КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 – Комп'ютерні науки,
освітньо-наукова програма «Управління проєктами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Дослідження процесів управління проєктом розробки інтелектуальної інформаційної системи управління людськими ресурсами в ІТ-командах»

Студента 2-го курсу групи УП-21

Олексія КОРИТНОГО
(ім'я, прізвище)

(підпис студента)

Науковий керівник:

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

Вадим ЗЮЗЮН
(ім'я, прізвище)

(дата) *(підпис)*

Попередній захист:

(Висновок: "До захисту в Екзаменаційній комісії")

Завідувач кафедри
технологій управління (підпис) Віктор МОРОЗОВ (ім'я, прізвище) (дата)

Київ 2026

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

_____
"8" грудня 2025 року

З А В Д А Н Н Я
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Студент: Коритний Олексій Тимурович
Група: УП-21

1. Тема кваліфікаційної роботи: «Дослідження процесів управління проєктом розробки інтелектуальної інформаційної системи управління людськими ресурсами в ІТ-командах». Затверджена протоколом кафедри ТУ від 5 грудня 2025 року № 4 .

2. Строк подання студентом готової роботи – "21" травня 2026 року.

3. Цільова установка та вихідні дані до роботи: дослідження різних методів та інструментів для управління проєктом розробки інформаційної системи, їх використання у плануванні проєкту, ресурсів, бюджету та управлінні ризиками.

4. Зміст роботи: дослідження конкурентного середовища, обґрунтування доцільності та життєздатності проєкту, загальний опис проєкту, аналіз зацікавлених сторін проєкту, проведення SWOT-аналізу, вибір методології управління проєктом, формування беклогу продукту проєкту, організаційна структура проєкту, планування робіт проєкту у вигляді спринтів, управління

ресурсами та бюджетування проєкту, управління ризиками проєкту. Розробка інформаційної структури проєкту, створення концептуальної моделі бази даних, проєктування інтерфейсу користувача.

5. Перелік графічного матеріалу: актуальність, концептуальна та математична модель інформаційної системи, процес функціонування системи, User Story проєкту, беклог продукту проєкту, OBS структура проєкту, планування спринтів та діаграма Ганта, ризики проєкту, архітектура інформаційної системи, інформація про базу даних, сторінки сайту.

6. Календарний план виконання роботи

№ з/п	Назва частин роботи	План виконання роботи
1	Вивчення літературних джерел з предмету дослідження	11.01.25-30.01.25
2	Складання плану кваліфікаційної роботи	15.01.26-23.01.26
3	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін	26.01.26
4	Підготовка розділу 1	27.01.26-26.02.26
5	Підготовка розділу 2	27.02.26-16.03.26
6	Підготовка розділу 3	17.03.26-06.04.26
7	Підготовка розділу 4	07.04.26-26.04.26
8	Оформлення кваліфікаційної роботи	27.04.26-03.05.26
9	Передача кваліфікаційної роботи науковому керівникові	04.05.26
10	Передача кваліфікаційної роботи рецензенту для рецензування	08.05.26
11	Захист кваліфікаційної роботи	25.05.26-26.05.26

Дата видачі завдання «10» грудня 2025 р.

Керівник роботи доцент Вадим ЗЮЗЮН
(посада, ім'я, прізвище)

_____
(підпис)

Завдання прийняв до виконання студент групи УП-21

Олексій КОРИТНИЙ
(ім'я, прізвище)

_____
(підпис)

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему

«Дослідження процесів управління проєктом розробки інтелектуальної інформаційної системи управління людськими ресурсами в IT-командах»

Студент: Коритний Олексій Тимурович

Науковий керівник: Зюзюн Вадим Ігорович

Рік захисту – 2026

Метою кваліфікаційної роботи є дослідження, аналіз та оптимізація процесів управління проєктом для розробки інтелектуальної інформаційної системи управління людськими ресурсами (HRM) для IT-команд, що дозволить підвищити ефективність реалізації проєктів IT сектору, мінімізувати ризики розробки та забезпечує найкращі умови робочої навантаженості для працівників.

Наукова новизна дослідження полягає у розробці адаптивної концептуальної моделі управління IT-проєктами, яка, на відміну від існуючих, враховує специфіку розробки та інтеграції алгоритмів штучного інтелекту (інтелектуальних компонентів) у процеси управління людськими ресурсами. Модель передбачає удосконалення процесу формування проєктних IT-команд та покращення існуючих HR методологій для забезпечення здорового робочого середовища для працівників.

Практична цінність отриманих результатів полягає в: 1) автоматизації процесу рекрутингу та внутрішнього відбору, що базується на багатокритеріальному аналізі компетенцій (ролі, технічного стека, а також поєднання hard та soft skills); 2) оптимізації управління людськими ресурсами шляхом моніторингу завантаженості персоналу, що дозволяє оперативно виявляти ризики професійного вигорання або простою фахівців; підвищенні ефективності проєктного менеджменту завдяки впровадженню системи

підтримки прийняття рішень, яка прискорює формування збалансованих команд для досягнення цілей проєкту.

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

У *першому розділі* проведено комплексне дослідження предметної області, аналіз існуючих програмних рішень, літературних та інформаційних джерел щодо способів до вирішення проблем описаних роботою. Виконано SWOT-аналіз проєкту, ідентифіковано зацікавлені сторони, визначено задачі дослідження та сформульовано технічне завдання у вигляді паспорту проєкту. Також побудовано дерево причин та наслідків проєкту.

У *другому розділі* розроблено концептуальну модель інформаційної системи, здійснено математичну постановку задачі відповідно до розробленої моделі та представлено застосування методів моделювання у контексті підбору персоналу для нового IT-проєкту «HRM».

В *третьому розділі* було обґрунтовано вибір технології управління проєктом, визначено функціональні та нефункціональні вимоги до продукту, сформовано беклог, описано організаційну структуру управління проєктом, сплановано ресурси, бюджет та спринти з використанням Діаграми Ганта. Окрему увагу приділено ідентифікації ризиків та розробці карти їх управління.

В *четвертому розділі* наведено опис алгоритмів роботи з аналізом оцінювання людських ресурсів та розподіл за проєктами, концептуальне та даталогічне моделювання бази даних продукту проєкту та розробка back-end частини платформи.

Робота містить 106 сторінок без додатків, 25 рисунків та 39 таблиць. Додатки складають 12 сторінок.

Ключові слова: управління IT-проєктами, інтелектуальна інформаційна система, управління людськими ресурсами, штучний інтелект, гібридні методології, предиктивна аналітика, професійне вигорання, управління ризиками, IT-команда.

ЗМІСТ

ВСТУП 8
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ 11
1.1 Дослідження існуючих програмних рішень та формування nnnnпредметної області 11
1.2 Проведення аналізу літературних та інформаційних джерел щодо можливостей вирішення виявлених проблем 13
1.3 Ідентифікація зацікавлених сторін проєкту 14
1.4 SWOT-аналіз проєкту 16
1.5 Побудова дерева причин та наслідків 22
1.6 Постановка задачі дослідження. Формулювання технічного завдання на розробку у вигляді паспорту проєкту 23
РОЗДІЛ 2. ОПИС КОНЦЕПТУАЛЬНИХ МОДЕЛЕЙ ТА nnnФОРМАЛІЗАЦІЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ 27
2.1 Розробка концептуальної моделі інформаційної системи 27
2.1.1 Критерізація параметрів ресурсів моделей 31
2.2 Формалізація математичних моделей та постановка задачі в математичному вигляді 35
2.3 Застосування розроблених моделей в межах інтелектуальної інформаційної системи управління людськими ресурсами в ІТ-командах 37
РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ЛЮДСЬКИМИ РЕСУРСАМИ 41
3.1 Вибір технології управління проєктом 41
3.2 Визначення функціональних та нефункціональних вимог до продукту проєкту 44

3.3 Формування беклогу продукту проєкту 47
3.4 Організаційна структура проєкту 49
3.5 Розробка WBS проєкту 51
3.6 Планування ресурсного забезпечення проєкту та бюджету 53
3.7 Планування спринтів та візуалізація їх виконання за допомогою ProjectLibre 56
3.8 Ідентифікація проєктних ризиків та розробка карти управління ними 62
3.9 Управління процесами контролю якості в проєкті 69
РОЗДІЛ 4. РОЗРОБКА ОСНОВНИХ СКЛАДОВИХ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ 76
4.1 Опис алгоритмів роботи з аналізом оцінювання людських ресурсів та розподіл за проєктами 76
4.2 Концептуальне та даталогічне моделювання бази даних продукту проєкту 79
4.3 Розробка back-end частини вебплатформи 87
4.4 Розробка front-end частини вебплатформи 91
ВИСНОВКИ 97
ДОДАТКИ 107

ВСТУП

Актуальність теми дослідження. Основною розвитку IT-галузі є висока динаміка ринку, високотехнологічні вимоги до кінцевого результату продукту та критичною залежністю бізнесу від кваліфікації і психоемоційного стану співробітників. У цих умовах людський капітал стає головним активом, а ефективне управління ним – визначальним фактором конкурентоспроможності компаній. Не дивно, що компанії зі штатом більше десятки тисяч співробітників поступово зменшують кількість своїх співробітників на користь ШІ та інтелектуальних технологій. Традиційні системи управління людськими ресурсами (HRM) здебільшого зосереджені на адміністративно-облікових функціях і не здатні в режимі реального часу забезпечувати багатокритеріальний підбір команд, прогнозування професійного вигорання та дотримання балансу між роботою і особистим життям розробників.

Гострота питання зумовлена тим, що класичні методи HRM, які роками слугували фундаментом для стабільних корпорацій, сьогодні фактично безсилі перед динамікою українського IT-ринку, що функціонує в режимі постійного «гасіння пожеж». Проблема не лише у дефіциті кваліфікованих кадрів, а у критичній відсутності інструментів, здатних прогнозувати вигорання фахівців чи неефективність команд ще до того, як це стане фінансовим збитком для компанії. Нинішній менеджер проєктів змушений балансувати між сухою метрикою завантаженості та крихким емоційним станом розробників, які працюють під тиском блекаутів, мобілізаційних процесів та загальної невизначеності. Відтак, створення інтелектуальної системи, що здатна врахувати ці нелінійні фактори, є не просто технологічною забаганкою, а стратегічною необхідністю для виживання вітчизняних продуктових та аутсорсингових компаній.

Крім того, актуальність підсилюється «кризою довіри» до автоматизації як такої. Існуючі рішення часто грішать надмірною алгоритмізацією,

перетворюючи управління людьми на механічний перерозподіл ресурсів, що викликає лише роздратування та опір у творчих колективах. В умовах, коли кожна помилка у підборі або утриманні фахівця коштує компанії місяців простою, пошук балансу між математичною точністю Python-бібліотек та живою інтуїцією управлінця стає головним викликом для сучасної комп'ютерної науки. На основі вище зазначеного була сформована мета дослідження.

Мета дослідження полягає у розробці та створенні HRM системи, що забезпечує оптимальний підбір працівників враховуючи стандарти та правила умови праці зі збереженням work-life балансу та визначення ключових працівників на основі проєктних ролей, технічних знань та особистісних якостей.

Завдання дослідження:

1. Дослідити методології до підбору та визначення оптимальної команди для виконання IT-проєктів, визначити слабкі та сильні сторони, а також сформувати предметну область.
2. Проаналізувати літературні та інформаційні джерела щодо можливостей вирішення виявлених проблем, а також здійснити SWOT-аналіз проєкту та ідентифікувати зацікавлені сторони.
3. Сформулювати технічне завдання у вигляді паспорта проєкту, розробити концептуальну модель інформаційної системи та визначити математичну постановку задачі.
4. Обґрунтувати вибір методології управління проєктом, визначити організаційну структуру управління, сформувати беклог продукту та спланувати спринти з використанням діаграми Ґанта.
5. Ідентифікувати ризики та забезпечити підходи до їх вирішення (карта управління ризиками).
6. Створити алгоритм роботи вебплатформи, розробити концептуальну та даталогічну модель, запропонувати підхід до реалізації бази даних.
7. Реалізувати концепти вебплатформи.

Об'єкт дослідження – процеси управління персоналом та кадрами у інноваційних ІТ-проєктах у сфері software engineering та artificial intelligence.

Предмет дослідження – методології, інструменти та управлінські практики, що застосовуються при розробці інтелектуальної системи управління людськими ресурсами на основі моделей ШІ.

Наукова новизна дослідження полягає у розробці адаптивної концептуальної моделі управління ІТ-проєктами, яка, на відміну від існуючих, враховує специфіку розробки та інтеграції алгоритмів штучного інтелекту (інтелектуальних компонентів) у процеси управління людськими ресурсами. Модель передбачає удосконалення процесу формування проєктних ІТ-команд та покращення існуючих HR методологій для забезпечення здорового робочого середовища для працівників.

Практичне значення отриманих результатів полягає у:

1) Автоматизації процесу рекрутингу та внутрішнього відбору, що базується на багатокритеріальному аналізі компетенцій (ролі, технічного стека, а також поєднання hard та soft skills).
2) Оптимізації управління людськими ресурсами шляхом моніторингу завантаженості персоналу, що дозволяє оперативно виявляти ризики професійного вигорання або простою фахівців.
3) Підвищенні ефективності проєктного менеджменту завдяки впровадженню системи підтримки прийняття рішень, яка прискорює формування збалансованих команд для досягнення цілей проєкту.

Апробація результатів роботи. Результати дослідження були представлені на конференціях: 1) 2st International scientific and practical conference «Information Systems and Technology: Results and Prospects» (IST 2025), Kyiv, Ukraine, 2025; 2) Information Technology and Implementation (IT&Is), 21 November, 2025, Kyiv, Ukraine; 3) 4th International Scientific and Practical Conference «Modern Scientific Research: Theoretical and Practical Aspects», Riga, Latvia, 11-13 May 2026; 4) 8th International Scientific and Practical Conference «Modern Perspectives on Global Scientific Solutions», Bergen, Norway, 18-20 May 2026.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ

1.1 Дослідження існуючих програмних рішень та формування предметної області

Задля досягнення бажаного результату для реалізації «HRM», необхідно провести детальний аналіз існуючих рішень, аби визначити їх сильні та слабкі сторони, щоб сконцентрувати увагу до вирішення проблемних точок роботи аналогів. В результаті такої роботи можна точно визначити перелік вимог та функцій, які мають бути включені до застосунку, що розробляється.

Проаналізувавши ринок, одним з найбільш наближених застосунків є «MuchSkills» [1] – інструмент проєктного управління для Agile проєктів та моніторингу виконання задач. Вебзастосунок є інтеграційної фріланс базою працівників з багатьма характеристиками, зокрема понад 12000 різновидів технічних навичок, 8000 сертифікацій, можливість створення власних наборів навичок, аналітика найбільш необхідних навичок сучасності та підкріплення методологій від Gartner, Deloitte, Accenture, Google та інших. Окрім цього, деякі компанії, зокрема «Harald Pihl», визнають, що використовуються застосунок як HR інструмент для актуалізації власної бази працівників та можливості їх професійного зростання. Однак, застосунок використовує виключно алгоритмічний та систематичний підхід без використання ШІ – система не пропонує використання інших ресурсів, наприклад, трохи слабшими за основних кандидатів задля покриття вимог проєкту.

Інший аналогічний та з високою оцінкою застосунок походить з Нідерландів та має назву «Epicflow» [2] – аналітична система для управління ресурсами в мультипроєктному середовищі, яка використовує алгоритми предиктивної аналітики. На відміну від MuchSkills, у Epicflow є ключовий інструмент «future load graph», який прогнозує вузькі місця у використанні ресурсів на майбутнє. Система має функцію вираховувати пріоритетність кожного завдання для усіх працівників в режимі реального часу, зважаючи на

дедлайни всіх проєктів одночасно. Одна з найбільш цікавих функцій це автоматичне відсікання кількості активних задач, оскільки за філософією компанії-виробника перемикання між багатьма проєктами вбиває продуктивність працівника і підвищує стрес.

Платформа «Screendragon» [3] використовує алгоритми для створення теплових карт ємності та підбору персоналу на основі навичок, але зосереджена більше на маркетинг ніж IT. Такі інструменти, як Scoro [4] та Wrike [5], пропонують розширене планування з передбаченням ризиків (передбачення ризиків на основі моделі ШІ), тоді як Productive [6] фокусується на прогнозуванні маржинальності проєктів у режимі реального часу. Інші системи, як Kantata [7], Clickup [8], ProjectManager [9] схожі між собою та забезпечують розподіл завдань та забезпечують візуалізацію даних у реальному часі, однак не всі програмні застосунки мають здібності до прогнозування вигорання та збереження «здорового» робочого середовища.

Існують системи, як Taskfino [10] та ActivTrak [11], є HRMS (human resource management systems) застосунками, що ведуть базу та актуальну інформацію по працівникам, зокрема поточне навантаження, бухгалтерію та ведення заробітних плат. У цих застосунках є модуль управління проєктами, але у порівнянні з іншими застосунками є значно слабший. Для формування предметної області необхідно визначити ключові компоненти системи – модуль підбору працівників до команди, управління даними по співробітникам, формування звітів щодо поточного навантаження усього штату компанії. Кожна підсистема має бути розроблена згідно передових технологій та алгоритмів ШІ (EBM, опитувальник MBI, LSTM тощо). Практично відсутні універсальні системи, які б інтегрували принципи збереження work-life balance безпосередньо у ядро алгоритмів багатоцільової оптимізації під час створення команд. Предметна область поточного проєкту полягає у синтезі цих напрямків: розробці системи, яка превентивно моделює IT-команди, обмежуючи максимальне навантаження розробника на рівні 85 %,

та використовує багатовимірні фільтри (роль, технічний стек, психологічна сумісність) для досягнення глобального оптимуму ефективності та добробуту.

Центральною інновацією запропонованої системи є імплементація суворого правила розподілу навантаження. Згідно з цим правилом, система не дозволяє призначати завдання працівнику, якщо його сумарне планове навантаження перевищує 85 % від його робочої ємності.

1.2 Проведення аналізу літературних та інформаційних джерел щодо можливостей вирішення виявлених проблем

Автор статті [12, 38] наводить приклади того, як впровадження ШІ в системи HRMS впливає на рівень роботи у телекомунікаційному секторі працівників ОАЕ. Дослідження явно показує, що подібне впровадження інтелектуальної системи позитивно впливає на QOL («Quality of life», рівень життя), і показує, що збереження QOL сприяє кращій продуктивності та вищій результативності.

У публікації [13] пропонується гібридний еволюційний алгоритм (EA), що поєднується з методом багатокритеріального аналізу PROMETHEE для формування команд. Модель застосовує штрафи до цільової функції у разі невідповідності навичок або порушення обмежень проєкту (фільтрів).

Роботи [14, 16-18] зосереджуються на методах машинного навчання та редукції простору станів задля швидкого підбору IT-експертів, а також у самому дослідженні [15, 19] описано модель прогнозування вигорання працівників на основі опитувальника Маслоу.

Автори статті [20] описують створення моделей машинного навчання, що направлені на фактори ризику виникнення емоційного виснаження працівників команди.

У статті [22] пропонується інноваційний підхід, що комбінує обробку природної мови (NLP) та нечітку логіку. Алгоритм KeyBERT вилучає ключові слова з описів завдань та профілів розробників для створення векторних ембедингів. Потім система нечіткої логіки оптимізує ці дані, класифікуючи

якість підбору пар «розробник-завдання», що дозволяє автоматично генерувати високоефективні команди.

Науковці у своїй публікації [23] описуються вирішення задачі підбору команди як проблему комбінаторної оптимізації за допомогою багатокритеріального цілочисельного програмування, а у публікації [24] описується розробка дата-центричної методології формування software engineering команд, що фокусується на психологічних профілях за моделлю особистості FFM (five-factor model) та яким чином фіксується оптимальний перелік працівників для проєкту.

Таким чином аналіз показує, що поточний рівень розвитку питання дає можливість ефективно підійти до вирішення задачі та імплементувати ШІ як ядро системи для формування оптимальних команд для ІТ-проєктів з врахування QOL.

1.3 Ідентифікація зацікавлених сторін проєкту

Основною задачею будь-якого бізнес аналітика є точне визначення усіх зацікавлених осіб проєкту, як з точки зору проєктної команди, так і з кінцевого бенефіціара (клієнта, замовника). Кожний стейкхолдер має кінцеве бачення вигляду та функціоналу системи, свої очікування від її роботи та рівень впливу та оптимізації існуючих бізнес-процесів команди (оптимізації операційних витрат).

У працях [20-24] описуються перелік зацікавлених сторін. Аналіз існуючих додатків також розширює перелік стейкхолдерів. Найвищий вплив на проєкти мають кінцеві користувачі застосунком – проєктні менеджери та HR. Окрім цього варто і враховувати вимоги замовників, команди розробки (як покращення до вимог) та дизайнерів.

Перелік зацікавлених сторін представлений у табл. 1.1.

Таблиця 1.1

Аналіз зацікавлених сторін проєкту

№	Зацікавлена сторона	Роль у проєкті	Інтереси та очікування	Вплив на проєкт	Ступінь залучено-сті	Стратегія взаємодії
1	2	3	4	5	6	7
1	Спонсори / Інвестори	Ініціатори, партнери	Отримання ROI (окупність інвестицій), підвищення загальної прибутковості бізнесу за рахунок оптимізації процесів.	Високий	Низька / Середня	Регулярні стратегічні звіти, презентація досягнутих результатів та метрик ефективності.
2	CEO, CTO	Стратегічні керівники	Успішне впровадження системи, зниження витрат на плинність кадрів, утримання ключових талантів у компанії.	Високий	Низький	Керівні комітети, затвердження бюджетів та ключових архітектурних рішень.
3	Проєктний менеджер	Основний користувач системи	Швидкий та ефективний багатокритеріальний підбір команд, дотримання строків релізів без перенавантаження команд.	Високий	Висока	Навчання, тестування алгоритмів формування команд, збір фідбеку щодо функціоналу конструктора команд.
4	HR-директор	Основний користувач системи	Отримання глобальної аналітики використання ресурсів компанії, впровадження нових програм підтримки працівників.	Середній	Середня	Затвердження алгоритмів психологічних метрик, аналіз дашбордів «здоров'я» компанії, стратегічні сесії.
5	HR-менеджери	Користувачі аналітичного блоку системи	Зменшення рутинної адміністративної роботи, інструменти для щоденного моніторингу work-life balance працівників.	Середній	Висока	Залучення до налаштування фільтрів системи, тестування панелей моніторингу, щомісячні опитування.
6	Проєктна команда	Об'єкти управління	Збереження work-life balance (гарантія ліміту навантаження у 85 %).	Середній	Низька	Анонімні опитування, презентація концепції ШІ

Завершення табл. 1.1

1	2	3	4	5	6	7
7	UI/UX Дизайнери	Технічні виконавці (проєкту-вання)	Створення зручного та інтуїтивного інтерфейсу, чіткі вимоги до користувацького досвіду (UX).	Середній	Висока	Спринт-планування, спільні рев'ю макетів, тестування юзабіліті (Usability testing) з фокус-групами.
8	Розробники	Технічні виконавці (реалізація)	Чітке технічне завдання, стабільність вимог, доступ до якісних історичних даних для навчання алгоритмів ШІ.	Високий	Висока	Щоденні стендапи (Scrum), планування спринтів, код-рев'ю, ведення технічної документації.
9	DevOps-інженери	Технічні виконавці (інфраструк-тура)	Надійна серверна архітектура, можливість розгортання баз даних PostgreSQL.	Високий	Середня	Планування інфраструктури, моніторинг стабільності системи, налаштування хмарних середовищ.
10	QA-інженери (Тестувальники)	Технічні виконавці (контроль якості)	Чіткі критерії приймання (Acceptance Criteria), стабільне тестове середовище для перевірки алгоритмів оптимізації.	Середній	Висока	Участь у плануванні спринтів, баг-трекінг, розробка сценаріїв автоматизованого та навантажувального тестування.

Таким чином було визначено перелік усіх зацікавлених сторін проєкту, їх ролі, робота та вплив на проєкт. Цей крок дає можливість рухатись до створення SWOT-аналізу, де учасники проєкту напряму впливають на його результативність.

1.4 SWOT-аналіз проєкту

Для стратегічного розуміння позиціонування розроблюваної системи «HRM» необхідно провести SWOT-аналіз, що є основним та важливим кроком будь-якого проєкту. Команди проводять цей крок аби визначити доцільність

проведення проєкту та можливі ризики під час його реалізації. У табл. 1.2 наведено опис сильних сторін проєкту.

Таблиця 1.2

Сильні сторони

Код	Характеристика сильних сторін (Strength)
1	2
S01	Підвищення ефективності управління ресурсами
S02	Багатокритеріальний пошук працівників за фільтрами (hard та soft skills, попередній досвід, рівень завантаженості, тощо).
S03	Збереження принципу «85 %» під час підбору працівників на проєкт
S04	Можливість інтеграції у зовнішні застосунки (Jira, Confluence та HR застосунки як SAP HCM)
S05	Проведення онлайн моніторингу (збір метрик).
S06	Автоматизація бізнес-процесів (зменшення адміністративного навантаження на HR-відділ та проєктних менеджерів).
S07	Структуризація даних працівників у єдиній системі для управління ресурсами.
S08	Можливість розширення модулю (створення повноцінного застосунку для ведення проєктів, або розширення під повноцінний HR модуль)
S09	Визнання додатку на ринку (розширення пропозицій на ринку)
S10	Оптимізація пошуку працівників під конкретний проєкт (також працює як система порад для пропозицій інших працівників, що можуть бути слабшими, але мають такий самий набір навичок)

У табл. 1.3 наведено перелік слабких сторін.

Таблиця 1.3

Слабкі сторони

Код	Характеристика слабких сторін (Weaknesses)
1	2
W01	Висока технічна складність проєкту (потреба у глибокій експертизі для розробки та підтримки баз даних і гібридних ШІ-моделей).
W02	Наявність даних для навчання моделі (алгоритмам машинного навчання потрібні великі масиви накопичених історичних даних компанії для точного прогнозування та алгоритмізації підходів).
W03	Висока кількість людино-годин розробки (інтеграція моделей ШІ у систему потребує значних витрат часу та фінансів на початкових етапах).
W04	Пілотні версії запускаються на території України, через що важко враховувати деякі аспекти доступності та прогнозування використання працівників через війну

Завершення табл. 1.3

1	2
W05	Супротив до переходу на нову систему – багато співробітників зазвичай виступають проти переходу до використання нових систем через звичність використання попередніх.
W06	Технічна застарілість моделей та підходів (оскільки системи ШІ в сьогоденні розвиваються з великою швидкістю, існує ризик можливості старіння обраних підходів до реалізації проєкту)
W07	Зміна законодавства щодо збереження та опрацювання особистих даних

В табл. 1.4 наведено перелік можливостей проєкту.

Таблиця 1.4

Можливості

Код	Характеристика можливостей (Opportunities)
1	2
O01	Зростаючий попит на ринку B2B (оскільки найдорожчим ресурсом будь-якої компанії є люди, бізнес зацікавлений в оптимізації використання такого ресурсу).
O02	Можливість продажу системи як SaaS-рішення (можливість масштабування та продажу доступу до платформи іншим технологічним підприємствам, або розширення до вигляду як EpicFlow).
O03	Розширення до рівня *decision support system* (можливість перетворення системи на глобальний інструмент стратегічного планування для топ-менеджменту та (або) його трансформація у повноцінний HR модуль).
O04	Позитивний вплив на бренд роботодавця (впровадження такої системи забезпечує компанії здорове робоче середовище та забезпечення збереження нормального балансу роботи та життя).
O05	Розвиток екосистеми інтеграцій (додавання нових модулів та синхронізація з ERP (SAP S/4HANA, MS Dynamics, Orcale тощо), CRM та маркетинговими платформами).

У табл. 1.5 наведено перелік загроз для проєкту.

Таблиця 1.5

Загрози

Код	Характеристика загроз (Threats)
1	2
T01	Війна в Україні
T02	Психологічний опір працівників компанії (можлива недовіра IT-фахівців до систем алгоритмічного моніторингу через побоювання за свою приватність).

