-аналітик (Business Analyst), який транслює потреби замовника у вигляді Product Backlog, пріоритизує вимоги, формує цілі спринтів і приймає результати інкрементів. Решта учасників – Tech Lead / Full Stack Developer, Frontend Developer, QA Engineer, DevOps Engineer, UI/UX Designer – утворюють єдину

кросфункціональну команду Developers, що колективно планує спринти, оцінює задачі, реалізує функціональність і забезпечує її якість.

Таким чином, у гібридній моделі проєкту IBAS одна й та сама група фахівців одночасно вбудована в класичну організаційну структуру управління та в більш «плоску» структуру Scrum-команди. Це дозволяє поєднати прозорий розподіл відповідальності й ресурсів на рівні всього проєкту з гнучкістю, швидкою адаптацією до змін і самоорганізацією команди на етапі розробки програмного продукту. Ролі Scrum Master, Product Owner та Developers у проєкті розподілено з урахуванням теоретичних положень Scrum та специфіки IBAS.

Водночас важливо підкреслити, що, незважаючи на повагу до теоретичних засад Scrum і колективну відповідальність команди за продукт, у рамках даного проєкту ключовою управлінською одиницею залишається керівник проєкту (Project Manager). Саме він несе кінцеву відповідальність за досягнення цілей проєкту, виконання робіт у межах затверджених строків і бюджету, а також за результат етапу розробки. Тобто, під час застосування гнучкої методології на рівні команди зберігається загальний проєктний контур управління, у якому Project Manager виступає основною відповідальною особою перед замовником.

3.4 Визначення функціональних та нефункціональних вимог до продукту

У цьому підрозділі уточнюються вимоги до програмного продукту IBAS, що випливають із результатів аналізу предметної області, зацікавлених сторін та бізнес-процесів страхового посередництва.

Функціональні вимоги формують перелік основних сервісів, які повинен забезпечувати веб-додаток для підтримки діяльності страхового посередника: авторизацію користувачів, управління довідниками контрагентів, роботу з договорами страхування, формування документів і звітів, а також автоматичні нагадування щодо стану договорів. У табл. 3.1 подано функціональні вимоги до продукту IBAS із зазначенням їх ідентифікаторів, типу та текстового опису очікуваної поведінки системи.

Таблиця 3.1

Функціональні вимоги до продукту

ID	Тип вимоги	Опис вимоги
1	**2**	**3**
FR001	Авторизація	Додаток має забезпечувати безпечний вхід до особистого кабінету користувача за допомогою email і паролю.
FR002	Управління користувачами	Система має підтримувати реєстрацію, редагування та видалення користувачів з різними ролями (адміністратор, агент).
FR003	Перегляд договорів	Користувач може переглядати список укладених страхових договорів із можливістю фільтрації, пошуку та сортування.
FR004	Створення договору по шаблону	Система має забезпечувати створення нового договору з введенням даних про страхувальника, продукт, компанію, суму, строки дії використовуючи створений адміністратором шаблон.
FR005	Генерація PDF-договору	Після заповнення шаблону система повинна генерувати договір у форматі PDF з можливістю автоматичного додавання цифрового підпису агента та печатки компанії.
FR006	Розрахунок премії	Після введення параметрів договору система автоматично розраховує страхову премію за допомогою вбудованого калькулятора.
FR007	Генерація документів	Система має автоматично формувати супутні документи (договір, рахунок, згоди) на основі шаблонів із підставленням введених даних.
FR008	Збереження даних	Всі введені дані про договори, клієнтів, агентів та компанії зберігаються в базі даних і доступні для подальшого редагування.
FR009	Повідомлення і нагадування	Система надсилає автоматичні повідомлення користувачам про статус договорів і нагадування про пролонгацію.
FR010	Управління шаблонами	Адміністратор має можливість додавати, редагувати та видаляти шаблони документів, адаптуючи їх під зміни законодавства та компаній.
FR011	Звіти та аналітика	Користувачі можуть отримувати звіти за укладеними договорами, активністю агентів та іншою статистикою у вигляді таблиць та Excel-файлів.
FR012	Управління страховими компаніями	Адміністратор може додавати, редагувати і видаляти інформацію про страхові компанії, з якими співпрацює брокер.
FR013	Управління клієнтами	Користувач має можливість додавати, редагувати та переглядати інформацію про клієнтів (фізичних і юридичних осіб).
FR014	Управління агентами	Система підтримує ведення інформації про агентів, їх комісії, зв’язки з брокерами та договорами.

Не менш важливими для успішної роботи системи є нефункціональні вимоги, які визначають такі аспекти, як надійність, безпека, зручність використання та продуктивність. У табл. 3.2 наведено їх основні категорії та характеристики.

Таблиця 3.2

Нефункціональні вимоги до продукту

ID	Тип вимоги	Опис вимоги
1	**2**	**3**
NFR001	Безпека	Система має забезпечувати захист персональних даних відповідно до вимог GDPR, з використанням сучасних методів шифрування.
NFR002	Надійність	Система повинна забезпечувати безперервну роботу з мінімальним часом простою (не більше 1 години на місяць).
NFR003	Масштабованість	Система має підтримувати збільшення кількості користувачів без втрати продуктивності.
NFR004	Зручність використання	Інтерфейс має бути інтуїтивно зрозумілим та адаптивним для різних пристроїв (ПК, планшет, смартфон).
NFR005	Логування	Всі дії користувачів повинні логуватися з метою аудиту та безпеки.
NFR006	Сумісність	Система має коректно працювати з популярними браузерами (Chrome, Firefox, Edge).
NFR007	Конфіденційність	Забезпечення розмежування прав доступу між різними ролями користувачів для захисту конфіденційної інформації.

Наведені нефункціональні вимоги задають рамкові обмеження для проєктування архітектури та вибору технологій реалізації системи IBAS. Серед них однією з ключових є вимога безпеки та конфіденційності (NFR001, NFR007), оскільки система працює з персональними даними клієнтів і фінансовою інформацією та має відповідати вимогам регулятора й міжнародним стандартам захисту даних.

3.5 Формування беклогу продукту

У цьому підрозділі описується процес формування беклогу продукту для системи IBAS. На основі визначених функціональних та нефункціональних вимог кожна вимога була трансформована у одну або декілька User Story,

орієнтованих на кінцевих користувачів системи: страхових агентів, адміністраторів брокера та менеджмент. Такий підхід дозволяє описати бажану поведінку продукту мовою користувацьких сценаріїв, а не лише технічних функцій.

Кожна User Story має унікальний код (US001–US015) та містить посилання на відповідну функціональну вимогу FR, формулювання у форматі «Як <роль>, я хочу <дія>, щоб <очікувана цінність>», а також набір критеріїв приймання (Acceptance Criteria). Повна таблиця User Story представлено у вигляді табл. В.1 додатку В кваліфікаційної роботи, а фрагмент для окремої User Story наведено в табл. 3.3.

Таблиця 3.3

Фрагмент User Story

Код US	**Формулювання US**	**Acceptance Criteria**
1	**2**	**3**
US001 (FR001) Авторизація	Як зареєстрований користувач (агент або адміністратор), я хочу мати можливість увійти в систему за email та паролем, щоб отримати доступ до свого робочого кабінету IBAS.	1. На сторінці входу відображаються поля «Email», «Пароль», чекбокс «Запам’ятати» та кнопка «Вхід». 2. Якщо користувач вводить некоректний email/пароль, система показує повідомлення про помилку й не пускає в особистий кабінет. 3. Після успішної авторизації користувач потрапляє на головну сторінку-дашборд. 4. Якщо встановлений чекбокс «Запам’ятати», при наступному відкритті системи email користувача підставляється автоматично. 5. Неактивні/заблоковані акаунти не можуть пройти авторизацію.
US002 (FR002) Управління користувачами	Як адміністратор брокера, я хочу створювати, редагувати та блокувати облікові записи користувачів із різними ролями, щоб керувати доступом до функцій системи.	1. Адміністратор має окремий розділ «Користувачі» з таблицею всіх існуючих акаунтів та фільтрами (ПІБ, email, роль, статус). 2. Адміністратор може створити нового користувача, вказавши ПІБ, email, роль і статус облікового запису. 3. Адміністратор може змінити роль та статус існуючого користувача (активний/заблокований). 4. Заблокований користувач не може авторизуватися в системі. 5. Всі створені/змінені користувачі відображаються у списку без перезавантаження всієї системи (оновлюється представлення даних).
...	...	...

Закінчення табл. 3.3

1	2	3
US015 (FR014) Управління агентами	Як адміністратор брокера, я хочу вести облік агентів та їхніх комісійних умов, щоб правильно розподіляти винагороду та контролювати їхню активність.	1. У розділі «Агенти» відображається список агентів із ПІБ, контактами, статусом і базовою комісійною ставкою. 2. Адміністратор може створити нового агента, вказавши необхідні реквізити та комісійну ставку/схему. 3. Адміністратор може змінити статус агента (активний/неактивний); неактивний агент не доступний для вибору в нових договорах. 4. У картці договору відображається прив'язаний агент, а також ця інформація враховується у звітах по агентам. 5. Зміни в довіднику агентів одразу впливають на доступні значення при оформленні нових договорів.

На наступному етапі User Story були декомпозовані на задачі та підзадачі рівня реалізації,що формують власне беклог продукту. Для кожної задачі визначено зміст робіт, необхідні ролі команди та залежності між елементами беклогу. Фрагмент беклогу продукту проєкту наведено в табл. 3.4. Першочерговий порядок розміщення User Story у беклозі визначався їх бізнес-цінністю для страхового посередника (авторизація, робота з договорами та клієнтами) та впливом на архітектурні рішення.

Повний результат розробки беклогу представлено у вигляді табл. Г.1 додатку Г кваліфікаційної роботи.

Таблиця 3.4

Фрагмент беклогу продукту проєкту

Код US	Формулювання US	Завдання, підзавдання
1	**2**	**3**
US001 (FR004) Створення договору із шаблону	Як страховий агент, я хочу мати можливість обрати шаблон договору зі списку, щоб швидко створити типовий договір	***T01. Аналіз шаблонів договорів*** ST01.1. Зібрати приклади шаблонів від агентів та страхових компаній. ST01.2. Уніфікувати структуру шаблонів (розділи, реквізити, змінні поля). ST01.3. Описати формат зберігання шаблонів у системі (БД / файлове сховище). ***T02. Реалізація вибору шаблону в інтерфейсі*** ST02.1. Спроєктувати UI вибору шаблону (список / модальне вікно).

Закінчення табл. 3.4

1	2	3
		ST02.2. Реалізувати завантаження обраного шаблону у форму договору. ST02.3. Забезпечити відображення лише актуальних (активних) шаблонів.
US002 (FR001) Авторизація користувачів	Як зареєстрований користувач, я хочу входити в систему за email та паролем, щоб отримати доступ до свого робочого кабінету IBAS.	***T01. Реалізація механізму авторизації*** ST01.1. Спроєктувати модель користувача та схему збереження облікових даних. ST01.2. Налаштувати механізм перевірки email/пароля (hashing, salting). ST01.3. Реалізувати обмеження доступу для неавторизованих користувачів. ***T02. Розробка форми входу*** ST02.1. Реалізувати UI сторінки логіну (Email, Пароль, «Запам'ятати»). ST02.2. Додати обробку помилок (неправильні дані, заблокований користувач). ST02.3. Налаштувати перенаправлення на дашборд після успішного входу.
...	...	...
US015 (FR014) Управління агентами	Як адміністратор брокера, я хочу вести облік агентів та їхніх комісійних умов, щоб правильно розподіляти винагороду та контролювати їхню активність.	***T01. Модель даних для агентів*** ST01.1. Визначити набір атрибутів агента (ПІБ, контакти, статус, базова комісія). ST01.2. Спроєктувати таблицю агентів та зв'язки з договорами. ***T02. Інтерфейс управління агентами*** ST02.1. Створити сторінку «Агенти» зі списком, пошуком і фільтрами. ST02.2. Реалізувати створення/редагування агента та налаштування комісійної ставки. ST02.3. Додати можливість змінювати статус агента (активний/неактивний). ST02.4. Налаштувати відображення агента у звітах та формах договорів тільки за активного статусу.

Сформований таким чином беклог продукту є базовим артефактом для подальшого планування спринтів, оцінювання трудомісткості та відстеження прогресу розробки. У процесі реалізації проєкту він може уточнюватися, доповнюватися або перепріоритезуватися залежно від зворотного зв'язку користувачів і змін зовнішніх умов.

3.6 Планування Спринту №1

Спринт №1 є першою ітерацією гнучкої частини життєвого циклу проєкту. Тривалість спринту становить два тижні (з 01.12.2025 по 14.12.2025), що відповідає рекомендаціям Scrum щодо коротких, але змістовних ітерацій.

Метою Спринту №1 є отримання початкового інкременту системи IBAS, який забезпечує можливість авторизації користувачів та базову роботу зі списками договорів і клієнтів, спираючись на єдину базу даних.

Під час Sprint Planning з беклогу продукту були відібрані найбільш пріоритетні User Story: US002 (авторизація), US004 (перегляд укладених договорів), US005 (збереження договору як чернетки), US009 (збереження даних у БД) та US014 (управління клієнтами). Кожну з обраних історій декомпозовано на задачі та підзадачі рівня реалізації, орієнтовані на окремі ролі команди: Project Manager, Business Analyst, UI/UX Designer, Tech Lead, Frontend Developer, QA Engineer, DevOps. Для кожної роботи визначено обсяг трудовитрат (у людино-годинах), відповідального виконавця та основні інструменти (Confluence, Jira, GitLab, Figma, Docker, PostgreSQL, Google Calendar, Microsoft Teams тощо). Фрагмент сформованого плану Спринту №1 наведено в табл. 3.5.

Повна таблиця інформації щодо планування Спринту 1 представлено у вигляді табл. Д.1 додатку Д кваліфікаційної роботи.

Таблиця 3.5

Фрагмент інформації щодо планування Спринту 1

№	Назва роботи	Трив. (год)	Виконавець	Матеріали, технології
1	**2**	**3**	**4**	**5**
1	Планування та координація реалізації US002, US004, US009, US014 (формування Sprint Backlog, узгодження задач з беклогу)	30	Project Manager	Confluence, Jira, Google Calendar, Microsoft Teams
...	...	...	...	...
6	Специфікація атрибутів БД і полів інтерфейсів для «Договорів» та «Клієнтів» (US004 T01, US014 T01, US009 T01)	20	Business Analyst	Confluence, Jira, Google Calendar, Microsoft Teams
...	...	...	...	...

Закінчення табл. 3.5

1	2	3	4	5
10	Проєктування форми «Створення договору із шаблону» та сценарію «Зберегти як чернетку» (US001 T02, US005 T02)	18	UI/UX Designer	Figma, Jira, Google Calendar, Microsoft Teams
...	...	...	...	...
18	Реалізація інтерфейсу «Зберегти як чернетку» та списку чернеток (US005 T02 – часткова реалізація)	10	Frontend Developer	Angular, TypeScript, HTML/CSS, GitLab, Jira, Google Calendar, Microsoft Teams
19	Реалізація сервісів БД і CRUD-операцій для користувачів, договорів, клієнтів (US009 T02, US004 T02, US014 T02)	24	Tech Lead	.NET, PostgreSQL, GitLab, Jira, Google Calendar, Microsoft Teams
20	Підготовка тест-плану та тест-кейсів для US002, US004, US014, US005	22	QA Engineer	Jira (test cases, defects), Google Calendar, Microsoft Teams
...	...	...	...	...
27	Налаштування тестового середовища (деплой через CI, змінні середовища) для інкременту спринту 1 (US002, US004, US014, US005)	12	DevOps	Docker, Docker Compose, GitLab, Jira, Google Calendar, Microsoft Teams

Запланований обсяг робіт забезпечує реалістичне завантаження членів команди (приблизно до 70 годин на кожного за спринт) та охоплює як технічні задачі (проєктування архітектури, налаштування середовищ, реалізація інтерфейсів і API), так і організаційні активності (планування, координація, тестування, оновлення документації). Такий розподіл дозволяє в межах першого спринту отримати цілісний інкремент продукту, який можна продемонструвати стейкхолдерам, а також створює основу для подальшого нарощування функціональності в наступних ітераціях.

3.7 Побудова діаграми Ганта

Для календарного планування робіт проєкту IBAS та контролю строків виконання було побудовано діаграму Ганта в середовищі Microsoft Project. Діаграма Ганта дає змогу візуалізувати послідовність і тривалість робіт,

визначити залежності між ними, виявити критичні ділянки графіка та оцінити можливості паралельного виконання завдань.

Побудова календарного плану виконувалася на основі розробленої WBS та обраної гібридної методології управління проєктом. У Microsoft Project було створено ієрархію робіт, задано тривалості, робочий календар, а також встановлено логічні зв'язки типу «кінець–початок» (FS) між роботами. Для ключових моментів застосовано віхи, що дозволяє контролювати проходження контрольних точок без штучного збільшення тривалостей.

Етап «Ініціація» стартує 01.09.2025 та завершується 30.09.2025 і включає формування концепції, дослідження діяльності страхових посередників, аналіз ринку, визначення зацікавлених сторін, вибір підходу до управління та підготовку/затвердження паспорта проєкту (див. рис. 3.4). Послідовність робіт вибудувана таким чином, щоб результати аналітики та досліджень слугували вхідними даними для формалізації цілей, підготовки паспорта та його узгодження із замовником.

1.1 Ініціація	22 днів	**01.09.25 8:00**	**30.09.25 17:00**
1.1.1 Початок проєкту	0 днів	01.09.25 8:00	01.09.25 8:00
1.1.2 Формування концепції	3 днів	**01.09.25 8:00**	**03.09.25 17:00**
1.1.2.1 Формування цілей	1 день	01.09.25 8:00	01.09.25 17:00
1.1.2.2 Постановка завдань	1 день	02.09.25 8:00	02.09.25 17:00
1.1.2.3 Затвердження місії та цілей проєкту	1 день	03.09.25 8:00	03.09.25 17:00
1.1.3 Дослідження діяльності страхових посередників	5 днів	**04.09.25 8:00**	**10.09.25 17:00**
1.1.3.1 Дослідження потреб	1 день	04.09.25 8:00	04.09.25 17:00
1.1.3.2 Аналіз поточних бізнес-процесів	1 день	05.09.25 8:00	05.09.25 17:00
1.1.3.3 Виявлення ключових проблем та вузьких місць у процесах	1 день	08.09.25 8:00	08.09.25 17:00
1.1.3.4 Побудова дерева проблем проєкту	1 день	09.09.25 8:00	09.09.25 17:00
1.1.3.5 Побудова дерева цілей на основі дерева проблем	1 день	10.09.25 8:00	10.09.25 17:00
1.1.4 Проведення аналізу ринку	4 днів	**11.09.25 8:00**	**16.09.25 17:00**
1.1.4.1 Аналіз цифрових рішень для посередників	1 день	11.09.25 8:00	11.09.25 17:00
1.1.4.2 Аналіз конкурентів	1 день	12.09.25 8:00	12.09.25 17:00
1.1.4.3 PEST-аналіз	1 день	15.09.25 8:00	15.09.25 17:00
1.1.4.4 SWOT-аналіз	1 день	16.09.25 8:00	16.09.25 17:00
1.1.5 Визначення зацікавлених сторін	3 днів	**17.09.25 8:00**	**19.09.25 17:00**
1.1.5.1 Ідентифікація внутрішніх та зовнішніх зацікавлених сторін	1 день	17.09.25 8:00	17.09.25 17:00
1.1.5.2 Аналіз впливу зацікавлених сторін	1 день	18.09.25 8:00	18.09.25 17:00
1.1.5.3 Розробка концептуальної моделі	1 день	19.09.25 8:00	19.09.25 17:00
1.1.6 Вибір підходу до управління	3 днів	**22.09.25 8:00**	**24.09.25 17:00**
1.1.6.1 Аналіз можливих підходів до управління	1 день	22.09.25 8:00	22.09.25 17:00
1.1.6.2 Вибір методології управління проєктом	1 день	23.09.25 8:00	23.09.25 17:00
1.1.6.3 Вибір та опис життєвого циклу проєкту	1 день	24.09.25 8:00	24.09.25 17:00
1.1.7 Паспорт проєкту та затвердження ініціації	4 днів	**25.09.25 8:00**	**30.09.25 17:00**
1.1.7.1 Розробка структури паспорта проєкту IBAS	1 день	25.09.25 8:00	25.09.25 17:00
1.1.7.2 Підготовка паспорта проєкту	1 день	26.09.25 8:00	26.09.25 17:00
1.1.7.3 Узгодження паспорта проєкту із замовником та ключовими стейкхолдерами	1 день	29.09.25 8:00	29.09.25 17:00
1.1.7.4 Затвердження паспорта проєкту	1 день	30.09.25 8:00	30.09.25 17:00
Завершення етапу ініціації, початок планування	0 днів	30.09.25 17:00	30.09.25 17:00

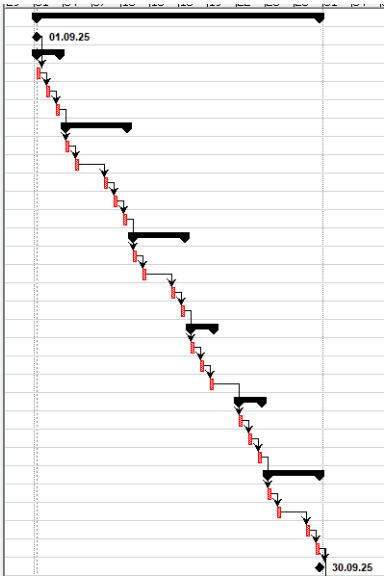
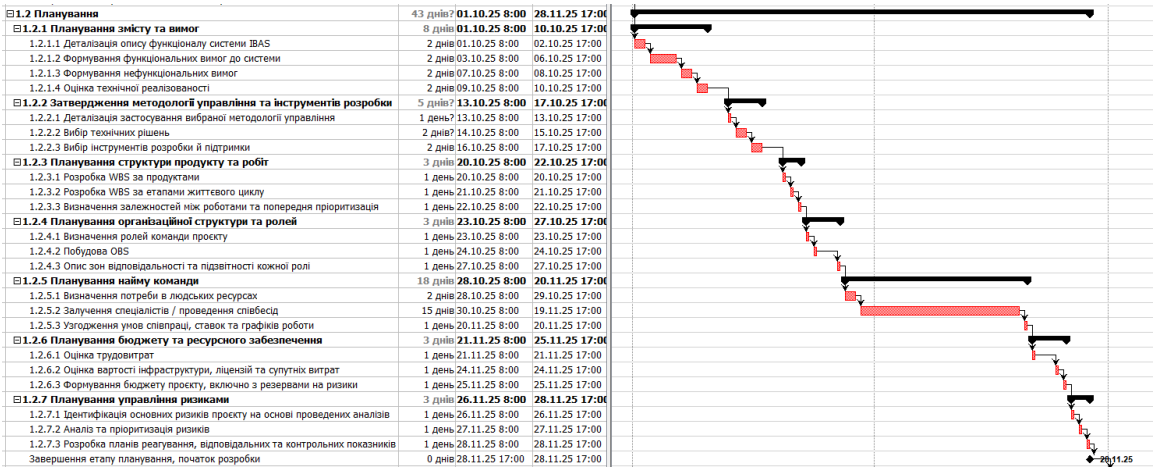


Рис. 3.4. Діаграма Ганта для етапу «Ініціація»

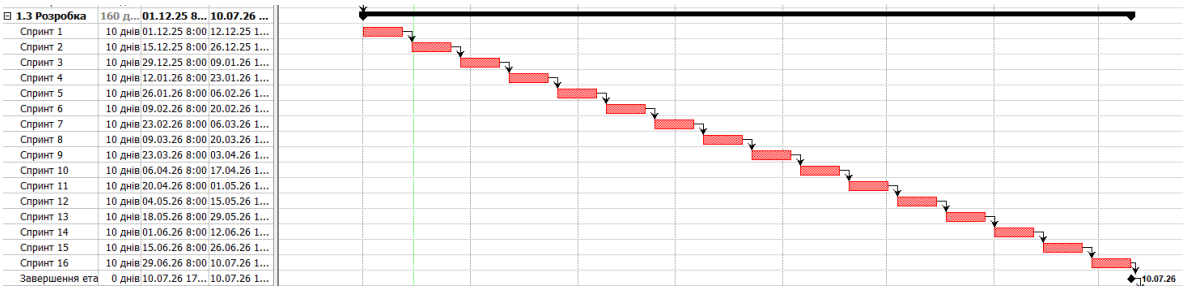
На рисунку 3.5 показано календарний план етапу 1.2 «Планування» (01.10.2025-28.11.2025). У межах цього етапу деталізуються функціональні та нефункціональні вимоги, затверджуються інструменти і технічні рішення, формуються WBS та OBS, план найму, бюджет і план управління ризиками.