Завершення табл. 1.5

1	2
T03	Суворі законодавчі обмеження (необхідність жорсткого дотримання норм обробки персональних даних, як Закон України Про захист персональних даних (№5491-VI від 20.11.2012, GDPR та європейський AI Act (система одразу розробляється для покриття вимог ЄС).
T04	Ризик неправдивого результату опрацювання запиту ШІ (працівники можуть бути дискриміновані, та (або) не включені до переліку через певні особливості особистісних якостей).
T05	Залежність від сторонніх систем (використання ресурсів AWS сприяє прямої залежності до роботи системи (як приклад, через поточну війну США та Ірану).
T06	Висока ринкова конкуренція (поява аналогічних HR-інструментів від мільярдних бізнесів значно ускладнює позиціювання застосунку на ринку).

SWOT-аналіз дає можливість проєктній команді якісно дослідити та обґрунтувати необхідність та доцільність виконання проєкту та результативного впровадження рішення. Для конкретного проєкту SWOT-аналіз дає можливість з високою точністю оцінити доцільність такого проєкту, його технічну та бізнес складність, а також покращення та інновації, які він вносить та оптимізує.

За результатами аналізу доцільно провести парне зіставлення елементів SWOT-аналізу для виявлення критичних взаємозв’язків між особливостями проєкту. Кожна сильна сторона може мати ризики, а слабка потенційний розвиток. Нижче наведений перелік таких попарних зіставлень та їх взаємозв’язок.

- S02–O01**.** Багатокритеріальний пошук працівників за фільтрами (S02) відповідає зростаючому попиту на ринку B2B щодо оптимізації використання співробітників (O01), що дозволить запропонувати бізнесу інструмент максимізації ефективності найдорожчого людино-ресурсу.
- S03–O04**.** Збереження принципу «85 %» під час підбору працівників на проєкт (S03) матиме прямий позитивний вплив на бренд роботодавця (O04), оскільки така політика на рівні алгоритмів підбору працівників (та і принципу роботи в компанії) гарантує створення здорового

робочого середовища, можливості навчання і розвитку, та підтримує work-life balance співробітників.

- S04–O05**.** Можливість інтеграції у зовнішні застосунки (Jira, Confluence) (S04) створює надійну технічну базу для подальшого розвитку екосистеми інтеграцій з ERP та CRM платформами (O05), що значно спростить продаж та впровадження системи у великі корпорації.
- S08–O02**.** Можливість розширення модулю під повноцінний застосунок (S08) дозволяє безперешкодно реалізувати можливість продажу системи як масштабованого SaaS-рішення іншим технологічним підприємствам (O02).
- S07–T03. Структуризація даних працівників у єдиній системі (S07) дозволить ефективніше контролювати та управляти інформацією і забезпечити відповідність суворим законодавчим обмеженням щодо захисту персональних даних та вимогам Закону України Про персональні дані та GDPR (T03).
- S10–T04. Оптимізація пошуку з функцією порад та пропозицій альтернативних працівників (S10) пом’якшує ризик поганого результату опрацювання запиту ШІ (T04).
- S02–T06. Багатокритеріальний пошук на основі багатьох фільтрів, що враховує рівень завантаженості (S02), є особливістю, яка допоможе виділитися на фоні високої ринкової конкуренції та аналогічних HR-інструментів та впровадити схожий підхід до HR в компаніях (T06).
- W02–O03. Нестача історичних даних для навчання моделі на початковому етапі (W02) може бути вирішена шляхом розширення до рівня decision support system (O03), що мотивуватиме топ-менеджмент компаній самостійно ініціювати накопичення та надання якісних даних для стратегічного планування.
- W03–O01. Висока кількість людино-годин розробки та фінансові витрати на інтеграцію моделей ШІ (W03) є виправданими та можуть

бути швидко окуплені завдяки стабільно зростаючому попиту на B2B ринку на інструменти оптимізації персоналу (O01).

- W05–O04. Супротив співробітників до переходу на нову систему (W05) можна успішно подолати, просуваючи позитивний вплив інструменту на бренд компанії пояснюючи, що система створена для захисту їхнього балансу роботи та життя (O04).
- W04–T01. Запуск пілотних версій в Україні (W04) в умовах війни (T01) робить прогнозування доступності працівників вкрай нестабільним, тому архітектура системи повинна передбачати можливість швидкого перерахунку ресурсів при непередбачуваних змінах.
- W01–T05. Висока технічна складність проєкту та глибока експертиза для інтеграції ШІ (W01) посилюють залежність від сторонніх систем, таких як AWS (T05), що вимагає створення надійних планів аварійного відновлення роботи застосунку.
- W06–T06. Ризик технічної застарілості ШІ-моделей (W06) може послабити позиції продукту в умовах високої ринкової конкуренції (T06), тому система має будуватися за принципом мікросервісної архітектури для швидкої заміни окремих модулів без переписування всього коду.
- W07–T03. Зміна законодавства щодо опрацювання даних (W07) у поєднанні з суворими існуючими нормами GDPR та AI Act (T03) створює критичні юридичні ризики, тому розробка баз даних має одразу відбуватися за принципом «privacy by design» (приватність за замовчуванням).

Такий аналіз комбінацій елементів SWOT-аналізу самостійно покриває сильні та слабкі сторони можливостями та загрозами проєкту. Створення основи принципу застосунку та поєднання нових технологій дасть можливість платформі стати однією з найкращих на ринку Project-HR рішень.

1.5 Побудова дерева причин та наслідків

Будь-який об’єкт вимагає створення діаграми дерева причин та наслідків для визначення способів та дій вирішення задач передбачених реалізацією. На рис 1.1 зображено дерево причин та наслідків.

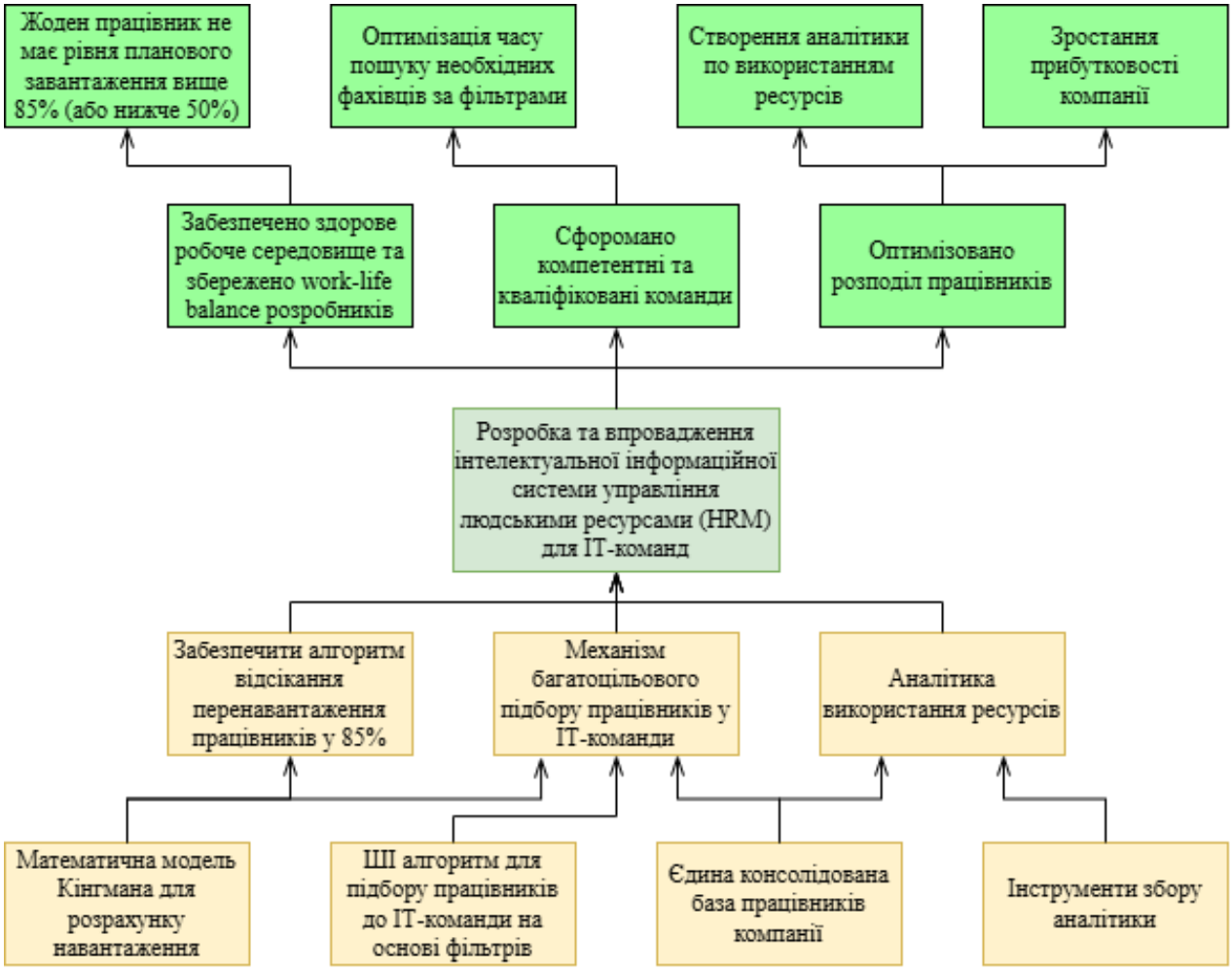


Рис 1.1. Дерево причин та наслідків проєкту

Побудоване дерево причини та наслідків дає можливість зрозуміти слабкі місця та можливості застосунку, окремі можливості якого будуть виконуватись в спринтах проєкту. Це також є основною для постановки задачі дослідження та формування паспорту проєкту

1.6 Постановка задачі дослідження. Формулювання технічного завдання на розробку у вигляді паспорту проєкту

Назва проєкту: Розробка та впровадження інтелектуальної інформаційної системи управління людськими ресурсами (HRM) для IT-команд.

Мета проєкту: проєктування та створення вебзастосунку для формування проєктної групи в IT-компаніях / командах, що автоматизує та оптимізує процес підбору та пошуку працівників на основі обраних фільтрів та залежностей, дотриманням правил навантаження працівників, та створення прозорої бази працівників для кращого управління ресурсами та зменшення операційних витрат.

Інноваційність проєкту: застосування моделей та алгоритмів ШІ для створення інтелектуальної системи пошуку та підбору працівників для IT-команд. Це дозволить IT-компаніям мати реальну аналітику та оптимізацію використання своїх ресурсів, що призводить до пришвидшення процесів підбору працівників, їх onboarding та вимірювання подальшого навантаження, що в комбінації призводить до збільшення прибутків компанії (за рахунок зменшення операційних витрат).

Цілі проєкту:

1. Розробити інтелектуальну вебплатформу управління людськими ресурсами для IT-команд з функціями багатокритеріального підбору працівників, алгоритмічного контролю завантаженості та адміністративного управління.
2. Забезпечити 100 % дотримання ліміту планового завантаження працівників на рівні не вище 85 % з моменту впровадження системи.
3. Скоротити час пошуку та формування проєктних команд на 40% протягом перших 6 місяців для автоматизованого підбору фахівців за визначеними фільтрами.

4. Завершити розробку та запустити вебплатформу протягом 10 місяців з моменту початку проєкту, з детальним планом, що містить ключові етапи та контрольні точки для відстеження прогресу.

Для кожної із зазначених вище цілей сформуємо перелік задач для досягнення поставлених майлстоунів. Перелік наведено задач відповідно до цілей наведено нижче.

Ціль 1. Розробити інтелектуальну вебплатформу управління людськими ресурсами для IT-команд з функціями багатокритеріального підбору працівників, алгоритмічного контролю завантаженості та адміністративного управління.

Задача 1.1. Провести збір і аналіз технічних та бізнес-вимог від ключових стейкхолдерів.

Задача 1.2. Спроектувати архітектуру системи та баз даних.

Задача 1.3. Розробити мікросервіс для забезпечення логіки роботи ШІ-моделей та обробки даних.

Задача 1.4. Розробити основний алгоритм підбору працівників за визначеними критеріями, фільтрами та обмеженнями.

Задача 1.5. Розробити модуль для збору метрик та формування аналітики.

Задача 1.6. Створити інтуїтивний користувацький інтерфейс.

Задача 1.7. Розробити панель адміністратора для управління правами доступу, профілями працівників та системними налаштуваннями.

Задача 1.8. Розробити концептуальну модель взаємодії систем та потоку даних.

Задача 1.9. Зробити систему доступною для легкої інтеграції зі сторонніми програмами.

Задача 1.10. Провести наскрізне тестування системи.

Ціль 2. Забезпечити 100% дотримання ліміту планового завантаження працівників на рівні не вище 85% з моменту впровадження системи.

Задача 2.1. Дослідити та формалізувати математичну модель розрахунку використання ресурсів на основі формули Кінгмана.

Задача 2.2. Розробити бекенд-модуль для безперервного підрахунку поточного та планового індивідуального завантаження кожного працівника.

Задача 2.3. Зашити обмеження у 85% навантаження працівників, що блокуватиме модуль підбору працівників до можливість залучення працівника до нових проєктів.

Задача 2.4. Реалізувати візуалізацію даних у вигляді capacity planning heatmaps для швидкого аналізу вільного ресурсу для PM.

Задача 2.6. Імплементувати алгоритм редукції простору станів SSR-TF з використанням методу one-hot encoding для миттєвого первинного відсіювання нерелевантних чи зайнятих кандидатів.

Задача 2.7. Розробити еволюційний алгоритм EA та інтегрувати метод PROMETHEE для багатокритеріального ранжування та підбору найефективніших комбінацій працівників.

Задача 2.8. Вбудувати алгоритмічні психологічні фільтри на основі п’ятифакторної моделі особистості FFM для підбору команд з високим рівнем сумісності та низькою конфліктністю.

Задача 2.9. Створити інтерактивний інструмент «Конструктор команд» з розширеними фільтрами (hard skills, soft skills, досвід, рольова модель) для автоматизації запитів проєктного менеджера.

Задача 2.10. Налаштувати модуль збору метрик для відстеження та порівняння часу, витраченого на формування команд до та після впровадження платформи, щоб верифікувати економію часу на 40%.

Ціль 3. Скоротити час пошуку та формування проєктних команд на 40% протягом перших 6 місяців для автоматизованого підбору фахівців за визначеними фільтрами.

Задача 3.1. Імплементувати алгоритм редукції простору станів SSR-TF з використанням методу one-hot encoding для миттєвого первинного відсіювання нерелевантних чи зайнятих кандидатів.

Задача 3.2. Розробити еволюційний алгоритм EA та інтегрувати метод PROMETHEE для багатокритеріального ранжування та підбору найефективніших комбінацій працівників.

Задача 3.3. Вбудувати алгоритмічні психологічні фільтри на основі п’ятифакторної моделі особистості FFM для підбору команд з високим рівнем сумісності та низькою конфліктністю.

Задача 3.4. Створити інтерактивний інструмент «Конструктор команд» з розширеними фільтрами (hard skills, soft skills, досвід, рольова модель) для автоматизації запитів проєктного менеджера.

Задача 3.5. Налаштувати модуль збору метрик для відстеження та порівняння часу, витраченого на формування команд до та після впровадження платформи, щоб верифікувати економію часу на 40%.

Ціль 4. Завершити розробку та запустити вебплатформу протягом 10 місяців з моменту початку проєкту, з детальним планом, що містить ключові етапи та контрольні точки для відстеження прогресу.

Задача 4.1. Розробити календарний план графік розробки (діаграму Ґанта) з розподілом завдань на двотижневі спринти та чіткими контрольними точками для моніторингу прогресу.

Задача 4.2. Спланувати та реалізувати запуск бета-версії (MVP) функціоналу конструктора команд на 6-му місяці для збору первинного зворотного зв’язку від PM та HR-відділу.

Задача 4.3. Здійснити фінальне калібрування математичних моделей, усунення виявлених дефектів та підготовку супровідної технічної документації.

Задача 4.4. Провести розгортання платформи на виробничому середовищі та організувати онбординг / навчання користувачів перед фінальним запуском на 10-й місяць.

В результаті опрацювання розділу було сформовано 4 цілі та їх задачі, які потім будуть перетворені у спринти та розподілені за командою розробки. Сформований перелік задач дає можливість перейти до проєктування концептуальної моделі та алгоритмізацію функцій та механізмів, які виконуються в межах системи.

РОЗДІЛ 2. ОПИС КОНЦЕПТУАЛЬНИХ МОДЕЛЕЙ ТА ФОРМАЛІЗАЦІЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ

2.1 Розробка концептуальної моделі інформаційної системи

У межах розробки інтелектуальної інформаційної системи управління людськими ресурсами в IT-командах чітке визначення вхідних та вихідних даних є ключовим етапом, що забезпечує функціонування концептуальної моделі. Вхідні дані формують інформаційний базис системи, який поділяється на кілька функціональних блоків.

Перший блок вхідної інформації складається з прямих метрик професійної кваліфікації, що включають оцінки знання мов програмування та фреймворків за десятибальною шкалою, дані про попередній галузевий досвід та результати перевірок якості коду.

Другий блок охоплює поведінкові та психологічні характеристики, такі як результати тестування за методикою Белбіна, рівень м’яких навичок та ступінь автономності у прийнятті рішень.

Третій блок вхідних даних має динамічний характер і надходить із суміжних систем управління проєктами та обліку робочого часу, надаючи інформацію про поточний відсоток завантаження працівника, кількість використаних днів відпустки та частоту випадків тимчасової непрацездатності.

Крім того, на вхід системи подаються параметри самого проєкту, а саме вимоги до проєктної ролі та необхідний технологічний стек, що виступають орієнтиром для подальшого порівняння.

В табл. 2.1 наведено вхідні параметри до системи, в табл. 2.2 проаналізовано процеси, в табл. 2.3 наведено вихідні дані, а в табл. 2.4 – параметри обмеження.

Таблиця 2.1

Вхідні дані

Блок моделі	Підблоки моделі	Елементи, що входять до підблоку
1	**2**	**3**
Вхідні дані	Дані працівників	1) ІД. 2) Прізвище. 3) Ім’я. 4) По-батькові. 5) Номер телефону. 6) Електронна адреса. 7) Дата прийому на роботу. 8) Дата звільнення.
	Відпуски та лікарняні	1) ІД. 2) Кількість відпусткових днів. 3) Кількість лікарняних днів. 4) Кількість відпусткових / лікарняних днів за останній квартал.
	Поточна зайнятість	1) % зайнятості. 2) ІД проєктів (до яких залучено працівника).

Таблиця 2.2

Процеси

Блок моделі	Модулі	Функціонал
1	**2**	**3**
Процеси	Підбір працівників до залучення на проєкт	1) Витяг переліку усіх працівників за обраними параметрами (фільтрами) в порядку ранжування за рівнем використання. 2) Автопідбір працівників за обраними критеріями (на основі обраних фільтрів, зокрема кількість ролей, рівень залученості та орієнтовна тривалість). 3) Формування остаточного переліку працівників під проєкт.
	Моніторинг та актуалізація	1) Автоматичне оновлення залишку відпусток та кількості лікарняних після кожної транзакції в HR-модулі. 2) Перерахунок відсотка зайнятості працівників (раз на добу).
	Управління даними	1) Автоматичне оновлення даних працівника (наприклад, % залучення) 2) Ручне коригування даних працівника (наприклад, soft / hard skills).
	Формування звітності	1) Формування звіту щодо використання ресурсів (факт + план) для визначення проблемних точок 2) Формування звіту щодо використання відпусток та лікарняних (для побудови аналіз факту вигорання).

Таблиця 2.3

Вихідні дані

Блок моделі	Підблоки моделі	Елементи, що входять до підблоку
1	**2**	**3**
Вихідні дані	Дані проєкту (за працівниками)	1) ІД проєкту. 2) Назва проєкту. 3) Оновлений відсоток завантаження працівників.
	Дані працівників	1) Оновлений відсоток залучення. 2) Доданий ІД проєкту. 3) Оновлення кількості відпусток та лікарняних.
	Звіти	1) Оновлені звіти щодо використання ресурсів (раз на добу).

Таблиця 2.4

Параметри обмеження

Блок моделі	Підблоки моделі	Елементи, що входять до підблоку
1	**2**	**3**
Вихідні дані	Організаційні обмеження	1) Рівень завантаженості працівника не може бути вищий за 90% (10% виділяється на навчання та розвиток). 2) У працівника не може бути більше ніж 10 лікарняних (3 ABAPS, та 7 за довідкою), окрім випадків ручної зміни. 3) Навчання працівників.
	Технічні обмеження	1) Формування бекапів (раз на добу) 2) Інтеграція з поточними системами ведення проєктів (Jira/Trello, Confluence) 3) Інструменти для управління даними на перших етапах MVP
	Юридичні обмеження	Відповідність GDPR та Закону України щодо опрацювання персональних даних.

В результаті, актуалізувавши вхідні/вихідні дані, інформацію щодо процесів та обмеження, а також літературні статті [25-29], маємо можливість скомпонувати та спроектувати концептуальну модель майбутнього вебзастосунку. Також, проведений аналіз літератури та публікації буде використаний для розробки концептуальної моделі інформаційної системи, яка зображена на рис 2.1.

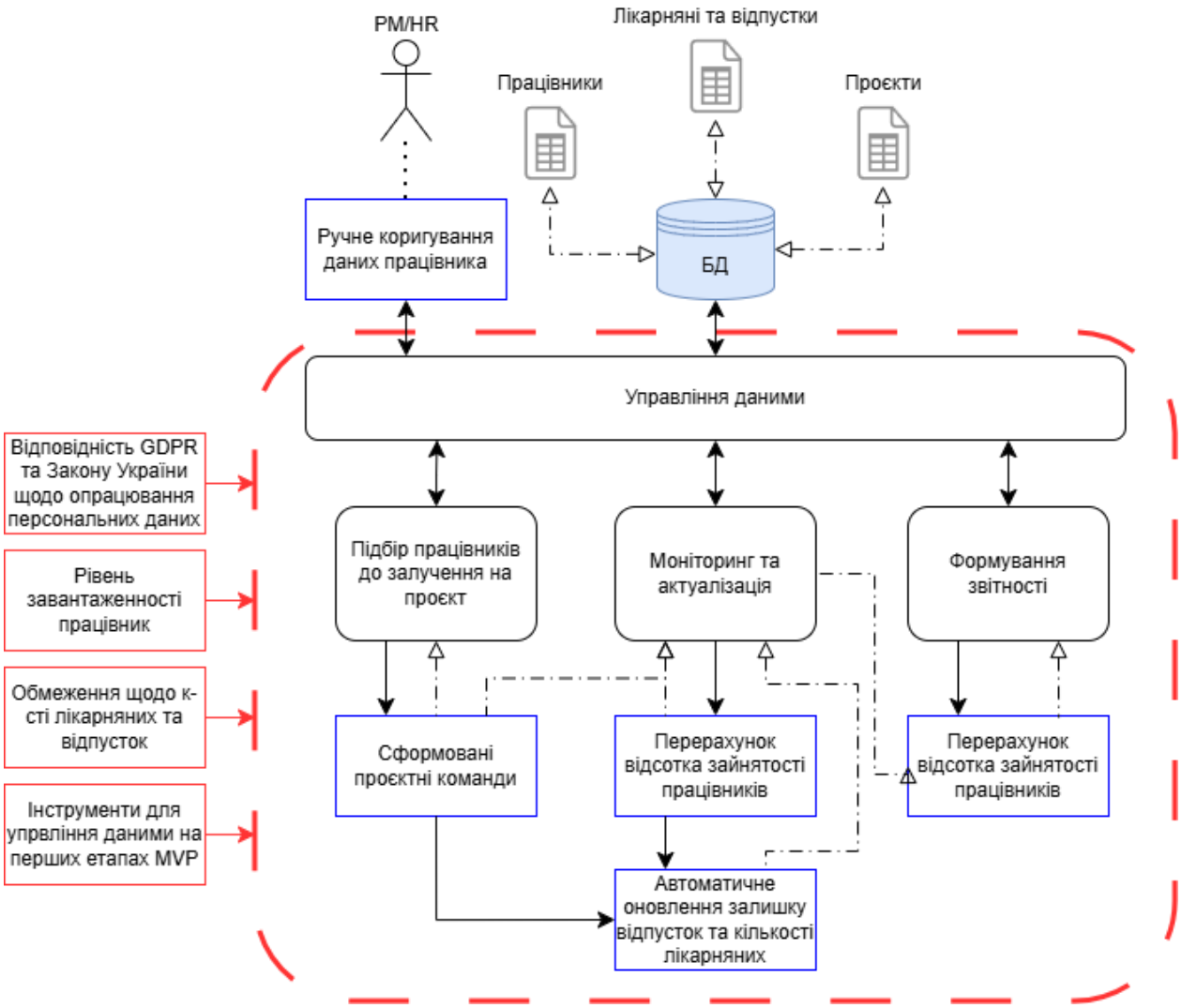


Рис. 2.1. Концептуальна модель інформаційної системи [30]

Прикладом використання моделі є звичайний сценарій формування команди для FinTech-проєкту, де проєктний менеджер ініціює запит у системі, вказуючи необхідні технічні параметри (наприклад, знання мови Python та досвід у FinTech розробці), та обмеження щодо поточної зайнятості фахівця протягом наступного кварталу.

Інтелектуальне ядро системи звертається до AWS Data Lake для отримання інформації про працівників, які підпадають за обраним фільтром. У результаті модуль порівняння видає ранжований список, де система може рекомендувати фахівця з трохи нижчим рівнем Hard Skills, але нульовим показником завантаження та релевантним доменним досвідом, що забезпечує швидкий старт без потреби в онбордингу. Після фінального затвердження

кандидатури модуль управління кадрами необхідно запустити процес оновлення даних працівника, змінюючи його статус завантаженості, що у майбутньому можливо автоматизувати.

2.1.1 Критерізація параметрів ресурсів моделей

Кожен працівник у системі може бути представлений як об'єкт E_i (Employee). Набір вхідних даних для кожного працівника D_i, до якого входять ПІБ, контактні дані та інша інформація із записів працівника, і описується як кортеж параметрів (2.1):

$$E_i = \{D_1, D_2, D_3, \dots, D_n\}, \tag{2.1}$$

де, E_i – працівник, а i – унікальний індекс працівника у загальній множині персоналу; D_i – унікальний набір даних працівник (ПІБ, контактні дані, тощо).

Для кожного проєкту P_i визначається функція підбору працівників E_i за формулою (2.2):

$$L_i = f\{F, T, C\}, \tag{2.2}$$

де, L_i – сформований перелік кандидатів на основі: F – набору фільтрів та параметрів відбору (роль, досвід, область знань); T – доступний об'єм навантаження в людино-днях (за основу береться 8 годин на добу для буднів); C – перелік кандидатів з бази.

Вихідні дані системи після формування команди можна представити як оновлений кортеж (2.3) стану проєкту S_P та працівника E_P:

$$S_P = \{D_i, U_i\}, \tag{2.3}$$

де, D_i – оновлені дані працівника згідно навантаження; U_i – загальне навантаження команди

Задля коректної роботи будь-якої моделі необхідно мати перелік параметрів, характеристик та ваг. Визначимо перелік та представимо його у вигляді табл. 2.5.

Таблиця 2.5

Ваги параметрів моделі підбору працівників

Назва	Критичність	Опис	Оцінка	Вага
1	2	3	4	5
Рівень Hard Skills	Висока	Технічні навички (мови, фреймворки). Визначає, чи може працівник виконати завдання.	Шкали від 0-10	0,25
Рівень Soft Skills	Середня	Комунікація, емоційний інтелект, робота в команді. Впливає на клімат у колективі.	Шкали від 0-10	0,2
Роль за Белбіном	Середня	Поведінкова роль (лідер, виконавець, критик). Важлива для балансування команди.	Шкали від 0-10	0,05
Проєктна роль	Висока	Проєктна роль працівника (аналітик, інженер тощо).	Не передбачено	Фільтр
Кількість днів відпустки	Висока	Залишок доступних днів відпочинку. Допомагає планувати графік релізів та уникати вигорання. Загальна кількість днів відсутності через хворобу. Ключовий маркер для оцінки доступності ресурсу.	Алгоритм розрахунку	0,15

Завершення табл. 2.5

1	2	3	4	5
Рівень автономності	Середня	Здатність працювати без детального нагляду. Визначає навантаження на team lead.	Шкала від 0-10	0,1
Поточне завантаження	Висока	Відсотковий показник зайнятості в інших проєктах. Запобігає перевантаженню.	Значення у %, де 85 – об'єктивний максимум з можливих 100	0,15
Проєктний досвід (Domain)	Висока	Досвід у конкретній бізнес-ніші (напр., FinTech).	Шкала від 0-10	0,2

Для визначення Hard Skills необхідно описати конкретні навички, що необхідні команді, наприклад, знання та вміння програмувати на Python/Java/C#.

Для опису кожної окремої спеціальності наводиться наступний набір параметрів, що описані у табл. 2.6 та табл. 2.7.

Таблиця 2.6

Перелік ваг параметрів для технічних навичок

Назва критерію	Критичність	Опис для оцінювання	Оцінка	Вага
1	2	3	4	5
Теоретичні знання	Середня	Рівень розуміння принципів роботи технології, алгоритмів та архітектурних патернів.	Шкала від 0-10	0,15
Практичний досвід	Висока	Кількість років/місяців роботи з технологією на реальних комерційних проєктах.	Шкала від 0-10	0,4
Якість коду	Висока	Дотримання стандартів написання коду (clean code, SOLID), відсутність критичних помилок.	Шкала від 0-10	0,3

Завершення табл. 2.6

1	2	3	4	5
Швидкість вирішення задач	Середня	Здатність вкладатися в оцінку часу при виконанні технічних завдань.	Шкала від 0-10	0.1
Наявність сертифікацій	Низька	Підтвердження знань зовнішніми вендорами (AWS, Google, Oracle тощо).	Перелік	0.05

Сформувавши технічні навички працівників, також варто перейти до визначення переліку м’яких навичок та їх ваг.

Таблиця 2.7

Перелік ваг параметрів для м’яких навичок

Назва критерію	Критичність	Опис для оцінювання	Оцінка	Вага
1	2	3	4	5
Якість комунікації	Висока	Здатність чітко формулювати думки, активне слухання, надання конструктивного фідбеку.	Шкала від 0-10	0,35
Командна взаємодія	Висока	Готовність допомогти колегам, участь у спільних обговореннях, відсутність токсичності.	Шкала від 0-10	0,3
Стресостійкість	Середня	Збереження продуктивності під час дедлайнів або різких змін вимог до проєкту.	Шкала від 0-10	0.2
Самостійність	Середня	Здатність знаходити рішення без постійних підказок з боку керівництва.	Шкала від 0-10	0,1
Наставництво	Низька	Бажання та вміння навчати менш досвідчених колег (важливо для рівня senior / lead).	Шкала від 0-10	0,05

Варто окремо визначити алгоритм для розрахунку відпусткових днів, до яких входять також і лікарняні дні (ABAPS – дні без необхідності лікарняної

завіреності, до 3 днів; та лікарняні, що оплачуються компанією та мають бути завірені лікарем, до 20 днів). Для розрахунку оцінки в математичній моделі, що буде показано у розділі 2.2.

2.2 Формалізація математичних моделей та постановка задачі в математичному вигляді

Деякі параметри не відповідають бальній оцінці, а слугують фільтром та точними правилами для відбору кандидатів до проєктної команди:

- Проєктна роль – критерій, що не має оцінки. Слугує лише критерієм для відбору працівників конкретної спеціальності. Наприклад, якщо за запитом необхідно знайти Devops інженера, то будуть відображені лише ті працівники, які в базі даних визначені як «Devops» в незалежності від інших критерії (hard skills, soft skills тощо).
- Алгоритм розрахунку даних для відпустко та лікарняних – для даної моделі пропонується використовувати наступний принцип: якщо кількість невикористаних/незапланованих днів відпустки працівника більше за 14 к.д., а кількість лікарняних за останній квартал менше 5, то надається мінімальна оцінка: 0; якщо вичерпано кількість відпусткових та лікарняних днів, то оцінка, відповідно, 10.
- Поточне завантаження – з поточними світовими практиками розподілення часу працівників, більшість компаній схильні до максимальної завантаженості своїх працівників робочими аспектами до 85% робочого часу, лишаючи інші 15 на саморозвиток. Саме тому для оцінки вільного часу співробітника буде використовуватись запас у 85%.

Для розрахунку загального показника працівника (R) для вибору на конкретний проєкт можливо розрахувати за наступною математичною моделлю (2.4):

$$R = \sum_{i=1}^{n} (Score_i \cdot w_i), \quad (2.4)$$

де, $Score_i$ – це нормалізована оцінка за шкалою від 0 до 10; w_i – це вага відповідного критерію таблиці.

Таким чином, розгорнута математична модель оцінки для кожного набору критеріїв окремо (I_{hr}) матиме вигляд (2.5):

$$I_{hr} = \sum S_i \cdot w_{\text{і}}, \tag{2.5}$$

де, S_i – нормалізована оцінка критерію; w_i – вага критерію.

Для того, щоб додати у формулу «Завантаження» (S_{load}) та «Відпустки»(S_{vac}), їх потрібно конвертувати в бали.

Оскільки максимальна можлива завантаженість працівника може становити 85% робочого часу, то маємо наступне систему рівнянь (2.6-2.7):

$$S_{load} = 1 - \frac{CurrentLoad}{85}\ 0,, \frac{\text{Якщо } Load<85}{\text{Якщо } Load>85}. \tag{2.6}$$

Чим менше завантажений працівник – тим вище бал. Для розрахунку лікарняних використовується понижуючий коефіцієнт. Якщо у працівника залишилося мало днів відпустки (ризик вигорання) або багато лікарняних (ризик відсутності), бал падає.

$$S_{vac} = \frac{DaysRemaining}{TotalYearPolicy} * (1 - Penalty_{sick}) . \tag{2.7}$$

В результаті маємо наступну табл. 2.8, на якій зображено верхньорівневий перелік ваг, з яких формується оцінка для підбору кандидата на проєкт.

Таблиця 2.8

Перелік параметрів та ваг для формування оцінки вибору кандидата

Параметр	**Вага (w)**	**Тип розрахунку в моделі (S_i)**
1	**2**	**3**
Hard Skills	0,25	Пряме значення (Score / 10)
Domain Experience	0,20	Пряме значення (Score / 10)
Поточне завантаження	0,15	Інверсоване значення (85 % – це 0 балів)
Відпустки/Лікарняні	0,15	Коефіцієнт доступності ресурсу
Soft Skills	0,10	Пряме значення (Score / 10)
Рівень автономності	0,10	Пряме значення (Score / 10)
Роль за Белбіном	0,05	Відповідність потрібній ролі (0 або 1)

Фільтр «Проєктна роль» слугує відсіканням зайвих результатів для зменшення кола відповідей. Цей параметр не входить у суму, а виступає як логічний множник (2.8):

$$R = I_{hr} \cdot P_{match}, \tag{2.8}$$

де, $P_{match} = 1$, якщо роль працівника відповідає ролі в проєкті;

$P_{match} = 0$, якщо роль працівника не відповідає ролі в проєкті.

2.3 Застосування розроблених моделей в межах інтелектуальної інформаційної системи управління людськими ресурсами в IT-командах

Основним та найбільш вживаним процесом HRM є пошук та підбір проєктної команди Проєктним менеджером, який має розуміння потенційних

необхідних учасників, їх кількість та роль, рівень технічних знань та особистості та, найважливіше, рівень завантаження співробітника. Системою передбачено правило 85 %, яке забороняє використовувати працівника на більше не 85 % робочого часу. Процес роботи проєктного менеджера починатись із необхідності визначення фільтрів та розуміння переліку та складу технічних знань проєктної команди.

Етап 1. Визначення фільтрів та ініціація запиту про працівників.

Проєктний менеджер ініціює процес, формуючи набір фільтрів до майбутніх кандидатів. Нехай *F* – множина заданих фільтрів (досвід, hard skills, soft skills, максимальне завантаження) (2.9):

$$F = \{f_{1i}, f_{2i,} f_{3i}, \dots, f_{ni}\}, \tag{2.9}$$

де, f_{1i} – *n* критерій пошуку працівника.

Сформувавши усі фільтри *F* проєктний менеджер має можливість створити запит *Q*, що складається з фільтрів *F* та ролей *R* (2.10)*:*

$$Q = \{(R_1, A_1, F_{1,}), (R_2, A_2, F_{2,}), (R_3, A_3, F_{3,}), \dots, (R_n, A_n, F_{n,})\}, \tag{2.10}$$

де, R_n – конкретна роль зазначена менеджером; A_n – необхідна кількість працівників.

В результаті проєктний менеджер створив запит до системи щодо формування команди для проєкту.

Етап 2. Ідентифікація потрібних співробітників та виведення результатів

У HRM системі передбачено уніфіковану БД, що містить таблиці із записами інформації щодо співробітників. Визначимо множину всіх працівників як *E*:

$$E = \{e_1, e_2, e_3, \dots, e_n\}. \tag{2.11}$$

Функція підбору f_{select} здійснює фільтрацію множини E на основі критеріїв F, застосовуючи алгоритм відсікання кандидатів, що не проходять за системними правилами. Відповідно функція відбору матиме наступний вигляд (2.12):

$$f_{select}(E, F) \rightarrow E' \tag{2.12}$$

де, $E' \in E$ – підмножина кандидатів, які відповідають вимогам та мають рівень утилізації p < 0.85.

Система використовує інтегральний показник I_{hr} для миттєвого ранжування кандидатів під конкретний проєкт: інтелектуальна модель фільтрує базу по «Проєктній ролі» та видає топ-список фахівців з найвищим балом (2.13):

$$I_{hr}(e_i) = \sum_{j=1}^{k}\left(w_j * s_{ij}\right) - P(p_i), \tag{2.13}$$

де, w_j – вага критерію j, s_{ij} – оцінка працівника i за критерієм j, $P(p_i)$ – штрафна функція за наближення до критичного рівня навантаження.

Якщо система бачить, що в команді вже є два «Генератори ідей», але немає «Реалізатора», вона автоматично підвищить пріоритет кандидата з відповідною роллю, навіть якщо його Hard Skills трохи нижчі.

Система виводить відсортований перелік працівників за спаданням показника I_{hr} і формує список рекомендованих працівників L (2.14):

$$L = |e_1, e_2, e_3, \dots, e_n|, = I_{hr}(e_1) > I_{hr}(e_2) > \cdots > I_{hr}(e_3). \tag{2.14}$$

Використання та індикація наявних, запланованих або не використаних відпусток, чи лікарняних дають проєктному менеджеру більш точне уявлення

про працівника. На розвиток моделі можна вивести аналіз відпусток / лікарняних співробітників за попередній період (наприклад, 1 рік, аби мати можливість скласти тенденцію про працівника).

Етап 3. Пропозиція використання альтернативних варіантів та затвердження переліку працівників.

Після затвердження проєктним менеджером фінального складу команди T_{final}, система оновлює глобальну матрицю завантаженості p для кожного обраного працівника $e_1 \in T_{final}$ (2.15):

$$p_i^{new} = p_i^{old} + \Delta p_{proj}, \tag{2.15}$$

де, Δp_{proj} – оцінка навантаження проєкту.

Також як окремий цікавий напрямок розвиток інтелектуальної моделі полягає у KPI та інших оцінках. Можливо збирати метрики та оновлювати їх за результатами проєкту аби нагороджувати співробітника, або навпаки понижувати. У кінцевому підсумку, саме якість пропрацювання концептуальної моделі визначає рівень довіри до висновків інформаційної системи, оскільки вона мінімізує вплив суб'єктивізму менеджерів і формує прозору, науково обґрунтовану методологію управління людськими ресурсами.

РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ЛЮДСЬКИМИ РЕСУРСАМИ

3.1 Вибір технології управління проєктом

Для розробки проєкту інтелектуальної HRM-системи найкраще підходить гнучкі методології [31] з елементами жорсткого контролю виконання та послідовності деяких задач, оскільки основою проєкту є розробка ядра для аналізу, пошуку та визначення необхідних осіб для IT-проєктів на основі ШІ. Однак для загального управління проєктом буде використовуватись фреймворк Scrum [32] із двотижневими спринтами, щотижневими статусними нарадами щодо виконання задач, проведення ретроспектив та демонстрації досягнутих результатів за визначений період. Scrum має гібридний підхід, під назвою Scrumban – поєднання моделі управління проєктами Scrum та Kanban [33-36].

У розрізі розробки проєкту можливо визначити наступні ключові вимоги:

- точне визначення правил, ваг та функції для розробки ядра ШІ;
- адаптивність до змін бізнес-процесу;
- залучення інвесторів до проміжних етапів оцінювання реалізації продукту;
- проведення керівних комітетів за фактом виконання чотирьох спринтів, або досягнення основних майлстоунів проєкту.
- Контроль виконання завдань та можливість перерозподілення ресурсів.

Для реалізації проєкту визначено наступні ролі:

- Scrum master – відповідає за управління проєктом, координацію команди та дотриманням термінів.
- Product owner – відповідає за формалізацію та контролю дотримання вимог замовника. Проводить пріоритезацію виконання задач, відповідає

за дотримання цінності проєкту та його позитивного впливу на компанію.

- Команда Scrum – кросфункціональна команда, що складається з системних та бізнес-аналітиків, ШІ розробників, фулстек розробників, UI/UX-дизайнерів та тестувальників.

Артефакти проєкту:

1. Статут проєкту – документ, що визначає старт та фініш проєкту, мету та бізнес цінність.
2. Реєстр зацікавлених осіб.
3. План проєкту.
4. Беклог проєкту – перелік задач проєкту, що описують бізнес, функціональні та нефункціональні вимоги включно з користувацькими кейсами.
5. Беклог спринту – перелік задач спринту для конкретної ітерації.
6. Події визначені проєктом.
7. Планування Sprint.
8. Щотижневий Scrum.
9. Sprint Review.
10. Ретроспектива.

Детальний опис подій наведено у табл. 3.1.

Таблиця 3.1

Характеристика подій Scrum

№	Подія Scrum	Мета	Учасники	Вхідні / формат роботи	Результат	Приклад у досліджуваному проєкті
1	**2**	**3**	**4**	**5**	**6**	**7**
1	Плану-вання Sprint	Визначити функціонал, що буде реалізовано в межах Sprint, та шлях його досягнення.	Повний склад Scrum команди	1. Мета продукту. 2. Беклог продукту. 3. Результати минулих спринтів.	Беклог спринту.	Планування задач: розробка алгоритму взаємодії БД та ядра ШІ.

Завершення табл. 3.1

1	2	3	4	5	6	7
2	Щотиж-невий Scrum	Статус виконання задач та синхронізація команди.	Scrum команда та Scrum Master	1. Що виконано за тиждень? 2. Плани на тиждень?	Спільне розуміння статусу роботи.	Розроблено API для оновлення БД.
3	Sprint review	Продемонструвати готовий функціонал, зібрати зворотний зв'язок у стейкхолдерів від замовника.	Повна Команда Scrum та замовники (інвестори)	1. Демо готового функціоналу. 2. Обговорення результатів. 3. Отримання фідбеку. 4. Наступні кроки.	1. Зворотний зв'язок від замовника. 2. Оновлений Беклог продукту та наступних спринтів	Продемонстровано сторінку формування команди на основі фільтрів та зазначених параметрів.
4	Ретро-спектива	Аналіз ефективності процесу роботи, пошук шляхів покращення.	Повна команда Scrum	Обговорення: 1. Що було добре, а що погано? 2. Що можливо покращити? 3. Яким чином я буду працювати над вирішенням проблеми?	1. Перелік думок для вдосконалення процесу. 2. План покращень підходів в наступному спринті (та поступовий аналіз).	Вирішено додати проміжну Scrum зустріч для оновлення статусів фулстек розробників.

Переваги обраної методології у розрізі реалізації проєкту полягає у наступному:

1. *Гнучкість до змін.* Хоча і реалізація модуля формування команди та ядра ШІ вимагає значного часу та точно визначених критеріїв, що являється основою проєкту, інші аспекти реалізації є гнучкими до змін та повторної пріоритезацію.
2. *Залученість Замовника.* Реалізація вимог Замовника проходить через Product Owner, який описує вимоги, задачі та мету проєкту. Постійне залучення Project Owner за подіями проєкту дає можливість точно слідкувати баченню кінцевих користувачів та Замовника.
3. *Ретроспективи та постійні покращення.* Проведення періодичних ретроспектив дає можливість команді знайти проблемні точки та способи їх вирішення/покращення.

4. *Контроль ризиків та моніторинг виконання задач.* Kanban підхід має чудово візуалізацію стану виконання задач та можливість використовувати теплові карти для визначення потенційних ризиків та їх контролю.
5. *Результативність.* Підхід двотижневих спринтів сприяє показовому факту виконання задач та досягнення поставлених цілей. Це дозволяє зацікавленим сторонам швидше отримувати оновлення щодо стану справ проєкту та мати можливість протестувати функціонал самостійно.

Таким чином, обрана методологія є найкращою для реалізації даного проєкту та дає можливість для його подальшого розвитку і оновлень.

3.2 Визначення функціональних та нефункціональних вимог до продукту проєкту

Як визначає BABOK [37], вимога це придатна (або точно описане бажана) необхідність чогось. Ця концепція першочергово зосереджується на розумінні того, яку саме цінність буде досягнуто, якщо цю вимогу виконати. Характер подання такої потреби може варіюватися. Це може бути традиційний текстовий документ, візуальна модель, макет, користувацька історія або навіть прототип – усе залежить від конкретних обставин проєкту, обраної методології та очікувань зацікавлених сторін.

Крім того, стандарт класифікує вимоги на кілька окремих рівнів. Серед них виділяють бізнес-вимоги, вимоги зацікавлених сторін, вимоги до рішення (які додатково поділяються на функціональні та нефункціональні), а також вимоги до переходу. Визначимо перелік бізнес вимог, що описані у табл. 3.2.

Таблиця 3.2

Бізнес-вимоги проєкту

Код	Назва вимоги	Характеристика бізнес-вимоги
1	2	3
BR01	Оптимізація формування команд	Платформа має скоротити час менеджерів на пошук і підбір фахівців у проєктні команди на 40 %.
BR02	Забезпечення work-life balance	Система повинна гарантувати дотримання балансу між роботою та особистим життям працівників не допускаючи перенавантаження.
BR03	Підтримка прийняття рішень (DSS)	Платформа повинна забезпечити топ-менеджмент та HR-відділ прозорою аналітикою використання ресурсів для максимізації фінансової ефективності проєктів.
BR04	Етичний контроль	Система має здійснювати моніторинг робочого стану виключно людяними методами, не порушуючи особисті кордони працівників.

Перелік функціональних вимог наведено у табл. 3.3.

Таблиця 3.3

Функціональні вимоги проєкту

Код	Назва вимоги	Характеристика функціональної вимоги
1	2	3
FR02	Розрахунок рівня завантаженості	Система повинна автоматично розраховувати поточний та плановий рівень завантаженості кожного працівника у відсотках.
FR02	Блокування перенавантаження	Програмний модуль повинен блокувати можливість призначення нових завдань, якщо розрахунковий рівень завантаженості працівника перевищує 85 %.
FR03	Фільтрований пошук	Система повинна мати передбачити можливість обирати декілька пошукових фільтрів, що дозволяють PM створювати запити на підбір за конкретними критеріями (hard skills, soft skills, досвід, роль).
FR04	Обмеження пошукових запитів	Система повинна автоматичного відсікати недоступних або нерелевантних кандидатів.
FR05	Пошук альтернативних варіантів	Система повинна генерувати до трьох (зробити цю опцію доступною для редагування) альтернативних варіантів складу команди для кожного запиту та виставляти їх за рангом.
FR06	Ручне коригування команд	Платформа повинна дозволяти ручне коригування складу згенерованої системою команди із миттєвим перерозрахунком загальних метрик та навантаження працівників.
FR07	Автоматичні сповіщення	Система має автоматично надсилати сповіщення коли працівник перенавантажений
FR08	Теплові карти завантаженості	Система повинна містити дашборди з візуалізацією даних у вигляді інтерактивних теплових карт завантаженості з можливістю експорту у Excel

Завершення табл. 3.3

1	2	3
FR09	Збір метрик та показників	Система повинна автоматично збирати метрики активності через API інтеграції із частотою 5 хвилин.
FR10	Рольова модель доступу (RBAC)	Платформа має підтримувати рольову модель доступу з відповідним розмежуванням прав на перегляд і редагування певних об’єктів системи.
FR11	Профілі працівників	Кожний працівник має мати власний профіль, де будуть вказані його навички та досвід. Право на редагування має як сам працівник, так і адміністратор.
FR12	Експорт звітів	Система повинна мати можливість генерувати звіти у Excel щодо поточного розподілу ресурсів та інших показників.

У табл. 3.4 наведено перелік нефункціональних вимог.

Таблиця 3.4

Нефункціональні вимоги проєкту

Код	Назва вимоги	Характеристика нефункціональної вимоги
1	2	3
NFR01	Продуктивність	Оптимізаційний підбір працівників під IT-проєкт має займати до 30 секунд.
NFR02	Архітектура БД	Система повинна використовувати PostgreSQL для транзакцій та Neo4j для графових обчислень.
NFR03	Масштабованість	Платформа повинна мати мікросервісну архітектуру для легкого масштабування.
NFR04	Надійність	Система повинна витримувати одночасну активну роботу не менше 500 користувачів без просадки продуктивності.
NFR05	Безпека	Доступ до системи має здійснюватися через SSO на базі O365.
NFR06	Захист даних	Система має відповідати законодавству GDPR щодо обробки та зберігання особистих даних.
NFR07	Конфіденційність	Збір даних має бути через інтеграційні точки та точки виклику системи.
NFR08	Точність ШІ	Моделі підбору працівників має мати щонайменше 85 % точності на основі тренувальної та реальної вибірок.
NFR09	Опис роботи алгоритму ШІ	Будь-які пропозиції ШІ мають супроводжуватись поясненням рішення.
NFR10	Резервне копіювання	Бази даних повинні автоматично проводити резервне копіювання EOD.
NFR11	Відновлення	Відновлення роботи системи має бути оперативним і займати до однієї робочої години.
NFR12	Адаптивний інтерфейс	Інтерфейс системи повинен бути адаптивним і коректно відображатися для Windows та MacOS ОС.

Завершення табл. 3.4

1	2	3
NFR13	Доступність	Користувацький інтерфейс має відповідати сучасним стандартам вебдоступу.
NFR14	Документація	Реалізація системи має супроводжуватись документацією та описовими рисунками/таблицями.

У табл. 3.5 наведено перелік перехідних вимог до продукту.

Таблиця 3.5

Перехідні вимоги проєкту

Код	Назва вимоги	Характеристика перехідної вимоги
1	2	3
TR01	Міграція даних	Необхідно розробити ETL-скрипти для перенесення історичних профілів працівників зі старих HR-систем чи таблиць Excel до нової БД.
TR02	Навчання персоналу	Необхідно провести навчання для кінцевих користувачів.
TR03	UAT	Перед впровадженням система повинна пройти етап тестування користувачами.

Точне визначення вимог дає можливість якісно та ефективно спроектувати архітектуру систем, інтерфейсів, бізнес-процеси та підходи до реалізації проєкту. Це також дає можливість ефективно спланувати задачі проєкту, їх послідовність та пріоритетність.

3.3 Формування беклогу продукту проєкту

Беклог продукту є важливим елементом Agile проєктів в являє собою впорядкований за пріоритетами список усього, що має бути створено, покращено або виправлено в продукті. Це дає команді повноцінне розуміння усього, що має бути виконано в проєкті, при цьому мати можливість оновлення за рахунок особливості Agile проєктів. Беклоги складаються з User Stories, що на основі спеціальної визначеної форми подачі кінцевої вимоги наводить розробників та дизайнерів на методи впровадження та рішення до реалізації. Гарно складені User Story – ключ до найкращої реалізації проєкту.

User story складаються на основі критеріїв прийняття, що англійською перекладаються як Acceptance Criteria, і визначають перелік кроків, станів та результатів, що мають бути досягнуті в межах користувацької історії. User story можуть мати декілька критеріїв прийняття. У табл. 3.6 наведено частковий перелік User Story (враховуючи об’єм інформації), а повний перелік наведено у табл. А.1 Додатку А.

Таблиця 3.6

Фрагмент переліку User Story проєкту

Код US	Назва та формулювання US	Acceptance Criteria
1	**2**	**3**
US01	*Авторизація в системі.* Як працівник компанії, я хочу авторизуватися через корпоративний SSO, щоб безпечно отримати доступ до свого робочого простору.	Scenario: Успішна авторизація. Given: Користувач на сторінці входу системи When: Вводить корпоративні облікові дані та підтверджує 2FA. Then: Система розпізнає роль користувача (PM, HR) And: Відкриває відповідний особистий кабінет.
US02	*Управління профілем компетенцій.* Як розробник, я хочу оновлювати інформацію про себе в особистому кабінеті, щоб система враховувала мої актуальні знання при підборі в команди.	Scenario: Додавання нової навички. Given: Користувач знаходиться в особистому профілі. When: Натискає кнопку «Додати нову навичку», обирає зі списку та натискає кнопку «Зберегти». Then: Навичка відображається у профілі і стає доступною для ШІ-алгоритму пошуку.
US03	***Інтеграція з трекерами задач.*** Як адміністратор, я хочу підключити Jira та GitHub через API, щоб система автоматично збирала метрики активності без ручного введення.	Scenario: Підключення Jira. Given: Користувач знаходиться в розділі «Налаштування інтеграцій». When: Вводить API-ключ Jira та натискає «Синхронізувати». Then: Статус змінюється на «Підключено», починається фоновий імпорт історичних даних працівників.
US04	*Створення профілю нового проєкту.* Як проєктний менеджер, я хочу створити новий проєкт та вказати необхідні ролі, щоб розпочати процес підбору команди.	Scenario: Створення проєкту. Given: Користувач на панелі управління проєктами. When: Заповнює назву, терміни, додає вакантні ролі та натискає «Створити». Then: Проєкт з’являється у списку зі статусом «Очікує формування команди».

Наведений вище перелік задач дає можливість структурувати та розподілити виконання задач та їх послідовність. Окрім цього, дотримання критеріїв прийняття напряму відповідає усім вимогам, що описано у розділі 3.2 «Визначення вимог проєкту».

3.4 Організаційна структура проєкту

Враховуючи Agile-підхід та високу технічну складність проєкту, найкраще підійде децентралізована крос-функціональна структура для реалізації системи. У традиційних моделях управління компанія поділяється на окремі відділи та підрозділи, що по-своєму створює бар'єри у взаємодії. До складу команди для проєкту мають увійти PO, Scrum майстер (PM), ШІ інженери, розробники, дизайнери, тестувальники та девопс.

Пласка ієрархія спирається на горизонтальні формати роботи, де рішення ухвалюються безпосередньо тими, хто створює цінність, а не проходять через довгі ланцюги керівництва. Команда не має класичного начальника, який роздає накази. Натомість власник продукту визначає що потрібно зробити, а розробники самоорганізуються, щоб вирішити як саме це реалізувати [38]. Склад команди наведено у табл. 3.7.

Таблиця 3.7

Склад проєктної команди

№	Роль в команді	Основні функціональні обов'язки
1	**2**	**3**
1	Product Owner	1.Формує бачення та стратегію розвитку продукту. 2.Працює з пріоритетами продуктового беклогу. 3.Узгоджує бізнес-вимоги зі стейкхолдерами 4.Приймає результати спринтів та проводить валідацію роботи.
2	Scrum майстер	1.Організовує планування, щотижневі-скрам, ретроспективи. 2.Сприяє самоорганізації крос-функціональної команди. 3.Усуває перешкоди в роботі. 4.Забезпечує дотримання Agile-практик.

Завершення табл. 3.7

1	2	3
3	Бізнес аналітик	1. Збирає та аналізує вимоги користувачів системи. 2.Перетворює HR-процеси у детальні технічні завдання. 3.Формує Acceptance Criteria для користувацьких історій.
4	Аналітик даних	1.Аналізує та очищує історичні дані телеметрії компанії. 2.Розробляє моделі машинного навчання. 3.Створює алгоритми прогнозування професійного вигорання. Навчає та тестує моделі для підвищення їхньої точності.
5	Backend / ML Розробник	1.Створює серверну логіку, мікросервіси та API. 2.Проєктує та адмініструє БД 3.Інтегрує алгоритми ШІ у бекенд платформи. 4.Налаштовує інтеграцію зі сторонніми сервісами.
6	Frontend Developer	1.Розробляє адаптивний клієнтський веб-інтерфейс. 2.Реалізує інтерактивність інструменту фільтрів оптимізаційного пошуку. 3.Інтегрує UI з серверними API та ШІ-модулями. 4.Розробляє візуалізації аналітики та теплових карт.
7	UI/UX Designer	1.Створює прототипи та макети користувацького інтерфейсу. 2.Проєктує логіку та зручність взаємодії користувача з системою. 3.Проводить тестування створених макетів. 4.Забезпечує зручне та зрозуміле представлення даних ШІ.
8	QA Engineer	1. Проводить ручне та автоматизоване тестування платформи. 2.Перевіряє коректність роботи математичних лімітів. 3.Валідує логіку багатокритеріальної видачі алгоритмів ШІ. 4.Забезпечує тестування безпеки та відповідність GDPR.
9	DevOps / MLOps Engineer	1.Налаштовує хмарну інфраструктуру та серверне середовище. 2.Автоматизує процеси CI/CD для програмного коду та ML-моделей. 3.Забезпечує моніторинг стабільності системи під навантаженням. 4.Керує резервним копіюванням та безпекою інфраструктури.