1.2 Планування	43 днів?	**01.10.25 8:00**	**28.11.25 17:0**
1.2.1 Планування змісту та вимог	8 днів	**01.10.25 8:00**	**10.10.25 17:0**
1.2.1.1 Деталізація опису функціоналу системи IBAS	2 днів	01.10.25 8:00	02.10.25 17:00
1.2.1.2 Формування функціональних вимог до системи	2 днів	03.10.25 8:00	06.10.25 17:00
1.2.1.3 Формування нефункціональних вимог	2 днів	07.10.25 8:00	08.10.25 17:00
1.2.1.4 Оцінка технічної реалізованості	2 днів	09.10.25 8:00	10.10.25 17:00
1.2.2 Затвердження методології управління та інструментів розробки	5 днів?	**13.10.25 8:00**	**17.10.25 17:0**
1.2.2.1 Деталізація застосування вибраної методології управління	1 день?	13.10.25 8:00	13.10.25 17:00
1.2.2.2 Вибір технічних рішень	2 днів?	14.10.25 8:00	15.10.25 17:00
1.2.2.3 Вибір інструментів розробки й підтримки	2 днів	16.10.25 8:00	17.10.25 17:00
1.2.3 Планування структури продукту та робіт	3 днів	**20.10.25 8:00**	**22.10.25 17:0**
1.2.3.1 Розробка WBS за продуктами	1 день	20.10.25 8:00	20.10.25 17:00
1.2.3.2 Розробка WBS за етапами життєвого циклу	1 день	21.10.25 8:00	21.10.25 17:00
1.2.3.3 Визначення залежностей між роботами та попередня пріоритизація	1 день	22.10.25 8:00	22.10.25 17:00
1.2.4 Планування організаційної структури та ролей	3 днів	**23.10.25 8:00**	**27.10.25 17:0**
1.2.4.1 Визначення ролей команди проєкту	1 день	23.10.25 8:00	23.10.25 17:00
1.2.4.2 Побудова OBS	1 день	24.10.25 8:00	24.10.25 17:00
1.2.4.3 Опис зон відповідальності та підзвітності кожної ролі	1 день	27.10.25 8:00	27.10.25 17:00
1.2.5 Планування найму команди	18 днів	**28.10.25 8:00**	**20.11.25 17:0**
1.2.5.1 Визначення потреби в людських ресурсах	2 днів	28.10.25 8:00	29.10.25 17:00
1.2.5.2 Залучення спеціалістів / проведення співбесід	15 днів	30.10.25 8:00	19.11.25 17:00
1.2.5.3 Узгодження умов співпраці, ставок та графіків роботи	1 день	20.11.25 8:00	20.11.25 17:00
1.2.6 Планування бюджету та ресурсного забезпечення	3 днів	**21.11.25 8:00**	**25.11.25 17:0**
1.2.6.1 Оцінка трудовитрат	1 день	21.11.25 8:00	21.11.25 17:00
1.2.6.2 Оцінка вартості інфраструктури, ліцензій та супутніх витрат	1 день	24.11.25 8:00	24.11.25 17:00
1.2.6.3 Формування бюджету проєкту, включно з резервами на ризики	1 день	25.11.25 8:00	25.11.25 17:00
1.2.7 Планування управління ризиками	3 днів	**26.11.25 8:00**	**28.11.25 17:0**
1.2.7.1 Ідентифікація основних ризиків проєкту на основі проведених аналізів	1 день	26.11.25 8:00	26.11.25 17:00
1.2.7.2 Аналіз та пріоритизація ризиків	1 день	27.11.25 8:00	27.11.25 17:00
1.2.7.3 Розробка планів реагування, відповідальних та контрольних показників	1 день	28.11.25 8:00	28.11.25 17:00
Завершення етапу планування, початок розробки	0 днів	28.11.25 17:00	28.11.25 17:00

Рис. 3.5. Діаграма Ганта для етапу «Планування»

У межах етапу 1.3 «Розробка» проєкт реалізується за гнучкою методологією Scrum зі спринтами довжиною в 2 тиждні(див. рис. 3.6). На діаграмі Ганта цей період відображено як послідовність спринтів, що дозволяє узгодити загальні терміни проєкту з іншими етапами життєвого циклу та виконати високорівневе календарне планування.



1.3 Розробка	160 д...	**01.12.25 8...**	**10.07.26 ...**
Спринт 1	10 днів	01.12.25 8:00	12.12.25 1...
Спринт 2	10 днів	15.12.25 8:00	26.12.25 1...
Спринт 3	10 днів	29.12.25 8:00	09.01.26 1...
Спринт 4	10 днів	12.01.26 8:00	23.01.26 1...
Спринт 5	10 днів	26.01.26 8:00	06.02.26 1...
Спринт 6	10 днів	09.02.26 8:00	20.02.26 1...
Спринт 7	10 днів	23.02.26 8:00	06.03.26 1...
Спринт 8	10 днів	09.03.26 8:00	20.03.26 1...
Спринт 9	10 днів	23.03.26 8:00	03.04.26 1...
Спринт 10	10 днів	06.04.26 8:00	17.04.26 1...
Спринт 11	10 днів	20.04.26 8:00	01.05.26 1...
Спринт 12	10 днів	04.05.26 8:00	15.05.26 1...
Спринт 13	10 днів	18.05.26 8:00	29.05.26 1...
Спринт 14	10 днів	01.06.26 8:00	12.06.26 1...
Спринт 15	10 днів	15.06.26 8:00	26.06.26 1...
Спринт 16	10 днів	29.06.26 8:00	10.07.26 1...
Завершення ета	0 днів	10.07.26 17...	10.07.26 1...

Рис. 3.6. Діаграма Ганта для етапу «Розробка»

Водночас слід підкреслити, що наведена кількість спринтів є прогнозом і використовується виключно як орієнтир для планування. Згідно з принципами Scrum, детальне планування здійснюється ітеративно: команда формує Sprint Backlog під час планування спринта, виходячи з пріоритетів Product Backlog, поточної готовності вимог та фактичної швидкості [22]. Отже, у процесі виконання проєкту кількість спринтів може змінюватися (бути більшою або меншою), оскільки обсяг робіт уточнюється, а пріоритети можуть переглядатися на підставі результатів інкрементів, зворотного зв’язку та ризиків. Таким чином, діаграма Ганта для Scrum-етапу не фіксує «остаточний» план усіх спринтів, а

демонструє попередню рамку виконання робіт у межах відведеного календарного часу.

Для демонстрації практичної організації робіт у Scrum у графіку детально розкрито приклад планування Спринт 1 (див. рис. 3.7). У межах цього спринта показано, що завдання виконуються паралельно всіма ролями команди, що відображає ключову перевагу гнучкого підходу — одночасну роботу над різними компонентами інкремента.

⊟Спринт 1	10 днів	01.12.25 8:00	12.12.25 17:00	
Планування та координація реалізації US002, US004, US009, US014 (формування Sprint Backlog, узгодження	4 днів	01.12.25 8:00	04.12.25 17:00	Проєктний менеджер
Проведення подій Scrum (daily, Sprint Review, Retrospective), комунікація із замовником щодо пріоритетів U	4 днів	05.12.25 8:00	10.12.25 17:00	Проєктний менеджер
Оновлення документації спринту (опис інкременту, оновлення беклогу)	2 днів	11.12.25 8:00	12.12.25 17:00	Проєктний менеджер
Аналіз та деталізація вимог для авторизації й управління користувачами (US002 T01, US003 T01)	3 днів	01.12.25 8:00	03.12.25 17:00	Бізнес-аналітик
Аналіз шаблонів договорів і процесу роботи з чернетками (US001 T01, US005 T01)	3 днів	04.12.25 8:00	08.12.25 17:00	Бізнес-аналітик
Специфікація атрибутів БД і полів інтерфейсів для «Договорів» та «Клієнтів» (US004 T01, US014 T01, US00	4 днів	09.12.25 8:00	12.12.25 17:00	Бізнес-аналітик
Проєктування макетів сторінки авторизації (US002 T02 – форма входу, стани помилки)	1 день	01.12.25 8:00	01.12.25 17:00	UI/UX Designer
Проєктування макета списку «Укладені договори» (US004 T02 – таблиця, фільтри, пагінація)	2 днів	02.12.25 8:00	03.12.25 17:00	UI/UX Designer
Проєктування макета списку «Клієнти» та картки клієнта (US014 T02)	4 днів	04.12.25 8:00	09.12.25 17:00	UI/UX Designer
Проєктування форми «Створення договору із шаблону» та сценарію «Зберегти як чернетку» (US001 T02, US	3 днів	10.12.25 8:00	12.12.25 17:00	UI/UX Designer
Проєктування логічної моделі БД для користувачів, клієнтів та договорів (US009 T01)	2 днів	01.12.25 8:00	02.12.25 17:00	Tech Lead
Проєктування архітектури модуля авторизації та безпеки (US002 T01)	1 день	03.12.25 8:00	03.12.25 17:00	Tech Lead
Проєктування REST API для списків договорів і клієнтів, CRUD-операцій (US004 T02, US014 T02, US009 T02)	2 днів	04.12.25 8:00	05.12.25 17:00	Tech Lead
Реалізація сервісів БД і CRUD-операцій для користувачів, договорів, клієнтів (US009 T02, US004 T02, US014	2 днів	08.12.25 8:00	09.12.25 17:00	Tech Lead
Код-рев'ю основних змін бекенду та фронтенду по спринту (US002, US004, US014, US005)	2 днів	10.12.25 8:00	11.12.25 17:00	Tech Lead
Підготока середовища, ознайомлення з проєктом та документацією	3 днів	01.12.25 8:00	03.12.25 17:00	Frontend Developer
Реалізація сторінки «Авторизація»	2 днів	04.12.25 8:00	05.12.25 17:00	Frontend Developer
Реалізація сторінки «Клієнти» (US014 T02 – таблиця, пошук, перехід у картку клієнта)	2 днів	08.12.25 8:00	09.12.25 17:00	Frontend Developer
Реалізація інтерфейсу «Зберегти як чернетку» та списку чернеток (US005 T02 – часткова реалізація)	3 днів	10.12.25 8:00	12.12.25 17:00	Frontend Developer
Підготовка тест-плану та тест-кейсів для US002, US004, US014, US005	10 днів	01.12.25 8:00	12.12.25 17:00	QA Engineer
Налаштування PostgreSQL та створення тестової БД для IBAS (US009 T02)	1 день	01.12.25 8:00	01.12.25 17:00	DevOps
Налаштування репозиторіїв і CI-пайплайнів для backend і frontend (US002, US004, US014)	9 днів	02.12.25 8:00	12.12.25 17:00	DevOps
Завершення Спринт1	0 днів	12.12.25 17:00	12.12.25 17:00	

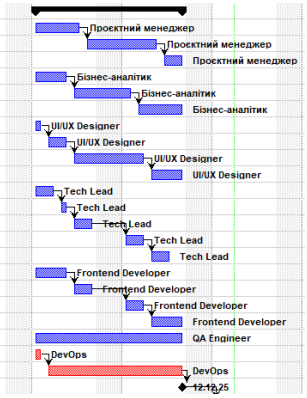
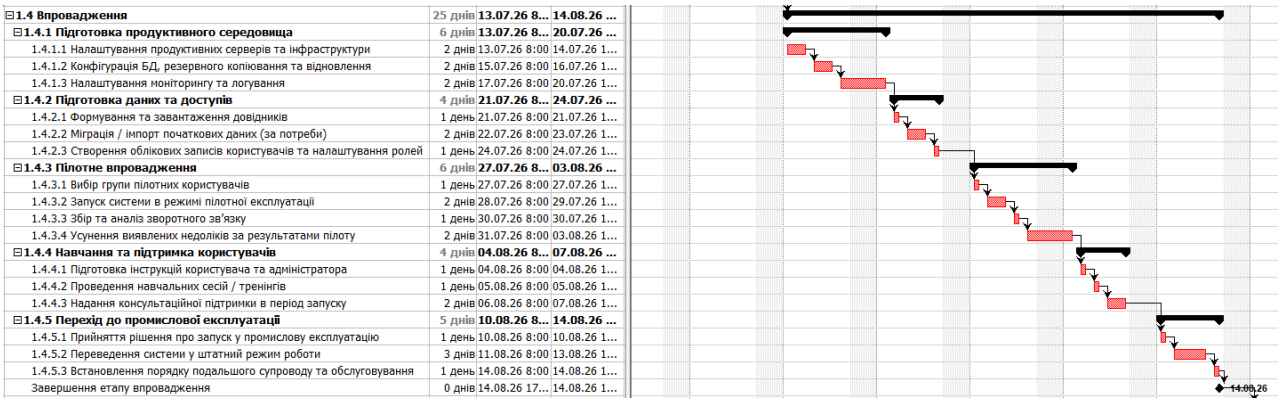


Рис. 3.7. Діаграма Ганта для Спринт 1

У Scrum завершення розробки окремого функціоналу в межах спринта не означає завершення всього циклу робіт над ним, оскільки обов'язковою складовою є перевірка якості та тестування інкремента. Тому після реалізації компонентів (наприклад, сторінки «Клієнти») у Product Backlog фіксуються задачі на тестування, які плануються до виконання в наступних спринтах відповідно до пріоритетів і місткості команди. У разі виявлення дефектів вони оформлюються як окремі backlog-елементи (bug/task) із визначеним пріоритетом, а не «повертаються» безпосередньо в поточний спринт, що допомагає зберігати стабільність Sprint Backlog. Під час наступного Sprint Planning команда відбирає найпріоритетніші виправлення та роботи, балансуючи між розвитком функціоналу, усуненням дефектів і досягненням цілей спринта. Такий підхід забезпечує прозорість, контроль змін та системне управління якістю продукту.

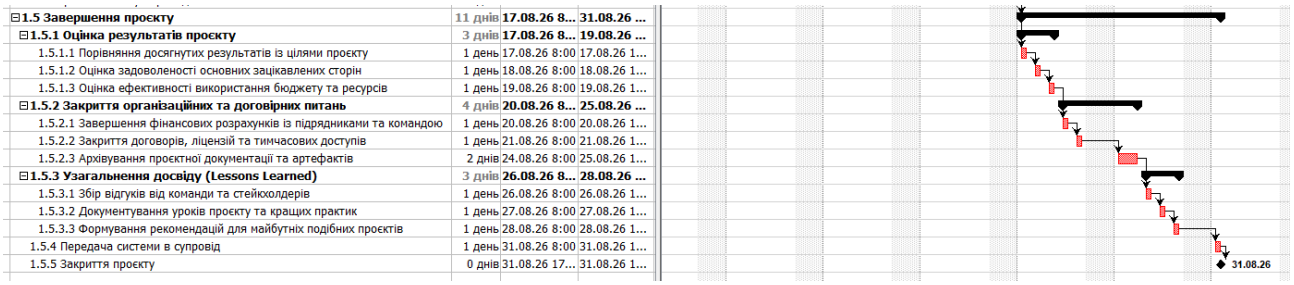
Подальший етап 1.4 «Впровадження» наведено на рисунку 3.8. Роботи на цьому етапі зосереджені на стабілізації рішення та готовності до експлуатації, тому задачі мають послідовні залежності та контрольні точки.



1.4 Впровадження	25 днів	**13.07.26 8...**	**14.08.26 ...**
1.4.1 Підготовка продуктивного середовища	6 днів	**13.07.26 8...**	**20.07.26 ...**
1.4.1.1 Налаштування продуктивних серверів та інфраструктури	2 днів	13.07.26 8:00	14.07.26 1...
1.4.1.2 Конфігурація БД, резервного копіювання та відновлення	2 днів	15.07.26 8:00	16.07.26 1...
1.4.1.3 Налаштування моніторингу та логування	2 днів	17.07.26 8:00	20.07.26 1...
1.4.2 Підготовка даних та доступів	4 днів	**21.07.26 8...**	**24.07.26 ...**
1.4.2.1 Формування та завантаження довідників	1 день	21.07.26 8:00	21.07.26 1...
1.4.2.2 Міграція / імпорт початкових даних (за потреби)	2 днів	22.07.26 8:00	23.07.26 1...
1.4.2.3 Створення облікових записів користувачів та налаштування ролей	1 день	24.07.26 8:00	24.07.26 1...
1.4.3 Пілотне впровадження	6 днів	**27.07.26 8...**	**03.08.26 ...**
1.4.3.1 Вибір групи пілотних користувачів	1 день	27.07.26 8:00	27.07.26 1...
1.4.3.2 Запуск системи в режимі пілотної експлуатації	2 днів	28.07.26 8:00	29.07.26 1...
1.4.3.3 Збір та аналіз зворотного зв'язку	1 день	30.07.26 8:00	30.07.26 1...
1.4.3.4 Усунення виявлених недоліків за результатами пілоту	2 днів	31.07.26 8:00	03.08.26 1...
1.4.4 Навчання та підтримка користувачів	4 днів	**04.08.26 8...**	**07.08.26 ...**
1.4.4.1 Підготовка інструкцій користувача та адміністратора	1 день	04.08.26 8:00	04.08.26 1...
1.4.4.2 Проведення навчальних сесій / тренінгів	1 день	05.08.26 8:00	05.08.26 1...
1.4.4.3 Надання консультаційної підтримки в період запуску	2 днів	06.08.26 8:00	07.08.26 1...
1.4.5 Перехід до промислової експлуатації	5 днів	**10.08.26 8...**	**14.08.26 ...**
1.4.5.1 Прийняття рішення про запуск у промислову експлуатацію	1 день	10.08.26 8:00	10.08.26 1...
1.4.5.2 Переведення системи у штатний режим роботи	3 днів	11.08.26 8:00	13.08.26 1...
1.4.5.3 Встановлення порядку подальшого супроводу та обслуговування	1 день	14.08.26 8:00	14.08.26 1...
Завершення етапу впровадження	0 днів	14.08.26 17...	14.08.26 1...

Рис. 3.8. Діаграма Ганта для етапу «Впровадження»

Завершальний етап 1.5 «Завершення проєкту» показаний на рисунку 3.9. Наявність цього етапу в діаграмі Ганта забезпечує повний цикл управління проєктом — від ініціації до формального закриття та передачі результатів.



1.5 Завершення проєкту	11 днів	**17.08.26 8...**	**31.08.26 ...**
1.5.1 Оцінка результатів проєкту	3 днів	**17.08.26 8...**	**19.08.26 ...**
1.5.1.1 Порівняння досягнутих результатів із цілями проєкту	1 день	17.08.26 8:00	17.08.26 1...
1.5.1.2 Оцінка задоволеності основних зацікавлених сторін	1 день	18.08.26 8:00	18.08.26 1...
1.5.1.3 Оцінка ефективності використання бюджету та ресурсів	1 день	19.08.26 8:00	19.08.26 1...
1.5.2 Закриття організаційних та договірних питань	4 днів	**20.08.26 8...**	**25.08.26 ...**
1.5.2.1 Завершення фінансових розрахунків із підрядниками та командою	1 день	20.08.26 8:00	20.08.26 1...
1.5.2.2 Закриття договорів, ліцензій та тимчасових доступів	1 день	21.08.26 8:00	21.08.26 1...
1.5.2.3 Архівування проєктної документації та артефактів	2 днів	24.08.26 8:00	25.08.26 1...
1.5.3 Узагальнення досвіду (Lessons Learned)	3 днів	**26.08.26 8...**	**28.08.26 ...**
1.5.3.1 Збір відгуків від команди та стейкхолдерів	1 день	26.08.26 8:00	26.08.26 1...
1.5.3.2 Документування уроків проєкту та кращих практик	1 день	27.08.26 8:00	27.08.26 1...
1.5.3.3 Формування рекомендацій для майбутніх подібних проєктів	1 день	28.08.26 8:00	28.08.26 1...
1.5.4 Передача системи в супровід	1 день	31.08.26 8:00	31.08.26 1...
1.5.5 Закриття проєкту	0 днів	31.08.26 17...	31.08.26 1...

Рис. 3.9. Діаграма Ганта для етапу «Завершення проєкту»

Побудована діаграма Ганта забезпечує цілісне бачення календарного плану проєкту IBAS та слугує базовим інструментом контролю строків виконання робіт.

3.8 Планування ресурсного забезпечення та бюджету

Планування ресурсного забезпечення та бюджету проєкту IBAS здійснюється відповідно до рекомендацій стандарту PMBOK, який передбачає послідовне визначення потреби в людських, матеріально-технічних та фінансових ресурсах, а також формування базового плану вартості[17].

Враховується тривалість життєвого циклу проєкту 12 місяців – з 1 вересня 2025 року по 31 серпня 2026 року – та використання гібридної моделі управління. На етапі планування визначається склад команди, рівень завантаженості кожної ролі, перелік необхідних сервісів та інфраструктури для дистанційної роботи, після чого виконуються кількісні розрахунки витрат і формується бюджет проєкту.

Планування людських ресурсів базується на сформованій структурі команди, до якої входять: проєктний менеджер / Scrum Master, бізнес-аналітик / Product Owner, Tech Lead/Full Stack, Frontend-розробник, інженер із забезпечення якості, UI/UX-дизайнер та фахівець з DevOps (див. табл. 3.6). Для кожної ролі визначено тривалість участі у проєкті та орієнтовний відсоток завантаженості, що дозволяє оцінити сумарний обсяг робіт, необхідний для досягнення цілей проєкту.

Таблиця 3.6

Людські ресурси

Роль	Кількість осіб	Орієнтовна завантаженість	Тривалість участі
Проєктний менеджер/Scrum Master	1	100%	12 місяців
Бізнес-аналітик/Product Owner	1	50%	8 місяців
Tech Lead/Full Stack	1	100%	10 місяців
Frontend Developer	1	100%	8 місяців
QA Engineer	1	100%	9 місяців
UI/UX Designer	1	50%	8 місяців
DevOps	1	100%	2 місяці

Проєктний менеджер залучений на повну зайнятість протягом усього життєвого циклу проєкту, оскільки відповідає за оперативне та стратегічне управління, координацію робіт, управління ризиками й змінами, а також за комунікації із замовником. Бізнес-аналітик працює з частковим навантаженням, фокусуючись на етапах уточнення вимог, формування та підтримки беклогу продукту, а також приймання інкрементів системи. Технічний лід, Frontend-розробник та QA-інженер забезпечують розробку та тестування

функціональності протягом більшої частини проєкту. UI/UX-дизайнер і DevOps-фахівець залучаються переважно у періоди активного проєктування інтерфейсів та підготовки інфраструктури до релізу, що відображено відповідним скороченим часом їх участі.

Кількісна оцінка витрат на людські ресурси виконувалася виходячи з припущення про 8-годинний робочий день, п’ятиденний робочий тиждень та умовну повну зайнятість 168 годин на місяць (див. табл. 3.7). Для кожної ролі було розраховано місячну кількість людино-годин залежно від відсотка завантаженості, далі ці значення множилися на тривалість участі в місяцях і погодинну ставку. У результаті отримано детальний розподіл вартості робіт за весь період реалізації IBAS. Загальні витрати на людські ресурси становлять 3 486 000,00 грн, що є основною статтею бюджету проєкту.

Таблиця 3.7

Витрати на людські ресурси

Роль	Кількість осіб	Год/міс	Тривалість (місяців)	Ставка (грн./год)	Вартість (грн.)
Проєктний менеджер/Scrum Master	1	168	12 місяців	400,00	806 400,00
Бізнес-аналітик/Product Owner	1	84	8 місяців	400,00	268 800,00
Tech Lead/Full Stack	1	168	10 місяців	500,00	840 000,00
Frontend Developer	1	168	8 місяців	450,00	604 800,00
QA Engineer	1	168	9 місяців	350,00	529 200,00
UI/UX Designer	1	84	8 місяців	400,00	268 800,00
DevOps	1	168	2 місяці	500,00	168 000,00
Загальна сума	-	-	-	-	3 486 000,00

Окремої уваги потребує планування матеріально-технічного забезпечення, яке у цьому проєкті враховує специфіку дистанційної роботи команди. Усі члени команди IBAS використовують власні ноутбуки, гарнітури та домашнє підключення до Інтернету, що входить до базової інфраструктури організації-виконавця і не закладається до бюджету окремого проєкту. Таким чином, у

кошторис включаються лише ті програмні засоби й хмарні сервіси, які потребують окремої оплати в межах цього проєкту.

Для підтримки процесів управління проєктом та розробки IBAS використовується поєднання платних і безкоштовних програмних продуктів. До платних віднесено ліцензії Jira Software (для ведення беклогу, планування спринтів і управління задачами за Scrum), Confluence (для ведення технічної та користувацької документації), а також хмарний хостинг (віртуальний сервер, на якому розгортається веб-застосунок, база даних та механізми резервного копіювання), статичну публічну IP-адресу, доменне ім’я, SSL-сертифікат для захищеного доступу через HTTPS і e-mail/SMS-шлюз для надсилання сповіщень користувачам системи (див. табл. 3.8).

Таблиця 3.8

Матеріальні ресурси та сервіси

Ресурс / сервіс	**Кількість (ліцензій / одиниць)**	**Період використання, міс.**	**Орієнтовна вартість, грн**
Jira Software (управління Scrum)	7 користувачів	12	≈ 28 000,00
Confluence (проєктна документація)	7 користувачів	12	≈ 22 000,00
Хмарний хостинг (app + БД + backup)	1 віртуальний сервер	12	≈ 50 000,00
Статична публічна IP-адреса	1 IP	12	≈ 2 000,00
Доменне ім’я	1 домен	12	≈ 1 000,00
SSL-сертифікат для HTTPS	1 сертифікат	12	≈ 1 000,00
E-mail/SMS-шлюз для сповіщень	1 сервіс	12	≈ 10 000,00
Загальна сума	-	-	≈ 114 000,00

Паралельно активно використовуються програмні засоби, які не створюють додаткових прямих витрат для бюджету проєкту, оскільки застосовуються в межах безкоштовних тарифів або вже наявних корпоративних ліцензій. До них належать: GitLab (репозиторій вихідного коду), безкоштовна версія Figma для проєктування інтерфейсів, Microsoft Teams для проведення

онлайн-зустрічей і щоденних зізвонів команди, а також безкоштовні середовища розробки та утиліти (наприклад, Visual Studio, браузери для тестування, Postman для API-перевірок). Ці інструменти відіграють важливу роль у забезпеченні продуктивної віддаленої роботи команди, але не потребують окремого фінансового резерву в рамках проєкту.