Такий підхід до команди ідеально підходить до реалізації даного проєкту, тим самим дає можливість:

- оперативно вести контроль виконання задач та дотримання часових дедлайнів;
- зменшити та ефективно контролювати ризики;
- ефективно планувати розвиток та впровадження кожної розробки;
- забезпечити ідеальну комунікацію та взаємодію в команді.

Взаємодія команди відображено на рис 3.1.

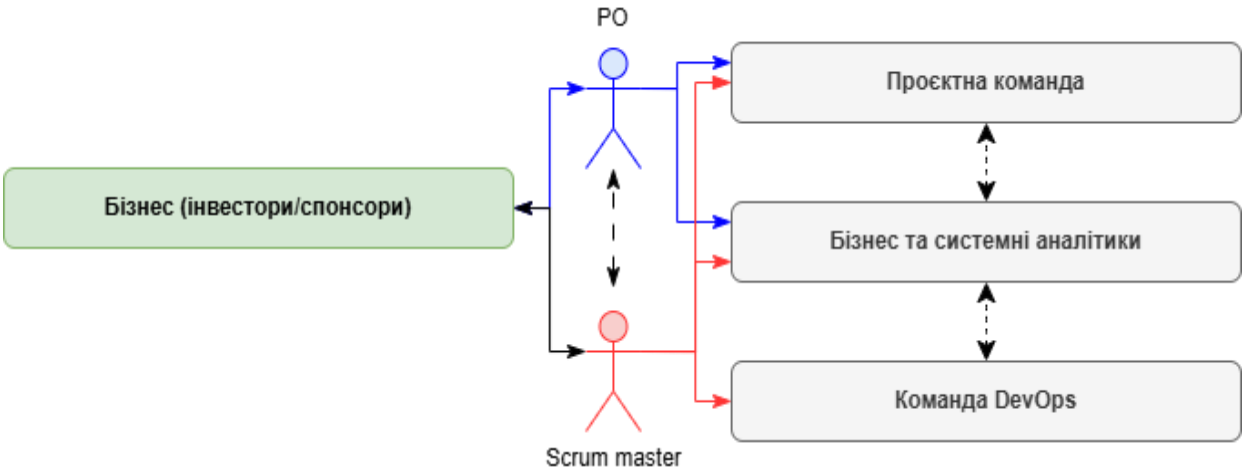


Рис 3.1. Організаційна структура управління проєктом

Така проєктна структура забезпечує використання Scrum-методології, тим самим мати високу ефективність, відповідальність та результативність.

3.5 Розробка WBS проєкту

Структура декомпозиції робіт (WBS – work breakdown structure) – це спеціалізована діаграма, що покликана для ієрархічного розбиття загального обсягу проєкту на окремі менші частини, що перетворюються в керовані компоненти спринтів. WBS [39] описує 100 % роботи, яка має бути виконана в розрізі проєкту, і допомагає візуалізувати проєкт та чітко відокремити завданнями за фазами життєвого циклу. Розбиття проєкту на дрібні роботи дозволяє детально оцінити час, бюджет та ресурси для кожного завдання. Набагато краще оцінювати витрати на рівні окремих підзадач, ніж намагатися оцінити весь проєкт цілком. Декомпозиція дає змогу потім закріпити виконання конкретної задачі за відповідним спеціалістом або відділом.

Життєвий цикл продукту для реалізації вебплатформи складається з наступних етапів, що зображені на рис 3.2.

Життєвий цикл проєкту розробки та впровадження інтелектуальної інформаційної системи управління людськими ресурсами (HRM) для IT-команд складається з повного переліку робіт передбачених проєктом.

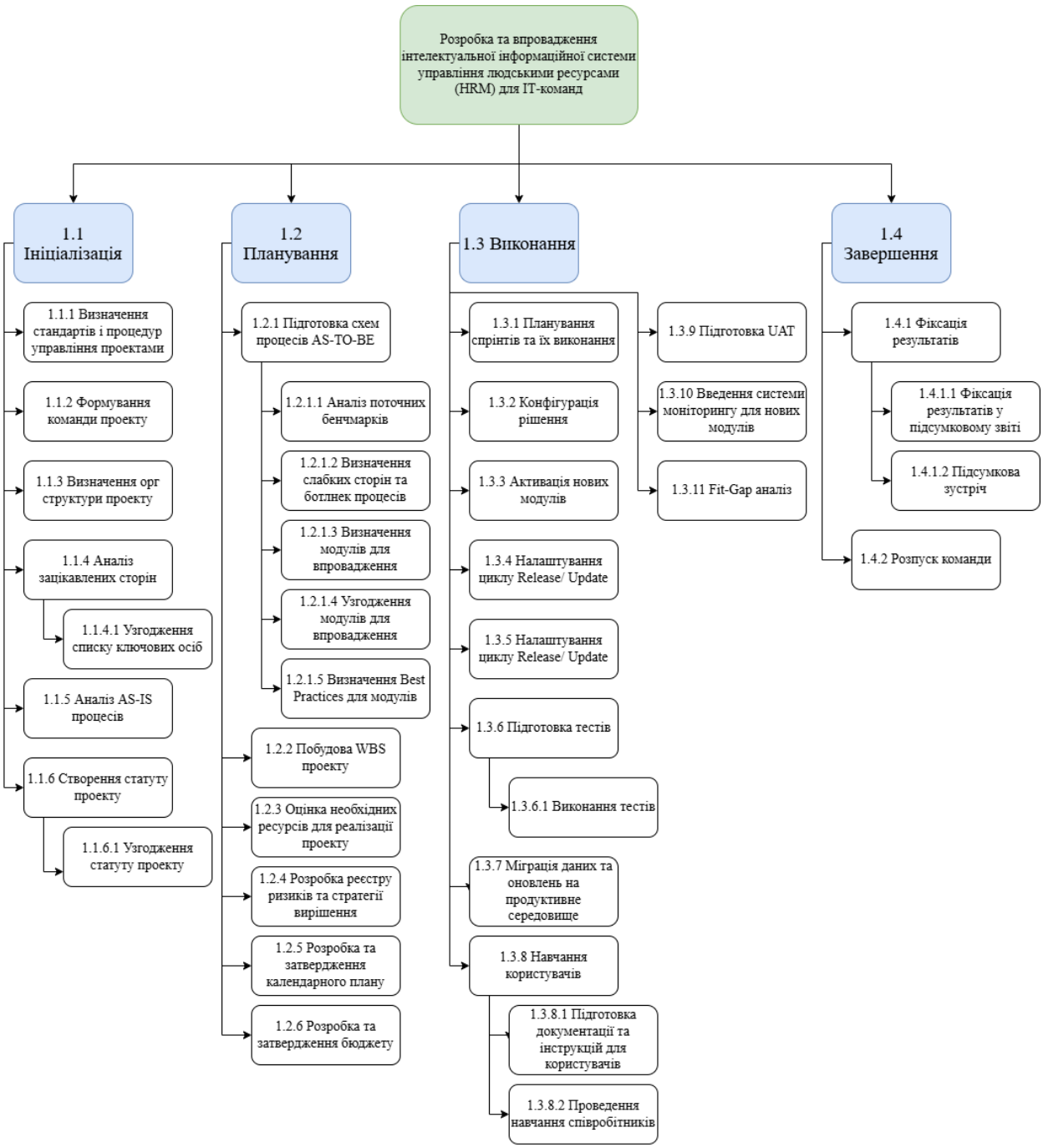


Рис. 3.2. Життєвий цикл проєкту

Запропонована структурована модель задач розбита за життєвими циклами проєкту дозволяє проєктному менеджеру мати візуальну картину виконання майлстоунів проєкту. Це також надає проєктній команді мати більш структурований та системний підхід до виконання поставлених задач.

3.6 Планування ресурсного забезпечення проєкту та бюджету

Реалізація HRM проєкту вимагає ретельного та детального планування фінансування, розподілення ресурсів та визначення оперативних, операційних та інструментних витрат передбачених реалізацією системи. Оскільки технічна складова проєкту включає розробка та навчання моделей ШІ, витрати можуть варіюватись в залежності від проведених тестувань та випробувань. Тому для реалізації такого проєкту потрібно обрати такий підхід фінансування та витрат ресурсів, аби досягти оптимальних та раціональних витрат.

Для оцінки вартості ІТ-проєкту потрібно використати гібрид методів Bottom-up та PERT. Як чітко було зазначено у SWOT-аналізі (зокрема, слабка сторона W03: «Висока кількість людино-годин розробки (інтеграція моделей ШІ у систему потребує значних витрат часу та фінансів на початкових етапах)»), проєкт має високий рівень складності та невизначеності. Робота з алгоритмами машинного навчання рідко піддається точній базовій оцінці, тому використання лише одного методу може призвести до серйозних похибок. За рахунок гібридизації можливо досягти наступних цілей:

1. Дати можливість адаптувати зміни до вимог, або потреб.
2. Швидкий перерозрахунок витрат на ресурси.

Базовий розрахунок вартості проєкту здійснюється методом Bottom-Up на основі створеної WBS, а для оцінки інтеграції ШІ-моделей (через високі ризики, ідентифіковані в W03) застосовано метод PERT. Такий гібридний підхід вважається найкращою практикою для цього проєкту.

При формуванні бюджету ІТ-проєкту також варто враховувати то факт, що майже всі сучасні проєкти мають модель «фіксованобюджетності», а тому важливо враховувати дану особливість при плануванні [40].

Сформувавши WBS, визначивши перелік зацікавлених сторін та систем, що приймають участь у процесі (визначені концептуальною моделлю), можливо створити перелік проєктної команди та її навантаження.

Табл. 3.8 містить перелік працівників та їх ролей, що будуть приймати участь в проєкті.

Таблиця 3.8

Перелік учасників проєктної команди

Роль	Кількість осіб	Орієнтовна завантаженість	Тривалість участі
1	**2**	**3**	**4**
Product Owner	1	50 %	10 міс.
Scrum Master	1	50 %	10 міс.
Business Analyst	1	50 %	3 міс.
Data Scientist	1	100 %	7 міс.
Backend / ML Engineer	3	100 %	9 міс.
Frontend Developer	2	100 %	8 міс.
UI/UX Designer	1	70 %	4 міс.
QA Engineer	2	100 %	6 міс.
DevOps / MLOps Engineer	1	50 %	10 міс.

Окрім цього, потрібно попіклуватись про вибір найкращих передових інструментів, що забезпечують ефективність та своєчасність виконання та роботи кінцевого продукту. У табл. 3.9 наведено відповідний перелік.

Таблиця 3.9

Технічне забезпечення проєкту

Категорія	Потреба	Коментарі
1	**2**	**3**
Хостинг/сервери та сервіси	AWS Cloud	Передбачає середовища DEV, TEST, PILOT, PROD
Система контролю версій	GitHub	Інструменти для командної розробки.
Таск-трекер	Jira	Ведення проєктної діяльності
Засоби комунікації	Teams	100 % віддалена робота (через ризик війни)
Дизайн-середовище	Figma	Для UI/UX прототипування.
CI/CD	GitHub Actions, Docker, Charlie	Автоматизація тестів і розгортання.

З урахуванням європейської середньостатистичної ставки співробітників станом на 2026, вартість витрат на людський ресурс прописано та прораховано у табл. 3.10.

Таблиця 3.10

Бюджет витрат на проєктну команду

Роль	К-ть	Год/міс	Міс.	Вартість людино-години (EUR)	Загальна вартість (EUR)
1	**2**	**3**	**4**	**5**	**6**
Product owner	1	80	11	22	19 360
Scrum master	1	80	10	22	17 600
Business analyst / HR expert	1	80	7	20	11 200
Data scientist	1	160	7	20	22 400
Backend / ML engineer	2	160	9	28	80 640
Frontend developer	1	160	7	28	31 360
UI/UX designer	1	112	4	18	8 064
QA engineer	1	160	6	18	17 280
DevOps / MLOps engineer	1	80	10	28	22 400
Загальна сума	-	-	-	-	*230 304*

Визначивши витрати на проєктну команду, можливо визначити бюджет, що виділяється на інфраструктурні та технологічні системи, а саму хостинг застосунку на хмарному сервері AWS, середовище для розробки та коміту коду GutHub. Інфраструктурні витрати прораховано у табл. 3.11.

Таблиця 3.11

Бюджет витрат на інфраструктуру команду

Матеріал	К-ть	Міс.	Вартість (EUR)
1	**2**	**3**	**4**
AWS Cloud (середовища DEV, TEST, PILOT, PROD)	1	10	2500
GitHub team	10 користувачів	10	400
Jira Software (платна версія)	10 користувачів	10	800
Microsoft teams	10 користувачів	10	600

Завершення табл. 3.11

1	2	3	4
Figma (Pro-акаунт)	1 користувач	4	60
GitHub Actions, Docker, Charlie (CI/CD)	1 підписка (командна)	10	300
Загальна сума	-	-	*4660*

В результаті оцінки, загальна сума проєкту становить 234964 євро, і триватиме 11 місяців.

3.7 Планування спринтів та візуалізація їх виконання за допомогою ProjectLibre

Планування спринту є фундаментальною частиною Agile-проєктів, в розрізі якої визначається напрямок та межі роботи команди на наступний період. Для розробки складних архітектурних рішень ця процедура є запобіжним інструментом, що трансформує верхньорівневе бачення продукту в контрольовані та вимірювані послідовні кроки. Планування дозволяє всім учасникам дійти єдиного розуміння того, яку саме бізнес-цінність буде створено найближчим часом. Під час спільного обговорення завдань команда може заздалегідь виявити потенційні блокери: на прикладі проєкту HRM це інтеграція ШІ-алгоритмів, чи нестача вузькопрофільних фахівців.

Великі системні блоки розбиваються на конкретні User Stories з чітко визначеними критеріями прийняття та відповідні технічні підзадачі. Це дозволяє команді детально продумати логіку реалізації – наприклад, як саме оптимізувати базу даних, що містить інформацію про навантаження працівників тощо. У табл. 3.12 наведено фрагмент найважливіших User Story, а повний перелік наведено у табл. А.2 Додатку А.

Таблиця 3.12

Фрагмент спринтів та задач проєкту

Код та назва US	**Задача**	**Підзадача**	**Роботи, які потрібно виконати**
1	**2**	**3**	**4**
US01: Авторизація	T.01.1 Налаштування SSO	ST.01.11 Конфігурація IdP	1.Реєстрація_додатку_в_Azure_AD.
			2.Обмін_SAML/OAuth_токенами.
			3.Перевірка_стану_2FA.
			4.Тестування_входу_через_SSO.
US01: Авторизація	T.01.2 Маршрутизація RBAC	ST.01.21 Налаштування редиректів	1.Читання_ролі_з_JWT-токена.
			2.Мапінг_ролей_до_сторінок.
			3.Створення_сесії_в_системі.
			4.Перевірка_редиректів.
US02: Профіль компетенцій	T.02.1 UI особистого кабінету	ST.02.11 Форма навичок	1.Макет_профілю_з_компетенціями.
			2.Компонент_вибору_навичок.
			3.Кнопка_збереження.
			4.Відображення_доданих_тегів.
US02: Профіль компетенцій	T.02.2 Оновлення індексів	ST.02.21 Логіка збереження	1.API_збереження_навичок.
			2.Синхронізація_з_ШІ-пошуком.
			3.Валідація_на_дублікати.
			4.Юніт-тест_доступності.
US03: Інтеграція з трекерами	T.03.1 Інтерфейс управління	ST.03.11 Сторінка налаштувань	1.UI_для_ключів_Jira/GitHub.
			2.Кнопка_синхронізації.
			3.Шифрування_ключів_на_клієнті.
			4.Валідація_порожніх_полів.
US03: Інтеграція з трекерами	T.03.2 Фоновий збір метрик	ST.03.21 Воркери імпорту	1.Мікросервіс_підключення_до_API.
			2.Черга_завантаження_історії.
			3.Мапінг_ID_користувачів.
			4.Стрес-тест_імпорту.
			4.Валідація_дат_початку/кінця.
US04: Профіль нового проєкту	T.04.2 Реєстрація проєкту	ST.04.21 Збереження в БД	1.Таблиці_Projects_і_Roles.
			2.API_прийому_даних.
			3.Присвоєння_статусу_очікування.
			4.Інтеграційні_тести_зв'язків.

Визначивши достатню кількість інформації за спринтами, можемо провести планування трьох спринтів та використання ресурсів. Приклади наведено у відповідних табл. 3.13 – 3.15.

Таблиця 3.13

Розподіл задач за першим спринтом

Задача / Підзадача	Назва роботи	Тривалість (год)	Виконавець	Матеріал, технології
1	2	3	4	5
ST.01.11	Реєстрація додатку в Azure AD	4	DevOps Engineer	Azure Portal
ST.01.11	Обмін SAML / OAuth токенами	6	Backend Developer	Node.js, MSAL
ST.01.11	Перевірка стану 2FA	4	Backend Developer	Node.js, REST API
ST.01.11	Тестування входу через SSO	4	QA Engineer	Postman, Cypress
ST.01.21	Читання ролі з JWT-токена	3	Frontend Developer	JavaScript
ST.01.21	Мапінг ролей до сторінок	3	Frontend Developer	React Router
ST.01.21	Створення сесії в системі	4	Backend Developer	Redis, Express.js
ST.01.21	Перевірка редиректів	3	QA Engineer	Selenium / Cypress
ST.02.11	Макет профілю з компетенціями	4	UI/UX Designer	Figma
ST.02.11	Компонент вибору навичок	5	Frontend Developer	React, Select2
ST.02.11	Кнопка збереження	2	Frontend Developer	JavaScript, CSS
ST.02.11	Відображення доданих тегів	3	Frontend Developer	React
ST.02.21	API збереження навичок	4	Backend Developer	PostgreSQL, Node.js
ST.02.21	Синхронізація з ШІ-пошуком	6	Data Engineer	ElasticSearch, Python
ST.02.21	Валідація на дублікати	2	Backend Developer	SQL
ST.02.21	Юніт-тест доступності	3	QA Automation	Charlie

Таблиця 3.14

Розподіл задач за другим спринтом

Задача / Підзадача	Назва роботи	Тривалість (год)	Виконавець	Матеріал, технології
1	2	3	4	5
ST.04.11	Макет списку проєктів	4	UI/UX Designer	Figma
ST.04.11	Форма: Назва, Опис, Терміни	5	Frontend Developer	React, HTML/CSS
ST.04.11	Блок вибору ролей	4	Frontend Developer	React
ST.04.11	Валідація дат початку/кінця	3	Frontend Developer	JavaScript
ST.04.21	Таблиці Projects і Roles	4	Backend Developer	PostgreSQL
ST.04.21	API прийому даних	5	Backend Developer	Node.js, Express
ST.04.21	Присвоєння статусу очікування	2	Backend Developer	JavaScript, SQL
ST.04.21	Інтеграційні тести зв'язків	4	QA Automation	Charlie
ST.03.11	UI для ключів Jira/GitHub	4	Frontend Developer	React, VS Code
ST.03.11	Кнопка синхронізації	2	Frontend Developer	JavaScript, CSS
ST.03.11	Шифрування ключів на клієнті	4	Frontend Developer	Crypto.js
ST.03.11	Валідація порожніх полів	2	QA Engineer	Chrome DevTools
ST.03.21	Мікросервіс підключення до API	8	Backend Developer	Python, Requests
ST.03.21	Черга завантаження історії	6	Backend Developer	Celery, RabbitMQ
ST.03.21	Мапінг ID користувачів	4	Backend Developer	SQL
ST.03.21	Стрес-тест імпорту	5	QA Automation	Apache JMeter

Таблиця 3.15

Розподіл задач за третім спринтом

Задача / Підзадача	Назва роботи	Тривалість (год)	Виконавець	Матеріал, технології
1	2	3	4	5
ST.05.11	Вибір тегів Hard Skills	4	Frontend Developer	React
ST.05.11	Повзунки Soft Skills	4	Frontend Developer	HTML5, CSS
ST.05.11	Скидання налаштувань	2	Frontend Developer	JavaScript
ST.05.21	Поля вимог у таблиці	3	Backend Developer	PostgreSQL
ST.05.21	Серіалізація фільтрів у JSON	3	Backend Developer	Node.js
ST.05.21	API оновлення вимог	4	Backend Developer	REST API
ST.05.21	Перевірка збереження	3	QA Engineer	Postman
ST.06.11	Формування Payload	4	Backend Developer	Node.js
ST.06.11	Відправка запиту до ШІ	5	Data Scientist	Python
ST.06.11	Обробка тайм-аутів	3	Backend Developer	Axios, JavaScript
ST.06.11	Обробка порожньої видачі	2	Backend Developer	Node.js
ST.06.21	Картка працівника	5	Frontend Developer	React
ST.06.21	Анімація очікування (Скелетон)	3	Frontend Developer	CSS Animations
ST.06.21	Повідомлення відсутності	2	Frontend Developer	React
ST.06.21	Рендеринг списку	4	Frontend Developer	JavaScript, React DOM

Для візуалізації виконання спринтів було використано аналог MS Project (через вартість ліцензії застосунку) ProjectLibre [40] – програмне забезпечення для управління проєктами, що містить усі базові інструменти традиційного управління проєктами, зокрема діаграми Ґанта, мережеві графіки, ієрархічну структуру робіт (WBS), управління задачами, алокацію ресурсів та відстеження витрат.

Графік складається з окремих майлстоунів до яких входять спринти. Кожен спринт містить задачі та підзадачі, що закріплені за певним працівником та має орієнтовну оцінену тривалість виконання задачі.

Повна діаграма Ґанта виконана у застосунку ProjectLibre, однак для прикладу першого спринту на рис. 3.3 наведено приклад візуалізації, а на рис. 3.4-3.5 проєктну команду та її навантаження.

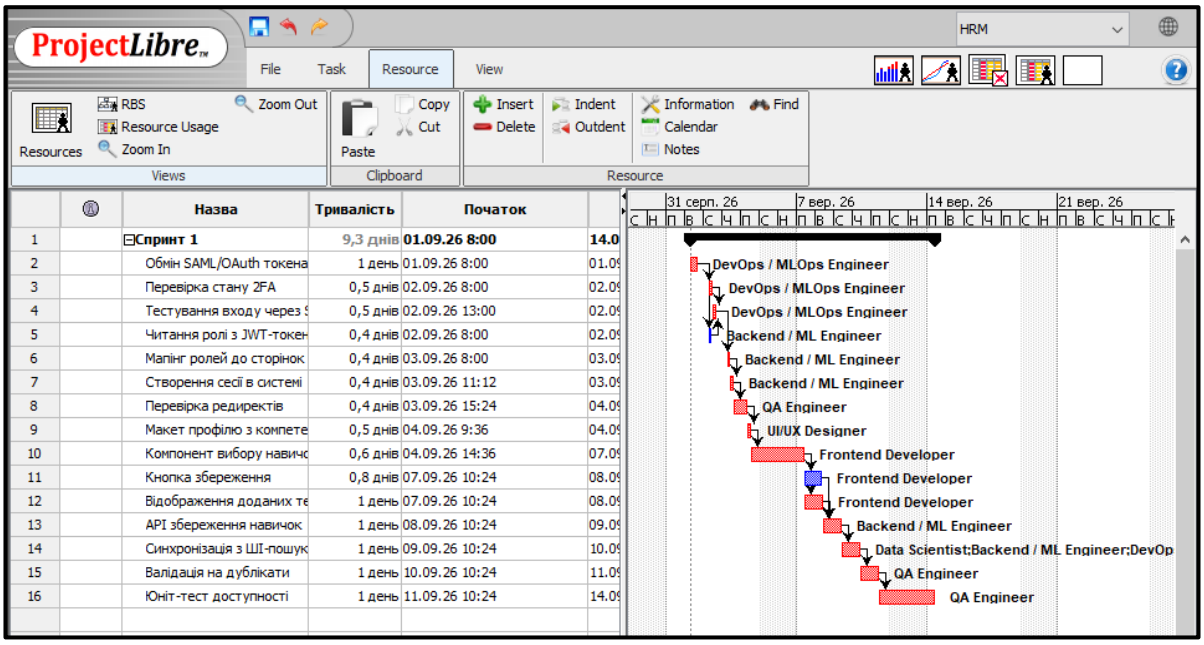


Рис 3.3. Візуалізація задач спринту та призначення співробітників

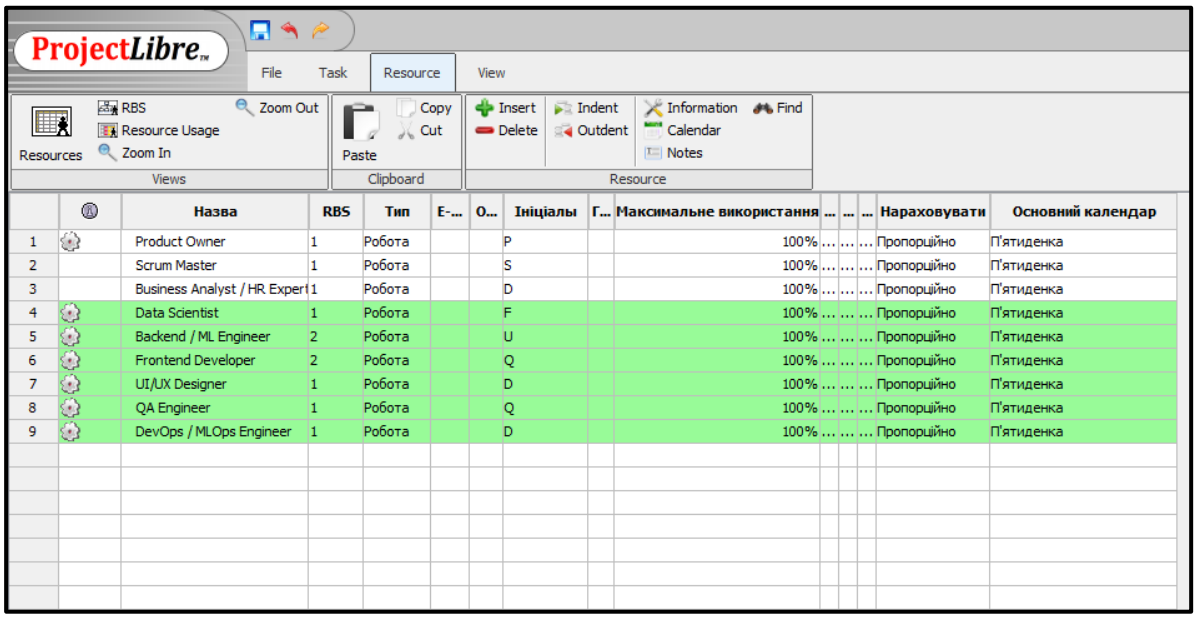
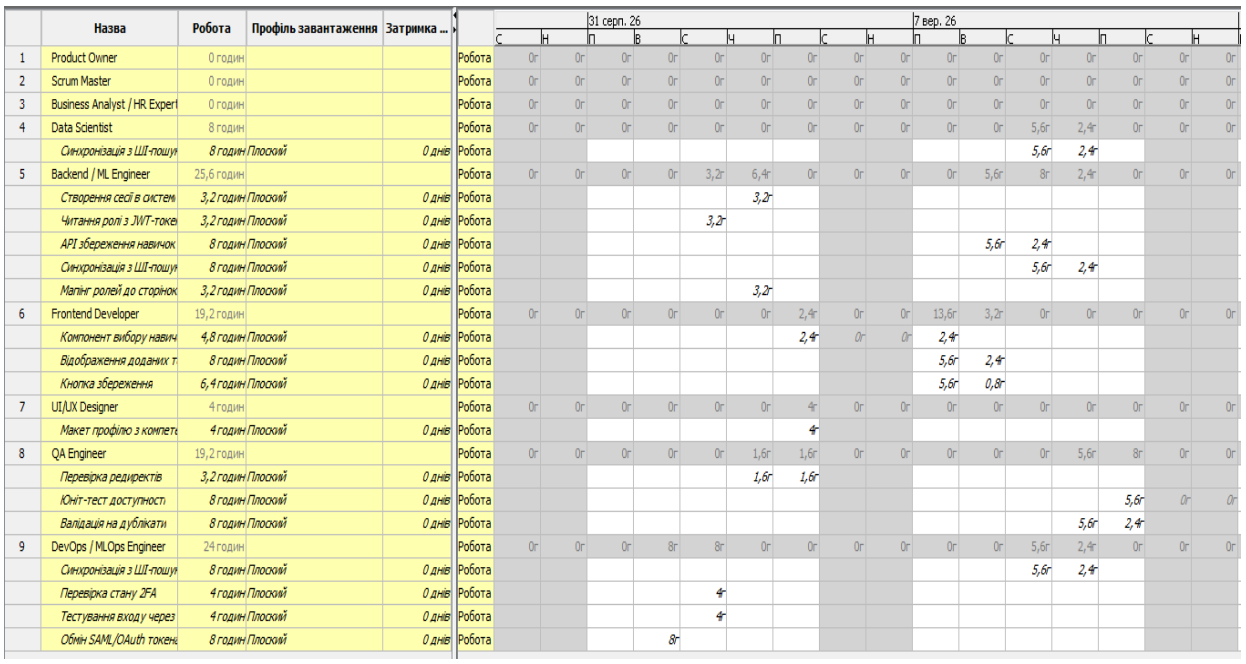


Рис 3.4. Візуалізація складу проєктної команди



	Назва	Робота	Профіль завантаження	Затримка ...
1	Product Owner	0 годин		
2	Scrum Master	0 годин		
3	Business Analyst / HR Expert	0 годин		
4	Data Scientist	8 годин		
	Синхронізація з ШІ-пошук	*8 годин*	*Плоский*	*0 днів*
5	Backend / ML Engineer	25,6 годин		
	Створення сесії в систем	*3,2 годин*	*Плоский*	*0 днів*
	Читання ролі з JWT-токе	*3,2 годин*	*Плоский*	*0 днів*
	API збереження навичок	*8 годин*	*Плоский*	*0 днів*
	Синхронізація з ШІ-пошук	*8 годин*	*Плоский*	*0 днів*
	Мапінг ролей до сторінок	*3,2 годин*	*Плоский*	*0 днів*
6	Frontend Developer	19,2 годин		
	Компонент вибору навич	*4,8 годин*	*Плоский*	*0 днів*
	Відображення доданих т	*8 годин*	*Плоский*	*0 днів*
	Кнопка збереження	*6,4 годин*	*Плоский*	*0 днів*
7	UI/UX Designer	4 годин		
	Макет профілю з компете	*4 годин*	*Плоский*	*0 днів*
8	QA Engineer	19,2 годин		
	Перевірка редиректів	*3,2 годин*	*Плоский*	*0 днів*
	Юніт-тест доступності	*8 годин*	*Плоский*	*0 днів*
	Валідація на дублікати	*8 годин*	*Плоский*	*0 днів*
9	DevOps / MLOps Engineer	24 годин		
	Синхронізація з ШІ-пошук	*8 годин*	*Плоский*	*0 днів*
	Перевірка стану 2FA	*4 годин*	*Плоский*	*0 днів*
	Тестування входу через	*4 годин*	*Плоский*	*0 днів*
	Обмін SAML/OAuth токен	*8 годин*	*Плоский*	*0 днів*

Рис 3.5. Навантаження проєктної команди на Спринт 1

Візуально сформована покроковість виконання задач та розподілення їх виконання між працівниками дає можливість Проєктному менеджеру швидко орієнтуватись у ході та результативності виконання поставлених завдань.

3.8 Ідентифікація проєктних ризиків та розробка карти управління ними

Ідентифікація ризиків є фундаментальним етапом управління будь-яким IT-проєктом, оскільки вона дозволяє команді перейти до проактивного планування та запобігання проблеми. Згідно з методологіями управління проєктами, як, наприклад, PMBOK, розуміння невизначеностей на ранніх етапах допомагає зберегти бюджет, дотриматися графіка та забезпечити якість кінцевого продукту, особливо для реалізації подібних проєктів в HR аспекті [42].

Спираючись на SWOT-аналіз, ідентифікація ризиків є критично необхідною для успішної реалізації проєкту.

У цьому проєкті передбачена розробка ШІ, робота з даними, що підпадають під жорсткі законодавчі обмеження як GDPR та європейський AI

Act. Окрім цього, треба мати на увазі прямі ризики пов’язані з війною в Україні, що робить прогнозування доступності працівників для майбутніх проєктів непередбачуваним. Також треба враховувати, що система залежить від сторонніх сервісів, зокрема хостинг на AWS, використання GitHub тощо. Саме тому необхідно підібрати правильний процес управління та реагуванням на ризики.

Процес управління ризиками в провідних IT-проєктах – це систематичний підхід до виявлення, оцінки та реагування на загрози і можливості проєкту. Для забезпечення успішного результату цей процес має бути безперервним та інтегрованим у загальний життєвий цикл розробки [43].

Згідно з міжнародними стандартами, процес управління ризиками зазвичай складається з кількох послідовних етапів. На етапі планування управління ризиками визначаються методологія, інструменти та джерела даних. Формується загальна стратегія, встановлюються критерії оцінки, такі як шкали ймовірності та впливу, а також розподіляються ролі та відповідальність серед учасників команди.

Наступним кроком є ідентифікація ризиків, що являє собою процес систематичного виявлення потенційних подій, які можуть позитивно або негативно вплинути на цілі. Враховуються як технічні загрози, наприклад, неконтрольоване розростання вимог або «feature creep», так і людський фактор, когнітивні упередження чи проблеми з розподілом ресурсів [44-45].

Після цього ідентифіковані ризики проходять аналіз та оцінку для визначення їхньої пріоритетності. Цей етап включає якісний аналіз – суб'єктивну оцінку ймовірності виникнення та рівня впливу кожного ризику, а також кількісний аналіз. Використання інструментів предиктивного моделювання під час кількісного аналізу дозволяє отримати більш точні прогнози щодо відхилень у бюджеті чи графіку [46].

Далі відбувається планування реагування на ризики, що передбачає розробку конкретних дій для мінімізації загроз та максимізації можливостей.

Для негативних ризиків застосовуються стратегії уникнення, передачі, зниження (передачі) або прийняття [43; 47].

Завершальним і постійним етапом є моніторинг та контроль, який включає безперервне відстеження ідентифікованих ризиків та оцінку ефективності впроваджених заходів реагування [48].

У табл. 3.15 наведено частковий перелік ризиків, а у табл. А.3 Додатку А наведено повний перелік ризиків.

Таблиця 3.15

Фрагмент таблиці з ідентифікацією ризиків проєкту

Код ризику	Тип ризику	Ризикова подія	Сила впливу	Керованість
1	2	3	4	5
Р01	Форс-мажор	Тривалі відключення електроенергії (блекаути), що зупиняють роботу команди	висока	середня
Р02	Форс-мажор	Мобілізація ключових співробітників проєкту	висока	середня
Р03	Форс-мажор	Фізична загроза життю команди через обстріли та повітряні тривоги	висока	низька
Р04	Форс-мажор	Нестабільність інтернет-з'єднання у регіонах перебування команди	висока	середня
Р05	Технічні	Складність реалізації технології миттєвих зрізів у реальному часі	висока	середня
Р06	Технічні	Неточності в роботі алгоритмів ШІ	висока	середня
Р07	Технічні	Висока затримка при обробці великих масивів даних	середня	висока
Р08	Технічні	Уразливість вебплатформи до кібератак та витоку даних	висока	середня
Р09	Технічні	Складність інтеграції платформи з існуючими HRM/ERP системами замовників	середня	середня
Р10	Технічні	Вибір архітектури, що не масштабується під зростаюче навантаження	висока	висока
Р11	Управлінські	Неконтрольоване розростання вимог до функціоналу	висока	середня
Р12	Управлінські	Помилки в оцінці строків та бюджету розробки модулів машинного навчання	висока	середня
Р13	Управлінські	Низький рівень комунікації між Data Science відділом та backend-розробниками	середня	висока

Завершення табл. 3.15

1	2	3	4	5
P14	Управлінські	Недостатня залученість стейкхолдерів на етапі тестування прототипу	середня	висока
P15	Управлінські	Відсутність або низька якість технічної документації API	середня	висока

Після ідентифікації ризиків, необхідно провести якісний та кількісний аналіз, аби отримати експертну думку щодо можливості настання ризиків, способів їх вирішення та визначення важливості. Оцінювання ризиків проводиться за параметрами можливих затримок у часі, фінансових витрат, ймовірності настання (на думку експерта), частота настання та важливість для проєкту.

У табл. 3.16 наведено оцінку частину таких ризиків та їх критичність, а у табл. А.4 Додатку А повний перелік:

Таблиця 3.16

Фрагмент таблиці з якісною оцінкою ризиків проєкту

№	Ризикова подія	Затримки у часі		Фінансові втрати		Ймовірність		Частота за проєкт		Важливість ризику
		ЯО	КО	ЯО	КО	ЯО	КО	ЯО	КО	
1	2	3	4	5	6	7	8	9	10	11
2	Мобілізація ключових співробітників проєкту	ВВ	9	ВВ	8	ВС	8	ВС	6	31
3	Фізична загроза життю команди через обстріли та повітряні тривоги	ВВ	10	ВВ	9	СВ	5	ВС	6	30
4	Нестабільність інтернет-з'єднання у регіонах	ВС	6	СВ	5	ВС	7	ВС	7	25
5	Складність реалізації технології миттєвих зрізів у реальному часі	ВС	7	ВС	6	ВС	6	ВС	6	25

Завершення табл. 3.16

1	2	3	4	5	6	7	8	9	10	11
6	Неточності в роботі алгоритмів ШІ при предиктивному аналізі ризиків	ВС	6	СВ	5	СВ	5	СВ	5	21
7	Висока затримка при обробці великих масивів даних	СС	4	СС	4	ВС	7	ВС	7	22
8	Уразливість вебплатформи до кібератак та витоку даних	ВВ	8	ВВ	9	СН	4	СС	4	25
9	Складність інтеграції платформи з існуючими HRM/ERP	ВС	6	ВС	6	ВС	6	ВС	6	24
10	Вибір архітектури, що не масштабується під зростаюче навантаження	ВС	7	ВВ	8	СН	4	СН	4	23

Здійснивши деталізовану оцінку якісних та кількісних показників визначених ризиків, необхідно провести реалізацію карти протиризикових заходів – спеціалізований інструмент управління проєктами, який структурує конкретні кроки для запобігання, раннього виявлення та оперативного реагування на ідентифіковані загрози. На відміну від класичного реєстру ризиків, що переважно фіксує ймовірність та наслідки, така карта має яскраво виражений практичний характер і перетворює теоретичний аналіз на покроковий план дій. Вона слугує інструкцією для команди, забезпечуючи чіткий розподіл стратегій на етапах до та після виникнення критичної ситуації [46-54].

У табл. 3.17 наведено фрагмент карти протиризикових заходів, а у табл. А.5 Додатку А повний перелік.

Таблиця 3.17

Фрагмент карти протиризикових заходів проєкту

№	Ризикова подія	ПРЗ 1 (профілактика)	Симптом (рання ознака)	ПРЗ 2 (при симптомі)	ПРЗ 3 (при проблемі)
1	2	3	4	5	6
1	Тривалі відключення електроенергії (блекаути)	Забезпечення команд автономним живленням (EcoFlow, генератори)	Публікація графіків жорстких відключень	Перехід на гнучкий/асинхронний графік роботи	Робота в режимі offline-розробки, перенесення релізів
2	Мобілізація ключових співробітників проєкту	Бронювання фахівців, перехресне навчання	Отримання повістки для уточнення даних	Термінова передача знань	Перерозподіл задач, екстрений пошук заміни на ринку
3	Фізична загроза життю через обстріли	Можливість релокації	Збільшення інтенсивності «сигналу повітряної тривоги» у регіонах	Зупинка нарад, перехід в укриття	Надання оплачуваних відпусток для відновлення, зсув дедлайнів
4	Нестабільність інтернет-з'єднання	Оплата резервних провайдерів, Starlink	Часті відключення на зустрічах, затримки пушів коду	Перехід на мобільний інтернет, текстові звіти	Компенсація коворкінгів з гарантованим зв’язком
5	Складність реалізації технології миттєвих зрізів	Створення PoC на ранніх етапах, залучення tech lead	Відставання в оцінках часу на розробку модуля	Парне програмування, архітектурні консультації	Перегляд архітектури, спрощення частоти зрізів
6	Неточності в роботі алгоритмів ШІ при аналізі	Використання якісних датасетів, A/B тестування моделей	Хибнопозитивні результати на етапі тестування	Додаткове тренування моделі	Тимчасове відключення ШІ-модуля, ручний аналіз ризиків
7	Висока затримка при обробці масивів даних	Використання асинхронних черг (Kafka / RabbitMQ)	Відгук API перевищує SLA (затримки > 1-2 с)	Профілювання коду, оптимізація запитів до БД	Кешування через Redis, масштабування серверів
8	Уразливість до кібератак та витоку даних	Аудит коду, впровадження WAF, наскрізне шифрування	Аномальна активність у логах, скарги користувачів	Блокування IP, активація жорстких правил брандмауера	Відключення вузлів, патчінг помилок
9	Складність інтеграції з існуючими ERP системами	Розробка універсального API, детальне вивчення документації	Помилки мапінгу даних під час першої інтеграції	Створення кастомних адаптерів для специфічних ERP	Залучення інтеграторів клієнта, обмеження списку підтримуваних систем

Завершення табл. 3.17

1	2	3	4	5	6
10	Вибір архітектури, що не масштабуєтьс я	Навантажувальн е тестування, cloud-native підходи	Зависання платформи при пікових навантаженя х	Автоматичне горизонтальне масштабування	Екстрене збільшення потужностей інстансів, рефакторинг
11	Неконтро-льоване розростання вимог	Жорсткий процес Change Request, фіксація Scope	Поява нових «критичних» вимог посеред спринту	Перенесення нових ідей у беклог	Перегляд бюджету / термінів, заморожування нових фіч до релізу MVP
12	Помилки в оцінці строків розробки модулів ШІ	Залучення Senior data scientists, закладання буферу 25 %	Відставання від графіка розробки понад 15 %	Декомпозиція задач, ескалація проблеми на Project Manager	Використання готових pre-trained моделей/API замість кастомних
13	Низький рівень комунікації Data Science та backend	Спільне планування архітектури, жорсткі API Contracts	Несумісність форматів даних під час злиття гілок	Фасилітовані зустрічі, уточнення контрактів у Swagger	Призначення єдиного архітектора за точку інтеграції
14	Недостатня залученість стейкхолдерів	Узгодження графіка демо-сесій, налаштування UAT середовища	Відсутність зворотного зв'язку після демо-релізів	Індивідуальні дзвінки, опитувальники для збору фідбеку	Блокування переходу на наступний етап до підписання актів UAT

У межах роботи з ризиками командою була проведена плідна праця щодо ідентифікації та визначення ризиків, якісній та кількісній оцінці та подальше створення карти протиризикових заходів, для можливості оперативно управляти та реагувати на ризики.

Варто окремо зазначити, що процес управління ризиками не є одноразовим – це тривалий ітераційний процес моніторингу, переоцінки та визначення нових проблем та способів реагування на них. Новітні методології та підходи ідеально підходять для подібного Agile-проєкту.

В результаті чого, це дає проєктній команді перейти до розробки основних складових програмного забезпечення.

3.9 Управління процесами контролю якості в проєкті

Управління якістю проєкту – це процеси, що об’єднують усю операційну діяльність в проєкті / компанії, що визначають правила, політику взаємовідносин, цілі та розподіл відповідальності з точки зору якості кожного функціонального напряму, аби проєкт відповідав кінцевим потребам та вимогам. Управління якістю проєктом є постійним процесом, що проводиться з певною інтенсивністю (наприклад раз на спринт), аби визначити чи відповідає хід розробки проєкту поставленого плану. Основними процесами управління є:

1) Планування якості – визначення основних стандартів якості, що закріплені за цим проєктом (та як їх притримуватись).
2) Забезпечення якості – виконання планових систематичних операцій щодо дотримання якості, що забезпечують виконання усіх процесів, що спрямовані на дотримання вимог проєкту.
3) Контроль якості – моніторинг конкретних результатів з ціллю визначення відповідності прийнятим стандартам проєкту та визначення шляхів уникнення / вирішення проблем та причин, які зумовлюють невиконання вимог.

Контроль якості проєкту має бути спрямований як на проєкт, так і на продукт його реалізації. Для прикладу з реалізацією HRM системи, як програмного продукту, потрібні конкретні методи та правила для контролю якості. Невиконання вимог та недотримання цілей може призвести до понаднормової роботи команди (що призводить до перевтоми працівників) та відхилення від графіку виконання проєкту (досягнення цілей проєкту та вимог є невчасними). На рис. 3.6 наведено загальну схему управління якістю проєкту. Модель побудована на основі моделі управління якістю в PMBOK [55].

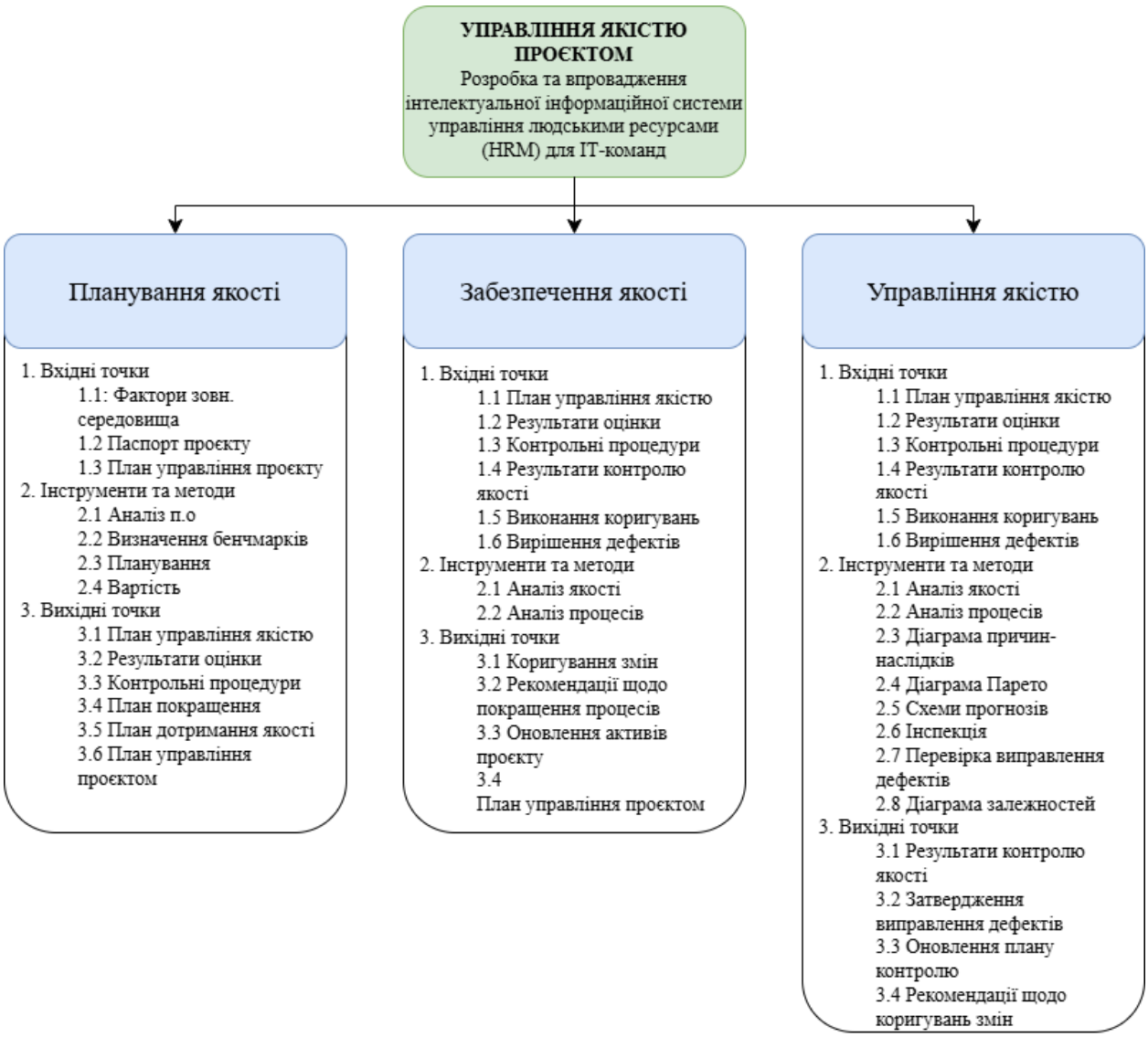


Рис 3.6. Загальна схема контролю якості проєктом HRM

Планування якості проєкту має базуватись на актах, правилах та політики компанії, які поширюються на проєкт. Також мають бути відображені відповідні практики та стандарти, наприклад стандарти розробки IT-проєктів, ведення мікросервісної архітектури тощо. На проєкт можуть впливати різні чинники, як, зокрема, війна в Україні, політична ситуація на близькому сході та ін. У акти може входити інформація щодо порогових витрат на проєкт (можливе % значення перевищення вартості проєкту), або пошук нових ресурсів. Важливим аспектом є критерії прийняття – вони мають бути досягнуті відповідним чином та позитивно впливати на результати проєкту. Під час планування проєкту має бути використаний аналіз витрат

(витрати на операційну діяльність, зокрема заробітні плати спеціалістів, підписки тощо), визначення бенчмарків (рівень написання коду, дотримання стандартів), проведення експериментів (статичний метод для визначення факторів, що мають зовнішній та невизначений рівень впливу), вартість якості (скільки коштує проводити операції з дотримання якості та чи вони релевантні) та інші інструменти планування якості (наприклад, мозковий штурм, діаграми відповідності процесів, аналіз силових полів, методи номінальних груп, матричні методи тощо). Як результат проєктна команда має повноцінний план контролю якості та методи оцінки процесів.

У заходах контрою якості прийматимуть участь основні представники Замовника, що формуватимуть спільну оцінку, Product Owner та зовнішній аудит. У табл. 3.18 наведено приклад плану для першого спринту проєкту.

Таблиця 3.18

Фрагмент плану управління якістю

№ Спринту	**Назва спринту**	**Дати контоолю**			**Експерти**		
		Дата 1	**Дата 2**	**Дата 3**	**COMP**	**PO**	**Audit**
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
S1.1-1	Створення основного функціоналу front-end для формування проєкту та команди	*01.09.2026*	*10.09.2026*	*15.09.2026*	(0-100)	(0-100)	(0-100)
S1.1-2	Програмування та навчання моделі оптимізаційного підбору працівників для проєкту	*01.09.2026*	*12.09.2026*	*21.09.2026*	(0-100)	(0-100)	(0-100)

Завершення табл. 3.18

1	2	3	4	5	6	7	8
S1.1-3	Створення мікросервісної архітектури та налаштування DEV, TEST, PROD середовищ	*01.09.2026*	*05.09.2026*	*10.09.2026*	(0-100)	(0-100)	(0-100)
S1.2-1	Налаштування взаємодії мікросервісної архітектури та front-end частини проєкту	*20.09.2026*	*29.09.2026*	*05.10.2026*	(0-100)	(0-100)	(0-100)
S1.2-2	Проєктування зовнішньої інтеграції та створення функціоналу для оновлення БД та підтримки записів	*21.09.2026*	*30.09.2026*	*10.10.2026*	(0-100)	(0-100)	(0-100)
…	…	…	…	…	…	…	…
Результат		*0д*	*1д*	*3д*	*95%*	*90%*	*98%*

Ціллю оцінки якості є досягнення 95 % усіма учасниками та проведення вчасних контролів із визначенням середньої кількості днів, коли контролі проводились раніше, або пізніше зазначеної дати. Експерти, що проводять оцінку, мають посилатись на план проєкту, його вимоги та цілі, аби якісно провести оцінку. Експерти мають визначити точки та дати контролю, що добре, а що погано та способи покращення.

Процеси забезпечення якості складається з плану, що описує певну послідовність дій як дотримуватись якості в проєкту. Він має складати загальний перелік робіт, виконавці, початкову та кінцеву дати, відношення до спринту, очікувана та фактична відповідність вимогам. Окрім цього процес має забезпечувати логіку запровадження змін до виявлених проблем проєкту

та вартість їх усунення, оскільки це є прямими ризиками проєкту. Для дотримання найвищого рівня якості, пропонується використання сторонньої організації, яка проводитиме аудит виконаних робіт і складатиме оцінку роботи, та способи покращення (за необхідності). Результатом є перелік запропонованих змін, bottleneck процесів та рекомендації щодо покращення. Окрім цього, проєктним менеджером має оновитись план управління проєктом та переглянуто дати (дедлайни).

Процеси контролю якості включають в себе моніторинг конкретних результатів (визначених у табл. 3.18), формулювання загальної аналітики та відповідності стандартам проєкту. Управління якістю має бути безперервним та проводитись на всіх етапах проєкту. Оскільки процес є Agile і розглядається використання лише наявних ресурсів (без використання сторонніх підрядників, або фріланс) в розрізі проєкту пропозиції щодо покращення будуть описуватись у конкретних спринтах, а загальна аналітика складатись по кожному спринту окремо, що потім сформує єдину загальну за проєктом. Конкретні процеси та функції для покращення мають розбиратись на ретроспективах.

Одним з найкращих інструментів для контролю якості є діаграма причино-наслідкових зв’язків та їх формування. На рис. 3.7 зображено приклад такої діаграми на імітованому дефекті проєкту:

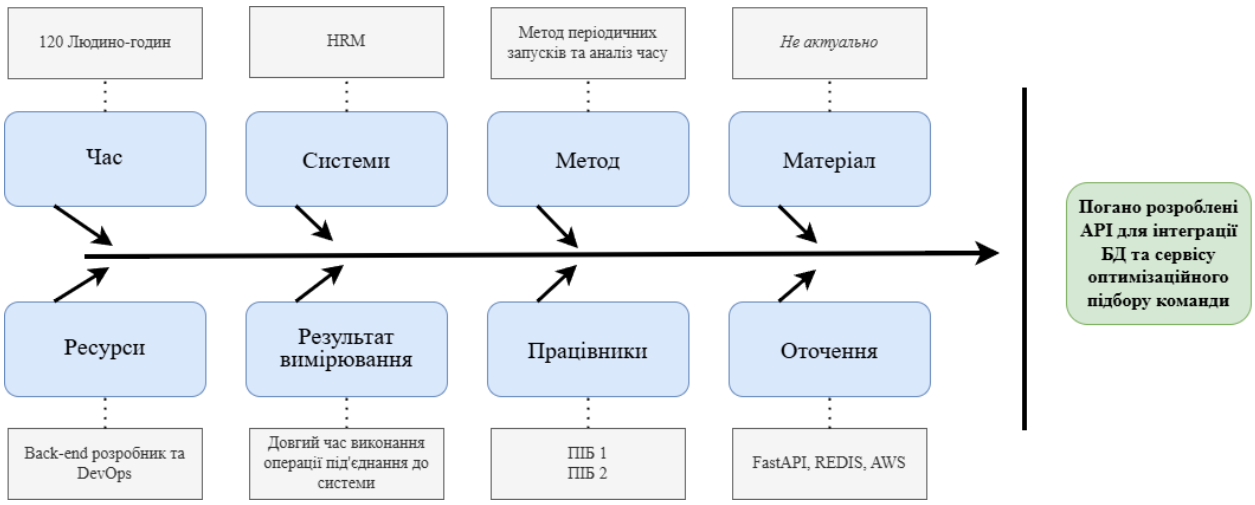


Рис 3.7. Діаграма причинно-наслідкових зв’язків

Контрольні діаграми потрібні для визначення стабільності проходження розробки та виконання цілей проєкту. Вони також можуть використовуватись для відображення життєвого циклу проєкту та продукту. До цих діаграм відносяться діаграми нижньої та верхньої межі (відхилення від норм проєкту), діаграми залежностей процесів, гістограми розподілення ресурсів, що закріплені за задачами та діаграма Парето (відображає частоту виникнення проблеми відносно факторів проблеми). Наприклад, процес оптимізаційного пошуку відображає не ті фільтри, що були використані проєктним менеджером під час вибору ролей. Такий приклад зображено на рис. 3.8.