На основі узгодженого плану використання людських та матеріальних ресурсів сформовано базовий план вартості проєкту IBAS і розраховано загальний бюджет. Сума прямих витрат на персонал і інфраструктуру становить 3 600 000,00 грн, з яких переважна частка припадає на оплату праці команди розробки й супроводу. Відповідно до підходів PMBOK, для підвищення стійкості проєкту до ризиків доцільно передбачити бюджетні резерви (див. табл. 3.9). У даній роботі резерв на ймовірні обставини (contingency reserve), що покриває ідентифіковані ризики та заплановані заходи реагування, встановлено на рівні 10 % від прямих витрат, що становить 360 000,00 грн. Додатково передбачено управлінський резерв (management reserve) у розмірі 5 % від прямих витрат – 178 700,00 грн, призначений для реагування на непередбачені події та зміни, які виходять за межі початково визначеного обсягу робіт. З урахуванням цих резервів підсумковий бюджет проєкту IBAS становить 4 140 000,00 грн, що в роботі інтерпретується як орієнтовно 4,1 млн грн.

Таблиця 3.9

Бюджет проєкту IBAS

Стаття витрат	Короткий опис	Сума, грн
Людські ресурси	Оплата праці команди (PM, BA/PO, TL, FE, QA, UI/UX, DevOps)	3 486 000,00
Платні матеріальні ресурси та сервіси	Jira, Confluence, хмарний хостинг, статична IP, домен, SSL, e-mail/SMS	114 000,00
Разом прямі витрати проєкту	Сума витрат на персонал та інфраструктуру	3 600 000,00
Резерв на ймовірні обставини (10%)	Покриття ідентифікованих ризиків та планових заходів реагування	360 000,00
Управлінський резерв (5%)	На непередбачені події поза межами початкового обсягу робіт	180 000,00
Підсумковий бюджет проєкту	Прямі витрати + резерви	4 140 000,00

Така структура бюджету узгоджується з рекомендаціями PMBOK щодо формування базового плану вартості та резервів і забезпечує достатню фінансову гнучкість під час реалізації проєкту у віддаленому форматі.

3.9 Ідентифікація ризиків і розробка карти управління ними

Ідентифікація ризиків у проєктах розробки інформаційних систем ґрунтується на положеннях стандартів з управління проєктами та ризиками, зокрема стандарту PMBOK та стандарту ISO 31000, а також рекомендаціях сучасних дослідників у сфері IT-проєктів [28]. На цьому етапі ризик розглядається як невизначена подія або умова, настання якої може негативно вплинути на строки, вартість, обсяг або якість результатів проєкту. Ідентифікація виконується за допомогою аналізу документації проєкту, експертних опитувань членів команди та представників замовника, аналізу попереднього досвіду реалізації подібних рішень, а також структуризації потенційних загроз за категоріями. Результатом є реєстр ризиків, який надалі використовується для їх оцінювання, планування заходів реагування та моніторингу.

Ризики згруповано за однорідними типами, що відображають як технологічні, так і організаційні особливості проєкту (див. табл. 3.10). До технічних віднесено ризики, пов’язані зі складністю мікросервісної архітектури, продуктивністю системи, інтеграціями з прикладними інтерфейсами страхових компаній, надійністю резервного копіювання та міграцією даних. Інфраструктурні та операційні ризики стосуються налаштування хмарного середовища, кластеру Kubernetes, конвеєрів безперервної інтеграції і доставки та можливих збоїв хостингу. Окремо виділені ризики інформаційної безпеки, регуляторні ризики, пов’язані зі змінами вимог Національного банку України, якісні ризики, що характеризують зручність інтерфейсу та коректність бізнес-логіки, а також ризики, пов’язані із замовником, управлінські, кадрові, ризики Scrum-процесу, комунікаційні, фінансові, календарні та зовнішні. Для кожного ризику визначено якісну силу впливу на проєкт та рівень керованості, що дозволяє в подальшому сформувати карту управління ризиками та визначити

пріоритетні загрози, які потребують першочергових превентивних і коригувальних дій.

У табл. 3.10 подано фрагмент ідентифікації ризиків проєкту створення інформаційної системи для страхових посередників IBAS. Повна таблиця ідентифікації ризиків проекту представлена у вигляді табл. Е.1 додатку Е кваліфікаційної роботи.

Таблиця 3.10

Фрагмент ідентифікації ризиків проекту

Код ризику	**Тип ризику**	**Ризикова подія**	**Сила впливу**	**Керованість**
1	**2**	**3**	**4**	**5**
P01	Технічні	Обраний мікросервісний підхід виявляється надто складним для підтримки невеликою командою	висока	середня
P02	Технічні	Проблеми з продуктивністю системи при одночасній роботі значної кількості агентів і брокерів	середня	висока
...	...	...	...	...
P06	Інфраструктурні	Затримка налаштування Kubernetes-кластера та CI/CD-процесів для розгортання IBAS	середня	середня
P07	Операційні	Збій хостингу, що призводить до простою системи	висока	низька
P08	Інформаційна безпека	Несанкціонований доступ або витік персональних та комерційних даних користувачів IBAS	висока	середня
P09	Регуляторні	Зміна нормативно-правових вимог НБУ до діяльності страхових посередників та обліку договорів після реалізації основного функціоналу	висока	низька
P10	Якісні	Низька зручність та інтуїтивність UI/UX інтерфейсів для агентів і брокерів, що знижує готовність користуватися IBAS	середня	висока
...	...	...	...	...
P14	Пов’язані із замовником	Пасивність замовника у прийнятті рішень, затримка погодження ключових артефактів проєкту	середня	середня

Закінчення табл. 3.10

1	2	3	4	5
Р15	Управлінські	Затримка затвердження паспорта проєкту, бюджету та плану робіт на етапі ініціації	середня	середня
...	...	...	...	...
Р20	Кадрові	Ускладнення з наймом потрібних фахівців на ринку праці (тривалий пошук Frontend / QA / DevOps)	середня	низька
Р21	Scrum-процес	Перевантаження команди задачами у спринтах, що призводить до вигорання та падіння продуктивності	середня	висока
Р22	Scrum-процес	Неефективне планування беклогу та оцінювання задач, що призводить до систематичного невиконання плану спринтів	середня	висока
...	...	...	...	...
Р26	Фінансові	Перевищення вартості пострелізної підтримки	середня	середня
Р27	Календарні	Відставання від 12-місячного графіка реалізації проєкту, зрив ключових контрольних точок	висока	середня
Р28	Зовнішні	Масовані обстріли	висока	низька
Р29	Зовнішні	Мобілізація, ВЛК	висока	низька
Р30	Зовнішні	Відключення електроенергії, що впливають на доступність команди та середовищ	висока	низька

Наступним кроком є оцінка ризиків проєкту, вона здійснювалася з використанням розширеної якісної шкали. Для кожної ризикової події експертно визначались чотири показники: можливі затримки у часі, орієнтовні фінансові витрати, ймовірність настання та очікувана частота прояву протягом життєвого циклу проєкту. Спочатку показникам надавали якісні оцінки за розширеною шкалою НН–ВВ і переводили у квазикількісні бали від 1 до 9 [29]. Інтегральний показник важливості ризику розраховувався як добуток квазикількісної оцінки фінансових витрат на квазикількісну оцінку ймовірності, що дозволило ранжувати ризики за пріоритетністю та визначити ті, що потребують першочергових заходів реагування.

У табл. 3.11 наведено результати оцінювання ризиків проєкту IBAS за описаною методикою. Аналіз отриманих результатів показує, що більшість ризиків належать до середнього або підвищеного рівня важливості, що є очікуваним для складного IT-проєкту зі значною часткою невизначеності. Найвищі значення важливості, рівні 54, отримали три ризики: часті зміни пріоритетів замовника щодо функціоналу системи, пасивність замовника під час ухвалення рішень і погодження ключових артефактів, а також перевантаження команди задачами у спринтах. У всіх трьох випадках поєднуються високі фінансові наслідки у вигляді додаткових трудовитрат та можливих переробок, висока ймовірність настання подій та частий характер їх прояву. Фактично саме взаємодія із замовником та організація роботи команди формують ядро ризикового поля проєкту і можуть критично впливати на строки, бюджет і якість результатів.

Таблиця 3.11

Результати оцінювання ризиків проєкту

№	Ризикова подія	Затримки у часі		Фінансові витрати		Ймовірність		Частота		Важливість
		ЯО	КО	ЯО	КО	ЯО	КО	ЯО	КО	
1	2	3	4	5	6	7	8	9	10	11
1	Обраний мікросервісний підхід складний	ВС	8	СВ	6	СН	4	НВ	3	24
2	Проблеми з продуктивністю	СВ	6	СС	5	СВ	6	СН	4	30
3	Затримка інтеграції API	ВС	8	ВС	8	СВ	6	СН	4	48
4	Втрата/пошкодження даних	ВВ	9	ВВ	9	СС	5	НС	2	45
5	Помилки міграції даних	СВ	6	СС	5	ВН	7	СН	4	35
6	Затримка Kubernetes / CI/CD	ВС	8	СС	5	СС	5	НВ	3	25
7	Збій хостингу	СС	5	СН	4	СВ	6	СВ	6	24
8	Витік персональних даних	СВ	9	СВ	9	СН	4	НН	1	36
9	Зміна вимог НБУ	СВ	6	СВ	6	СС	5	СН	4	30
10	Проблеми UI/UX	СС	5	СС	5	ВН	7	СВ	6	35
11	Помилки в преміях/договорах	СС	5	СН	4	СВ	6	СН	4	24

Закінчення табл. 3.11

1	2	3	4	5	6	7	8	9	10	11
12	Часті зміни пріоритетів	ВВ	9	СВ	6	ВВ	9	ВН	7	54
13	Відсутність вихідних даних	СВ	6	СН	4	ВВ	9	ВН	7	36
14	Пасивність замовника	ВВ	9	СВ	6	ВВ	9	ВН	7	54
15	Затримка затвердження документації	ВВ	9	СН	4	СВ	6	СН	4	24
16	Низький рівень комунікації	СС	5	СН	4	ВВ	9	ВВ	9	36
17	Недостатній досвід РМ	СС	5	СС	5	СВ	6	СВ	6	30
18	Вихід ключового фахівця	ВВ	9	ВВ	9	СН	4	НВ	3	36
19	Недостатня предметна компетентність	СВ	6	СС	5	СВ	6	СВ	6	30
20	Складність найму	ВВ	9	ВН	7	СВ	6	СН	4	42
21	Перевантаження команди	ВВ	9	СВ	6	ВВ	9	ВН	7	54
22	Погане планування беклогу	ВВ	9	СС	5	ВС	8	ВВ	9	40
23	Конфлікти в команді	СВ	6	СН	4	СВ	6	СС	5	24
24	Перевищення бюджету	СС	5	ВВ	9	СС	5	СН	4	45
25	Невчасне фінансування	ВВ	9	СС	5	СН	4	НС	2	20
26	Перевищення вартості підтримки	НВ	3	ВН	7	СС	5	СН	4	35
27	Відставання від графіка	СВ	6	ВВ	9	СС	5	СН	4	45
28	Масовані обстріли	ВВ	9	СВ	6	ВН	7	НН	1	42
29	Мобілізація, ВЛК	ВВ	9	СС	5	СС	5	СН	4	25
30	Відключення електроенергії	СВ	6	СН	4	ВВ	9	ВН	7	36

Важливе місце в картині ризиків посідають зовнішні чинники воєнного часу: масовані обстріли, мобілізація членів команди, відключення електроенергії. Їх фінансові наслідки та потенційний вплив на доступність середовищ оцінені як високі, проте через обмежену керованість вони не перевищують за важливістю внутрішні організаційні та комунікаційні ризики.

Наступним етапом є розробка карти управління ризиками та конкретних протиризикових заходів. Фрагмент розробки протиризикових заходів подано в табл. 3.12. Повна таблиця розробки протиризикових заходів проєкту представлена у вигляді табл. Ж.1 додатку Ж кваліфікаційної роботи.

Для кожної ризикової події визначено комплекс протиризикових заходів. Для кожного ризику виділено ранній симптом, що сигналізує про наближення проблеми, а також три групи дій: превентивні заходи, дії у разі появи симптомів та дії у разі фактичної реалізації ризику. Такий підхід дозволяє перейти від суто аналітичного опису ризиків до практичної системи їх проактивного менеджменту, де заздалегідь визначені як механізми профілактики, так і сценарії реагування у кризових ситуаціях.

Таблиця 3.12

Фрагмент розробки протиризикових заходів проєкту

№	Ризикова подія	ПРЗ 1	Симптом (рання ознака)	ПРЗ 2	ПРЗ 3
		Профілактика		Дії при симптомі	Дії при проблемі
1	2	3	4	5	6
1	Обраний мікросервісний підхід складний	Архітектурний аудит, спрощення сервісів, стандартизація стеку, документація; навчання команди	Зростає час розробки й релізів, багато блокерів через інтеграції між сервісами	Перегляд архітектури, декомпозиція/об'єднання сервісів, залучення архітектора-консультанта	Поетапний рефакторинг у бік більш монолітного/модульного рішення, замороження нових фіч до стабілізації
2	Проблеми з продуктивністю	Нефункціональні вимоги, проєктування під навантаження, регулярні performance-тести, моніторинг	Падіння швидкодії у пікові години, збільшення часу відгуку, скарги користувачів	Оптимізація найкритичніших запитів, масштабування ресурсів, додатковий performance-аналіз	Запуск тимчасового масштабування (вертикального/горизонтального), введення обмежень на навантаження, позаплановий реліз виправлень

Для технічних та інфраструктурних ризиків зроблено акцент на формалізованих інженерних практиках. Для ризику складності мікросервісної архітектури передбачено проведення архітектурного аудиту, стандартизацію стеку технологій, належну документацію та навчання команди, а у разі появи симптомів, таких як зростання часу розробки та кількості блокерів, заплановано перегляд архітектури і поетапний рефакторинг у бік більш керованого рішення.

Важливими ризиками є ті, що пов’язані із замовником, управлінням проєктом, командною взаємодією та кадровими аспектами. Для частих змін пріоритетів і пасивності замовника передбачено формалізацію процесу управління змінами, фіксацію спринтів і бачення продукту, регулярні статус-зустрічі та протоколи рішень, а на рівні реагування – введення чітких правил зміни вже погоджених задач, ескалація до стейкхолдерів вищого рівня, перегляд релізних планів і трикутника обсяг-час-вартість. Для ризиків перевантаження команди, низького рівня комунікації, конфліктів у команді та недостатнього досвіду керівника проєкту заплановано регулярні зустрічі, ретроспективи, уточнення ролей та відповідальностей, залучення фасилітаторів і менторів.

Розробка карти управління ризиками та протиризикових заходів забезпечила перехід від декларативного опису загроз до практичного інструментарію їх проактивного менеджменту. Для пріоритетних ризиків визначено ранні симптоми, превентивні дії, сценарії реагування у разі появи ознак реалізації та дії у випадку настання події, а також відповідальних осіб.

РОЗДІЛ 4. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

У цьому розділі представлено практичну реалізацію інформаційної системи IBAS, що була спроєктована на попередніх етапах роботи. Спершу розглядаються технічні рішення, які визначають архітектуру програмного продукту, інструменти розробки та засоби забезпечення надійності й безпеки. Далі подано концептуальну, логічну та фізичну моделі бази даних, що формують основу для зберігання та обробки інформації в межах системи. Завершальними елементами розділу є опис алгоритму роботи застосунку та побудова інтерфейсу веб-додатку, які забезпечують коректність бізнес-логіки та зручність користувацької взаємодії.

4.1. Технічні рішення та інструменти розробки

У цьому підрозділі описано вибір основних технологій і сервісів для побудови системи IBAS, а також обґрунтування їхнього застосування.

4.1.1 Архітектурний підхід: мікросервісна програма

У процесі розробки інформаційної системи IBAS було обрано мікросервісний архітектурний підхід, що передбачає розподіл функціональності застосунку на невеликі, ізольовані та незалежно розгортані сервіси [30]. На рисунку 4.1 представлено прототип дизайну архітектури, який демонструє логічну структуру компонентів, взаємодію між ними та загальні принципи побудови системи. Центральним елементом архітектури є компонент Back for Frontend, що виконує роль проміжного шару між інтерфейсами користувача та мікросервісами. Такий підхід дозволяє спростити логіку фронтенду, уніфікувати точки входу та забезпечити гнучке масштабування. Крім того, саме в BFF зосереджено механізми авторизації та перевірки доступу до бізнес-функцій системи. Користувацькі інтерфейси згруповані за основними напрямами діяльності страхового посередника: інтерфейси управління ролями й користувачами, шаблонами документів, договорами, звітністю та клієнтськими

даними. Кожний із цих модулів взаємодіє з BFF, який передає запити до відповідних мікросервісів.

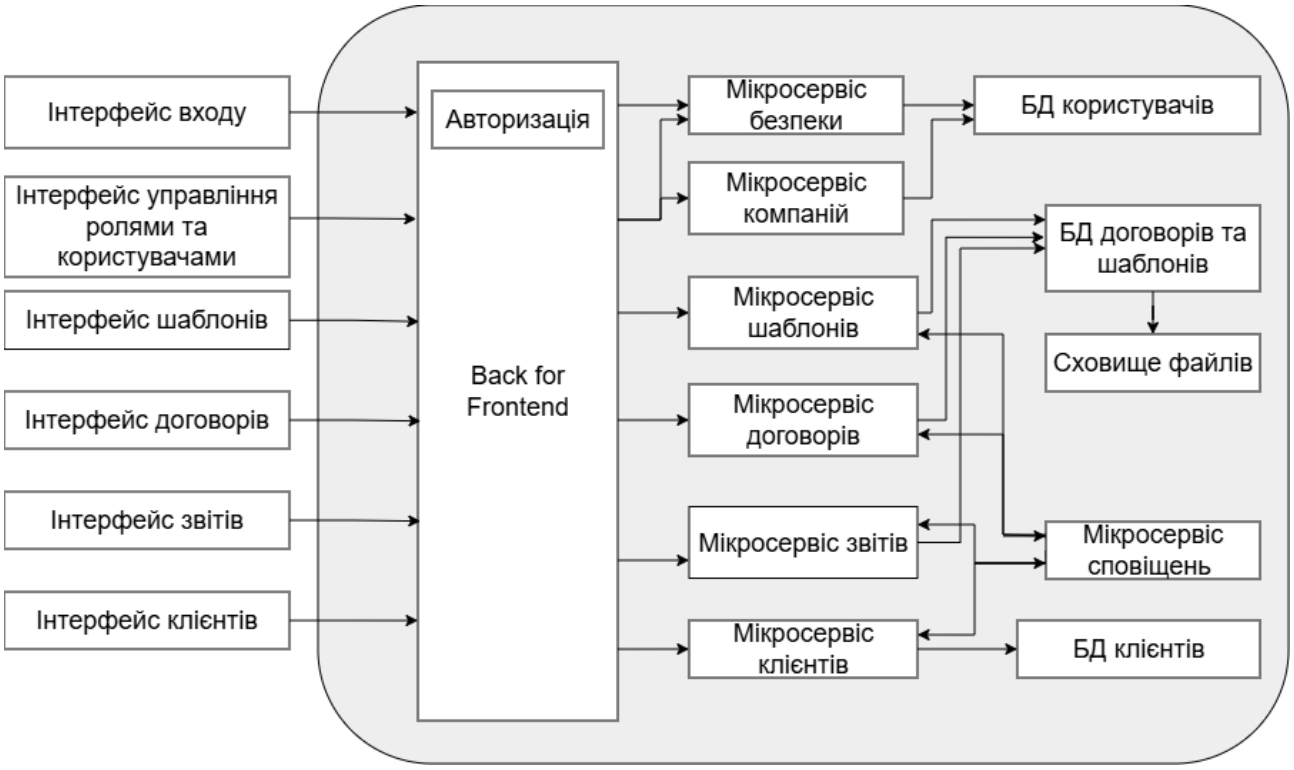


Рис. 4.1. Прототип дизайну архітектури застосунку

З правої частини рисунка представлено набір мікросервісів, кожен з яких відповідає за окремий домен:

- мікросервіс безпеки забезпечує валідацію токенів, управління користувачами та ролями;
- мікросервіс компаній опрацьовує інформацію про страхові компанії;
- мікросервіс шаблонів відповідає за збереження та рендеринг шаблонів договорів;
- мікросервіс договорів реалізує бізнес-логіку створення, зміни та пролонгації страхових договорів;
- мікросервіс звітів генерує управлінські, фінансові та операційні звіти;
- мікросервіс клієнтів забезпечує роботу з клієнтськими профілями та історією договорів;

- мікросервіс сповіщень відповідає за формування e-mail та інших типів повідомлень, зокрема нагадувань щодо строків дії договорів.

Кожен сервіс має власне сховище даних, що відповідає принципам незалежності та слабкого зв'язування. Використання окремих реляційних баз даних , в даному випадку PostgreSQL, забезпечує ізоляцію даних та можливість масштабувати сервіси автономно. Наприклад, мікросервіс користувачів має окрему БД, мікросервіс договорів – власну БД шаблонів і договорів, а мікросервіс клієнтів – свою БД клієнтів. Для збереження статичних документів передбачено окреме файлове сховище, яке використовується сервісами шаблонів і договорів.

Комунікація між сервісами організована через внутрішню мережу середовища контейнеризації, що дозволяє забезпечити високу відмовостійкість, автоматичний перезапуск сервісів, балансування навантаження та гнучке масштабування кожного компонента відповідно до потреб. На рисунку умовно показано логічні з'єднання між сервісами та їхніми базами даних, які гарантують, що кожен компонент працює автономно та виконує лише свою доменну функцію.

Таким чином, мікросервісна архітектура дозволяє розробляти систему IBAS як масштабований, модульний та легко підтримуваний застосунок. Поділ на окремі сервіси спрощує оновлення функціональності, дає можливість розгортати та тестувати компоненти незалежно, а також забезпечує високу гнучкість у розвитку продукту відповідно до потреб страхового ринку.

4.1.2 Клієнтська частина: Angular SPA

Клієнтська частина інформаційної системи IBAS реалізується у форматі single-page application (SPA) на основі фреймворку Angular з використанням TypeScript [31; 32]. Такий підхід забезпечує високу швидкодію інтерфейсу, чітке розділення компонентів за функціональними модулями та можливість динамічного оновлення даних без перезавантаження сторінки, що є критично важливим для щоденної роботи страхового посередника.

Архітектура фронтенду побудована за модульним принципом: для окремих доменів — договори, клієнти, шаблони документів, звіти, управління ролями та користувачами — створено власні Angular-модулі з набором компонентів, сервісів і маршрутів. Це дозволяє масштабувати інтерфейс поступово, не порушуючи загальної структури застосунку, а також спрощує підтримку та розширення функціоналу.

Маршрутизація в Angular забезпечує швидке перемикання між основними розділами системи. Користувач може виконувати такі операції, як перегляд картки клієнта, створення нового договору або формування звіту, без необхідності повного перезавантаження сторінки. Така модель роботи особливо зручна під час оформлення страхового договору: після введення параметрів страхування SPA автоматично відправляє запит до REST API відповідного мікросервісу, отримує розраховану страхову премію та відображає її у реальному часі [33].

Для роботи з даними застосовується стандартний Angular HttpClient, а інтерцептори реалізують централізовану обробку ключових процесів:

- автоматичне додавання JWT-токена до кожного запиту;
- перенаправлення на сторінку авторизації у разі отримання помилок 401/403;
- уніфіковану обробку серверних повідомлень про помилки.

Такий підхід підвищує безпеку клієнтської частини та зменшує дублювання коду у компонентах.

4.1.3 Реляційна база даних PostgreSQL

Як СУБД для IBAS обрано PostgreSQL [34]. Вибір зумовлений її надійністю, повною підтримкою транзакцій ACID, розвиненими засобами індексації та можливістю роботи з типом даних JSONB для гнучкого зберігання конфігурацій (наприклад, плейсхолдерів шаблонів або специфікацій тарифів). Це дозволяє поєднати класичну реляційну модель із потребою зберігати напіструкуровані дані.

Структура таблиць розроблена відповідно до логічної моделі даних: для первинних ключів використовуються UUID-ідентифікатори, для фінансових показників – тип NUMERIC, для часових міток – TIMESTAMP WITH TIME ZONE. Схему БД супроводжують за допомогою міграцій (інструменти Flyway/Liquibase чи ORM-міграції), що дає змогу відстежувати версії структури та відтворювати її на різних середовищах. Налаштовані ролі й привілеї дотримуються принципу найменших повноважень: кожен мікросервіс має доступ лише до власних схем та операцій, що підвищує безпеку й цілісність даних.

4.1.4 CI/CD та DevOps-процеси

Для проєкту IBAS передбачається використання практик CI/CD з автоматизацією збірки, тестування та розгортання [35]. Будь-яка зміна в репозиторії (push або pull request) запускає пайплайн, який виконує:

- збірку застосунку та залежностей;
- запуск юніт- і за потреби інтеграційних тестів;
- створення Docker-образів сервісів і публікацію їх у реєстрі контейнерів;
- автоматичний деплой у тестове середовище.

Для розгортання у production може використовуватися стратегія blue-green або canary-deployments, що дає змогу оновлювати систему з мінімальними простоями та швидким відкатом у разі збоїв. Конфігурації для середовищ dev / staging / prod зберігаються окремо (файли конфігурацій, змінні середовища, сховища секретів), що спрощує керування налаштуваннями. У CI/CD-процес інтегруються лінтери, статичний аналіз коду та сканування вразливостей контейнерів, що підвищує якість і безпеку релізів.