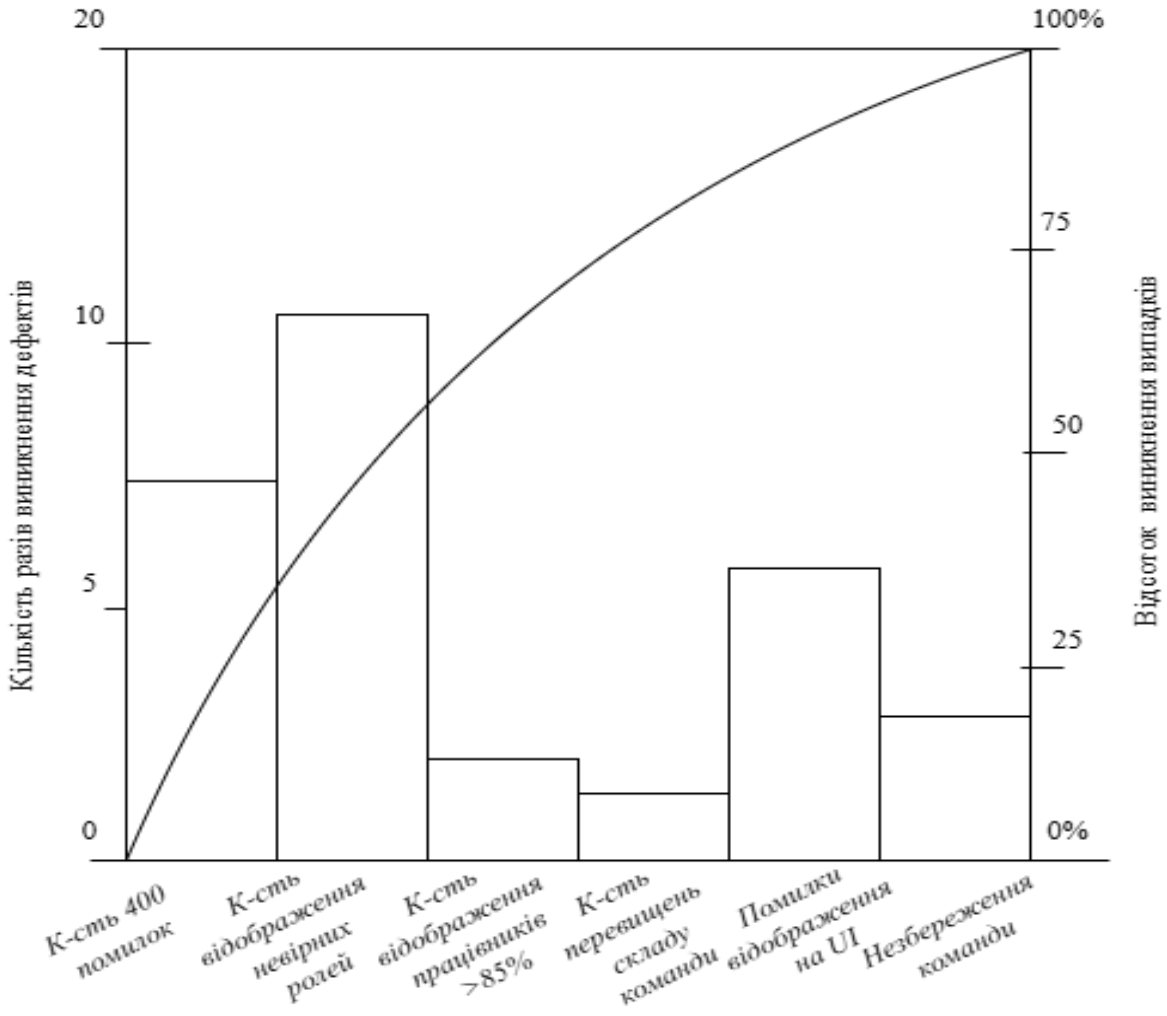


Рис. 3.8. Діаграма Парето

Окрім цього, можуть використовуватись схеми прогнозів, що відображають історію та модель змін (свого роду лінійний графік візуалізації зміни продукту). До моделі входять категоризації технічного виконання та відповідність витратами та часу проєкту.

Контроль якості є важливим постійним процесом будь-якого проєкту, а особливо технічних проєктів, які спрямовані на розвиток та використання моделей ШІ та машинного навчання. Постійний контроль призводить до оновлення планів, потреб та дотримання цілей проєкту.

РОЗДІЛ 4. РОЗРОБКА ОСНОВНИХ СКЛАДОВИХ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ

4.1 Опис алгоритмів роботи з аналізом оцінювання людських ресурсів та розподіл за проєктами

Усі системи працюють за певно визначеною послідовністю кроків, які складаються в алгоритм та процеси, якими користувачі виконують свої задачі. Навіть простий алгоритм «якщо – то» є послідовністю та правилом реагування на ті чи інші події. Наприклад, правило 85 % є обмеженням, яке не дає модулю підбору обирати працівників, навантаження яких є більшим або рівним 85 %.

Алгоритми по суті складають собою бізнес-правила та концепцію, за якою працюватимуть кінцеві користувачі. Однак для такого потрібно розуміти вхідні точки, події та вихідні точки. Окрім «best case» необхідно враховувати усі альтернативні шляхи, які виникають через помилки, відхилення або інші умови спричинені користувачем, або системою. Це важливо враховувати.

Основним алгоритмом HRM системи є пошук та формування проєктної команди на основі заданих фільтрів проєктним менеджером, який точно має знати ролі, склад, технічні знання (включно з досвідом роботи у сфері), кількість учасників проєктної команди та орієнтовну тривалість проєкту [56].

На рис. 4.1 зображено основний алгоритм процесу пошуку та формування проєктної команди. Він складається із взаємодії користувача та вебплатформи, що використовує API для зв’язку із модулем підбору працівників, який у свою чергу використовує SQL скрипти для створення запиту до БД. Як результат виконання проєктному менеджеру буде виведено зображення працівників відповідно до кожної ролі. Якщо працівника не було знайдено на, то іконка ролі буде підсвічена червоним, та виводитиме текст «Працівника не знайдено. Чи бажаєте знайти альтернативу?». Натисненням на це повідомлення почне інший процес підбору альтернативних працівників.

Іншим основним процесом є адміністрування інформації щодо працівників. Наприклад, front-end розробник став senior рівня та отримав нові

кваліфікації. На особистому профілі вебплатформи працівник має можливість додати нову інформацію про знання та здібності. Важливо, змінення рівня працівника та його ролі доступне лише HR та адміністраторам. В результаті зміни інформації, на веб-платформі у картці працівника збережеться оновлена інформація у нього, а у БД оновиться інформація. Такий процес зображено на рис. 4.2.

Вище наведені процеси є із залученням користувачів, однак алгоритми в першу чергу являють собою послідовність кроків для взаємодії між системами. Розглянемо приклад взаємодії БД та модулю підбору працівників. Необхідно виділяти усі побічні реагування системи на помилки. Алгоритм цього процесу описано на рис. 4.3.

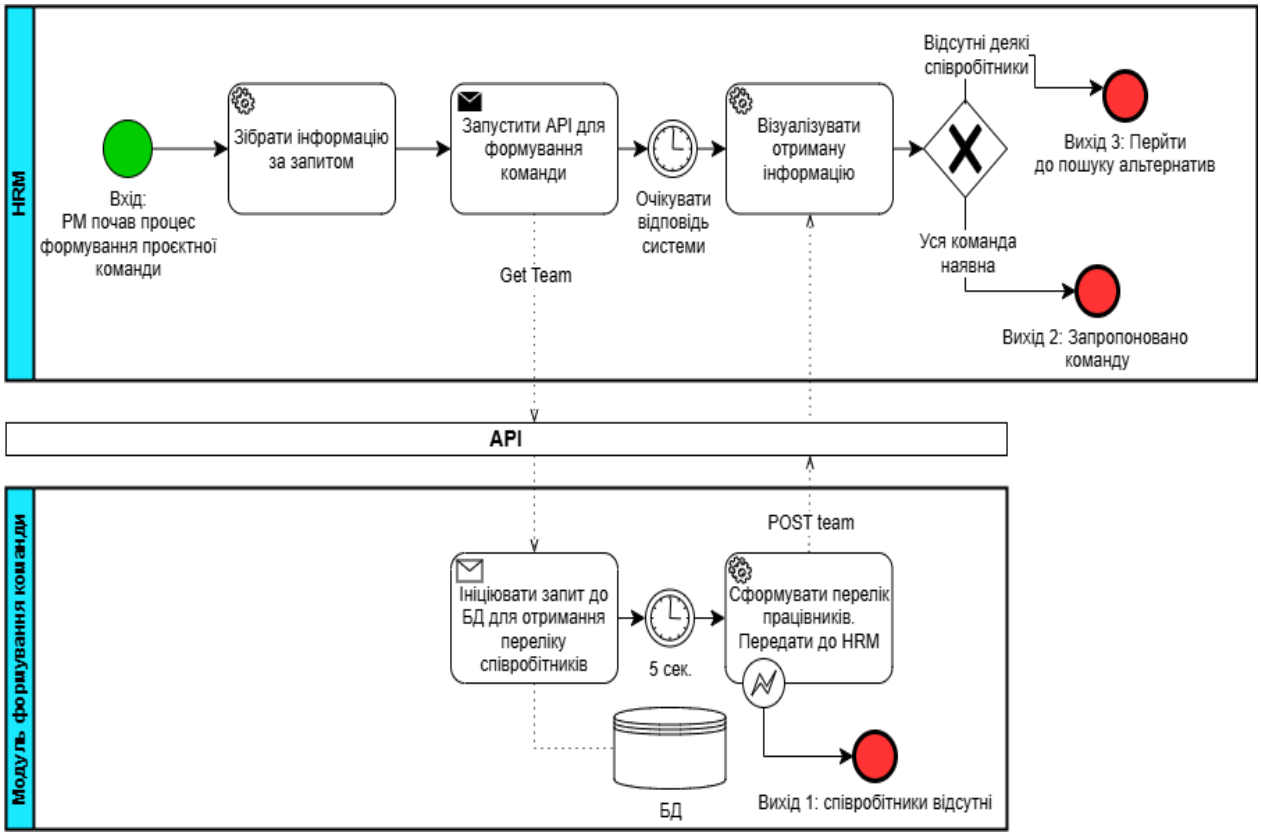


Рис. 4.1. Алгоритм підбору працівників для проєкту

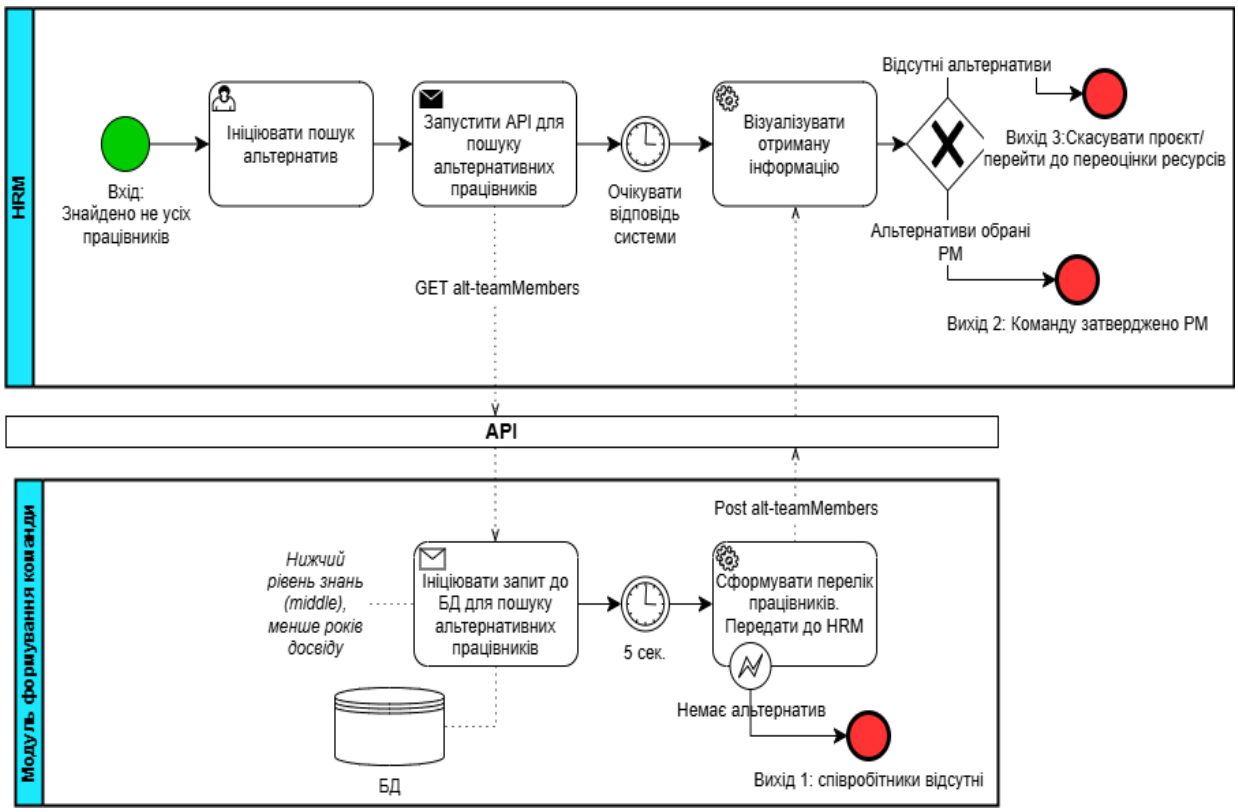


Рис. 4.2. Пошук альтернативних працівників

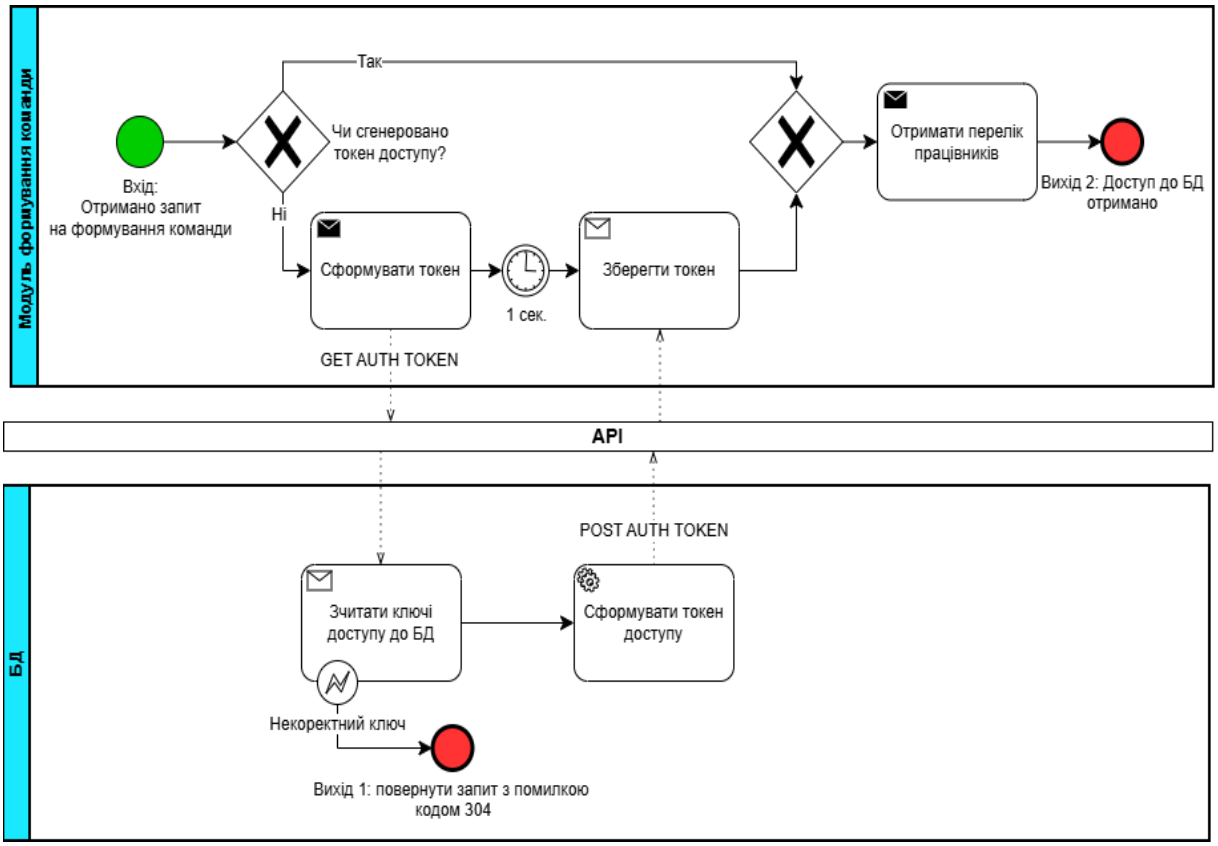


Рис. 4.3. Налаштування доступу до БД

Оскільки HRM система є великим об’єктом, що охоплює багато різних алгоритмів, решта ключових процесів наведена у Додатку Б цього документу. До цих алгоритмів належать: адміністрування інформації працівників (рис. Б.1) та формування аналітичної звітності за використанням ресурсів (рис. Б.2).

4.2 Концептуальне та даталогічне моделювання бази даних продукту проєкту

У розділі 2 даної роботи було спроєктовано концептуальну модель HRM системи, яка візуалізує взаємодію БД та сервісів системи. Даталогічна модель спрямована на організацію та управління даними БД, які використовуються для роботи системи та команди. Основним завдання є ідентифікувати зв’язки, розробити складові кожної таблиці та їх взаємодії у БД [56-60].

БД HRM системи складається з таких таблиць:

1. c_users – містить перелік та інформацію про усіх співробітників, що зареєстровані в системі.
2. ss_techskills – містить перелік усіх технічних навичок співробітників.
3. ss_softskills– містить перелік усіх якісних характеристик співробітників.
4. c_userskills – містить перелік технічних та якісних навичок та характеристик співробітників
5. sp_projects – містить перелік усіх проєктів та їх власників (проєктних менеджерів).
6. sp_projectteams – містить перелік працівників за проєктом.
7. c_activitylog – містить інформацію про взаємодію користувача з системою.
8. c_whitelistusers – містить перелік працівників, що не є фізично працевлаштованими до компанії, однак мають доступ до її атрибутів.

Основною сутністю є співробітники (users), які можуть належати до 0 або багатьох проєктів (projects). До цієї таблиці також входять сторонні користувачі (c_whitelistusers), що у таблиці мають спеціалізований тип

«external». Співробітники мають до 3 основних переважаючих технічних особливостей (ss_techskills) та до 3 якісних характеристик (ss_softskills), які враховує модуль підбору працівників під час формування команди. Проєкти (sp_projects) мають одну проєктну команду (sp_projects). До логу подій на платформі (c_activitylog) входять 0 або багато записів про дії користувачів. Ці дії зберігають у собі інформацію за місяць, та з такою самою інтенсивністю її видаляють.

Працівник (1) може належати до одного, або багатьох проєктів (сутність 1:Б). Така сутність відображена на рис. 4.4.

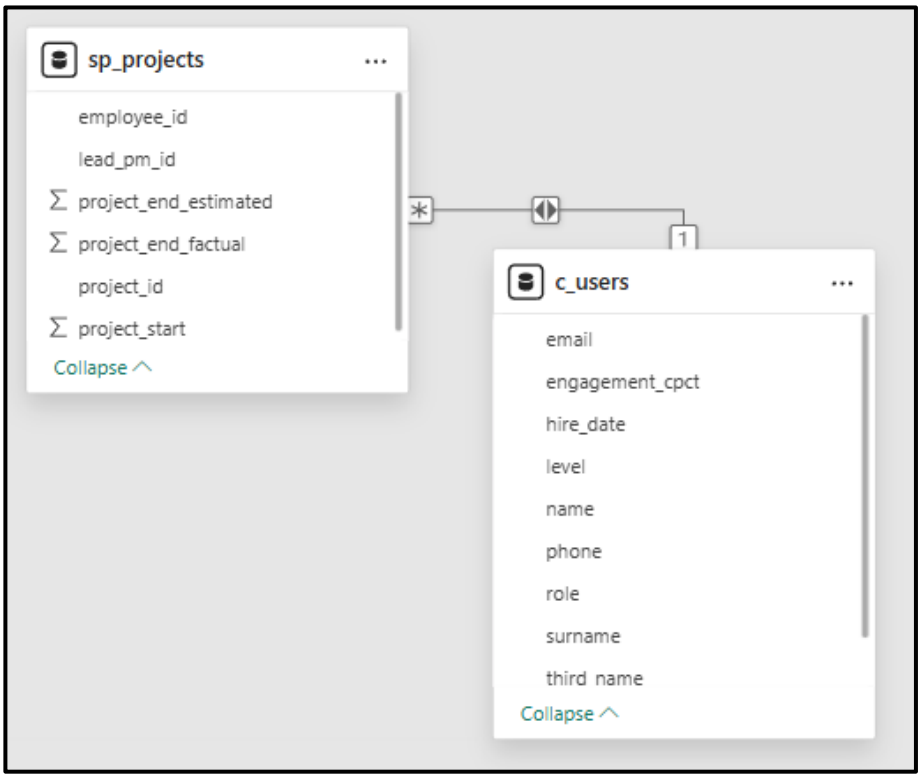


Рис 4.4. Сутність працівник до проєкту

До співробітників також можуть відноситись сторонні особи, які є, наприклад ФОПами, або сторонніми рекрутерами. Багато сторонніх осіб може належати до багатьох користувачів таблиці c_users (взаємозв’язок Б:Б), і візуалізовано на рис. 4.5.

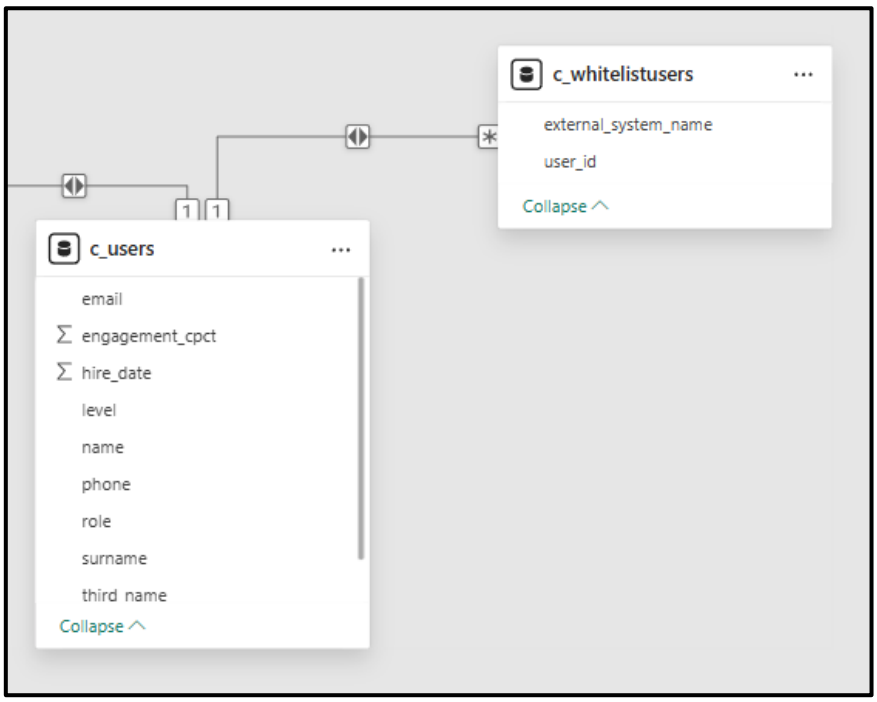


Рис. 4.5. Сутність сторонній працівник до працівника

Наступними взаємозв'язками є прив'язка технічних та якісних особливостей співробітників. Кожна з таблиць містить три основні навички користувача, відповідно має зв'язок 1:1. Такий взаємозв'язок зображено на рис. 4.6.

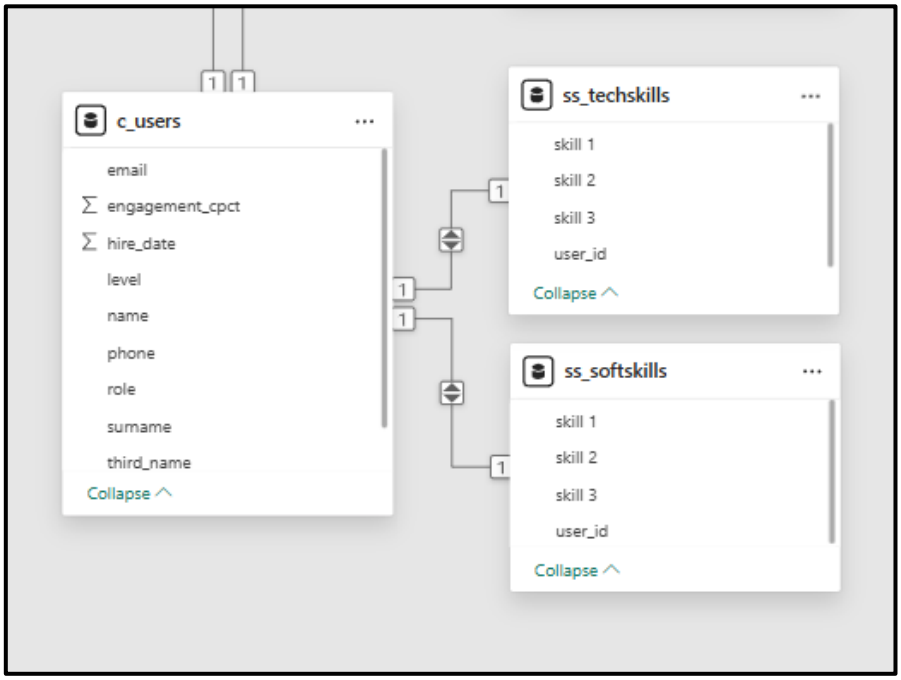


Рис. 4.6. Сутність таблиць навичок до працівника

Як було сказано раніше, платформа має вести логування дій користувачів. Таким чином, у таблиці c_whitelistusers може бути багато записів про одного користувача (відношення 1:Б). Така взаємодія відображена на рис. 4.7.

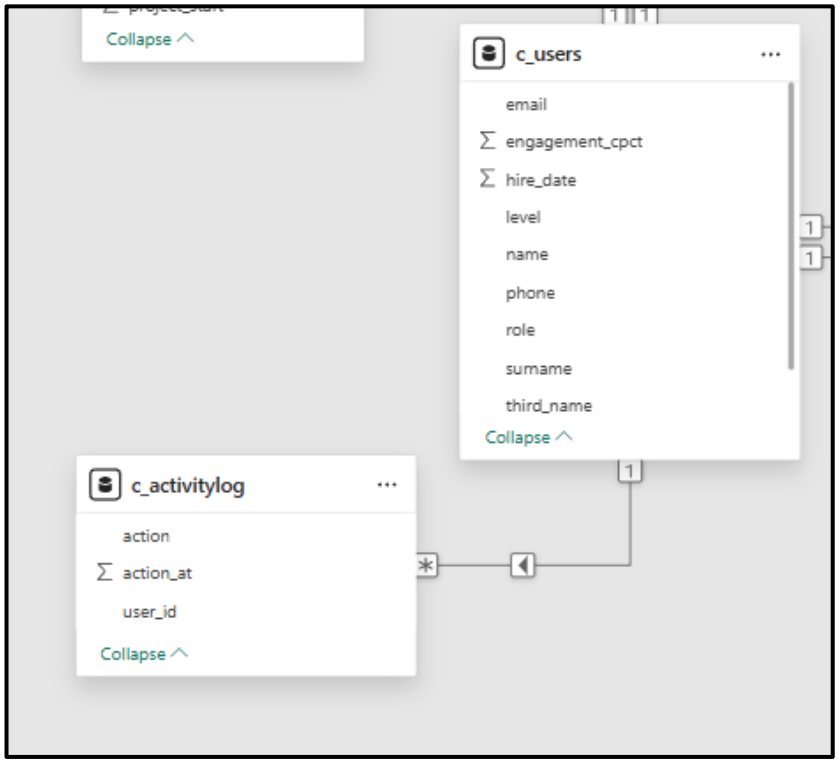


Рис. 4.7. Сутність таблиць дії до працівника

Остання сутність це взаємозалежність проєктної команди до проєкту та загального переліку працівників. Відповідно команди можуть відповідати різним проєктам (сутність Б:Б) та працівники команди можуть відповідати переліку працівників (Б:Б). Така взаємозалежність зображена на рис. 4.8.