4.1.5 Моніторинг, логування та безпека

Моніторинг стану й продуктивності сервісів IBAS планується реалізувати за допомогою систем збору метрик і логів (наприклад, стек Prometheus + Grafana

для метрик та централізоване логування на базі Elasticsearch / Loki). Це дозволить відстежувати час відповіді, кількість помилок, навантаження на ресурси, будувати дашборди та налаштовувати сповіщення про аномалії чи перевищення порогових значень.

Логи запитів, помилок і критичних подій з мікросервісів будуть збиратися в єдиний журнал, що спрощує діагностику інцидентів і аналіз поведінки системи.

Безпека забезпечується на кількох рівнях:

- шифрування трафіку між компонентами SSL/TLS [36];
- чіткі політики доступу до БД та сервісів (ролі, права, принцип найменших привілеїв);
- аудит операцій через спеціальні таблиці audit_logs та тригери;
- за потреби – шифрування чутливих полів на рівні стовпців.

Асинхронні задачі (масові розсилки сповіщень, складні розрахунки) виконуються через черги повідомлень або брокер (наприклад, RabbitMQ чи інший message broker), що підвищує надійність, розвантажує основні сервіси та запобігає блокуванню користувацьких операцій.

4.2. Моделі бази даних

Для системи IBAS, у якій обробка договорів, клієнтів, посередників та фінансових операцій є критично важливою побудова послідовних моделей бази даних забезпечує прозоре та кероване формування інформаційної архітектури.

З точки зору управління проєктом, поетапне формування моделей даних має низку переваг. Концептуальна модель дозволяє узгодити бачення предметної області між замовником, аналітиками та командою розробки, мінімізуючи ризики неправильного трактування бізнес-правил. Логічна модель структурує інформацію відповідно до вимог нормалізації та майбутніх сценаріїв використання, що спрощує планування обсягів даних, продуктивності та навантаження на систему. Фізична модель, у свою чергу, трансформує ці рішення у конкретні технічні реалізації, що дозволяє точніше оцінити

інфраструктурні ресурси, масштабованість, витрати на підтримку та потенційні ризики експлуатації.

4.2.1 Концептуальна модель бази даних

На рис. 4.2 зображено концептуальну модель бази даних, вона описує сутності предметної області та їх взаємозв'язки на абстрактному рівні, без деталей реалізації. У ній ми визначаємо, що існує в системі, що зберігається, і як ці об'єкти логічно пов'язані.

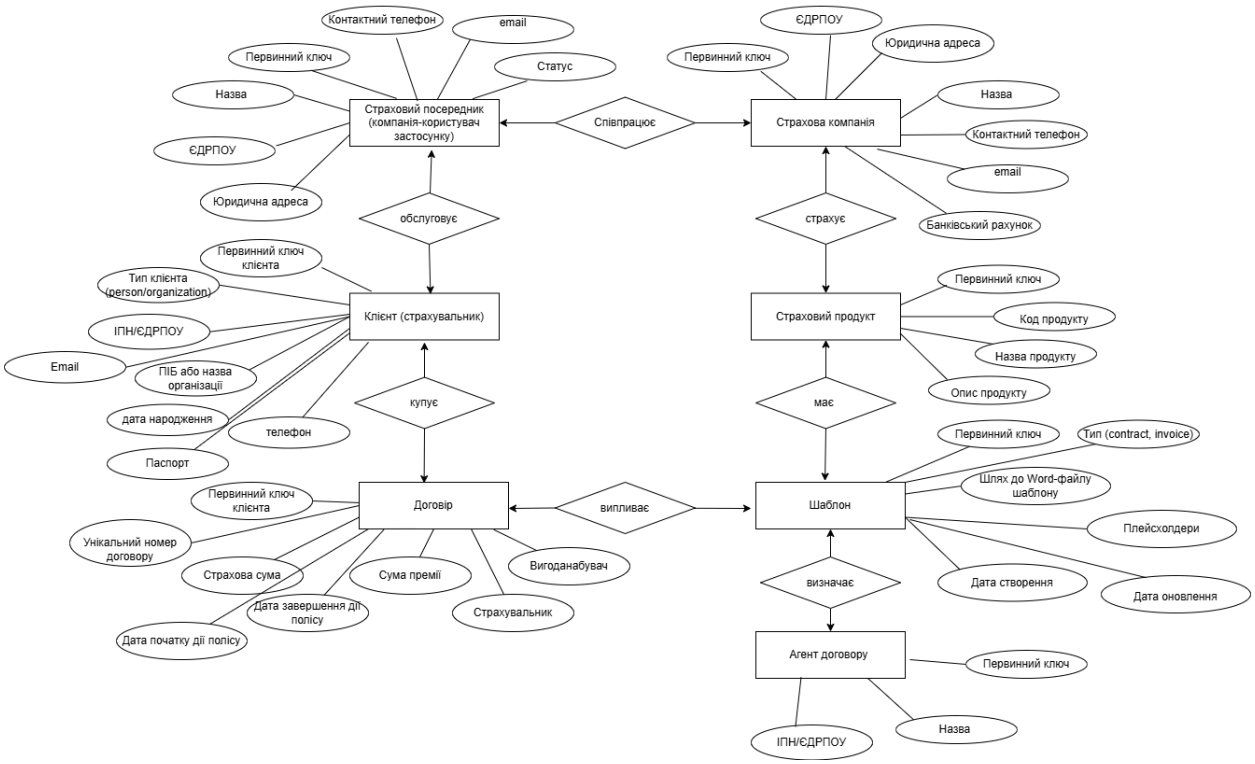


Рис. 4.2. Концептуальна модель бази даних

Наведемо опис кожної сутності із переліком ключових атрибутів та пояснення зв'язків між сутностями.

1. Страховий посередник

Призначення: представляє брокерську компанію або страхового посередника, що користується системою.

Атрибути:

- Первинний ключ (унікальний ідентифікатор посередника).
- Назва компанії.

- Реєстраційний код (ЄДРПОУ).
- Юридична адреса.
- Контактний телефон.
- Email.
- Статус (наприклад, active/inactive) або інші ознаки активності/налаштування.

Зв’язки**:**

- Співпрацює зі страховими компаніями (InsuranceCompany). Це M:N-зв’язок: один брокер може співпрацювати з багатьма страховиками, і одна страхова компанія може співпрацювати з багатьма брокерами. У концептуальній моделі це відображено як ромб «співпрацює» між «Страховий посередник» і «Страхова компанія».
- Обслуговує клієнтів (InsuranceParticipant). Один брокер може мати багато клієнтів, яких він обслуговує.
- Укладає договори через своїх агентів (Agent) – у концептуальній діаграмі це можна розуміти як участь «Агент договору» у моделі брокера.

2. Страхова компанія (InsuranceCompany)

Призначення: представляє страхову компанію, що надає страхові продукти.

Атрибути:

- Первинний ключ (унікальний ідентифікатор страховика).
- Назва компанії.
- Реєстраційний код (ЄДРПОУ).
- Юридична адреса.
- Контактний телефон.
- Email.
- Банківський рахунок (якщо фіксується у концептуальній моделі).
- Статус співпраці (active/inactive) або інші загальні ознаки.

Зв’язки:

- Співпрацює з брокерами (через відповідний зв’язок M:N «співпрацює»).
- Страхує клієнтів (через продукти і договори): у концептуальній моделі очевидно, що страхова компанія бере участь у договорі як сторона, яка страхує.
- Має страхові продукти (InsuranceProduct). Зв’язок 1:N: кожна страхова компанія пропонує один або декілька продуктів, а продукт належить конкретній страхові.

3. Клієнт (страхувальник) (InsuranceParticipant)

Призначення: представляє особу або організацію, яка укладає договір страхування.

Атрибути:

- Первинний ключ (унікальний ідентифікатор клієнта).
- Тип клієнта (person / organization) – вказує, чи це фізична особа, чи юридична.
- ІПН/ЄДРПОУ або інший ідентифікаційний код.
- Для фізичної особи: ПІБ, дата народження, паспортні дані (якщо зазначено на концептуальному рівні).
- Для юридичної особи: назва організації, реквізити.
- Email, телефон, адреса (контактні дані).
- Статус облікового запису клієнта (active/inactive) – якщо моделюється.

Зв’язки:

- Обслуговується брокером (Broker) – один клієнт може обслуговуватися певним брокером або декількома (через M:N).
- Купує страхові продукти у формі договорів: у концептуальній діаграмі роль «купує» пов’язує «Клієнт» та «Страховий продукт» через проміжну сутність «Договір».

4. Страховий продукт (InsuranceProduct)

Призначення: описує конкретний страховий продукт/програму, яку пропонує страхова компанія.

Атрибути:

- Первинний ключ (унікальний ідентифікатор продукту).
- Код продукту (внутрішній ідентифікатор у межах страхової).
- Назва продукту.
- Опис продукту (умови, особливості).

Зв’язки:

- Має страхова компанія – зв’язок 1:N: продукт належить одній страховій.
- Купується клієнтом через договір: «Клієнт купує Страховий продукт» відображається через проміжну сутність «Договір».
- За необхідності, зв’язок із шаблонами документів, які використовуються для цього продукту (частково може розглядатися у концептуальній моделі).

5. Шаблон документа (Template)

Призначення: містить опис шаблону для документів, що генеруються під час укладання договору (наприклад, Word-файл договору, рахунок, форма згоди).

Атрибути:

- Первинний ключ (унікальний ідентифікатор шаблону).
- Тип шаблону (наприклад, contract, invoice, consent).
- Шлях до Word-файлу шаблону або його умовне позначення на концептуальному рівні.
- Плейсхолдери (перелік полів, які потрібно підставляти) – вказані концептуально як атрибут «Плейсхолдери».
- Дата створення шаблону.
- Дата останнього оновлення.
- Інші ознаки (версія, статус активності) можуть бути відображені концептуально.

Зв’язки:

- Визначає/використовується для «Договору»: договір генерується на основі певного шаблону. У концептуальній діаграмі «Шаблон» пов’язаний з «Договір» через відношення «визначає» або подібне.
- Може бути зв’язок із «Страховий продукт» або «Страхова компанія», якщо потрібне уточнення, що шаблон належить конкретному продукту чи страховій. На концептуальній діаграмі це приховане, але загальна ідея: шаблон прив’язаний до умов конкретного продукту/страхової.

6. Договір (Contract)

Призначення: представляє укладений договір страхування між клієнтом і страховою компанією, ініційований через брокера/агента.

Атрибути:

- Первинний ключ (унікальний ідентифікатор договору).
- Унікальний номер договору.
- Страхова сума (Sum Insured).
- Сума премії (PremiumAmount).
- Дата початку дії полісу (StartDate).
- Дата закінчення дії полісу (EndDate).
- Статус договору (draft, active, expired) можна моделювати концептуально.

Зв’язки:

- Випливає із (або «генерується на основі») шаблону (Template): кожен договір створюється через конкретний шаблон, отже зв’язок із «Шаблон».
- Стосунки з клієнтом: «Клієнт купує Страховий продукт» реалізується через «Договір»; тобто в діаграмі є відношення між «Клієнтом» і «Договором» (часто через асоціативні таблиці у фізичній реалізації, але на концептуальному рівні це відображається як прямий зв’язок «Клієнт» ↔ «Договір»).
- Зв’язок зі страховою компанією: договір укладається з певною страховою (страхова страхує клієнта). У концептуальній моделі це може бути

позначено через відношення «страхує» між «Страхова компанія» та «Договір» або через проміжне «Страховий продукт».

- Зв’язок зі страхуванням продуктом: договір стосується конкретного «Страхового продукту»; у концептуальній діаграмі клієнт купує продукт через договір.
- Зв’язок з агентом: «Агент договору» бере участь у створенні та оформленні договору. У концептуальній діаграмі є сутність «Агент договору», пов’язана з «Договором».
- Вигодонабувач: можуть бути окремі учасники договору, які отримують виплати за полісом. У концептуальній моделі це може бути позначено через відношення «Вигодонабувач» ↔ «Договір» або як атрибут зв’язку клієнта з договором. На діаграмі вказано «Вигодонабувач» поряд із «Страхувальник».

7. Агент договору (фактично юр. або фіз особа)

Призначення: представляє страхового агента (фізичну чи юридичну особу), через якого брокер укладає договір з клієнтом.

Атрибути:

- Первинний ключ (унікальний ідентифікатор агента).
- ІПН/ЄДРПОУ або інший ідентифікатор.
- Назва (ПІБ для фізособи або назва організації).
- Контактні дані (телефон, email) можуть бути зазначені концептуально.

Зв’язки:

- Укладає або бере участь у «Договорі», отримує комісійну винагороду.

4.2.2 Логічна (даталогічна) модель бази даних

На рисунку 4.3 показано логічну (даталогічну) модель бази даних, яка відображає перетворення концептуальних сутностей у конкретні таблиці, їх атрибути та зв’язки між ними у реляційному вигляді. Логічна модель слугує

проміжним етапом між концептуальною схемою та фізичною реалізацією в СУБД.

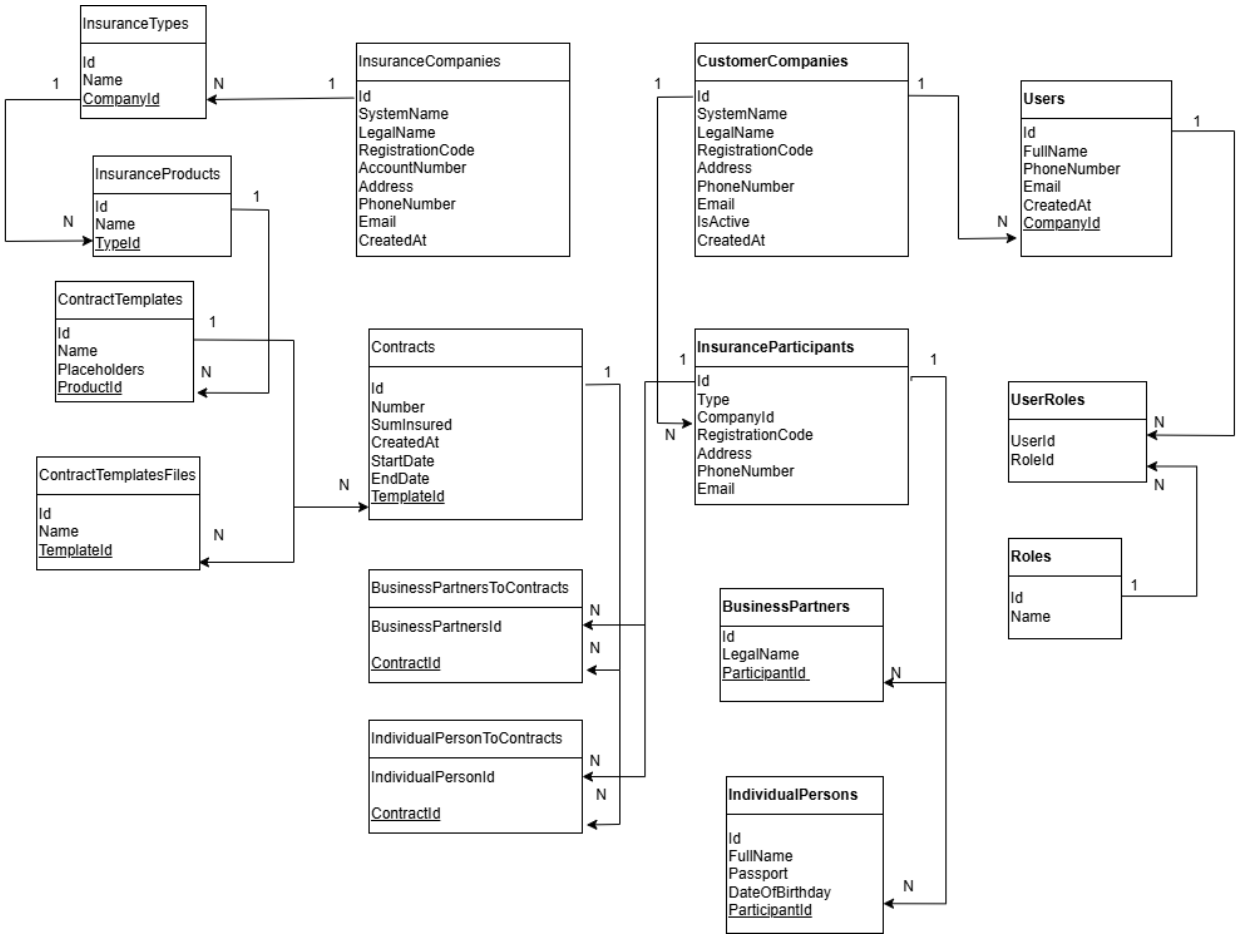


Рис. 4.3. Логічна (даталогічна) модель бази даних

На рисунку 4.3 показана логічна модель бази даних, що відображає основні таблиці та зв’язки між ними. Таблиця InsuranceCompanies містить інформацію про страхові компанії, а CustomerCompanies — про компанії-клієнти, які пов’язані зі своїми користувачами через таблицю Users.

Таблиця InsuranceParticipants зберігає дані про учасників договорів різних типів (страхувальники, вигодонабувачі, агенти) із прив’язкою до посередників. Страхові продукти та їх типи представлені таблицями InsuranceProducts та InsuranceTypes, що пов’язані зі страховиками.

Для шаблонів договорів використані таблиці ContractTemplates і ContractTemplatesFiles, які зберігають шаблони та відповідні файли. Основні дані

про договори зберігаються в таблиці Contracts, що містить інформацію про номер, строки дії, суму страхування та посилання на шаблон.

Проміжні таблиці BusinessPartnersToContracts і IndividualPersonToContracts реалізують зв’язки багато-до-багатьох між партнерами, фізичними особами та договорами. Таблиці BusinessPartners і IndividualPersons пов’язані з InsuranceParticipants, що дозволяє підтримувати єдину структуру учасників.

Модель забезпечує ефективне зберігання і управління даними, необхідними для роботи системи укладання страхових договорів.

4.2.3 Фізична моделі бази даних

На рис. 4.4 схематичне зображення фізичної моделі бази даних. Вона базується на раніше затвердженій логічній моделі і доповнюється конкретними типами даних.

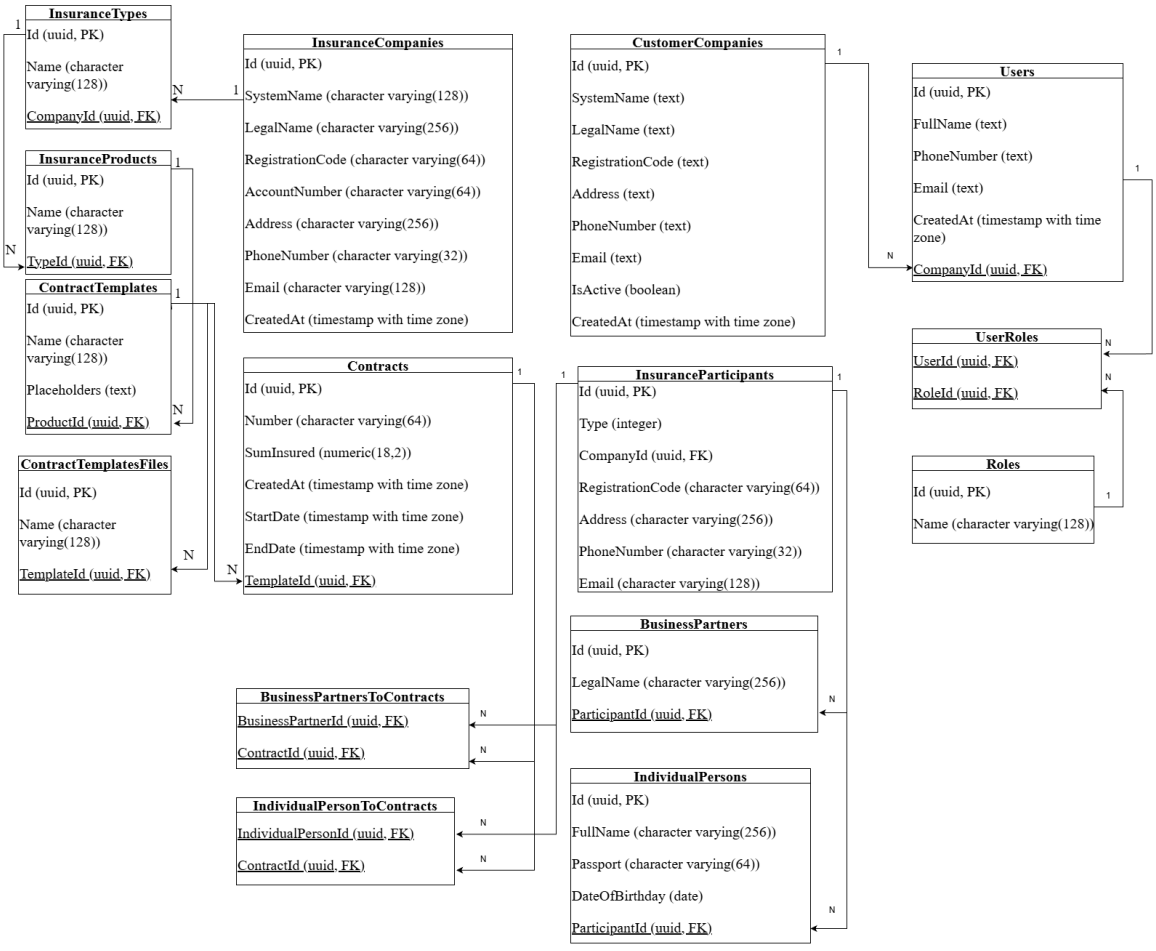


Рис. 4.4. Фізична модель бази даних

На рис. 4.5 показано фізичну модель бази даних, яка була реалізована в PostgreSQL.

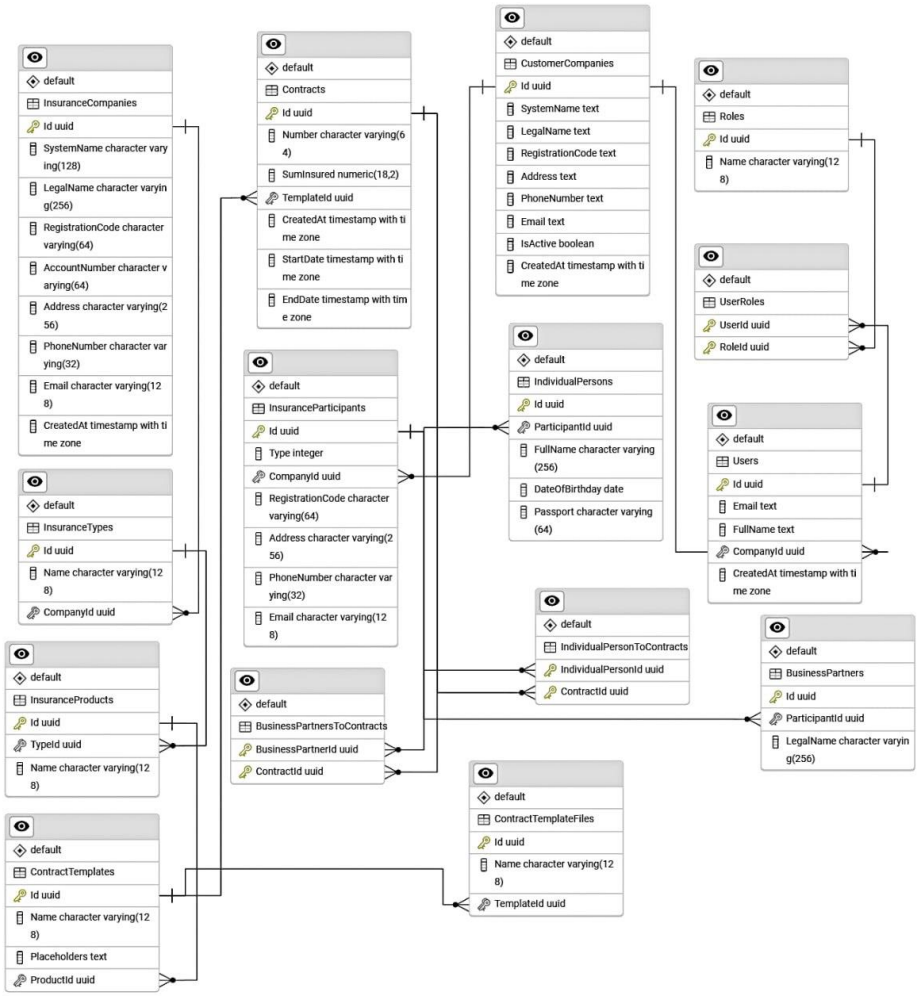


Рис. 4.5. Вигляд реалізованої фізичної моделі бази даних у PostgreSQL

Для ключових сутностей, наприклад, Id в таблицях Contracts, Users, InsuranceCompanies та інших, було обрано тип UUID, що дозволяє генерувати унікальні ідентифікатори на рівні клієнта чи сервісу.

SQL-скрипт для створення БД наведено у Додатку И.

4.3 Алгоритм роботи додатку

Алгоритм роботи системи IBAS починається з моменту, коли користувач заходить у додаток (див. рис. 4.6).