Повна візуалізація взаємозв’язків таблиці відображена на рис. В.1 Додатку В.

Сформувавши взаємозв’язки таблиць, необхідно перейти до процесу формування даталогічної моделі – вона описує сутності, класи або об'єкти даних, що мають відношення до предметної області, атрибути, які використовуються для їх опису, та зв’язки між ними, щоб забезпечити загальний набір семантики для аналізу та реалізації.

Даталогічна модель є важливою для сучасних реляційних БД [61-62]. Вона дозволяє перевірити структуру даних на відповідність нормальним формам, усуває дублювання даних і захищає базу від аномальних залежностей між таблицями, оновлення чи видалення інформації.

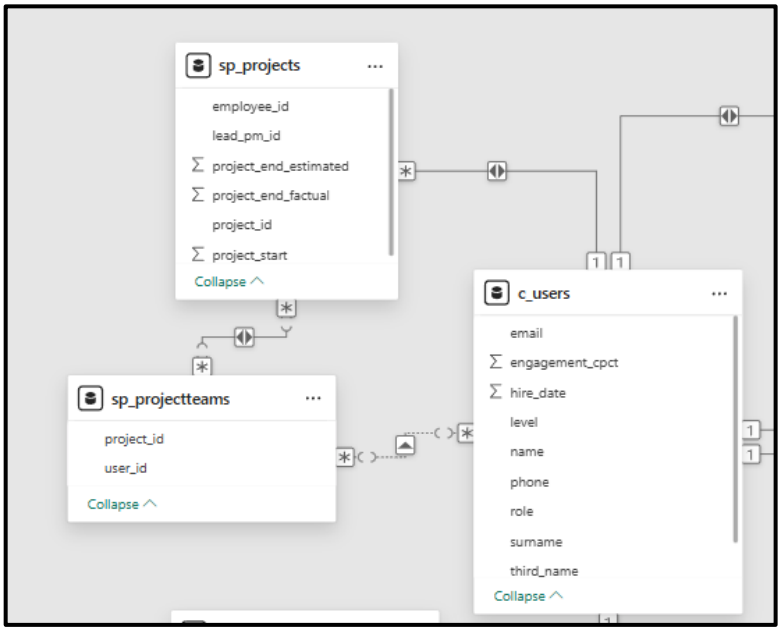


Рис. 4.8. Сутність таблиць проєктна команда до працівників та проєктів

Модель чітко визначає, які таблиці залежать одна від одної. Створена логічна схема є універсальною, що дозволяє розробникам девопс команді розгорнути базу даних на будь-якій реляційній платформі (PostgreSQL, MySQL, Oracle чи SQL Server) не нехтуючи архітектурою системи.

Відповідно, необхідно сформувати перелік атрибутів колонок сутностей БД для створення даталогічної моделі. Повний перелік наведено у табл. 4.1-4.8.

Таблиця 4.1

Перелік атрибутів таблиці «c_users»

Field	Type	Null	Key	Default
1	**2**	**3**	**4**	**5**
user_id	uuid	NO	PRI	NULL
name	varchar(50)	NO		NULL

Завершення табл. 4.1

surname	varchar(50)	NO		NULL
third_name	varchar(50)	YES		NULL
email	varchar(100)	NO	UNI	NULL
phone	varchar(20)	YES		NULL
role	varchar(100)	NO		NULL
level	varchar(50)	NO		NULL
engagement_cpct	decimal(3,2)	NO		NULL
hire_date	date	NO		NULL

Таблиця 4.2

Перелік атрибутів таблиці «sp_projects»

Field	Type	Null	Key	Default
1	**2**	**3**	**4**	**5**
project_id	uuid	NO	PRI	NULL
lead_pm_id	uuid	NO	MUL	NULL
project_start	timestamp	NO	-	NULL
project_end_factual	timestamp	YES	-	NULL
project_end_estimated	timestamp	NO	-	NULL
employee_id	uuid	NO	MUL	NULL

Таблиця 4.3

Перелік атрибутів таблиці «sp_projectteams»

Field	Type	Null	Key	Default
1	**2**	**3**	**4**	**5**
project_id	uuid	NO	PRI	NULL
user_id	uuid	NO	PRI	NULL

Таблиця 4.4

Перелік атрибутів таблиці «ss_techskills»

Field	Type	Null	Key	Default
1	**2**	**3**	**4**	**5**
user_id	uuid	NO	PRI	NULL
skill 1	varchar(50)	YES	-	NULL
skill 2	varchar(50)	YES	-	NULL
skill 3	varchar(50)	YES	-	NULL

Таблиця 4.5

Перелік атрибутів таблиці «ss_softskills»

Field	Type	Null	Key	Default
1	**2**	**3**	**4**	**5**
user_id	uuid	NO	PRI	NULL
skill 1	varchar(50)	YES	-	NULL
skill 2	varchar(50)	YES	-	NULL
skill 3	varchar(50)	YES	-	NULL

Таблиця 4.6

Перелік атрибутів таблиці «c_activitylog»

Field	Type	Null	Key	Default
1	**2**	**3**	**4**	**5**
user_id	uuid	NO	MUL	NULL
action_at	timestamp	NO	-	NULL
action	varchar(100)	NO	-	NULL

Таблиця 4.7

Перелік атрибутів таблиці «c_whitelistusers»

Field	Type	Null	Key	Default
1	**2**	**3**	**4**	**5**
user_id	uuid	NO	MUL	NULL
external_system_name	varchar(100)	NO	-	NULL

Сучасні методі та інструменти дозволяють використовувати 2 способи для формування, створення та налаштувань баз даних:

1. *UI-підхід*. Більшість сучасних СУБД дають можливість використовувати візуальний редактор для створення таблиці, їх подальшого налаштування, чи видалення, створення взаємозв’язків між таблицями, організувати та змінювати дані, модифікувати рядки тощо. Цей підхід є зручним та інтуїтивним для працівників, оскільки програма самостійно використовує SQL-скрипти не залучаючи до цього людину (яка, наприклад не має достатнього володіння та навичок написання SQL-скриптів для генерації табличок).

2. *SQL-скрипти*. Вимагають знань та функцій мови програмування, не мають широкого UI для зручності працівників, однак забезпечує більшу точність та результативність створених таблиці.

   Нижче наведено приклад SQL скрипту для створення таблиці c_users:

```
CREATE TABLE c_users (
    user_id UUID PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    surname VARCHAR(50) NOT NULL,
    third_name VARCHAR(50),
    email VARCHAR(100) NOT NULL UNIQUE,
    phone VARCHAR(20),
    role VARCHAR(100) NOT NULL,
    level VARCHAR(50) NOT NULL,
    engagement_cpct DECIMAL(3,2) NOT NULL,
    hire_date DATE NOT NULL
);
```

SQL скрипт складається із функції CREATE, яка створює новий запис (таблицю) у СУБД. У дужках вказуються назва колонки, тип атрибуту, чи є колонка основним зв’язком з іншими таблицями (PRIMARY KEY) та чи може бути значення колонки не нульовим.

Усі унікальні ідентифікатори в системі використовують тип UUID, тому що він має довжину у 32 символи та 4 дефіси, що збільшує кількість можливих записів і уніфікує підхід до ідентифікації об’єктів записів БД.

PRIMARY KEY вказує, що конкретна колонка (для прикладу вище з user_id) є унікальним ідентифікатором кожного рядка. Окремими важливим аспектом у СУБД є налаштування зав’язків таблиць між собою. Для цього використовується атрибут FOREIGN KEY, який визначає які колонки утворюють зовнішній зв’язок із таблицею. Для прикладу, таблиця sp_projectteam має такий зв’язок:

```
CREATE TABLE sp_projects (
    project_id VARCHAR(36) PRIMARY KEY,
    lead_pm_id VARCHAR(36) NOT NULL,
    project_start TIMESTAMP NOT NULL,
    project_end_factual TIMESTAMP,
    project_end_estimated TIMESTAMP NOT NULL,
    employee_id VARCHAR(36) NOT NULL,
    -- Визначення зовнішніх ключів
    CONSTRAINT fk_projects_pm FOREIGN KEY (lead_pm_id) REFERENCES c_users(user_id),
    CONSTRAINT fk_projects_emp FOREIGN KEY (employee_id) REFERENCES c_users(user_id)
);
```

Однак, для правильного використання FOREIGN KEY необхідно вказувати REFERENCES – зв'язок з іншою колонкою пов'язаної таблиці. Це потрібно для налаштування зв'язку та можливості створювати об'єднані запити до БД до кількох таблиць одночасно.

4.3 Розробка back-end частини вебплатформи

Реалізація backend частини застосунку є одним з найскладніших завдань, оскільки система у собі компонує елементи моделі ШІ, різні мікросервіси, API та СУБД – платформа має здати працювати асинхронно, виконуючи користувацькі та системні задачі. Найдоцільнішим стандартом для backend є використання мови Python, яка домінує у сфері AI розробок. Для реалізації backend пропонується використовувати два підходи одночасно:

1. FastAPI – це сучасний, швидкий (високопродуктивний) веб-фреймворк для створення API з Python на основі стандартних підказок типів Python [63]. Його основними перевагами є:

1) Дуже висока продуктивність на рівні з NodeJS та Go завдяки Starlette та Pydantic. І до того ж є одним з найшвидших доступних фреймворків для Python.
2) Збільшує швидкість розробки функцій приблизно на 200 %–300 %.
3) Зменшує приблизно на 40% помилки, спричинені розробником.
4) Менше часу на налагодження.
5) Створений для легкого використання та навчання. Менше часу на читання документації.
6) Мінімізація дублювання коду.
7) Відповідний стандартам. Заснований на відкритих стандартах для API та повністю сумісний з ними: OpenAPI (раніше відомий як Swagger) та JSON Schema.

2. Django – це високорівневий веб-фреймворк на Python, який сприяє швидкій розробці та чистому, прагматичному дизайну [64-66]. Розроблений досвідченими розробниками, він бере на себе багато труднощів веб-розробки, тому проєктна команда може більше зосередитися на написанні свого додатку. Він безкоштовний і open source. Django має великий набір інструментів: ORM для зручної обробки структурованих даних у PostgreSQL, готову панель адміністратора для управління моделями, а також вбудовані системи автентифікації та middleware.

Django використовує підхід мікросервісів, що ідеально підходить для реалізації HRM системи. Оскільки платформою планується обробка великих масивів даних, є вкрай доцільним використовувати такий підхід. Окрім цього, Django поєднується із FastAPI, обробляє вхідні дані та здійснює маршрутизацію API, тоді як модуль підбору працівників працює за межами основного навантаження на систему.

Варто також виокремити сенситивність інформації, що використовується – особисті даних працівників, що підпадають під Закони України щодо управління та зберігання особистих даних, та Європейський

закон GDPR. Django має потужні вбудовані механізми захисту від найпоширеніших загроз.

Реалізація backend та frontend частини відбувається у середовищі VS Code, яке дає можливість дуже просто встановити розширення Python, Django та FastAPI (рис. 4.9).

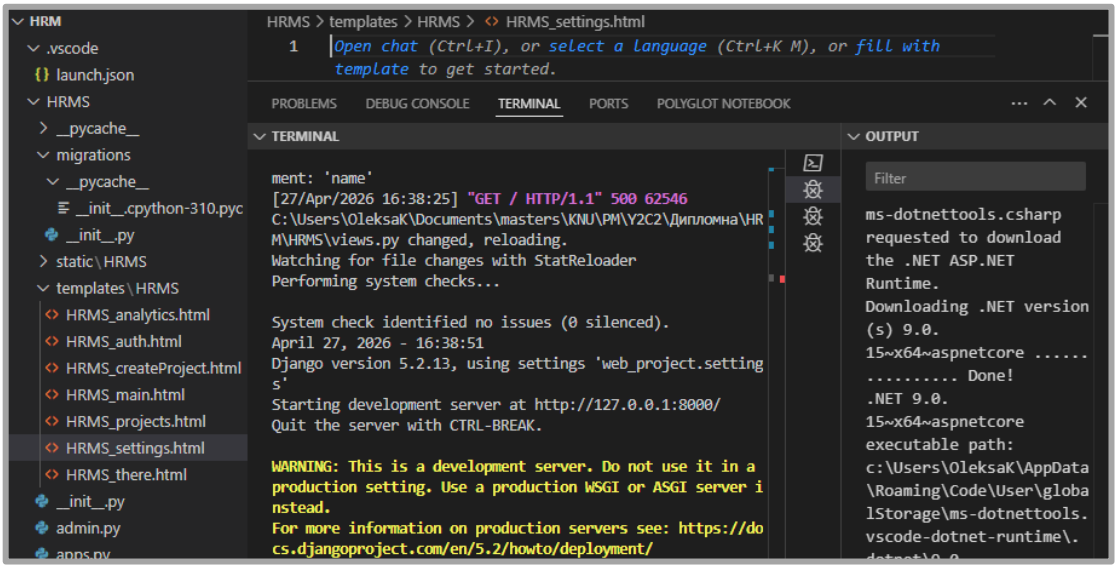


Рис. 4.9. Платформа Microsoft VS Code

Архітектура та фрагмент коду back-end частини зображена на рис. 4.10.

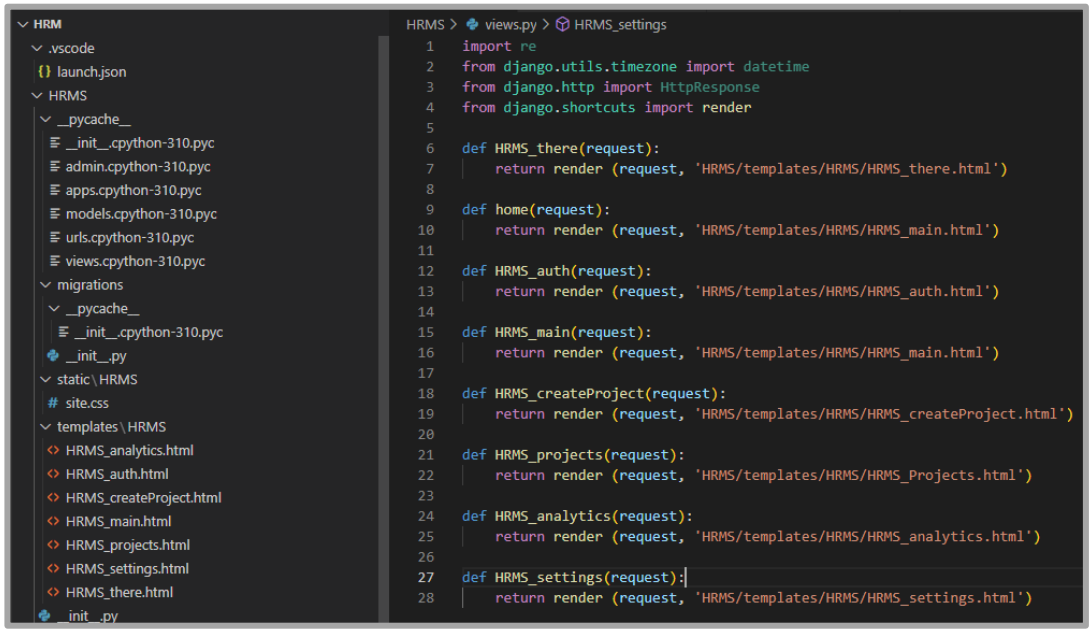


Рис. 4.10. Архітектура та фрагмент коду back-end HRM

Нижче наведено фрагмент коду функції виклику API GET_USERS, яка збирає перелік працівників на основі заданих проєктним менеджером фільтрів:

```
from django.db import models
class ProjectTeamMember(models.Model):
project_id = models.IntegerField() # Прив'язка до конкретного проєкту
full_name = models.CharField(max_length=255, null=True, blank=True)
role = models.CharField(max_length=100) # Наприклад: 'Senior Product Owner', 'Data Analyst'
years_of_experience = models.IntegerField(null=True, blank=True) # Досвід (yow)
capacity_percent = models.IntegerField(null=True, blank=True) # Завантаженість (cpst)
is_found = models.BooleanField(default=True) # True - працівника знайдено, False - вакансія відкрита
def __str__(self):
return f"{self.role} - {self.full_name if self.is_found else 'Not Found'}"
from rest_framework.decorators import api_view
from rest_framework.response import Response
from rest_framework import status
from.models import ProjectTeamMember
from.serializers import TeamMemberSerializer
@api_view()
def get_assembled_team(request, project_id):
"""
API-ендпоінт (Get_USERS) для отримання складу згенерованої команди.
Віддає список як знайдених фахівців, так і незакритих позицій,
які відображатимуться помаранчевими картками на фронтенді.
"""
try:
# У реальній системі тут може викликатися ваш алгоритм SSR-TF для
# генерації команди, якщо її ще не створено, із перевіркою правила 85 %
# Отримуємо всіх учасників для запитаного проєкту з БД
team_members = ProjectTeamMember.objects.filter(project_id=project_id)
# Якщо команду ще не сформовано
if not team_members.exists():
return Response(
{"message": "Для цього проєкту ще не згенеровано команду."},
status=status.HTTP_404_NOT_FOUND)
# Серіалізуємо дані у JSON
serializer = TeamMemberSerializer(team_members, many=True)
return Response({
"status": "success",
"project_id": project_id,
```

```
"team_data": serializer.data
}, status=status.HTTP_200_OK)
except Exception as e:
return Response(
{"error": str(e)},
status=status.HTTP_500_INTERNAL_SERVER_ERROR
)
```

Django та FastAPI є чудовим рішенням для розробки програмного продукту, оскільки розподілення функціоналу за мікросервісами спрощує оновлення, налаштування та покращення, і не вимагає перевіряти та модифікувати інші функції платформи.

4.4 Розробка front-end частини вебплатформи

Front-end відповідає за зручність та добробут роботи кінцевих користувачів системи, забезпечуючи інтуїтивно зрозумілий інтерфейс доступний для різних девайсів під малі та великі розміри екранів. UI є одним з найпершим критерієм прийняття та взаємодії із системою, що пришвидшує, або сповільнює, процес погодження та адаптації до застосунку користувачів. Чим зручніша платформа – тим більше зацікавлені працівники.

Спираючись на обраний гібридний підхід поєднання Django та Python, необхідно підібрати методи та інструменти, що будуть взаємодіяти найкраще з ними при цьому зберігати кінцеву якість та можливість подальшого розширення, налаштувань і розробок.

Для реалізації front-end частини було обрано React.js [67] для створення платформи типу SPA [68].

React має найкращу підтримку бібліотек Recharts та Nive, які потрібні для роботи зі складними графіками та діаграмами, і подальшого формування excel-звітів. Інтерфейс інструменту будується окремими модулями / блоками – розробнику не потрібно змінювати загальний код, а лише маленьку частину, тим самим не збільшуючи ризик втручання в інші елементи платформи. React працює з WebSockets, що тільки грає на руку із інтеграцією із Django. Оскільки

інструмент створений для роботи з GraphQL та RestAPI, то робота з FastAPI не матиме жодних проблем. React використовує такі мови програмування:

1. JSX/HTML – виконує роль наповнення інформацією сторінок вебплатформи. Використовує теги, описи та інші елементи для створення об’єктів на сторінці. Однак HTML є лише основою, а самим фреймворком є JSX - спеціальне синтаксичне розширення, яке дозволяє писати HTML-подібну розмітку всередині JavaScript-коду для створення динамічних компонентів. JSX є розширенням розробленими спеціально для полегшення роботи ReactJs.
2. Tailwind CSS – це сучасний CSS-фреймворк, який дозволяє швидко створювати та стилізувати користувацькі інтерфейси безпосередньо у HTML-розмітці або JSX-коді компонентів. Tailwind надає низькорівневі допоміжні класи з яких розробник самостійно та гнучко «збирає» унікальний дизайн. Фреймворк містить вбудовані інструменти для створення адаптивного дизайну і дозволяє уникнути написання довгих, заплутаних і важких користувацьких CSS-файлів
3. JavaScript – базова мова програмування, на якій побудовано бібліотеки React. JS по суті є функціоналом front-end, як наприклад кнопки, початок використання функцій, інтерактивні взаємодії з вебсайтом тощо.

На рис. 4.11-4.15 зображено мокапи скріншотів застосунку. Для створення використовувався ресурс penpot.app [69].