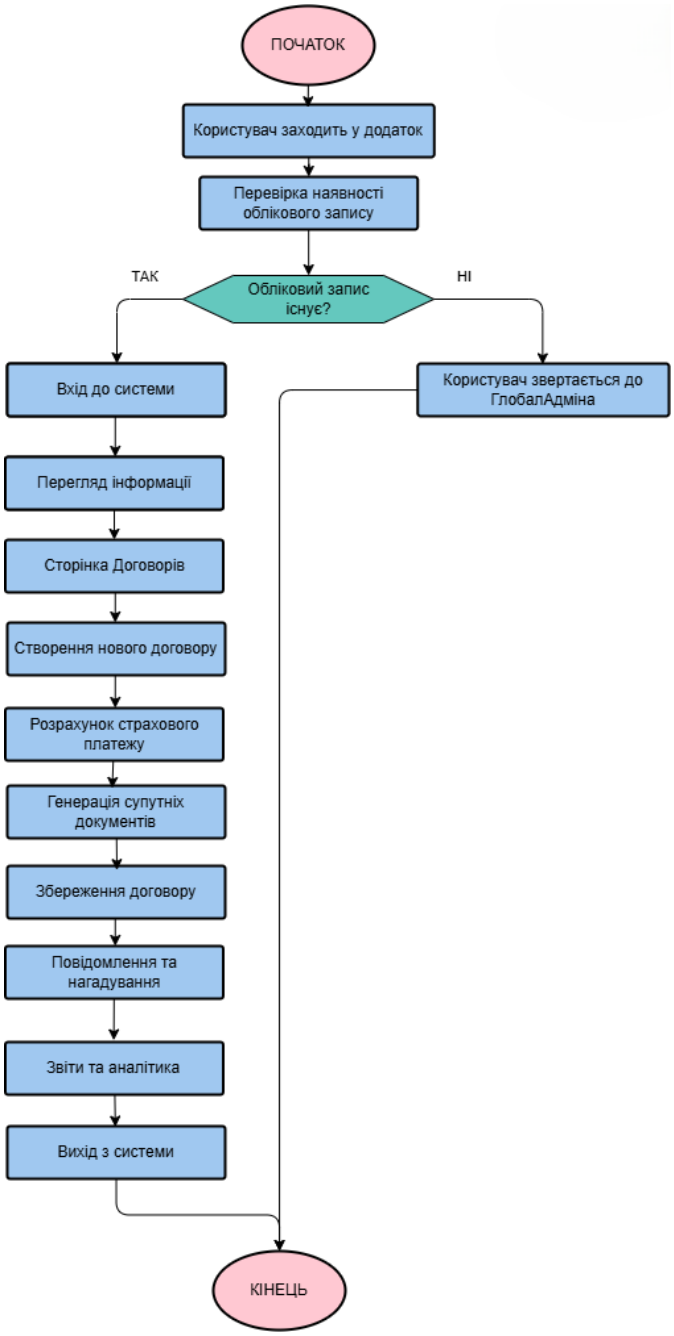


Рис. 4.6. Алгоритм роботи додатку

Система перевіряє наявність облікового запису користувача. Якщо обліковий запис існує, користувач проходить процедуру автентифікації, вводячи свої дані для входу. Після успішного входу він отримує доступ до основного функціоналу.

У разі відсутності облікового запису користувач звертається до глобального адміністратора для реєстрації та отримання доступу до системи.

Після входу користувач може переглядати інформацію, зокрема сторінку договорів, на якій відображаються всі укладені страхові договори з детальною інформацією. За необхідності користувач має змогу створити новий договір, заповнивши всі необхідні поля.

Далі система автоматично розраховує страхову премію за допомогою вбудованого калькулятора, а також генерує супутні документи на основі шаблонів із підставленням введених даних. Усі дані договору зберігаються в базі даних.

Користувач отримує сповіщення та нагадування про важливі події, зокрема про необхідність пролонгації договорів. Окрім цього, система надає звіти та аналітичні дані, що допомагають оцінювати діяльність страхових агентів і контролювати укладені договори.

В будь-який момент користувач може вийти з системи, завершуючи роботу з додатком.

4.4. Інтерфейс веб-додатку

Інтерфейс веб-додатку Insurance Broker Assistant System (IBAS) створений з урахуванням зручності користувача та логіки роботи страхового посередника. На рис. 4.7 зображено сторінку авторизації застосунку.

Рис. 4.7. сторінка авторизації застосунку

На рис. 4.8 бачимо сторінку укладених договорів. На сторінці присутня таблиця, яка містить ключову інформацію по кожному договору: номер договору, дату укладання, строки дії, страхову компанію, продукт страхування, страхувальника, страхову суму, платіж, статус оплати, та агента, який оформив договір.

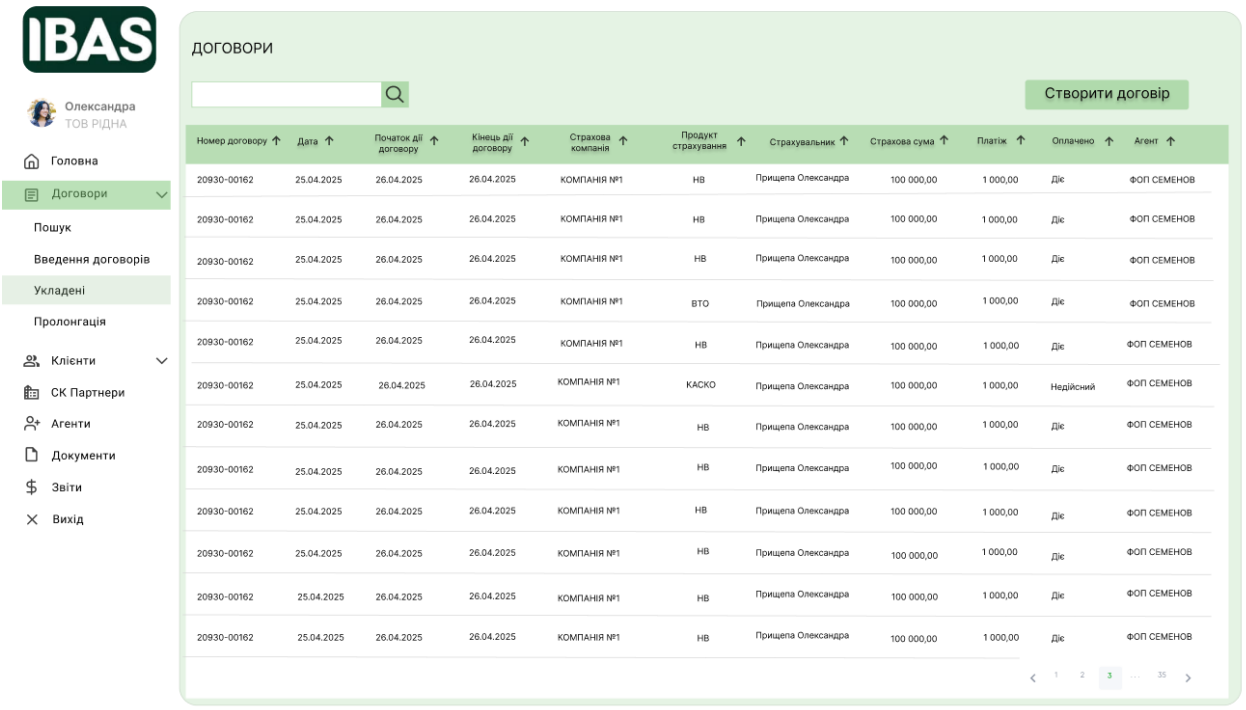


Рис. 4.8. Сторінка договорів

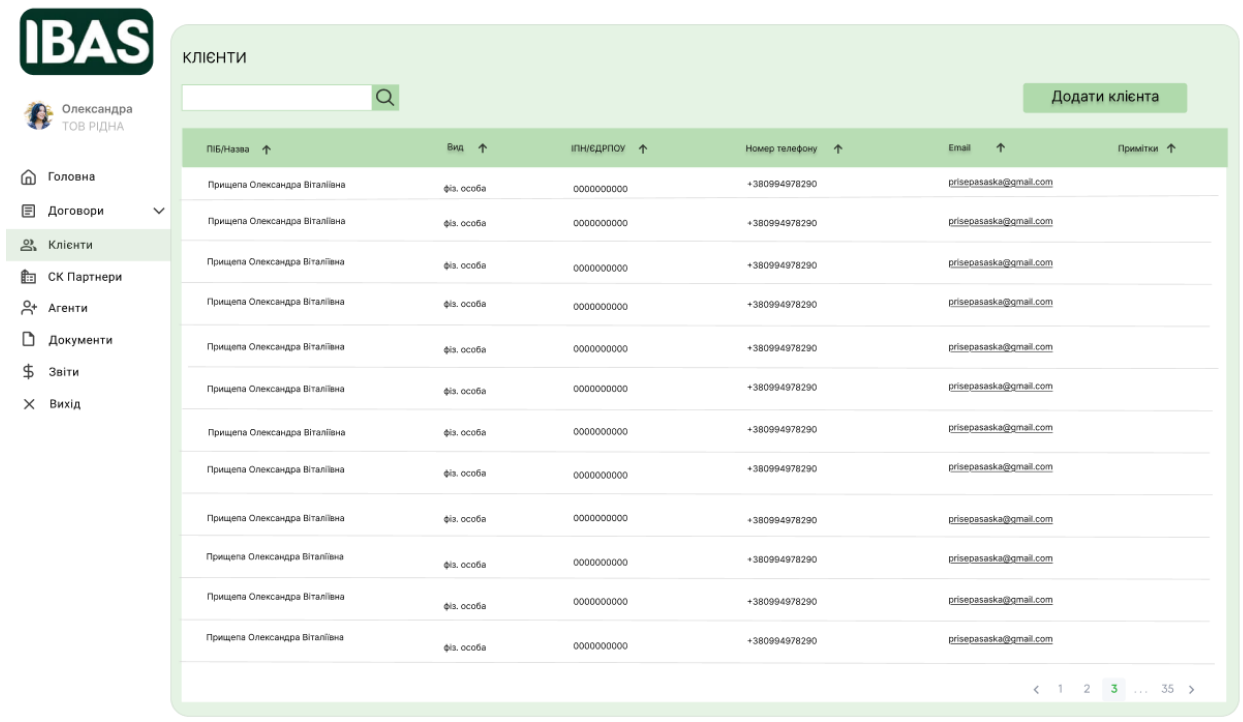


Рис. 4.9. Сторінка клієнтів

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було досягнуто поставлену мету та послідовно виконано всі поставлені задачі:

Дослідження сучасного стану українського страхового ринку та нормативно-правових вимог дало змогу виявити ключові тенденції, ризики та обмеження, що формують професійне середовище страхових посередників. Аналіз законодавства дозволив визначити проблеми, які потребують автоматизації: складність ведення договорів, контроль строків, взаємодія з клієнтами, вимоги до звітності та прозорого аудиту. Це підтвердило актуальність створення вебплатформи для брокерів і агентів.

PEST-аналіз та SWOT-аналіз проєкту, а також ідентифікація внутрішніх і зовнішніх зацікавлених сторін забезпечили комплексне розуміння впливу ринкового, регуляторного, економічного та технологічного середовища на розвиток системи IBAS. У роботі визначено ключові групи стейкхолдерів — страхових посередників, клієнтів, страхові компанії та команду розробки, що дало змогу сформувати чіткі очікування та вимоги до функціональності системи.

Сформульовано технічне завдання у вигляді паспорта проєкту, визначено функціональні та нефункціональні вимоги, побудовано концептуальну модель інформаційної системи. У роботі систематизовано основні бізнес-процеси: формування договорів, ведення клієнтів і посередників, автоматичні нагадування, пролонгації, генерація додатків і звітів. Це створило фундамент для подальшого проєктування архітектури та модулів системи.

Обґрунтовано вибір гібридної технології управління проєктом Water-Scrum-Fall, що поєднує передбачуваність планування з гнучкістю ітераційної розробки. Визначено організаційну структуру команди, проведено декомпозицію робіт проєкту, що забезпечило прозорість розподілу відповідальності, чіткість етапів робіт і можливість ефективного контролю виконання проєкту.

Проведено ідентифікацію, якісну та кількісну оцінку ризиків, визначено їхні тригери, наслідки та можливі стратегії реагування. Сформовано карту управління ризиками, що охоплює технічні, організаційні, фінансові, інтеграційні та інформаційні ризики, що дозволило підвищити передбачуваність процесу розробки та зменшити імовірність виникнення критичних проблем.

Розроблено алгоритми роботи ключових бізнес-процесів, створено концептуальну та даталогічну моделі бази даних, визначено особливості використання обраного технологічного стеку. Спроєктовано архітектуру вебплатформи IBAS, яка забезпечує масштабованість, безпеку, надійність і можливість подальшого розширення функціоналу.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Рейтинг страхових компаній України - FORINSURER. FORINSURER журнал про страхування, перестрахування та InsurTech. URL: https://forinsurer.com/ratings/nonlife (дата звернення: 11.10.2025).

2. Нагляд за ринком небанківських фінансових послуг. Національний банк України. URL: https://bank.gov.ua/ua/supervision/nonbanks (дата звернення: 11.10.2025).

3. International Association of Insurance Supervisors. International Association of Insurance Supervisors. URL: https://www.iais.org/ (date of access: 11.10.2025).

4. Страхування. Національний банк України. URL: https://bank.gov.ua/ua/supervision/regulation-nonbank-fs-market/insurance (дата звернення: 11.10.2025).

5. Комплексна інформаційна система Національного банку України. Національний банк України. URL: https://kis.bank.gov.ua/search-fu (дата звернення: 11.10.2025).

6. Про страхування. Офіційний вебпортал парламенту України. URL: https://zakon.rada.gov.ua/laws/show/1909-20#Text (дата звернення: 11.10.2025).

7. Sammut-Bonnici, T., & Galea, D. (2015). PEST analysis. У Wiley Encyclopedia of Management (Vol. 12, Issue 1, С. 1-1). Wiley. doi.org.

8. Second IDD application report 2022/2023. European Insurance and Occupational Pensions Authority. URL: https://www.eiopa.europa.eu/publications/second-idd-application-report-20222023_en (date of access: 11.10.2025).

9. Solvency II Overview – Frequently asked questions. European Commission - European Commission. URL: https://ec.europa.eu/commission/presscorner/detail/et/memo_15_3120 (date of access: 11.10.2025).

10. Directive - 2009/103 - EN - EUR-Lex. EUR-Lex – Access to European Union law – choose your language. URL: https://eur-lex.europa.eu/eli/dir/2009/103/oj/eng (date of access: 11.10.2025).

11. Interfax-Ukraine. Страховий ринок доводить, що попри пандемію та велику війну, можна досягти видатних результатів – голова НБУ. Інтерфакс-Україна. URL: https://interfax.com.ua/news/interview/1127016.html (дата звернення: 11.10.2025).

12. Федерація страхових об'єднань України відкрила офіс у Брюсселі. Мінфін - все про фінанси: новини, курси валют, банки. URL: https://minfin.com.ua/ua/2025/10/09/160131022/ (дата звернення: 11.10.2025).

13. Leading Software Development Company | ProfITsoft. ProfITsoft. URL: https://profitsoft.dev/ (date of access: 11.10.2025).

14. EWA страховой маркетплейс #1. EWA. URL: https://www.ewa.ua/ (дата звернення: 11.10.2025).

15. Kenton W. SWOT: What Is It, How It Works, and How to Perform an Analysis. Investopedia. URL: https://www.investopedia.com/terms/s/swot.asp (date of access: 11.10.2025).

16. China C. R. What Is Software as a Service (SaaS)? | IBM. IBM. URL: https://www.ibm.com/think/topics/saas (date of access: 11.10.2025).

17. PMI Ukraine Chapter | PMBOK 7. PMI Ukraine Chapter | Управління проєктами та сертифікації PMI. URL: https://pmiukraine.org/pmbok7 (дата звернення: 07.11.2025).

18. Eawag - das Wasserforschungsinstitut des ETH-Bereichs - Eawag. URL: https://www.eawag.ch/fileadmin/Domain1/Abteilungen/sandec/schwerpunkte/sesp/CLUES/Toolbox/t8/D8_1_Problem_Tree_Analysis.pdf (дата звернення: 11.10.2025).

19. Limited S. FTM Business Template - Project Passport | PDF | Project Management | Risk. Scribd.

URL: https://www.scribd.com/document/17253469/FTM-Business-Template-Project-Passport (date of access: 11.10.2025).

20. Project Life Cycle [Phases & Best Practices] | The Workstream. Atlassian. URL: https://www.atlassian.com/work-management/project-management/project-life-cycle (date of access: 11.10.2025).

21. A Conceptual Project Management Maturity Model for the South African Power Sector. IntechOpen - Open Science Open Minds | IntechOpen. URL: https://www.intechopen.com/chapters/82218 (date of access: 11.10.2025).

22. Agile Documentation: Methodology & Best Practices. Document360. URL: https://document360.com/blog/agile-documentation/ (date of access: 11.10.2025).

23. Verheul Consultants. URL: https://www.verheulconsultants.nl/water-scrum-fall_Forrester.pdf (дата звернення: 11.10.2025).

24. Moein M. The Official Kanban Guide | PDF. Scribd. URL: https://www.scribd.com/document/914679103/The-Official-Kanban-Guide (date of access: 11.10.2025).

25. A Guide to Water-Scrum-Fall Project Management | Cloud Coach. Cloud Coach. URL: https://cloudcoach.com/blog/a-guide-to-water-scrum-fall-project-management/ (date of access: 11.10.2025).

26. Work Breakdown Structure. workbreakdownstructure.com. URL: https://www.workbreakdownstructure.com/ (date of access: 11.10.2025).

27. Moein M. The Official Kanban Guide | PDF. Scribd. URL: https://www.scribd.com/document/914679103/The-Official-Kanban-Guide (date of access: 11.10.2025).

28. ISO 31000:2018. *ISO*. URL: https://www.iso.org/standard/65694.html (date of access: 11.10.2025).

29. Тімінський О.Г., Коломієць А.С. Методи управління ризиками в IT проєктах [Текст]: методичні вказівки до виконання практичних, лабораторних робіт та самостійної роботи для студентів освітньої програми «Управління проєктами» спеціальності 122 «Комп'ютерні науки» для денної і заочної форм

навчання / Тімінський О.Г., Коломієць А.С. – К. : КНУ імені Тараса Шевченка, 2021. – 40 с.

30. Microservices Architecture | Atlassian. *Atlassian*. URL: https://www.atlassian.com/microservices/microservices-architecture (date of access: 11.10.2025).

31. Home • Angular. *Home • Angular*. URL: https://angular.dev/ (date of access: 11.10.2025).

32. JavaScript With Syntax For Types. *TypeScript: JavaScript With Syntax For Types.* URL: https://www.typescriptlang.org/ (date of access: 11.10.2025).

33. What is RESTful API? - RESTful API Explained - AWS. *Amazon Web Services, Inc.* URL: https://aws.amazon.com/what-is/restful-api/ (date of access: 11.10.2025).

34. PostgreSQL. *PostgreSQL*. URL: https://www.postgresql.org/ (date of access: 11.10.2025).

35. What Is Continuous Delivery? Guide to CI/CD | Atlassian. *Atlassian*. URL: https://www.atlassian.com/continuous-delivery (date of access: 11.10.2025).

36. What is SSL/TLS: An In-Depth Guide - SSL.com. *SSL.com*. URL: https://www.ssl.com/article/what-is-ssl-tls-an-in-depth-guide/ (date of access: 11.10.2025).

37. Maxym Z. Моделювання даних (Data Modelling). Maxym Zosym. URL: https://www.maxzosim.com/data-modelling/ (дата звернення: 11.10.2025).

38. Kubiavka L., Teslia I. Hlevna J., Ivanov E., Latysheva T., Yehorchenkova N., Yehorchenkov O., Boyko N., PrimaDoc-Enterprise Information Management System: the project creation and implementation, The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 21-23 September, 2017, Bucharest, Romania

ДОДАТКИ

Додаток А

Таблиця А.1

Аналіз потенційних конкурентів

Параметр	Опис компанії
1	**2**
Назва	**ProfITsoft**
Основні послуги	ProfITsoft пропонує повний цикл розробки програмного забезпечення й тестування: від підготовки й організації процесу розробки до підтримки успішно реалізованих проєктів і оптимізації продуктивності існуючих рішень [13].
Особливості	ProfITsoft має понад 20 років досвіду у розробці високотехнологічних рішень, зокрема глибоку експертизу в Java та веб-технологіях. Компанія позиціонує себе як провідний розробник рішень для страхової галузі, включно з власною платформою “Complex System of Automation of the Insurance Company” (Integrated Insurance Company Automation System). Наголошується на високому рівні компетенцій, корпоративній культурі, довгострокових відносинах із клієнтами та бізнес-гнучкості. Використовуються сучасні фронтенд-технології (ReactJS, Angular, TypeScript тощо), бекенд на Java зі Spring, REST/SOAP, бази даних SQL/NoSQL, хмари (AWS, Azure, Google Cloud), інструменти тестування й CI/CD-практики
Цільова аудиторія	Основні клієнти ProfITsoft – середні й великі компанії, зокрема страховики та фінансові установи, які потребують комплексних рішень автоматизації бізнес-процесів. Крім того, компанія орієнтується на європейський ринок і працює з міжнародними замовниками, а також на проєкти в галузі Travel & Hospitality та FinTech. Для страхового сегмента це, насамперед, страхові компанії та, можливо, великі брокерські мережі, які потребують інтегрованих систем для автоматизації договорів, калькуляцій і звітності
Доступні платформи	Кросплатформений доступ через браузер, з охопленням хмарної інфраструктури і можливістю інтеграції з мобільними або десктопними клієнтами через API.
Сильні сторони	1. Досвід і експертиза: понад 20 років на ринку, глибока компетенція в Java та веб-технологіях, доведені кейси в InsurTech-сегменті. 2. Галузева спрямованість: власна система автоматизації для страхових компаній, що підтверджує розуміння специфіки бізнесу й можливість швидше впроваджувати типові процеси (генерація договорів, розрахунок премій, звітність). 3. Повний цикл послуг: від розробки «з нуля» до підтримки легасі, оптимізації продуктивності, що важливо для клієнтів із існуючими системами. 4. Технологічний стек: сучасні рішення у фронтенді (React, Angular), бекенді (Java, Spring), хмарна орієнтація, що дає гнучкість і масштабованість. 5. Міжнародна присутність: фокус на європейському ринку робить компанію привабливою для замовників за межами України.

Закінчення табл А.1

1	2
Слабкі сторони	1. Орієнтація на великі проєкти: фокус на enterprise-рішення може бути надмірним для невеликих брокерських компаній із обмеженим бюджетом або потребою швидкого MVP. 2. Вартість послуг: через високий рівень експертизи та комплексність підходу ціни можуть бути вищими порівняно з локальними невеликими розробниками або готовими SaaS-рішеннями. 3. Менша увага до “легких” продуктів: якщо потрібен простий інструмент для невеликих агентів або стартапів insurtech, ProfITsoft може не мати готових легковагових рішень. 4. Залежність від зовнішніх факторів: як українська компанія з європейською експансією, потенційно відчуває ризики, пов’язані з регіональною нестабільністю, що може впливати на довгострокові контракти та ресурси. 5. Непрозорість комерційних пропозицій: на сайті немає публічних демонстрацій продукту чи цінових рівнів, що утруднює попередню оцінку для потенційних клієнтів.
Назва	**EWA**
Основні послуги	EWA – це маркетплейс, у якому з одного боку страховики розміщують свої продукти, а з іншого боку підключаються страхові агенти та брокери. Тут вони мають можливість продавати всі продукти, які страховики розмістили на маркетплейсі.
Особливості	1. Налаштування агентських мереж 2. Вибір та роздача страхових продуктів 3. Облік оплат та звітність 4. Інструменти автоматизації лонгації договорів страхування
Цільова аудиторія	Страхові компанії, страхові агенти
Доступні платформи	Фронт-офіс, мобільний додаток, API
Сильні сторони	1. Одна публічна оферта для доступу 2. Страхові продукти 30+ компаній 3. Найкраща комісійна винагорода на ринку 4. Користування маркетплейсом EWA для агентів безкоштовне 5. Налаштування власної агентської мережі 6. Мобільний додаток 7. Відкрите API для інтеграцій
Слабкі сторони	1. Страховий агент не може створювати свої шаблони та працювати з ними. 2. Немає можливості страховому агенту продавати продукт страхової компанії, яка не співпрацює з EWA. 3. Неможливо агенту продати нетиповий продукт, який бажає клієнт

Додаток Б

Рис. Б.1 Декомпонована WBS по життєвому циклу проєкту

Додаток В

Таблиця В.1

User Story для функціональних вимог

Код US	Формулювання US	Acceptance Criteria
1	2	3
US001 (FR001) Авторизація	Як зареєстрований користувач (агент або адміністратор), я хочу мати можливість увійти в систему за email та паролем, щоб отримати доступ до свого робочого кабінету IBAS.	1. На сторінці входу відображаються поля «Email», «Пароль», чекбокс «Запам'ятати» та кнопка «Вхід». 2. Якщо користувач вводить некоректний email/пароль, система показує повідомлення про помилку й не пускає в особистий кабінет. 3. Після успішної авторизації користувач потрапляє на головну сторінку-дашборд. 4. Якщо встановлений чекбокс «Запам'ятати», при наступному відкритті системи email користувача підставляється автоматично.
US002 (FR002) Управління користувачами	Як адміністратор брокера, я хочу створювати, редагувати та блокувати облікові записи користувачів із різними ролями, щоб керувати доступом до функцій системи.	1. Адміністратор має окремий розділ «Користувачі» з таблицею всіх існуючих акаунтів та фільтрами . 2. Адміністратор може створити нового користувача, вказавши ПІБ, email, роль (агент, адміністратор) і статус облікового запису. 3. Адміністратор може змінити роль та статус існуючого користувача (активний/заблокований). 4. Заблокований користувач не може авторизуватися в системі. 5. Всі створені/змінені користувачі відображаються у списку без перезавантаження всієї системи.
US003 (FR003) Перегляд договорів	Як страховий агент, я хочу бачити список укладених договорів з можливістю пошуку, фільтрації та сортування, щоб швидко знаходити потрібний договір.	1. На сторінці «Укладені договори» відображається таблиця з основними полями (номер, дати, страхова компанія, продукт, страхувальник, сума, платіж, статус, агент). 2. Користувач може здійснювати пошук за номером договору або ПІБ/назвою клієнта через поле пошуку над таблицею. 3. Для кожної колонки доступне сортування за зростанням/спаданням. 4. Доступні базові фільтри: за датами, статусом, страховою компанією, агентом. 5. Якщо кількість договорів перевищує розмір сторінки, відображається пагінація .
US004 (FR004) Створення договору із шаблону	Як страховий агент, я хочу мати можливість обрати шаблон договору зі списку, щоб швидко створити типовий договір.	1. Користувач повинен мати доступ до списку доступних шаблонів у вигляді випадаючого списку або меню. 2. При виборі шаблону структура договору (розділи, стандартні умови) автоматично підставляється у форму договору.

Продовження табл. В.1

1	2	3
US004 (FR004) Створення договору із шаблону	Як страховий агент, я хочу мати можливість обрати шаблон договору зі списку, щоб швидко створити типовий договір.	1. Користувач повинен мати доступ до списку доступних шаблонів у вигляді випадаючого списку або меню. 2. При виборі шаблону структура договору (розділи, стандартні умови) автоматично підставляється у форму договору. 3. Дані клієнта, страхувальника та продукту автоматично заповнюють відповідні поля, якщо вони вже збережені в системі. 4. Користувач може відредагувати автоматично заповнені поля перед збереженням договору. 5. Після успішного створення договору система показує підтвердження (спливаюче повідомлення або інший індикатор).
US005 (FR004) Створення договору із шаблону	Як агент, я хочу зберегти створений договір у чернетку, щоб мати змогу повернутися до редагування пізніше.	1. Доступна кнопка «Зберегти як чернетку» поруч із кнопкою «Створити». 2. Чернетки мають окрему вкладку в особистому кабінеті. 3. Дані в чернетці повинні зберігатися не менше 30 днів. 4. Можна продовжити редагування договору з чернетки. 5. Можна видалити чернетку за бажанням.
US006 (FR005) Генерація PDF-договору	Як страховий агент, я хочу згенерувати договір у форматі PDF, щоб завантажити, надіслати клієнту або роздрукувати його.	1. Після заповнення та збереження договору на сторінці договору доступна дія «Згенерувати PDF». 2. Згенерований PDF містить всі реквізити договору, дані клієнта, страхувальника, агента та компанії згідно обраного шаблону. 3. У PDF передбачені місця для підписів і печаток, а також можливість додавання електронного підпису агента. 4. Користувач може завантажити PDF-файл на свій комп’ютер або відкрити для попереднього перегляду у браузері. 5. У разі помилки генерації система показує зрозуміле повідомлення про помилку.
US007 (FR006) Розрахунок страхової премії	Як страховий агент, я хочу вводити параметри ризику та миттєво отримувати розрахунок страхової премії, щоб швидко озвучити клієнту вартість договору.	1. На формі створення/редагування договору є блок «Розрахунок премії» з полями вводу параметрів ризику. 2. Після заповнення обов’язкових полів користувач може запустити розрахунок натисканням кнопки «Розрахувати». 3. Система використовує збережені тарифні ставки для обраного продукту й коректно обчислює премію. 4. Розрахована премія автоматично підставляється у відповідне поле договору й відображається користувачу. 5. При зміні ключових параметрів договору користувач може повторно виконати перерахунок.

Продовження табл. В.1

1	2	3
US008 (FR007) Генерація супровідних документів	Як страховий агент, я хочу автоматично формувати супровідні документи (рахунки, заяви, згоди тощо) на основі шаблонів, щоб не готувати їх вручну.	1. На сторінці договору доступний список типів супровідних документів, які можна створити. 2. Після вибору типу документа система підставляє у шаблон дані клієнта, договору й компанії. 3. Користувач може переглянути згенерований документ перед збереженням/експортом і за потреби відредагувати окремі поля. 4. Документ можна зберегти в системі та/або завантажити у PDF/іншому форматі. 5. Згенеровані документи прив'язуються до відповідного договору і відображаються в його картці.
US009 (FR008) Збереження даних договорів та контрагентів	Як користувач системи, я хочу щоб усі введені дані по договорах, клієнтах, агентах та страховиках зберігалися в єдиній базі, щоб мати до них постійний доступ і уникнути повторного введення.	1. Після натискання кнопки «Зберегти» система записує дані договору/клієнта/агента/компанії в базу даних без втрати інформації. 2. Збережені записи доступні для пошуку й перегляду у відповідних розділах («Договори», «Клієнти», «СК партнери», «Агенти»). 3. Користувач може відкрити існуючий запис, відредагувати поля й повторно зберегти зміни. 4. Система не допускає створення дублюючих записів за унікальними ідентифікаторами. 5. У разі проблеми збереження система показує повідомлення про помилку й не втрачає введені користувачем дані.
US010 (FR009) Повідомлення та нагадування	Як страховий агент, я хочу отримувати автоматичні нагадування про важливі події по договорах (закінчення строку, прострочені платежі), щоб вчасно зв'язуватися з клієнтами.	1. Система щоденно перевіряє статуси договорів та формує події для нагадувань (закінчення дії, планова пролонгація, прострочення премії). 2. На головній сторінці-дашборді відображається блок із ключовими нагадуваннями (кількість договорів із простроченою премією, які закінчуються найближчим часом). 3. Користувач має можливість перейти з нагадування безпосередньо в картку відповідного договору. 4. За налаштуванням брокера система може надсилати email-повідомлення про окремі події (наприклад, за 30 днів до закінчення договору). 5. Уже опрацьовані нагадування можуть бути позначені як виконані або приховані користувачем.
US011 (FR010) Управління шаблонами	Як адміністратор, я хочу додавати, редагувати та видаляти шаблони документів, щоб оперативно адаптувати їх до змін законодавства	1. У розділі «Шаблони» доступний список усіх шаблонів із зазначенням типу документа, назви, дати останньої зміни та статусу (активний/деактивований). 2. Адміністратор може створити новий шаблон, вказавши назву, тип документа й текст шаблону з маркерами для підстановки даних. 3. Адміністратор може змінювати існуючий шаблон, зберігаючи нову версію.

Продовження табл. В.1

1	2	3
US011 (FR010) Управління шаблонами	Як адміністратор, я хочу додавати, редагувати та видаляти шаблони документів, щоб оперативно адаптувати їх до змін законодавства й вимог страхових компаній.	4. Шаблон, позначений як неактивний, не відображається у списку вибору під час створення договору/документа. 5. Система не дозволяє видалити шаблон, який використовується в існуючих договорах, але дозволяє його деактивувати.
US012 (FR011) Звіти та аналітика	Як менеджер брокера, я хочу формувати звіти за договорами, клієнтами та активністю агентів, щоб аналізувати ефективність продажів і портфель договорів.	1. У розділі «Звіти» доступний перелік типових звітів (реєстр договорів, пролонгації, прострочені премії, звіти по агентам). 2. Для кожного звіту користувач може обрати період, страхову компанію, агента, продукт та інші фільтри. 3. Після формування звіту дані відображаються у табличному вигляді в інтерфейсі системи. 4. Користувач може експортувати сформований звіт у формат Excel (або інший табличний формат). 5. Якщо за вибраними фільтрами дані відсутні, система відображає повідомлення «Дані відсутні» замість пустої сторінки з помилкою.
US013 (FR012) Управління страховими компаніями	Як адміністратор брокера, я хочу вести довідник страхових компаній-партнерів, щоб агенти могли швидко обирати потрібну компанію при оформленні договорів.	1. У розділі «СК партнери» відображається список страхових компаній з основними реквізитами (назва, код, контакти, статус співпраці). 2. Адміністратор може додати нову страхову компанію, заповнивши обов’язкові поля (назва, код, контактні дані). 3. Адміністратор може редагувати або деактивувати існуючу компанію. 4. У випадаючих списках при створенні договору відображаються лише активні компанії. 5. Зміни в довіднику одразу враховуються при оформленні нових договорів.
US014 (FR013) Управління клієнтами	Як страховий агент, я хочу мати окремий розділ із клієнтами, щоб швидко знаходити клієнта й не вводити його дані щоразу при створенні договору.	1. У розділі «Клієнти» відображається таблиця з клієнтами (ПІБ/назва, тип – фіз./юр. особа, ІПН/ЄДРПОУ, телефон, email тощо). 2. Агент може додати нового клієнта, заповнивши обов’язкові поля й зберігши запис. 3. При спробі додати клієнта з уже існуючим ІПН/ЄДРПОУ система повідомляє про дубль та пропонує відкрити існуючий запис. 4. Агент може редагувати контактні дані клієнта, а зміни будуть використовуватися у всіх нових договорах. 5. На формі створення договору агент може обрати клієнта зі списку, після чого дані клієнта автоматично підставляються в поля договору.

Закінчення табл. В.1

1	2	3
US015 (FR014) Управління агентами	Як адміністратор брокера, я хочу вести облік агентів та їхніх комісійних умов, щоб правильно розподіляти винагороду та контролювати їхню активність.	1. У розділі «Агенти» відображається список агентів із ПІБ, контактами, статусом і базовою комісійною ставкою. 2. Адміністратор може створити нового агента, вказавши необхідні реквізити та комісійну ставку/схему. 3. Адміністратор може змінити статус агента (активний/неактивний); неактивний агент не доступний для вибору в нових договорах. 4. У картці договору відображається прив’язаний агент, а також ця інформація враховується у звітах по агентам. 5. Зміни в довіднику агентів одразу впливають на доступні значення при оформленні нових договорів.

Додаток Г

Таблиця Г.1

Беклогу продукту проєкту

Код US	Формулювання US	Завдання, підзавдання
1	**2**	**3**
US001 (FR004) Створення договору із шаблону	Як страховий агент, я хочу мати можливість обрати шаблон договору зі списку, щоб швидко створити типовий договір без ручного введення тексту.	***T01. Аналіз шаблонів договорів*** ST01.1. Зібрати приклади шаблонів від агентів та страхових компаній. ST01.2. Уніфікувати структуру шаблонів (розділи, реквізити, змінні поля). ST01.3. Описати формат зберігання шаблонів у системі (БД / файлове сховище). ***T02. Реалізація вибору шаблону в інтерфейсі*** ST02.1. Спроєктувати UI вибору шаблону (список / модальне вікно). ST02.2. Реалізувати завантаження обраного шаблону у форму договору. ST02.3. Забезпечити відображення лише актуальних (активних) шаблонів.
US002 (FR001) Авторизація користувачів	Як зареєстрований користувач, я хочу входити в систему за email та паролем, щоб отримати доступ до свого робочого кабінету IBAS.	***T01. Реалізація механізму авторизації*** ST01.1. Спроєктувати модель користувача та схему збереження облікових даних. ST01.2. Налаштувати механізм перевірки email/пароля (hashing, salting). ST01.3. Реалізувати обмеження доступу для неавторизованих користувачів. ***T02. Розробка форми входу*** ST02.1. Реалізувати UI сторінки логіну (Email, Пароль, «Запам’ятати»). ST02.2. Додати обробку помилок (неправильні дані, заблокований користувач). ST02.3. Налаштувати перенаправлення на дашборд після успішного входу.
US003 (FR002) Управління користувачами	Як адміністратор, я хочу створювати, редагувати та блокувати облікові записи користувачів, щоб керувати доступом до функцій системи.	***T01. Проєктування моделі ролей та прав*** ST01.1. Визначити ролі (адміністратор, агент тощо) та повноваження. ST01.2. Спроєктувати схему збереження ролей і зв’язків «користувач–роль». ***T02. Розробка інтерфейсу управління користувачами*** ST02.1. Створити сторінку зі списком користувачів і базовими фільтрами. ST02.2. Реалізувати форми створення/редагування користувача. ST02.3. Додати функцію блокування/розблокування користувача.

Продовження табл. Г.1

1	2	3
US004 (FR003) Перегляд укладених договорів	Як страховий агент, я хочу бачити список укладених договорів з пошуком, фільтрацією та сортуванням, щоб швидко знаходити потрібний договір.	***T01. Проєктування таблиці договорів*** ST01.1. Визначити перелік колонок (номер, дати, СК, продукт, страхувальник тощо). ST01.2. Спроєктувати структуру даних і необхідні запити до БД.T02. Реалізація інтерфейсу списку договорів ST02.1. Розробити UI сторінки «Укладені договори» з пагінацією. ST02.2. Реалізувати пошук за номером договору/ПІБ клієнта. ST02.3. Додати сортування за колонками та базові фільтри (період, статус, СК, агент). ST02.4. Налаштувати перехід до картки договору зі списку.
US005 (FR004) Збереження договору як чернетки	Як агент, я хочу зберегти створений договір у чернетку, щоб мати змогу повернутися до редагування пізніше.	***T01. Підтримка статусу «чернетка»*** ST01.1. Додати в модель договору статуси, включаючи «чернетка». ST01.2. Описати правила зміни статусів (чернетка → укладений тощо). ***T02. Реалізація збереження чернеток*** ST02.1. Додати кнопку «Зберегти як чернетку» у формі договору. ST02.2. Реалізувати збереження неповного набору даних договору. ST02.3. Налаштувати окрему вкладку/фільтр «Чернетки» у списку договорів або в особистому кабінеті. ST02.4. Реалізувати можливість продовження редагування договору з чернетки. ST02.5. Додати можливість видалити чернетку та механізм автоматичного очищення старіших за 30 днів (за потреби).
US006 (FR005) Генерація PDF-договору	Як страховий агент, я хочу згенерувати договір у форматі PDF, щоб завантажити, надіслати клієнту або роздрукувати його.	***T01. Вибір бібліотеки для PDF-генерації*** ST01.1. Порівняти варіанти бібліотек для формування PDF (за можливостями та ліцензіями). ST01.2. Обрати підхід до формування макета договору (шаблони/HTML → PDF). ***T02. Реалізація сервісу генерування PDF*** ST02.1. Створити сервіс, що формує PDF на основі шаблону та даних договору. ST02.2. Перевірити коректність відображення реквізитів, підписів, печаток. ST02.3. Додати можливість завантажити або відкрити PDF в браузері. ST02.4. Налаштувати обробку помилок генерації та логування.

Продовження табл. Г.1

1	2	3
US007 (FR006) Розрахунок страхової премії	Як страховий агент, я хочу вводити параметри ризику та миттєво отримувати розрахунок страхової премії, щоб швидко озвучити клієнту вартість договору.	***T01. Опис бізнес-правил розрахунку*** ST01.1. Зібрати тарифні ставки та формули для основних продуктів. ST01.2. Узгодити з бізнесом алгоритми розрахунку премії та округлення. ***T02. Реалізація калькулятора премії*** ST02.1. Створити сервіс розрахунку премії з урахуванням параметрів договору. ST02.2. Інтегрувати сервіс у форму договору (кнопка «Розрахувати»). ST02.3. Автоматично підставляти розраховану премію в поле «Платіж». ST02.4. Додати юніт-тести для перевірки коректності формул.
US008 (FR007) Генерація супровідних документів	Як страховий агент, я хочу автоматично формувати супровідні документи (рахунки, заяви, згоди тощо), щоб не готувати їх вручну.	***T01. Опис шаблонів супровідних документів*** ST01.1. Визначити перелік типів документів (рахунок, згода, заява тощо). ST01.2. Підготувати текстові шаблони з маркерами для підстановки даних. ***T02. Реалізація механізму генерування документів*** ST02.1. Створити сервіс заповнення шаблонів даними договору/клієнта. ST02.2. Реалізувати UI для вибору типу документа з картки договору. ST02.3. Забезпечити збереження згенерованих документів та можливість експорту в PDF/Word. ST02.4. Налаштувати прив’язку документів до відповідного договору в системі.
US009 (FR008) Збереження даних у БД	Як користувач, я хочу, щоб усі введені дані по договорах, клієнтах, агентах та страховиках зберігалися в єдиній базі, щоб уникнути повторного введення.	***T01. Проєктування бази даних*** ST01.1. Спроєктувати сутності (клієнти, договори, агенти, СК, шаблони тощо). ST01.2. Визначити унікальні ключі (ID, номер договору, ІПН/ЄДРПОУ) та зв’язки між таблицями. ***T02. Реалізація CRUD-операцій*** ST02.1. Реалізувати сервіси створення, читання, оновлення та видалення для основних сутностей. ST02.2. Додати перевірку дублювання даних (наприклад, клієнт з тим самим ІПН). ST02.3. Налаштувати транзакційність для критичних операцій (створення договору з пов’язаними записами). ST02.4. Реалізувати обробку помилок БД та зрозумілі повідомлення користувачу.

Продовження табл. Г.1

1	2	3
US010 (FR009) Повідомлення та нагадування	Як страховий агент, я хочу отримувати автоматичні нагадування про важливі події по договорах, щоб вчасно зв'язуватися з клієнтами.	***T01. Визначення правил сповіщень*** ST01.1. Сформувати перелік подій для нагадувань (закінчення дії, простої премії, планова пролонгація тощо). ST01.2. Визначити строки сповіщень (за 30/14/7 днів тощо). ***T02. Реалізація механізму нотифікацій*** ST02.1. Створити фоновий процес/планувальник для перевірки умов сповіщень. ST02.2. Реалізувати запис подій у таблицю нагадувань. ST02.3. Показувати нагадування на дашборді головної сторінки. ST02.4. Додати можливість позначати нагадування як опрацьовані. ST02.5. За потреби налаштувати відправку email-сповіщень.
US011 (FR010) Управління шаблонами	Як адміністратор, я хочу додавати, редагувати та деактивувати шаблони документів, щоб адаптувати їх до змін законодавства й вимог СК.	***T01. Інтерфейс довідника шаблонів*** ST01.1. Створити сторінку зі списком шаблонів (тип, назва, статус, дата зміни). ST01.2. Додати фільтрацію за типом документа та статусом. ***T02. Операції з шаблонами*** ST02.1. Реалізувати створення нового шаблону з редактором тексту. ST02.2. Реалізувати редагування існуючого шаблону з веденням історії змін (мінімум дата/користувач). ST02.3. Реалізувати деактивацію шаблону замість фізичного видалення. ST02.4. Забезпечити використання тільки активних шаблонів при формуванні договорів та документів.
US012 (FR011) Звіти та аналітика	Як менеджер брокера, я хочу формувати звіти за договорами, клієнтами та активністю агентів, щоб аналізувати ефективність продажів.	***T01. Визначення переліку звітів*** ST01.1. Узгодити з бізнесом список стандартних звітів (реєстр договорів, пролонгації, прострочені премії, звіти по агентам тощо). ST01.2. Описати набір полів та фільтрів для кожного звіту. ***T02. Реалізація звітного модуля*** ST02.1. Реалізувати каталог звітів і сторінку параметрів звіту. ST02.2. Створити SQL-запити/представлення для формування звітів. ST02.3. Відобразити звіт у табличному вигляді в інтерфейсі. ST02.4. Реалізувати експорт у Excel.ST02.5. Оптимізувати запити для великих обсягів даних.

Закінчення табл. Г.1

1	2	3
US013 (FR012) Управління страховими компаніями	Як адміністратор, я хочу вести довідник страхових компаній-партнерів, щоб агенти могли швидко обирати потрібну компанію при оформленні договорів.	***T01. Структура довідника СК*** ST01.1. Визначити обов'язкові реквізити страхової компанії (назва, код, контакти, статус співпраці). ST01.2. Спроєктувати таблицю в БД для збереження СК. ***T02. Інтерфейс управління СК*** ST02.1. Створити сторінку зі списком СК з фільтрами за назвою/статусом. ST02.2. Реалізувати форми створення/редагування компаній. ST02.3. Додати можливість деактивувати компанію (без видалення історії). ST02.4. Забезпечити використання тільки активних компаній у формах створення договорів.
US014 (FR013) Управління клієнтами	Як страховий агент, я хочу мати окремий розділ із клієнтами, щоб швидко знаходити клієнта й не вводити його дані щоразу при створенні договору.	***T01. Проєктування довідника клієнтів*** ST01.1. Визначити набір полів для фізичних і юридичних осіб (ПІБ/назва, ІПН/ЄДРПОУ, контакти тощо). ST01.2. Спроєктувати структуру таблиць і унікальні обмеження (наприклад, по ІПН). ***T02. Реалізація UI для роботи з клієнтами*** ST02.1. Створити сторінку «Клієнти» з таблицею, пошуком і фільтрами. ST02.2. Реалізувати форму додавання/редагування клієнта. ST02.3. Додати перевірку дублювання клієнтів за ІПН/ЄДРПОУ. ST02.4. Налаштувати вибір клієнта з довідника у формі створення договору з автозаповненням полів.
US015 (FR014) Управління агентами	Як адміністратор брокера, я хочу вести облік агентів та їхніх комісійних умов, щоб правильно розподіляти винагороду та контролювати їхню активність.	***T01. Модель даних для агентів*** ST01.1. Визначити набір атрибутів агента (ПІБ, контакти, статус, базова комісія тощо). ST01.2. Спроєктувати таблицю агентів та зв'язки з договорами. ***T02. Інтерфейс управління агентами*** ST02.1. Створити сторінку «Агенти» зі списком, пошуком і фільтрами. ST02.2. Реалізувати створення/редагування агента та налаштування комісійної ставки. ST02.3. Додати можливість змінювати статус агента (активний/неактивний). ST02.4. Налаштувати відображення агента у звітах та формах договорів тільки за активного статусу.

Додаток Д

Таблиця Д.1

Інформація щодо планування Спринту 1

№	Назва роботи	Трив. (год)	Виконавець	Матеріали, технології
1	2	3	4	5
1	Планування та координація реалізації US002, US004, US009, US014 (формування Sprint Backlog, узгодження задач Т з беклогу)	30	Project Manager	Confluence, Jira, Google Calendar, Microsoft Teams
2	Проведення подій Scrum (daily, Sprint Review, Retrospective), комунікація із замовником щодо пріоритетів User Story	24	Project Manager	Confluence, Jira, Google Calendar, Microsoft Teams
3	Оновлення документації спринту (опис інкременту, оновлення беклогу)	16	Project Manager	Confluence, Jira, Google Calendar, Microsoft Teams
4	Аналіз та деталізація вимог для авторизації й управління користувачами (US002 T01, US003 T01)	30	Business Analyst	Confluence, Jira, Google Calendar, Microsoft Teams
5	Аналіз шаблонів договорів і процесу роботи з чернетками (US001 T01, US005 T01)	20	Business Analyst	Confluence, Jira, Google Calendar, Microsoft Teams
6	Специфікація атрибутів БД і полів інтерфейсів для «Договорів» та «Клієнтів» (US004 T01, US014 T01, US009 T01)	20	Business Analyst	Confluence, Jira, Google Calendar, Microsoft Teams
7	Проєктування макетів сторінки авторизації (US002 T02 – форма входу, стани помилки)	16	UI/UX Designer	Figma, Jira, Google Calendar, Microsoft Teams
8	Проєктування макета списку «Укладені договори» (US004 T02 – таблиця, фільтри, пагінація)	18	UI/UX Designer	Figma, Jira, Google Calendar, Microsoft Teams
9	Проєктування макета списку «Клієнти» та картки клієнта (US014 T02)	18	UI/UX Designer	Figma, Jira, Google Calendar, Microsoft Teams
10	Проєктування форми «Створення договору із шаблону» та сценарію «Зберегти як чернетку» (US001 T02, US005 T02)	18	UI/UX Designer	Figma, Jira, Google Calendar, Microsoft Teams
11	Проєктування логічної моделі БД для користувачів, клієнтів та договорів (US009 T01)	24	Tech Lead	.NET, PostgreSQL, GitLab, Jira, Google Calendar, Microsoft Teams
12	Проєктування архітектури модуля авторизації та безпеки (US002 T01)	18	Tech Lead	.NET, PostgreSQL, GitLab, Jira, Google Calendar, Microsoft Teams

Закінчення табл. Д.1

1	2	3	4	5
13	с	18	Tech Lead	.NET, PostgreSQL, GitLab, Jira, Google Calendar, Microsoft Teams
14	Код-рев'ю основних змін бекенду та фронтенду по спринту (US002, US004, US014, US005)	10	Tech Lead	GitLab, Jira, Google Calendar, Microsoft Teams
15	Реалізація сторінки авторизації згідно макетів (US002 T02 – валідація форм, обробка помилок)	18	Frontend Developer	Angular, TypeScript, HTML/CSS, GitLab, Jira, Google Calendar, Microsoft Teams
16	Реалізація сторінки «Укладені договори» (US004 T02 – таблиця, сортування, пошук, пагінація)	24	Frontend Developer	Angular, TypeScript, HTML/CSS, GitLab, Jira, Google Calendar, Microsoft Teams
17	Реалізація сторінки «Клієнти» (US014 T02 – таблиця, пошук, перехід у картку клієнта)	18	Frontend Developer	Angular, TypeScript, HTML/CSS, GitLab, Jira, Google Calendar, Microsoft Teams
18	Реалізація інтерфейсу «Зберегти як чернетку» та списку чернеток (US005 T02 – часткова реалізація)	10	Frontend Developer	Angular, TypeScript, HTML/CSS, GitLab, Jira, Google Calendar, Microsoft Teams
19	Реалізація сервісів БД і CRUD-операцій для користувачів, договорів, клієнтів (US009 T02, US004 T02, US014 T02)	24	Tech Lead	.NET, PostgreSQL, GitLab, Jira, Google Calendar, Microsoft Teams
20	Підготовка тест-плану та тест-кейсів для US002, US004, US014, US005	22	QA Engineer	Jira (test cases, defects), Google Calendar, Microsoft Teams
21	Функціональне тестування авторизації (US002 – позитивні й негативні сценарії, блокування користувача)	18	QA Engineer	Jira, браузери, Postman, Google Calendar, Microsoft Teams
22	Тестування списку договорів і клієнтів (US004, US014 – пошук, сортування, пагінація)	18	QA Engineer	Jira, браузери, Postman, Google Calendar, Microsoft Teams
23	Тестування збереження договору як чернетки (US005 – створення, відкриття, видалення)	12	QA Engineer	Jira, браузери, Postman, Google Calendar, Microsoft Teams
24	Налаштування PostgreSQL та створення тестової БД для IBAS (US009 T02)	16	DevOps	PostgreSQL, Docker, Docker Compose, GitLab, Jira, Google Calendar, Microsoft Teams
25	Налаштування репозиторіїв і CI-пайплайнів для backend і frontend (US002, US004, US014)	24	DevOps	GitLab, Docker, Docker Compose, Jira, Google Calendar, Microsoft Teams

Додаток Е

Таблиця Е.1

Ідентифікація ризиків проекту

Код ризику	Тип ризику	Ризикова подія	Сила впливу	Керованість
1	2	3	4	5
P01	Технічні	Обраний мікросервісний підхід виявляється надто складним для підтримки невеликою командою	висока	середня
P02	Технічні	Проблеми з продуктивністю системи при одночасній роботі значної кількості агентів і брокерів	середня	висока
P03	Технічні	Неможливість або суттєва затримка інтеграції з API страхових компаній-партнерів	висока	середня
P04	Технічні	Втрата або пошкодження даних через помилки процесу резервного копіювання БД та файлового сховища	висока	висока
P05	Технічні	Помилки при міграції даних клієнтів і договорів зі старих систем, що призводять до некоректності інформації	середня	середня
P06	Інфраструктурні	Затримка налаштування Kubernetes-кластера та CI/CD-процесів для розгортання IBAS	середня	середня
P07	Операційні	Збій хостингу, що призводить до простою системи	висока	низька
P08	Інформаційна безпека	Несанкціонований доступ або витік персональних та комерційних даних користувачів IBAS	висока	середня
P09	Регуляторні	Зміна нормативно-правових вимог НБУ до діяльності страхових посередників та обліку договорів після реалізації основного функціоналу	висока	низька
P10	Якісні	Низька зручність та інтуїтивність UI/UX інтерфейсів для агентів і брокерів, що знижує готовність користуватися IBAS	середня	висока
P11	Якісні	Помилки в розрахунку страхових премій або формуванні умов договорів	висока	висока

Продовження табл. Е.1

1	2	3	4	5
Р12	Пов’язані із замовником	Часті зміни пріоритетів замовника щодо функціоналу IBAS	висока	середня
Р13	Пов’язані із замовником	Невчасне надання замовником необхідної вихідної інформації (тарифи, шаблони договорів, регламенти)	середня	середня
Р14	Пов’язані із замовником	Пасивність замовника у прийнятті рішень, затримка погодження ключових артефактів проєкту	середня	середня
Р15	Управлінські	Затримка затвердження паспорта проєкту, бюджету та плану робіт на етапі ініціації	середня	середня
Р16	Управлінські	Низькій рівень комунікації в команді	середня	висока
Р17	Управлінські	Недостатній досвід керівника проєкту в управлінні IT-проєктами за змішаною моделлю Water–Scrum–Fall	висока	середня
Р18	Кадрові	Вихід з команди ключового спеціаліста (Tech Lead, Business Analyst, DevOps)	висока	середня
Р19	Кадрові	Недостатня компетентність окремих членів команди в страховій предметній області	середня	середня
Р20	Кадрові	Ускладнення з наймом потрібних фахівців на ринку праці (тривалий пошук Frontend / QA / DevOps)	середня	низька
Р21	Scrum-процес	Перевантаження команди задачами у спринтах, що призводить до вигорання та падіння продуктивності	середня	висока
Р22	Scrum-процес	Неефективне планування беклогу та оцінювання задач, що призводить до систематичного невиконання плану спринтів	середня	висока
Р23	Комунікаційні	Непорозуміння в команді, конфлікти між PM, BA, Tech Lead та розробниками, що уповільнюють прийняття рішень	середня	висока
Р24	Фінансові	Перевищення затвердженого бюджету	висока	середня
Р25	Фінансові	Невчасне фінансування	висока	середня

Закінчення табл. Е.1

1	2	3	4	5
Р26	Фінансові	Перевищення вартості пострелізної підтримки	середня	середня
Р27	Календарні	Відставання від 12-місячного графіка реалізації проєкту, зрив ключових контрольних точок	висока	середня
Р28	Зовнішні	Масовані обстріли	висока	низька
Р29	Зовнішні	Мобілізація, ВЛК	висока	низька
Р30	Зовнішні	Відключення електроенергії, що впливають на доступність команди та середовищ	висока	низька

Додаток Ж

Таблиця Ж.1

Протиризикові заходи проєкту

№	Ризикова подія	ПРЗ 1	Симптом (рання ознака)	ПРЗ 2	ПРЗ 3
		Профілактика		Дії при симптомі	Дії при проблемі
1	2	3	4	5	6
1	Обраний мікросервісний підхід складний	Архітектурний аудит, спрощення сервісів, стандартизація стеку, документація; навчання команди	Зростає час розробки й релізів, багато блокерів через інтеграції між сервісами	Перегляд архітектури, декомпозиція/об’єднання сервісів, залучення архітектора-консультанта	Поетапний рефакторинг у бік більш монолітного/модульного рішення, замороження нових фіч до стабілізації
2	Проблеми з продуктивністю	Нефункціональні вимоги, проєктування під навантаження, регулярні performance-тести, моніторинг	Падіння швидкодії у пікові години, збільшення часу відгуку, скарги користувачів	Оптимізація найкритичніших запитів, масштабування ресурсів, додатковий performance-аналіз	Запуск тимчасового масштабування (вертикального/горизонтального), введення обмежень на навантаження, позаплановий реліз виправлень
3	Затримка інтеграції API	Попередній аналіз API, пілотні інтеграції, письмові домовленості з партнерами, буфер часу в плані	Затримки з отриманням техдокументації, нестабільність тестових середовищ партнерів	Ескалація до менеджменту партнерів, перегляд пріоритетів, реалізація тимчасових напівручних процесів	Реалізація альтернативних каналів обміну (файли, веб-форми), перенесення частини функцій у напівручний режим до завершення інтеграції
4	Втрата/пошкодження даних	Резервне копіювання за регламентом, регулярні тестові відновлення, моніторинг backup	Помилки або попередження у логах backup-процесів, тривале виконання джоб	Негайна перевірка конфігурацій, ручний запуск резервування, залучення DevOps	Відновлення даних з останніх валідних копій, інформування замовника, аналіз втрат і компенсаційні заходи, посилення політики резервування

Продовження табл. Ж.1

1	2	3	4	5	6
5	Помилки міграції даних	План міграції, тестові міграції на стендах, валідаційні скрипти, погоджені правила трансформації	Зростання кількості неконсистентних записів, скарги на некоректні дані	Зупинка поточного етапу міграції, корекція правил, перезапуск на тестовому середовищі	Частковий/повний rollback, виправлення даних скриптами, можливе повторне завантаження з перевіркою кожного етапу
6	Затримка Kubernetes / CI/CD	Раннє планування DevOps-робіт, виділений DevOps, використання готових шаблонів, документований pipeline	Багато ручних розгортань, часті збої стендів, затримки з релізами	Пріоритизація DevOps-задач, тимчасове спрощення pipeline (мінімальний CI/CD), залучення зовнішнього експерта	Перехід на тимчасову простішу інфраструктуру (наприклад, VM без K8s), далі поступове повернення до цільової схеми
7	Збій хостингу	Вибір надійного провайдера, SLA, багатозонне розміщення, моніторинг доступності	Короткочасні недоступності середовища, нестабільна робота сервісів	Звернення до провайдера, перемикання трафіку на резервний вузол, збільшення моніторингових алертів	Активація DR-плану, міграція на резервну інфраструктуру, інформування користувачів і замовника, післяінцидентний аналіз
8	Витік персональних даних	Впровадження ролей та прав, шифрування, логування доступів, двофакторна автентифікація, політики безпеки	Підозріла активність у логах, нетипові авторизації, зростання кількості помилок авторизації	Оперативний аудит логів, блокування підозрілих облікових записів, зміна паролів, посилення політик	Локалізація інциденту, інформування регулятора та клієнтів (за вимогами НБУ/ЗУ), технічне закриття діри, перегляд архітектури безпеки
9	Зміна вимог НБУ	Моніторинг регуляторних змін, участь у профільних робочих групах, закладання регуляторного резерву в бюджеті й термінах	Поява проєктів нових нормативних актів, листи-роз'яснення НБУ	Оцінка впливу змін, оновлення беклогу регуляторних задач, погодження з замовником коригування плану	Формування окремого релізу для імплементації вимог, перерозподіл бюджету й пріоритетів

Продовження табл. Ж.1

1	2	3	4	5	6
10	Проблеми UI/UX	Залучення UX-дизайнера, прототипування у Figma, юзабіліті-тести з агентами/брокерами	Низькі оцінки демо-версій, скарги на складність інтерфейсу, тривалий онбординг	Ретроспектива UX-рішень, швидкі UX-поліпшення у найбільш критичних сценаріях, додаткове навчання користувачів	Переробка ключових екранів, впровадження полегшеного режиму/майстрів, оновлення навчальних матеріалів
11	Помилки в преміях/договорах	Формалізовані формули й тарифи, подвійна перевірка, автотести розрахунків	Виявлення розбіжностей між розрахунками системи й ручними, збільшення кількості запитань від агентів	Зупинка використання проблемних продуктів, виправлення формул, додаткове тестування	Перерахунок постраждалих договорів, інформування клієнтів, компенсаційні заходи, затвердження оновлених алгоритмів
12	Часті зміни пріоритетів	Формалізація процесу управління змінами (Change Request), фіксація спринтів, product-vision	Часті запити на зміну вже погоджених задач, зростання Work in Progress	Переговори з замовником, введення правил: зміни тільки через CR і з перенесенням інших задач	Ре-планування релізів, скорочення обсягу менш критичних функцій, можливий перегляд трикутника «обсяг–час–вартість»
13	Відсутність вихідних даних	Чіткий список необхідних артефактів, дедлайни, відповідальні зі сторони замовника	Прострочені дедлайни надання тарифів/шаблонів, блокуючі задачі в беклозі	Нагадування та ескалація до керівництва замовника, тимчасова робота з тестовими/чорнетковими даними	Перенесення залежних функцій у наступні релізи, фіксація впливу в протоколах, можливий перегляд календарного плану
14	Пасивність замовника	Регулярні статус-зустрічі, затверджений графік рев'ю, протоколи рішень	Відсутність зворотного зв'язку по надісланих документах, перенесення зустрічей	Ескалація до стейкхолдерів вищого рівня, встановлення «дефолтних» рішень у разі відсутності відповіді	Фіксація рішень про прийняття артефактів за замовчуванням, офіційний лист про вплив затримок на строки та бюджет

Продовження табл. Ж.1

1	2	3	4	5	6
15	Затримка затвердження документації	Підготовка документів завчасно, узгодження структури паспорта, попередні неформальні погодження	Документи довго «на підписі», немає офіційного старту	Додаткові зустрічі для проходження по документам «голосом», розбиття затвердження на етапи	Фіксація зміщення стартових дат, перегляд плану, можливе поетапне фінансування з частковим запуском робіт
16	Низький рівень комунікації	Регулярні daily, ретроспективи, прозорі канали комунікації,	Нерозуміння статусу задач, дублювання роботи, зростання кількості конфліктів	Фасилітовані командні зустрічі, уточнення ролей і відповідальностей, введення стандартів оновлення статусів	Заміна/перерозподіл ролей, залучення зовнішнього фасилітатора/HR, можлива ротація окремих учасників
17	Недостатній досвід PM	Менторинг від більш досвідченого PM, навчання, участь у професійних спільнотах	PM складно аргументує вибір підходів, проблеми з плануванням спринтів і фаз	Залучення ментора/консультанта, спільне планування критичних етапів, рев'ю планів	У разі системних проблем – часткове делегування функцій (release-менеджер, Scrum-Master) або заміна PM
18	Вихід ключового фахівця	Документація рішень, знання-sharing, парне програмування, план наступності (back-up на роль)	Зростання bus-factor, накопичення завдань, які «вміє тільки він/вона»	Прискорене передавання знань, шадовінг наступника, корекція пріоритетів	Терміновий пошук заміни/аутсорс, тимчасове зниження обсягу релізів, можливе перегрупування архітектури для зменшення залежності
19	Недостатня предметна компетентність	Навчання по продукту, спільні сесії з експертами ринку, глосарій термінів	Багато уточнюючих питань, помилки у вимогах та моделях даних	Додаткові воркшопи з предметної області, посилення участі BA, рев'ю рішень експертами	Зафіксовані помилки – виправлення рішень, уточнення функціоналу, можливе долучення зовнішнього доменного консультанта

Продовження табл. Ж.1

1	2	3	4	5	6
20	Складність найму	Аналіз ринку, early-recruiting, резервний пул кандидатів, гнучкі умови співпраці	Довго не закриваються вакансії, затримка старту технічних задач	Перегляд вимог до кандидатів, підсилення команди аутсорсингом/аутстафом, тимчасове розширення ролей	Суттєве перепланування графіка виконання, скорочення обсягу на перший реліз, перехід частини задач у супровідну фазу
21	Перевантаження команди	Адекватне планування спринтів, ліміти WIP, контроль over-time, прозорі пріоритети	Регулярні переробки, падіння продуктивності, зниження мотивації	Перерозподіл задач, тимчасове зменшення обсягу спринтів, додаткові day-off / відгул	Перебудова плану релізів, найм додаткових фахівців, зміна підходу до оцінки й пріоритизації
22	Погане планування беклогу	Встановлення стандарту оцінки (story points), участь всієї команди в плануванні, регулярні ретроспективи	Постійне невиконання плану спринтів, великий carry-over	Детальний аналіз причин, переоцінка складності задач, зміна підходу до формування спринтів	Перебудова всієї дорожньої карти, можлива зміна методології (наприклад, більший акцент на Kanban)
23	Конфлікти в команді	Чітко описані ролі (RACI), кодекс командної взаємодії, фасилітовані зустрічі	Регулярні суперечки, затягнуті обговорення, «протягування» рішень	Медіація конфліктів, робота з очікуваннями стейкхолдерів, оновлення RACI	Якщо конфлікти не вирішуються – заміна або розведення учасників по різних напрямках, перегляд структури управління
24	Перевищення бюджету	Резерв на ризики, контроль бюджету щомісяця, облік витрат по статтях	Прогноз перевищення бюджету за результатами звітного періоду	Оптимізація витрат, відсікання low-value функцій, переговори щодо перерозподілу бюджету	Офіційний перегляд бюджету/обсягу/строків, можлива фіксація мінімально життєздатного продукту (MVP)

Закінчення табл. Ж.1

1	2	3	4	5	6
25	Невчасне фінансування	Фінансовий календар, договори з чіткими датами оплат, інформування замовника	Затримка оплати чергового етапу, сигнали від фінслужби	Нагадування та офіційні листи, тимчасове обмеження робіт до критичних	Призупинка проєкту до надходження платежів, погодження нового графіка фінансування
26	Перевищення вартості підтримки	Узгоджений SLA та обсяг підтримки, прогноз навантаження, автоматизація моніторингу	Зростання кількості інцидентів, що не вкладаються в узгоджений обсяг	Оптимізація процесу підтримки, запровадження self-service-функцій,	Перегляд умов SLA та тарифів, перехід на іншу модель обслуговування
27	Відставання від графіка	Реалістичний план, буфери, контроль прогресу по віхах	Регулярне невиконання проміжних планів, попередження про зсув релізів	Аналіз критичного шляху, перерозподіл ресурсів на критичні задачі,	Офіційний перегляд загального графіка, зміна обсягу проєкту (MVP/фази), додатковий найм при згоді замовника
28	Масовані обстріли	Розподілена команда по різних регіонах/країнах, віддалений режим, резервні канали зв’язку	Часті повітряні тривоги, простої роботи, неможливість працювати з офісу	Перехід у п віддалений формат, гнучкий графік, перерозподіл задач на менш уражені регіони	Тимчасова зупинка частини робіт, зміщення дедлайнів, перенесення критичних функцій на більш безпечні ділянки/команди
29	Мобілізація, ВЛК	План наступності по ключових ролях, дублювання функцій, кадровий резерв	Інформація про можливу мобілізацію членів команди, часті відлучки на ВЛК	Швидке передавання знань дублеру, оновлення планів з урахуванням можливої відсутності	Перерозподіл ролей, найм заміни (в т.ч. з інших країн), коригування графіка робіт
30	Відключення електроенергії	Ноутбуки, UPS/генератори для ключових точок, хмарні середовища, мобільний інтернет	Перебої зі зв’язком, регулярні відключення світла в окремих членів команди	Перерозподіл задач на тих, хто має стабільне живлення, корекція робочих годин	Переведення критичних сервісів у стійкі дата-центри, перегляд робочої моделі на користь більш асинхронної роботи

Додаток И

SQL-скрипт для створення БД

```
--
-- PostgreSQL database dump
--
-- Dumped from database version 16.3
-- Dumped by pg_dump version 17.5
-- Started on 2025-06-17 03:18:59
SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET transaction_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;
--
-- TOC entry 6 (class 2615 OID 144265)
-- Name: default; Type: SCHEMA; Schema: -; Owner: postgres
--
CREATE SCHEMA "default";
ALTER SCHEMA "default" OWNER TO postgres;
SET default_tablespace = '';
SET default_table_access_method = heap;
--
-- TOC entry 224 (class 1259 OID 144329)
-- Name: BusinessPartners; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."BusinessPartners" (
    "Id" uuid NOT NULL,
    "ParticipantId" uuid NOT NULL,
    "LegalName" character varying(256) NOT NULL
);
ALTER TABLE "default"."BusinessPartners" OWNER TO postgres;
--
-- TOC entry 231 (class 1259 OID 144406)
-- Name: BusinessPartnersToContracts; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."BusinessPartnersToContracts" (
    "BusinessPartnerId" uuid NOT NULL,
    "ContractId" uuid NOT NULL
);
ALTER TABLE "default"."BusinessPartnersToContracts" OWNER TO postgres;
--
-- TOC entry 230 (class 1259 OID 144396)
-- Name: ContractTemplateFiles; Type: TABLE; Schema: default; Owner: postgres
```

```
--
CREATE TABLE "default"."ContractTemplateFiles" (
    "Id" uuid NOT NULL,
    "Name" character varying(128) NOT NULL,
    "TemplateId" uuid NOT NULL
);
ALTER TABLE "default"."ContractTemplateFiles" OWNER TO postgres;
--
-- TOC entry 228 (class 1259 OID 144374)
-- Name: ContractTemplates; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."ContractTemplates" (
    "Id" uuid NOT NULL,
    "Name" character varying(128) NOT NULL,
    "Placeholders" text NOT NULL,
    "ProductId" uuid NOT NULL
);
ALTER TABLE "default"."ContractTemplates" OWNER TO postgres;
--
-- TOC entry 229 (class 1259 OID 144386)
-- Name: Contracts; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."Contracts" (
    "Id" uuid NOT NULL,
    "Number" character varying(64) NOT NULL,
    "SumInsured" numeric(18,2) NOT NULL,
    "TemplateId" uuid NOT NULL,
    "CreatedAt" timestamp with time zone NOT NULL,
    "StartDate" timestamp with time zone NOT NULL,
    "EndDate" timestamp with time zone NOT NULL
);
ALTER TABLE "default"."Contracts" OWNER TO postgres;
--
-- TOC entry 217 (class 1259 OID 144271)
-- Name: CustomerCompanies; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."CustomerCompanies" (
    "Id" uuid NOT NULL,
    "SystemName" text NOT NULL,
    "LegalName" text NOT NULL,
    "RegistrationCode" text NOT NULL,
    "Address" text,
    "PhoneNumber" text,
    "Email" text,
    "IsActive" boolean NOT NULL,
    "CreatedAt" timestamp with time zone NOT NULL
);
ALTER TABLE "default"."CustomerCompanies" OWNER TO postgres;
--
-- TOC entry 232 (class 1259 OID 144421)
```

```
-- Name: IndividualPersonToContracts; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."IndividualPersonToContracts" (
    "IndividualPersonId" uuid NOT NULL,
    "ContractId" uuid NOT NULL
);
ALTER TABLE "default"."IndividualPersonToContracts" OWNER TO postgres;
--
-- TOC entry 225 (class 1259 OID 144339)
-- Name: IndividualPersons; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."IndividualPersons" (
    "Id" uuid NOT NULL,
    "ParticipantId" uuid NOT NULL,
    "FullName" character varying(256) NOT NULL,
    "DateOfBirthday" date NOT NULL,
    "Passport" character varying(64) NOT NULL
);
ALTER TABLE "default"."IndividualPersons" OWNER TO postgres;
--
-- TOC entry 218 (class 1259 OID 144278)
-- Name: InsuranceCompanies; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."InsuranceCompanies" (
    "Id" uuid NOT NULL,
    "SystemName" character varying(128) NOT NULL,
    "LegalName" character varying(256) NOT NULL,
    "RegistrationCode" character varying(64) NOT NULL,
    "AccountNumber" character varying(64) NOT NULL,
    "Address" character varying(256) NOT NULL,
    "PhoneNumber" character varying(32) NOT NULL,
    "Email" character varying(128) NOT NULL,
    "CreatedAt" timestamp with time zone NOT NULL
);
ALTER TABLE "default"."InsuranceCompanies" OWNER TO postgres;
--
-- TOC entry 221 (class 1259 OID 144297)
-- Name: InsuranceParticipants; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."InsuranceParticipants" (
    "Id" uuid NOT NULL,
    "Type" integer NOT NULL,
    "CompanyId" uuid NOT NULL,
    "RegistrationCode" character varying(64) NOT NULL,
    "Address" character varying(256) NOT NULL,
    "PhoneNumber" character varying(32) NOT NULL,
    "Email" character varying(128) NOT NULL
);
ALTER TABLE "default"."InsuranceParticipants" OWNER TO postgres;
--
```

```
-- TOC entry 227 (class 1259 OID 144364)
-- Name: InsuranceProducts; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."InsuranceProducts" (
    "Id" uuid NOT NULL,
    "TypeId" uuid NOT NULL,
    "Name" character varying(128) NOT NULL
);
ALTER TABLE "default"."InsuranceProducts" OWNER TO postgres;
--
-- TOC entry 223 (class 1259 OID 144319)
-- Name: InsuranceTypes; Type: TABLE; Schema: default; Owner: postgres
--

CREATE TABLE "default"."InsuranceTypes" (
    "Id" uuid NOT NULL,
    "Name" character varying(128) NOT NULL,
    "CompanyId" uuid NOT NULL
);
ALTER TABLE "default"."InsuranceTypes" OWNER TO postgres;
--
-- TOC entry 219 (class 1259 OID 144285)
-- Name: OutboxMessage; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."OutboxMessage" (
    "Id" uuid NOT NULL,
    "Type" text NOT NULL,
    "Content" text NOT NULL,
    "OccuredOnUtc" timestamp with time zone NOT NULL,
    "ProcessedOnUtc" timestamp with time zone,
    "Error" text
);
ALTER TABLE "default"."OutboxMessage" OWNER TO postgres;
--
-- TOC entry 220 (class 1259 OID 144292)
-- Name: Roles; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."Roles" (
    "Id" uuid NOT NULL,
    "Name" character varying(128) NOT NULL
);
ALTER TABLE "default"."Roles" OWNER TO postgres;

--
-- TOC entry 226 (class 1259 OID 144349)
-- Name: UserRoles; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."UserRoles" (
    "UserId" uuid NOT NULL,
    "RoleId" uuid NOT NULL
```

```
);
ALTER TABLE "default"."UserRoles" OWNER TO postgres;
--
-- TOC entry 222 (class 1259 OID 144307)
-- Name: Users; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."Users" (
    "Id" uuid NOT NULL,
    "Email" text NOT NULL,
    "FullName" text NOT NULL,
    "CompanyId" uuid NOT NULL,
    "CreatedAt" timestamp with time zone NOT NULL
);
ALTER TABLE "default"."Users" OWNER TO postgres;
--
-- TOC entry 216 (class 1259 OID 144266)
-- Name: __EFMigrationsHistory; Type: TABLE; Schema: default; Owner: postgres
--
CREATE TABLE "default"."__EFMigrationsHistory" (
    "MigrationId" character varying(150) NOT NULL,
    "ProductVersion" character varying(32) NOT NULL
);
ALTER TABLE "default"."__EFMigrationsHistory" OWNER TO postgres;
--
-- TOC entry 4975 (class 0 OID 144329)
-- Dependencies: 224
-- Data for Name: BusinessPartners; Type: TABLE DATA; Schema: default; Owner:
postgres
--
COPY "default"."BusinessPartners" ("Id", "ParticipantId", "LegalName") FROM stdin;
\.
--
-- TOC entry 4982 (class 0 OID 144406)
-- Dependencies: 231
-- Data for Name: BusinessPartnersToContracts; Type: TABLE DATA; Schema: default;
Owner: postgres
--
COPY "default"."BusinessPartnersToContracts" ("BusinessPartnerId", "ContractId")
FROM stdin;
\.
--
-- TOC entry 4981 (class 0 OID 144396)
-- Dependencies: 230
-- Data for Name: ContractTemplateFiles; Type: TABLE DATA; Schema: default; Owner:
postgres
--
COPY "default"."ContractTemplateFiles" ("Id", "Name", "TemplateId") FROM stdin;
\.
--
-- TOC entry 4979 (class 0 OID 144374)
```

```
-- Dependencies: 228
-- Data for Name: ContractTemplates; Type: TABLE DATA; Schema: default; Owner:
postgres
--
COPY "default"."ContractTemplates" ("Id", "Name", "Placeholders", "ProductId") FROM
stdin;
\.
--
-- TOC entry 4980 (class 0 OID 144386)
-- Dependencies: 229
-- Data for Name: Contracts; Type: TABLE DATA; Schema: default; Owner: postgres
--
COPY "default"."Contracts" ("Id", "Number", "SumInsured", "TemplateId",
"CreatedAt", "StartDate", "EndDate") FROM stdin;
\.
--
-- TOC entry 4968 (class 0 OID 144271)
-- Dependencies: 217
-- Data for Name: CustomerCompanies; Type: TABLE DATA; Schema: default; Owner:
postgres
--
COPY "default"."CustomerCompanies" ("Id", "SystemName", "LegalName",
"RegistrationCode", "Address", "PhoneNumber", "Email", "IsActive", "CreatedAt")
FROM stdin;
\.
--
-- TOC entry 4983 (class 0 OID 144421)
-- Dependencies: 232
-- Data for Name: IndividualPersonToContracts; Type: TABLE DATA; Schema: default;
Owner: postgres
--
COPY "default"."IndividualPersonToContracts" ("IndividualPersonId", "ContractId")
FROM stdin;
\.
--
-- TOC entry 4976 (class 0 OID 144339)
-- Dependencies: 225
-- Data for Name: IndividualPersons; Type: TABLE DATA; Schema: default; Owner:
postgres
--
COPY "default"."IndividualPersons" ("Id", "ParticipantId", "FullName",
"DateOfBirthday", "Passport") FROM stdin;
\.

--
-- TOC entry 4969 (class 0 OID 144278)
-- Dependencies: 218
-- Data for Name: InsuranceCompanies; Type: TABLE DATA; Schema: default; Owner:
postgres
--
```

```
COPY "default"."InsuranceCompanies" ("Id", "SystemName", "LegalName",
"RegistrationCode", "AccountNumber", "Address", "PhoneNumber", "Email",
"CreatedAt") FROM stdin;
\.
--
-- TOC entry 4972 (class 0 OID 144297)
-- Dependencies: 221
-- Data for Name: InsuranceParticipants; Type: TABLE DATA; Schema: default; Owner:
postgres
--
COPY "default"."InsuranceParticipants" ("Id", "Type", "CompanyId",
"RegistrationCode", "Address", "PhoneNumber", "Email") FROM stdin;
\.
--
-- TOC entry 4978 (class 0 OID 144364)
-- Dependencies: 227
-- Data for Name: InsuranceProducts; Type: TABLE DATA; Schema: default; Owner:
postgres
--
COPY "default"."InsuranceProducts" ("Id", "TypeId", "Name") FROM stdin;
\.
--
-- TOC entry 4974 (class 0 OID 144319)
-- Dependencies: 223
-- Data for Name: InsuranceTypes; Type: TABLE DATA; Schema: default; Owner:
postgres
--
COPY "default"."InsuranceTypes" ("Id", "Name", "CompanyId") FROM stdin;
\.
--
-- TOC entry 4970 (class 0 OID 144285)
-- Dependencies: 219
-- Data for Name: OutboxMessage; Type: TABLE DATA; Schema: default; Owner: postgres
--
COPY "default"."OutboxMessage" ("Id", "Type", "Content", "OccuredOnUtc",
"ProcessedOnUtc", "Error") FROM stdin;
\.
--
-- TOC entry 4971 (class 0 OID 144292)
-- Dependencies: 220
-- Data for Name: Roles; Type: TABLE DATA; Schema: default; Owner: postgres
--
COPY "default"."Roles" ("Id", "Name") FROM stdin;
\.
--
-- TOC entry 4977 (class 0 OID 144349)
-- Dependencies: 226
-- Data for Name: UserRoles; Type: TABLE DATA; Schema: default; Owner: postgres
--
COPY "default"."UserRoles" ("UserId", "RoleId") FROM stdin;
```

```
\.
--
-- TOC entry 4973 (class 0 OID 144307)
-- Dependencies: 222
-- Data for Name: Users; Type: TABLE DATA; Schema: default; Owner: postgres
--
COPY "default"."Users" ("Id", "Email", "FullName", "CompanyId", "CreatedAt") FROM
stdin;
\.
--
-- TOC entry 4967 (class 0 OID 144266)
-- Dependencies: 216
-- Data for Name: __EFMigrationsHistory; Type: TABLE DATA; Schema: default; Owner:
postgres
--
COPY "default"."__EFMigrationsHistory" ("MigrationId", "ProductVersion") FROM
stdin;
20250617001522_Init 9.0.3
\.
--
-- TOC entry 4781 (class 2606 OID 144333)
-- Name: BusinessPartners PK_BusinessPartners; Type: CONSTRAINT; Schema: default;
Owner: postgres
--
ALTER TABLE ONLY "default"."BusinessPartners"
    ADD CONSTRAINT "PK_BusinessPartners" PRIMARY KEY ("Id");
--
-- TOC entry 4805 (class 2606 OID 144410)
-- Name: BusinessPartnersToContracts PK_BusinessPartnersToContracts; Type:
CONSTRAINT; Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."BusinessPartnersToContracts"
    ADD CONSTRAINT "PK_BusinessPartnersToContracts" PRIMARY KEY
("BusinessPartnerId", "ContractId");
--
-- TOC entry 4802 (class 2606 OID 144400)
-- Name: ContractTemplateFiles PK_ContractTemplateFiles; Type: CONSTRAINT; Schema:
default; Owner: postgres
--
ALTER TABLE ONLY "default"."ContractTemplateFiles"
    ADD CONSTRAINT "PK_ContractTemplateFiles" PRIMARY KEY ("Id");
--
-- TOC entry 4795 (class 2606 OID 144380)
-- Name: ContractTemplates PK_ContractTemplates; Type: CONSTRAINT; Schema: default;
Owner: postgres
--
ALTER TABLE ONLY "default"."ContractTemplates"
    ADD CONSTRAINT "PK_ContractTemplates" PRIMARY KEY ("Id");
--
-- TOC entry 4799 (class 2606 OID 144390)
```

```
-- Name: Contracts PK_Contracts; Type: CONSTRAINT; Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."Contracts"
    ADD CONSTRAINT "PK_Contracts" PRIMARY KEY ("Id");
--
-- TOC entry 4755 (class 2606 OID 144277)
-- Name: CustomerCompanies PK_CustomerCompanies; Type: CONSTRAINT; Schema: default;
Owner: postgres
--
ALTER TABLE ONLY "default"."CustomerCompanies"
    ADD CONSTRAINT "PK_CustomerCompanies" PRIMARY KEY ("Id");
--
-- TOC entry 4808 (class 2606 OID 144425)
-- Name: IndividualPersonToContracts PK_IndividualPersonToContracts; Type:
CONSTRAINT; Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."IndividualPersonToContracts"
    ADD CONSTRAINT "PK_IndividualPersonToContracts" PRIMARY KEY
("IndividualPersonId", "ContractId");
--
-- TOC entry 4784 (class 2606 OID 144343)
-- Name: IndividualPersons PK_IndividualPersons; Type: CONSTRAINT; Schema: default;
Owner: postgres
--
ALTER TABLE ONLY "default"."IndividualPersons"
    ADD CONSTRAINT "PK_IndividualPersons" PRIMARY KEY ("Id");
--
-- TOC entry 4761 (class 2606 OID 144284)
-- Name: InsuranceCompanies PK_InsuranceCompanies; Type: CONSTRAINT; Schema:
default; Owner: postgres
--
ALTER TABLE ONLY "default"."InsuranceCompanies"
    ADD CONSTRAINT "PK_InsuranceCompanies" PRIMARY KEY ("Id");
--
-- TOC entry 4771 (class 2606 OID 144301)
-- Name: InsuranceParticipants PK_InsuranceParticipants; Type: CONSTRAINT; Schema:
default; Owner: postgres
--
ALTER TABLE ONLY "default"."InsuranceParticipants"
    ADD CONSTRAINT "PK_InsuranceParticipants" PRIMARY KEY ("Id");
--
-- TOC entry 4791 (class 2606 OID 144368)
-- Name: InsuranceProducts PK_InsuranceProducts; Type: CONSTRAINT; Schema: default;
Owner: postgres
--
ALTER TABLE ONLY "default"."InsuranceProducts"
    ADD CONSTRAINT "PK_InsuranceProducts" PRIMARY KEY ("Id");
--
-- TOC entry 4778 (class 2606 OID 144323)
```

```
-- Name: InsuranceTypes PK_InsuranceTypes; Type: CONSTRAINT; Schema: default;
Owner: postgres
--
ALTER TABLE ONLY "default"."InsuranceTypes"
    ADD CONSTRAINT "PK_InsuranceTypes" PRIMARY KEY ("Id");
--
-- TOC entry 4763 (class 2606 OID 144291)
-- Name: OutboxMessage PK_OutboxMessage; Type: CONSTRAINT; Schema: default; Owner:
postgres
--
ALTER TABLE ONLY "default"."OutboxMessage"
    ADD CONSTRAINT "PK_OutboxMessage" PRIMARY KEY ("Id");
--
-- TOC entry 4766 (class 2606 OID 144296)
-- Name: Roles PK_Roles; Type: CONSTRAINT; Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."Roles"
    ADD CONSTRAINT "PK_Roles" PRIMARY KEY ("Id");
--
-- TOC entry 4787 (class 2606 OID 144353)
-- Name: UserRoles PK_UserRoles; Type: CONSTRAINT; Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."UserRoles"
    ADD CONSTRAINT "PK_UserRoles" PRIMARY KEY ("UserId", "RoleId");
--
-- TOC entry 4774 (class 2606 OID 144313)
-- Name: Users PK_Users; Type: CONSTRAINT; Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."Users"
    ADD CONSTRAINT "PK_Users" PRIMARY KEY ("Id");
--
-- TOC entry 4753 (class 2606 OID 144270)
-- Name: __EFMigrationsHistory PK___EFMigrationsHistory; Type: CONSTRAINT; Schema:
default; Owner: postgres
--
ALTER TABLE ONLY "default"."__EFMigrationsHistory"
    ADD CONSTRAINT "PK___EFMigrationsHistory" PRIMARY KEY ("MigrationId");
--
-- TOC entry 4803 (class 1259 OID 144437)
-- Name: IX_BusinessPartnersToContracts_ContractId; Type: INDEX; Schema: default;
Owner: postgres
--
CREATE INDEX "IX_BusinessPartnersToContracts_ContractId" ON
"default"."BusinessPartnersToContracts" USING btree ("ContractId");
--
-- TOC entry 4779 (class 1259 OID 144436)
-- Name: IX_BusinessPartners_ParticipantId; Type: INDEX; Schema: default; Owner:
postgres
--
```

```
CREATE INDEX "IX_BusinessPartners_ParticipantId" ON "default"."BusinessPartners"
USING btree ("ParticipantId");
--
-- TOC entry 4800 (class 1259 OID 144440)
-- Name: IX_ContractTemplateFiles_TemplateId; Type: INDEX; Schema: default; Owner:
postgres
--
CREATE INDEX "IX_ContractTemplateFiles_TemplateId" ON
"default"."ContractTemplateFiles" USING btree ("TemplateId");
--
-- TOC entry 4792 (class 1259 OID 144441)
-- Name: IX_ContractTemplates_Name; Type: INDEX; Schema: default; Owner: postgres
--
CREATE UNIQUE INDEX "IX_ContractTemplates_Name" ON "default"."ContractTemplates"
USING btree ("Name");
--
-- TOC entry 4793 (class 1259 OID 144442)
-- Name: IX_ContractTemplates_ProductId; Type: INDEX; Schema: default; Owner:
postgres
--
CREATE INDEX "IX_ContractTemplates_ProductId" ON "default"."ContractTemplates"
USING btree ("ProductId");
--
-- TOC entry 4796 (class 1259 OID 144438)
-- Name: IX_Contracts_Number; Type: INDEX; Schema: default; Owner: postgres
--
CREATE UNIQUE INDEX "IX_Contracts_Number" ON "default"."Contracts" USING btree
("Number");
--
-- TOC entry 4797 (class 1259 OID 144439)
-- Name: IX_Contracts_TemplateId; Type: INDEX; Schema: default; Owner: postgres
--
CREATE INDEX "IX_Contracts_TemplateId" ON "default"."Contracts" USING btree
("TemplateId");
--
-- TOC entry 4806 (class 1259 OID 144444)
-- Name: IX_IndividualPersonToContracts_ContractId; Type: INDEX; Schema: default;
Owner: postgres
--
CREATE INDEX "IX_IndividualPersonToContracts_ContractId" ON
"default"."IndividualPersonToContracts" USING btree ("ContractId");
--
-- TOC entry 4782 (class 1259 OID 144443)
-- Name: IX_IndividualPersons_ParticipantId; Type: INDEX; Schema: default; Owner:
postgres
--
CREATE INDEX "IX_IndividualPersons_ParticipantId" ON "default"."IndividualPersons"
USING btree ("ParticipantId");
--
-- TOC entry 4756 (class 1259 OID 144445)
```

```
-- Name: IX_InsuranceCompanies_AccountNumber; Type: INDEX; Schema: default; Owner:
postgres
--
CREATE UNIQUE INDEX "IX_InsuranceCompanies_AccountNumber" ON
"default"."InsuranceCompanies" USING btree ("AccountNumber");
--
-- TOC entry 4757 (class 1259 OID 144446)
-- Name: IX_InsuranceCompanies_Email; Type: INDEX; Schema: default; Owner: postgres
--
CREATE UNIQUE INDEX "IX_InsuranceCompanies_Email" ON "default"."InsuranceCompanies"
USING btree ("Email");
--
-- TOC entry 4758 (class 1259 OID 144447)
-- Name: IX_InsuranceCompanies_RegistrationCode; Type: INDEX; Schema: default;
Owner: postgres
--
CREATE UNIQUE INDEX "IX_InsuranceCompanies_RegistrationCode" ON
"default"."InsuranceCompanies" USING btree ("RegistrationCode");

-- TOC entry 4759 (class 1259 OID 144448)
-- Name: IX_InsuranceCompanies_SystemName; Type: INDEX; Schema: default; Owner:
postgres
--
CREATE UNIQUE INDEX "IX_InsuranceCompanies_SystemName" ON
"default"."InsuranceCompanies" USING btree ("SystemName");
--
-- TOC entry 4767 (class 1259 OID 144449)
-- Name: IX_InsuranceParticipants_CompanyId; Type: INDEX; Schema: default; Owner:
postgres
--
CREATE INDEX "IX_InsuranceParticipants_CompanyId" ON
"default"."InsuranceParticipants" USING btree ("CompanyId");
--
-- TOC entry 4768 (class 1259 OID 144450)
-- Name: IX_InsuranceParticipants_Email; Type: INDEX; Schema: default; Owner:
postgres
--
CREATE UNIQUE INDEX "IX_InsuranceParticipants_Email" ON
"default"."InsuranceParticipants" USING btree ("Email");
--
-- TOC entry 4769 (class 1259 OID 144451)
-- Name: IX_InsuranceParticipants_RegistrationCode; Type: INDEX; Schema: default;
Owner: postgres
--
CREATE UNIQUE INDEX "IX_InsuranceParticipants_RegistrationCode" ON
"default"."InsuranceParticipants" USING btree ("RegistrationCode");
--
-- TOC entry 4788 (class 1259 OID 144452)
-- Name: IX_InsuranceProducts_Name; Type: INDEX; Schema: default; Owner: postgres
--
```

```
CREATE UNIQUE INDEX "IX_InsuranceProducts_Name" ON "default"."InsuranceProducts"
USING btree ("Name");

--
-- TOC entry 4789 (class 1259 OID 144453)
-- Name: IX_InsuranceProducts_TypeId; Type: INDEX; Schema: default; Owner: postgres
--
CREATE INDEX "IX_InsuranceProducts_TypeId" ON "default"."InsuranceProducts" USING
btree ("TypeId");
--
-- TOC entry 4775 (class 1259 OID 144454)
-- Name: IX_InsuranceTypes_CompanyId; Type: INDEX; Schema: default; Owner: postgres
--
CREATE INDEX "IX_InsuranceTypes_CompanyId" ON "default"."InsuranceTypes" USING
btree ("CompanyId");
--
-- TOC entry 4776 (class 1259 OID 144455)
-- Name: IX_InsuranceTypes_Name; Type: INDEX; Schema: default; Owner: postgres
--
CREATE UNIQUE INDEX "IX_InsuranceTypes_Name" ON "default"."InsuranceTypes" USING
btree ("Name");
--
-- TOC entry 4764 (class 1259 OID 144456)
-- Name: IX_Roles_Name; Type: INDEX; Schema: default; Owner: postgres
--
CREATE UNIQUE INDEX "IX_Roles_Name" ON "default"."Roles" USING btree ("Name");
--
-- TOC entry 4785 (class 1259 OID 144457)
-- Name: IX_UserRoles_RoleId; Type: INDEX; Schema: default; Owner: postgres
--
CREATE INDEX "IX_UserRoles_RoleId" ON "default"."UserRoles" USING btree ("RoleId");
--
-- TOC entry 4772 (class 1259 OID 144458)
-- Name: IX_Users_CompanyId; Type: INDEX; Schema: default; Owner: postgres
--

CREATE INDEX "IX_Users_CompanyId" ON "default"."Users" USING btree ("CompanyId");
--
-- TOC entry 4820 (class 2606 OID 144411)
-- Name: BusinessPartnersToContracts
FK_BusinessPartnersToContracts_Contracts_ContractId; Type: FK CONSTRAINT; Schema:
default; Owner: postgres
--
ALTER TABLE ONLY "default"."BusinessPartnersToContracts"
    ADD CONSTRAINT "FK_BusinessPartnersToContracts_Contracts_ContractId" FOREIGN
KEY ("ContractId") REFERENCES "default"."Contracts"("Id") ON DELETE CASCADE;
--
-- TOC entry 4821 (class 2606 OID 144416)
```

```
-- Name: BusinessPartnersToContracts
FK_BusinessPartnersToContracts_InsuranceParticipants_BusinessP~; Type: FK
CONSTRAINT; Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."BusinessPartnersToContracts"
    ADD CONSTRAINT
"FK_BusinessPartnersToContracts_InsuranceParticipants_BusinessP~" FOREIGN KEY
("BusinessPartnerId") REFERENCES "default"."InsuranceParticipants"("Id") ON DELETE
CASCADE;
--
-- TOC entry 4812 (class 2606 OID 144334)
-- Name: BusinessPartners FK_BusinessPartners_InsuranceParticipants_ParticipantId;
Type: FK CONSTRAINT; Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."BusinessPartners"
    ADD CONSTRAINT "FK_BusinessPartners_InsuranceParticipants_ParticipantId"
FOREIGN KEY ("ParticipantId") REFERENCES "default"."InsuranceParticipants"("Id") ON
DELETE CASCADE;
--
-- TOC entry 4819 (class 2606 OID 144401)
-- Name: ContractTemplateFiles
FK_ContractTemplateFiles_ContractTemplates_TemplateId; Type: FK CONSTRAINT; Schema:
default; Owner: postgres
--
ALTER TABLE ONLY "default"."ContractTemplateFiles"
    ADD CONSTRAINT "FK_ContractTemplateFiles_ContractTemplates_TemplateId" FOREIGN
KEY ("TemplateId") REFERENCES "default"."ContractTemplates"("Id") ON DELETE
CASCADE;
--
-- TOC entry 4817 (class 2606 OID 144381)
-- Name: ContractTemplates FK_ContractTemplates_InsuranceProducts_ProductId; Type:
FK CONSTRAINT; Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."ContractTemplates"
    ADD CONSTRAINT "FK_ContractTemplates_InsuranceProducts_ProductId" FOREIGN KEY
("ProductId") REFERENCES "default"."InsuranceProducts"("Id") ON DELETE CASCADE;
--
-- TOC entry 4818 (class 2606 OID 144391)
-- Name: Contracts FK_Contracts_ContractTemplates_TemplateId; Type: FK CONSTRAINT;
Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."Contracts"
    ADD CONSTRAINT "FK_Contracts_ContractTemplates_TemplateId" FOREIGN KEY
("TemplateId") REFERENCES "default"."ContractTemplates"("Id") ON DELETE CASCADE;
--
-- TOC entry 4822 (class 2606 OID 144426)
-- Name: IndividualPersonToContracts
FK_IndividualPersonToContracts_Contracts_ContractId; Type: FK CONSTRAINT; Schema:
default; Owner: postgres
--
```

```
ALTER TABLE ONLY "default"."IndividualPersonToContracts"
    ADD CONSTRAINT "FK_IndividualPersonToContracts_Contracts_ContractId" FOREIGN
KEY ("ContractId") REFERENCES "default"."Contracts"("Id") ON DELETE CASCADE;
--
-- TOC entry 4823 (class 2606 OID 144431)
-- Name: IndividualPersonToContracts
FK_IndividualPersonToContracts_InsuranceParticipants_Individua~; Type: FK
CONSTRAINT; Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."IndividualPersonToContracts"
    ADD CONSTRAINT
"FK_IndividualPersonToContracts_InsuranceParticipants_Individua~" FOREIGN KEY
("IndividualPersonId") REFERENCES "default"."InsuranceParticipants"("Id") ON DELETE
CASCADE;
--
-- TOC entry 4813 (class 2606 OID 144344)
-- Name: IndividualPersons
FK_IndividualPersons_InsuranceParticipants_ParticipantId; Type: FK CONSTRAINT;
Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."IndividualPersons"
    ADD CONSTRAINT "FK_IndividualPersons_InsuranceParticipants_ParticipantId"
FOREIGN KEY ("ParticipantId") REFERENCES "default"."InsuranceParticipants"("Id") ON
DELETE CASCADE;
--
-- TOC entry 4809 (class 2606 OID 144302)
-- Name: InsuranceParticipants
FK_InsuranceParticipants_CustomerCompanies_CompanyId; Type: FK CONSTRAINT; Schema:
default; Owner: postgres
--
ALTER TABLE ONLY "default"."InsuranceParticipants"
    ADD CONSTRAINT "FK_InsuranceParticipants_CustomerCompanies_CompanyId" FOREIGN
KEY ("CompanyId") REFERENCES "default"."CustomerCompanies"("Id") ON DELETE CASCADE;
-
-- TOC entry 4816 (class 2606 OID 144369)
-- Name: InsuranceProducts FK_InsuranceProducts_InsuranceTypes_TypeId; Type: FK
CONSTRAINT; Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."InsuranceProducts"
    ADD CONSTRAINT "FK_InsuranceProducts_InsuranceTypes_TypeId" FOREIGN KEY
("TypeId") REFERENCES "default"."InsuranceTypes"("Id") ON DELETE CASCADE;
--
-- TOC entry 4811 (class 2606 OID 144324)
-- Name: InsuranceTypes FK_InsuranceTypes_InsuranceCompanies_CompanyId; Type: FK
CONSTRAINT; Schema: default; Owner: postgres
--
ALTER TABLE ONLY "default"."InsuranceTypes"
    ADD CONSTRAINT "FK_InsuranceTypes_InsuranceCompanies_CompanyId" FOREIGN KEY
("CompanyId") REFERENCES "default"."InsuranceCompanies"("Id") ON DELETE CASCADE;
--
```

```
-- TOC entry 4814 (class 2606 OID 144354)
-- Name: UserRoles FK_UserRoles_Roles_RoleId; Type: FK CONSTRAINT; Schema: default;
Owner: postgres
--
ALTER TABLE ONLY "default"."UserRoles"
    ADD CONSTRAINT "FK_UserRoles_Roles_RoleId" FOREIGN KEY ("RoleId") REFERENCES
"default"."Roles"("Id") ON DELETE CASCADE;
--
-- TOC entry 4815 (class 2606 OID 144359)
-- Name: UserRoles FK_UserRoles_Users_UserId; Type: FK CONSTRAINT; Schema: default;
Owner: postgres
--
ALTER TABLE ONLY "default"."UserRoles"
    ADD CONSTRAINT "FK_UserRoles_Users_UserId" FOREIGN KEY ("UserId") REFERENCES
"default"."Users"("Id") ON DELETE CASCADE;
--
-- TOC entry 4810 (class 2606 OID 144314)
-- Name: Users FK_Users_CustomerCompanies_CompanyId; Type: FK CONSTRAINT; Schema:
default; Owner: postgres
--
ALTER TABLE ONLY "default"."Users"
    ADD CONSTRAINT "FK_Users_CustomerCompanies_CompanyId" FOREIGN KEY ("CompanyId")
REFERENCES "default"."CustomerCompanies"("Id") ON DELETE CASCADE;
-- Completed on 2025-06-17 03:18:59
--
-- PostgreSQL database dump complete
--
```