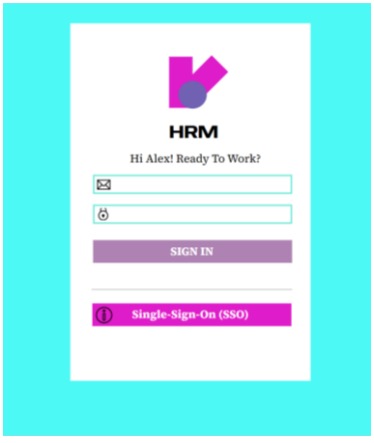


Рис. 4.11. Фрагмент сторінки аутентифікації до системи

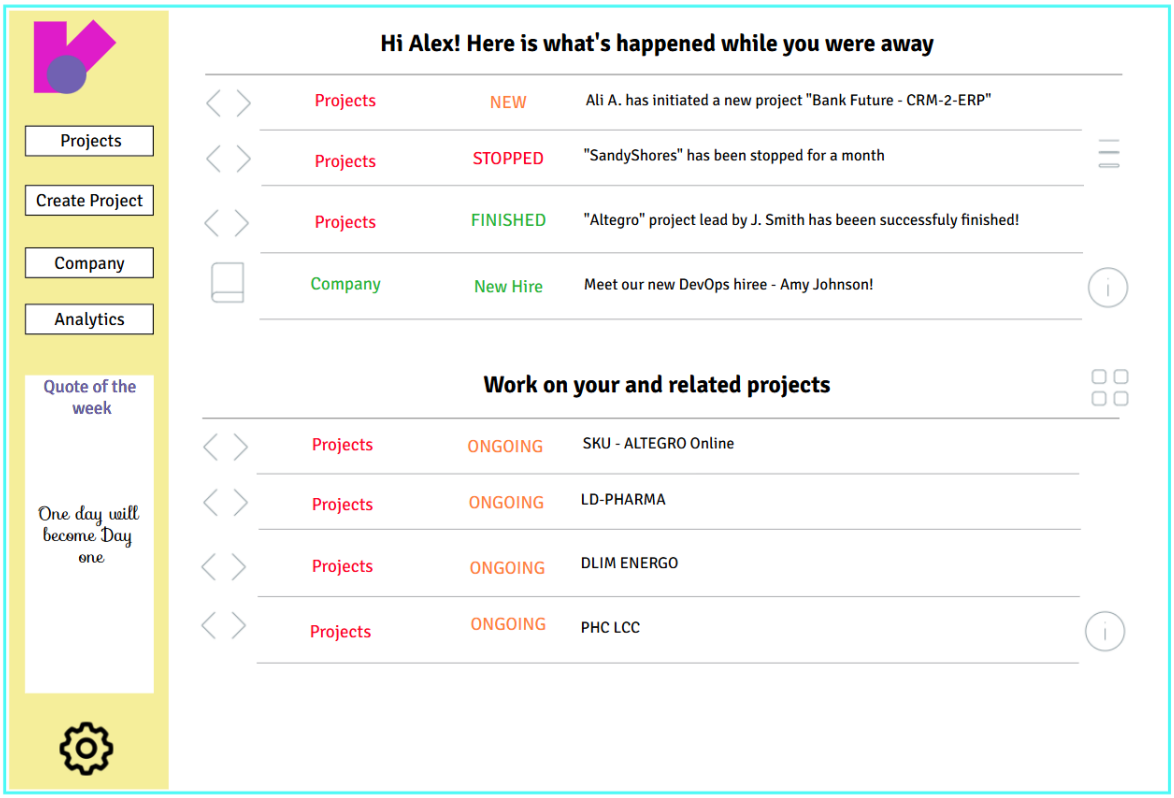


Рис. 4.12. Головна сторінка системи

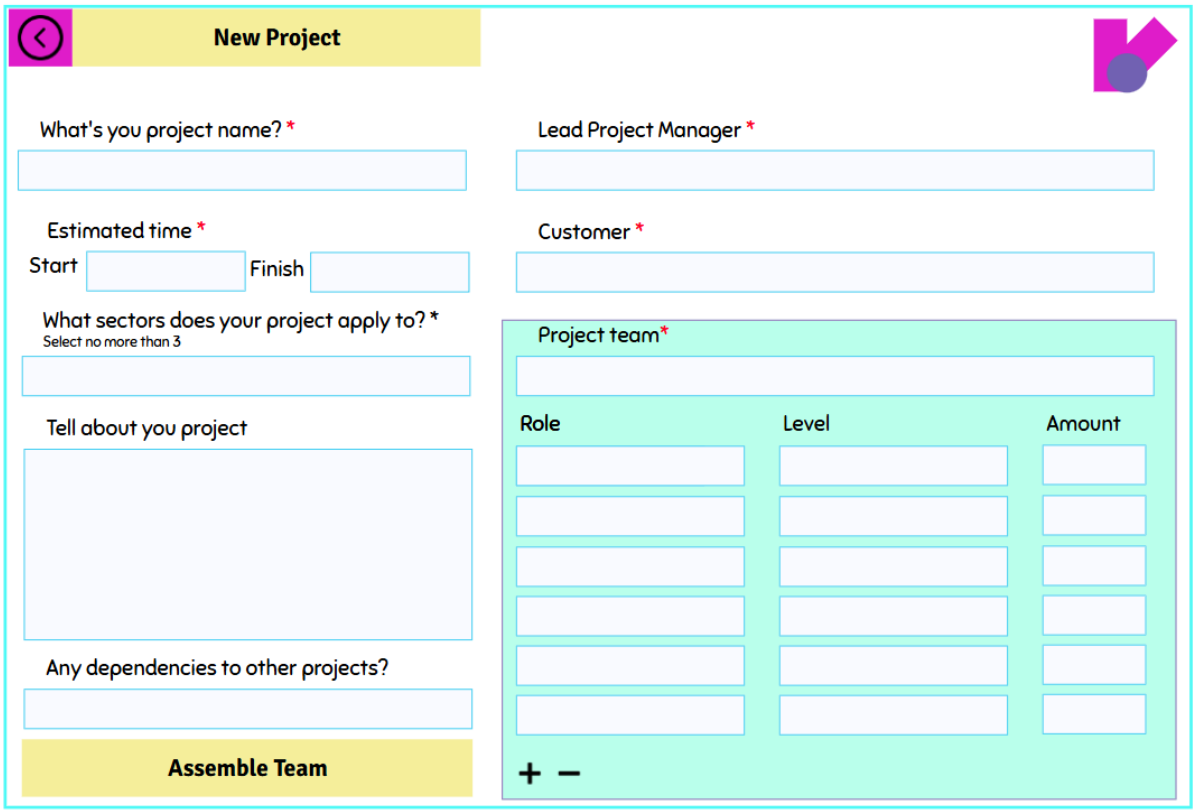


Рис. 4.13. Сторінка створення проєкту та вибору ролей команди

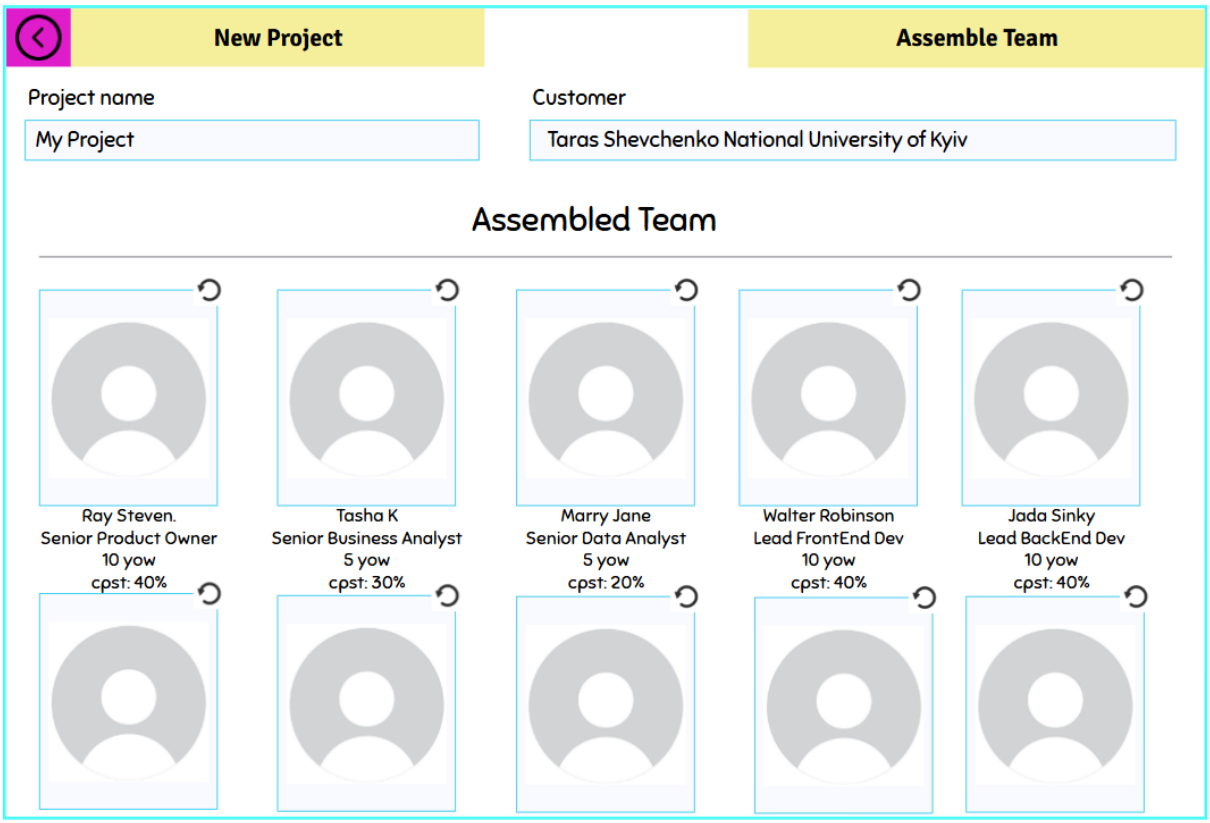


Рис. 4.14. Сторінка підібраної команди

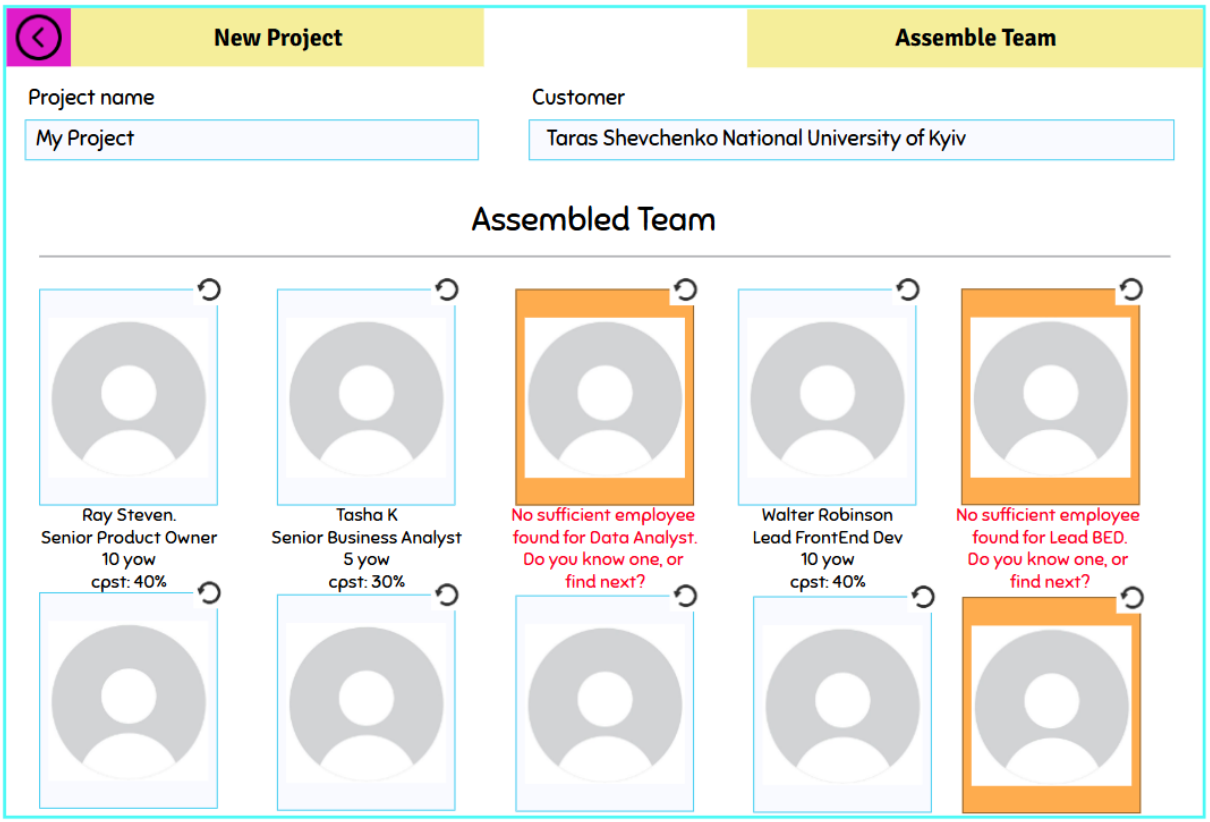


Рис. 4.15. Відсутні працівники відповідних проєктних ролей

Нижче наведено фрагмент CSS коду, що описує дизайн сторінки на рис. 4.15.

```
<!DOCTYPE html>
<html lang="uk">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>HRM Team Builder UI</title>
  <script src="https://cdn.tailwindcss.com"></script>
  </head>
   <body class="bg-gray-50 p-8 flex justify-center items-start min-h-screen font-sans">
<div class="w-full max-w-5xl bg-white border border-cyan-400 p-4 shadow-sm">
<div class="flex items-center gap-4 mb-6">
<button class="w-10 h-10 shrink-0 bg-fuchsia-500 rounded-full flex items-center justify-center border-2 border-black hover:bg-fuchsia-600 transition">
<svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6 text-black" fill="none" viewBox="0 0 24 24" stroke="currentColor" stroke-width="2">
path stroke-linecap="round" stroke-linejoin="round" d="M15 19l-7-7 7-7" />
</svg>
</button>
<div class="flex-1 flex gap-2">
<div class="flex-1 bg-yellow-200 text-center py-2 font-semibold border border-transparent">
New Project
</div>
<div class="flex-1 bg-yellow-200 text-center py-2 font-semibold border border-transparent">
Assemble Team
</div>
</div>
<div class="grid grid-cols-2 gap-8 mb-8">
<div>
<label class="block text-sm font-medium text-gray-800 mb-1">Project name</label>
<input type="text" value="My Project" class="w-full border border-cyan-400 px-3 py-1.5 focus:outline-none focus:ring-1 focus:ring-cyan-500">
</div>
<div>
```

```
            <label class="block text-sm font-medium text-gray-800 mb-1">Customer</label>
            <input type="text" value="Taras Shevchenko National University of Kyiv" class="w-full border border-cyan-400 px-3 py-1.5 focus:outline-none focus:ring-1 focus:ring-cyan-500">
            </div>
```

Інтуїтивно зрозумілий дизайн є основою та показником успішності продукту в обличчях зацікавлених сторін та кінцевих користувачів. Хороший симбіоз кольорів та відсутність інформативного перенавантаження сторінки спрощує та впорядковує роботу користувачів платформи. Дякуючи обраним методам розробки front-end частини, розробники та команда підтримки мають можливість розширення та оновлення завдяки модульному та сервісному підходах реалізації продукту.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи магістра було здійснено комплексне дослідження процесів управління проєктом розробки інтелектуальної інформаційної системи управління людськими ресурсами (HRM) для IT-команд. Виконані завдання дозволили сформувати цілісну методологію, що поєднує інноваційні математичні моделі з передовими практиками управління IT-проєктами.

Протягом виконання було досягнуто поставлених цілей та виконано наступні завдання:

1. Досліджено методології до підбору та визначення оптимальної команди для виконання IT-проєктів, визначено слабкі та сильні сторони, а також сформовано предметну область.
2. Проаналізовано літературні та інформаційні джерела щодо можливостей вирішення виявлених проблем, а також здійснено SWOT-аналіз проєкту та ідентифіковано зацікавлені сторони.
3. Сформульовано технічне завдання у вигляді паспорта проєкту, розроблено концептуальну модель інформаційної системи та визначено математичну постановку задачі.
4. Обґрунтовано вибір методології управління проєктом, визначено організаційну структуру управління, сформовано беклог продукту та сплановано спринти з використанням діаграми Ґанта.
5. Ідентифіковано ризики та забезпечено підходи до їх вирішення (карта управління ризиками).
6. Створено алгоритм роботи вебплатформи, розроблено концептуальну та даталогічну модель, запропоновано підхід до реалізації бази даних.
7. Реалізовано концепти вебплатформи.

Окрім цього проведена робота дає розуміння потенційному розширенню вебплатформи та її розвитку, зокрема можливість інтеграції із зовнішніми сервісами або її монетизація у вигляді B2B застосунку, що може стати

глобальною базою працівників для, наприклад, фрілансу. Також варто не забувати про можливі оновлення та підтримку моделей машинного навчання, які створюють та підбирають команди із кваліфікованих працівників для проєктних менеджерів. У світі да панують технології є вкрай важливим бути швидким до адаптації та вміти приймати зміни майже миттєво.

Практичне впровадження розробленої інтелектуальної платформи надасть проєктним менеджерам та HR потужний інструмент підтримки прийняття рішень, що дозволить скоротити час формування проєктних команд на 40 %, знизити рівень плинності кадрів через емоційне виснаження та перетворити етичне ставлення до людського капіталу на головну стратегічну конкурентну перевагу ІТ-компанії.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. MuchSkills, 2026. URL: https://www.muchskills.com/team-builder
2. Epicflow, 2025. URL: https://www.epicflow.com/
3. Screendragon, 2026. URL: https://www.screendragon.com/
4. Scoro, 2026. URL: https://www.scoro.com/
5. Wrike, 2026. URL: https://www.wrike.com/
6. Productive, 2026. URL: https://productive.io/
7. Kantata, 2026. URL: https://www.kantata.com/
8. ClickUp, 2026. URL: https://clickup.com/
9. ProjectManager, 2024. URL: https://www.projectmanager.com/
10. Taskfino, 2026. URL: https://www.projectmanager.com/
11. ActivTrak, 2025. URL: https://www.projectmanager.com/
12. Ramy H.M.M. AI-Enabled Human Resource Practices and Quality of Work Life: Evidence from the Saudi Telecommunications Sector. Journal of Organizational Behavior Research, 2025, 10(4), P. 14–27. https://doi.org/10.51847/J4pmwUH6Jf
13. Stavrou G., Adamidis P., Papathanasiou J. Multi Criteria Evolutionary Algorithm for Research Team Formation. XIV Balkan Conference on Operational Research (BALCOR 2020), 2020, P. 126–135.
14. Gillani S.M.Z.R., Zamli K.Z., Almutairi M., Nawi N.M.A novel state space reduction algorithm for team formation in social networks. PLoS ONE, 2021, 16(12), e0259786. https://doi.org/10.1371/journal.pone.0259786
15. Hernandez A.A., Albina E.M., Perez R.P. Development of Occupational Burnout Prediction Models Using Machine Learning Techniques and Maslach Burnout Inventory. 2024 IEEE 15th Control and System Graduate Research Colloquium (ICSGRC), 2024, P. 47–51. https://doi.org/10.1109/ICSGRC62081.2024.10690950
16. Kingman J.F.C. The single server queue in heavy traffic. Mathematical Proceedings of the Cambridge Philosophical Society, 1961, 57(4), P. 902–909.

17. Karimi R., Baghalzadeh Shishehgarkhaneh M., Moehler R.C., Fang Y. The Evolution of Work-Life Balance: Redefining Priorities in Human Resource Management. Dinasti International Journal of Economics, Finance & Accounting, 5(1), 2024, P. 23–34.

18. Koç H., Hynes J., Acar A.B. Organizational strategies for employee well-being: Balancing work-life demands in the tech industry. Joint Proceedings of the BIR 2025 Workshops and Doctoral Consortium, 2025, P. 170–183.

19. Tyagi S., Tomar A., Mohit M. Prediction of Mental Burnout Using Machine Learning. International Journal of Scientific Research in Engineering and Management, 2025, 9(1). https://doi.org/10.55041/ijsrem53497

20. Van Zyl-Cillié M., Bührmann J., Blignaut A., Demirtas D., Coetzee S. A machine learning model to predict the risk factors causing feelings of burnout and emotional exhaustion amongst nursing staff in South Africa. BMC Health Services Research, 2024, 24. https://doi.org/10.1186/s12913-024-12184-5

21. Zhernova P., Bodyanskiy Y., Yatsenko B., Zavgorodnii I. Detection and Prevention of Professional Burnout Using Machine Learning Methods. 2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 2020, 218–221. https://doi.org/10.1109/tcset49122.2020.235426

22. Munir Y., Umer Q., Faheem M., Akram S., Jaffar A. Developer Recommendation and Team Formation in Collaborative Crowdsourcing Platforms. IEEE Access, 2025, 13, P. 63170–63185. https://doi.org/10.1109/ACCESS.2025.3558557.

23. Singh P., Huynh P.K., Nguyen D., Le T.Q., Moreno W. Leveraging Multi-Criteria Integer Programming Optimization for Effective Team Formation. TechRxiv, 2023. https://doi.org/10.36227/techrxiv.24547984.

24. Lopez-Herrejon R.E. Team Formation in Software Engineering: A Data-Centric Methodology Based on Personality Traits. Electronics, 2024, 13(1), 178. https://doi.org/10.3390/electronics13010178

25. Ziuziun V., Bredikhin D. Conceptual and Mathematical Modeling in Managing a Project for Developing a Web Platform to Enhance Environmental Awareness 2025 IEEE 5th International Conference on Smart Information Systems and Technologies (SIST), Astana, Kazakhstan, 2025, P. 1-6. https://doi.org/10.1109/SIST61657.2025.11139157

26. Kovalchuk N., Zachko O., Kovalchuk O., Kobylkin D. Project Management of the Information System for the Selection of Project Teams. 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany, 2023, P. 1054-1057, https://doi.org/10.1109/IDAACS58523.2023.10348656

27. Ziuziun V., Danilina T. Conceptual and Mathematical Justification for the «PETHEALTH» Information System Development Project, Taurida Scientific Herald. Series: Technical Sciences, 2025, 4, part 1, P. 94-102. https://doi.org/10.32782/tnv-tech.2025.4.1.10

28. Kolomiiets A., Miroshnychenko I. et al. Development of Project Management Models for Information Systems to Improve Website SEO Metrics. XI International Scientific Conference «Information Technology and Implementation» (IT&I 2024). CEUR Workshop Proceedings, 2024, Vol-3909, P. 334-345. https://ceur-ws.org/Vol-3909/Paper_27.pdf

29. Ziuziun V., Koziuk Yu. Conceptual Modeling of an Information System for Professional Development and Promotion of Educational Services in the Field of Digital Design, Scientific journal Transactions of Kremenchuk Mykhailo Ostrohradskyi National University, 2025, 4(153), P. 156-163. https://doi.org/10.32782/1995-0519.2025.4.19

30. Ziuziun V., Korytnyi O. Intelligent Model for Ranking IT Specialists Based on Competencies, Team Roles, and Resource Workload Indicators. Proceedings of the 4th International Scientific and Practical Conference «Modern Scientific Research: Theoretical and Practical Aspects», Riga, Latvia, 11-13 May 2026, P. 246-257. https://doi.org/10.70286/EOSS-11.05.2026.009.246-257

31. Кон Майкл. Оцінювання і планування в Agile / пер. з англ. Г. Якубовська. Харків : Вид-во «Ранок» : Фабула, 2019. С. 336.

32. Sutherland J. Scrum : The Art of Doing Twice the Work in Half the Time. New York : Random House, 2014. 256 p.

33. Scrumban: Mastering two Agile methodologies, Atlassian, 2026. URL: https://www.atlassian.com/agile/project-management/scrumban

34. What is Scrumban?, Geek for Geeks, 2025. URL: https://www.geeksforgeeks. org/software-engineering/what-is-scrumban/

35. What is Scrumban?, Corey Ladas, 2021. URL: https://agilealliance.org/ scrumban/

36. Scrumban: The best of two Agile methodologies, Sarah Laoyan, 2026. URL: https://asana.com/resources/scrumban

37. BABOK Guide, International Institute of Business Analysis (IIBA), 2026. URL: https://www.iiba.org/career-resources/a-business-analysis-professionals-foundation-for-success/babok/

38. Ziuziun V., Timinskiy O., Kolomiiets A. et al. Research of Management Models for Commercial IT Project Development in a Remote Team Environment. Management of Development of Complex Systems, 2025, 61, pp. 26-34. https://doi.org/10.32347/2412-9933.2025.61.26-34

39. Зачко О. Б., Івануса А.І., Кобилкін Д.С. Управління проектами: теорія, практика, інформаційні технології. Львів: ЛДУ БЖД, 2019,173 с.

40. Morozov V., Kulyk R. Building integration models for the efficiency of managing complex fixed-budget IT projects. Management of Development of Complex Systems, 2025, 62, 97–106, dx.doi.org\10.32347/2412-9933.2025.62.97-106.

41. ProjectLibre. Transform Your Projects with Artificial Intelligence, 2026. URL: https://www.projectlibre.com

42. Ziuziun V., Kolomiiets A., Aspects of Decision-Making in the Management of Human Resources in IT Projects of Organizations, Conference: International Scientific and Practical Conference «Intelligent Information Systems

in Project and Program Management», Koblevo, September 12–15, 2023. Proceedings. - Kharkiv: O. M. Beketov National University of Urban Economy in Kharkiv, 2023. P. 45-49.

43. Reshetnyak O.I., Danko N.I., Yurchenko O.K. Risk management of IT projects: The essence and features of the application of the PMBoK and AGILE approaches. Business Inform, 2023, 10, P. 366–374. https://doi.org/10.32983/2222-4459-2023-10-366-374

44. Tanim S.H., Ahmad M.S. AI-driven strategic decision-making in IT project management: Enhancing risk assessment, cost control, and efficiency. SSRN Electronic Journal, 2025. https://doi.org/10.2139/ssrn.5202585

45. Pirogov P.I., Saitbagina L.A. Risk management in Agile IT projects involving Generation Z. Economics and Management, 2025, 31(12), P. 1578–1588. https://doi.org/10.35854/1998-1627-2025-12-1578-1588

46. Korytnyi O., Ziuziun V. Digital Ways of Controlling Risk Management and Assessment in Projects. Proceedings of the 2st International scientific and practical conference «Information Systems and Technology: Results and Prospects» (IST 2025), Kyiv, Ukraine, 2025, P. 178-181. URL: https://ist.fit.knu.ua/_files/ugd/016074_36d0f427916c46abb6491a7572bb63ec.pdf

47. Ziuziun V., Korytnyi O., Risk Controling and Monitoring Softwares that Consider Human Bias as a Major Risk. Information Technology and Implementation (Satellite): Conference Proceedings, November 21, 2025, Kyiv, Ukraine / Ministry of Education and Science of Ukraine, Taras Shevchenko National University of Kyiv and [etc]; Vitaliy Snytyuk (Editor). – Kyiv: Publishing House «Caravela», 2025, P. 326-327. URL: https://www.researchgate.net/publication/399285290_Risk_Controling_and_Monitoring_Softwares_that_Consider_Human_Bias_as_a_Major_Risk

48. Testorelli R., Tiso A., Verbano, C. Value creation with project risk management: A holistic framework. Sustainability, 2024, 16(2), P. 753. https://doi.org/10.3390/su16020753

49. Katende N., Kibe A., Kubwimana D. Implementing risk mitigation, monitoring, and management in IT projects. International Journal of Scientific & Technology Research, 2019, 8(2), P. 15-22.

50. Tavares B.G., de Souza A.D. Risk management in IT projects in the framework of Agile development. Proceedings of the 25th International Conference on Enterprise Information Systems, 2023, 1, P. 654-661.

51. Korytnyi O., Ziuziun V. Justification of the need to create an information system for risk management of IT projects. Proceedings of the Information Technology and Implementation (Satellite): Conference Proceedings, November 21, 2024, Kyiv, Ukraine / Taras Shevchenko National University of Kyiv and [etc]; Vitaliy Snytyuk (Editor). Kyiv: Publishing House «Caravela», P. 236-237. URL: https://www.researchgate.net/publication/387682561_Justification_of_the_Need_to_Create_an_Information_System_for_Risk_Management_of_IT_Projects

52. Morozov V., Striletskyi Ye., Stryzhak S. Study of risk management models in IT projects held by the distributed teams working asynchronously. The proceedings of IEEE 5th International Conference on Smart Information Systems and Technologies (SIST) 2025. https://doi.org/10.1109/SIST61657.2025.11139236

53. Ziuziun V. Analysis of Possible Risks in Human Resources Management in IT Companies, Conference: 5th International Scientific and Practical Internet Conference «Integration of Education, Science and Business in Modern Environment: Summer Debates», August 3-4, 2023. P.79-81.

54. Тімінський О.Г., Коломієць А.С. Управління ризиками та можливостями проєкту [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. – К. : КНУ імені Тараса Шевченка, 2021, 40 с.

55. Project Management Institute (PMI). A Guide to the Project Management Body of Knowledge (PMBOK® Guide). Sixth Edition, 2017.

56. Ziuziun V., Korytnyi O. Information Support and Architecture of an Intelligent Platform for Human Capital Management in IT Projects. Proceedings of

the 8th International Scientific and Practical Conference «Modern Perspectives on Global Scientific Solutions», Bergen, Norway, 18-20 May 2026, P. 248-259. https://doi.org/10.70286/EOSS-18.05.2026.005.248-259

57. Torres-Jimenez J., Rangel-Valdez N., De-la-Torre M., Avila-George H. An Approach to Aid Decision-Making by Solving Complex Optimization Problems Using SQL Queries. Applied Sciences, 2022, vol 12. MDPI. https://doi.org/10.3390/app12094569

58. Katsogiannis-Meimarakis G., Koutrika G. A survey on deep learning approaches for text-to-SQL. The VLDB Journal, 2023, vol 32. Springer, Berlin. https://doi.org/10.1007/s00778-022-00776-8

59. Eido W. M., Yasin H. M. Machine Learning Approaches for Enhancing Query Optimization in Large Databases. Engineering And Technology Journal, 2025, vol 10. https://doi.org/10.47191/etj/v10i03.39

60. Codd E.F. A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, 1970, 13(6), P. 377–387. https://doi.org/10.1145/362384.362685

61. Ng P.A. Further Analysis of the Entity-Relationship Approach to Database Design. IEEE Transactions on Software Engineering, 1981, 7(1), P. 85–99. https://doi.org/10.1109/TSE.1981.234522

62. Al-Fedaghi S. An Enhanced Framework for Object-Relational Database Design – An Experimental Study. 2024 IEEE International Conference on Database Technologies and Applications (ICDTA), 2024, P. 105–112. https://doi.org/10.1109/icdta.2024.10808665

63. FastAPI, USA, 2026, URL: https://fastapi.tiangolo.com/

64. Django, USA, 2026, URL: https://www.djangoproject.com/

65. Chandiramani A., Singh P. Management of Django Web Development in Python, 2021, J. Manage. Serv. Sci., 1(2), P. 1–17, https://doi.org/10.54060/JMSS/001.02.005

66. Chen S., Ahmmed S., Lal K., Deming C. Django Web Development Framework: Powering the Modern Web. American Journal of Trade and Policy, 2020, 7(3), P. 99-106. https://doi.org/10.18034/ajtp.v7i3.675

67. React, 2026. URL: https://react.dev/

68. What Are Single Page Applications? Examples, Frameworks, and More, Durga Prasad Acharya, GEEKFLARE, 2025. URL: https://geekflare.com/dev/single-page-applications/

69. Penpot. Penpot is the open-source design platform for teams that need scalable collaboration, 2026. URL: https://penpot.app

ДОДАТКИ

Додаток А

Таблиця А.1

Код US	Назва та формулювання US	Acceptance Criteria
US01	*Авторизація в системі.* Як працівник компанії, я хочу авторизуватися через корпоративний SSO, щоб безпечно отримати доступ до свого робочого простору.	Scenario: Успішна авторизація. Given: Користувач на сторінці входу системи When: Вводить корпоративні облікові дані та підтверджує 2FA. Then: Система розпізнає роль користувача (PM, HR) And: Відкриває відповідний особистий кабінет.
US02	*Управління профілем компетенцій.* Як розробник, я хочу оновлювати інформацію про себе в особистому кабінеті, щоб система враховувала мої актуальні знання при підборі в команди.	Scenario: Додавання нової навички. Given: Користувач знаходиться в особистому профілі. When: Натискає кнопку «Додати нову навичку», обирає зі списку та натискає кнопку «Зберегти». Then: Навичка відображається у профілі і стає доступною для ШІ-алгоритму пошуку.
US03	*Інтеграція з трекерами задач.* Як адміністратор, я хочу підключити Jira та GitHub через API, щоб система автоматично збирала метрики активності без ручного введення.	Scenario: Підключення Jira. Given: Користувач знаходиться в розділі «Налаштування інтеграцій». When: Вводить API-ключ Jira та натискає «Синхронізувати». Then: Статус змінюється на «Підключено», починається фоновий імпорт історичних даних працівників.
US04	*Створення профілю нового проєкту.* Як проєктний менеджер, я хочу створити новий проєкт та вказати необхідні ролі, щоб розпочати процес підбору команди.	Scenario: Створення проєкту. Given: Користувач на панелі управління проєктами. When: Заповнює назву, терміни, додає вакантні ролі та натискає «Створити». Then: Проєкт з'являється у списку зі статусом «Очікує формування команди».
US05	*Налаштування фільтрів підбору.* Як проєктний менеджер, я хочу задати фільтри, щоб алгоритм шукав лише релевантних кандидатів.	Scenario: Застосування фільтрів Given: Користувач у відкритому профілі нового проєкту. When: Обирає необхідні теги навичок та бажаний рівень комунікабельності Then: Фільтри зберігаються як базові вимоги для роботи.
US06	*Автоматична генерація команд.* Як проєктний менеджер, я хочу натиснути кнопку генерації, щоб ШІ-алгоритм згенерував перелік працівників на проєкт.	Scenario: Успішна генерація. Given: Фільтри проєкту налаштовані. When: Користувач натискає кнопку «Згенерувати команду». Then: Протягом 5 секунд на екрані з'являються картки працівників підібраних до проєкту. Or: Якщо працівників немає, система відображає повідомлення про відсутність відповідних співробітників.
US07	*Відображення навантаження працівника.* Як проєктний менеджер, я хочу бачити завантаження кандидатів на проєкт, щоб візуально оцінити їхню доступність.	Scenario: Перегляд навантаження кандидата. Given: Користувач переглядаю згенерований варіант команди. When: Наводить курсор на картку запропонованого розробника. Then: З'являється спливаючий графік його планового завантаження по тижнях у відсотках.
US08	*Блокування перенавантаження.* Як розробник, я хочу, щоб система технічно забороняла призначати на мене задачі понад 85 % мого робочого часу.	Scenario: Перевищення 85 % ліміту. Given: Поточне планове навантаження користувача становить 80 %. When: Менеджер намагається додати мене на проєкт, який вимагає ще 10 % часу.

Код US	Назва та формулювання US	Acceptance Criteria
		Then: Дія блокується, система видає повідомлення «Помилка: Перевищення ліміту утилізації 85 %».
US09	*Ручне коригування складу команди.* Як проєктний менеджер, я хочу мати можливість замінити запропонованого алгоритмом учасника вручну, щоб врахувати особисті фактори.	Scenario: Заміна працівника Given: Користувач переглядає згенеровану команду. When: Перетягує іншого кандидата з резервного списку на місце поточного. Then: Система миттєво перераховує загальний Match Score команди та перевіряє ліміти навантаження.
US10	*Дашборд здоров'я компанії.* Як HR-директор, я хочу бачити глобальну аналітику рівня використання ресурсів, щоб приймати стратегічні та управлінські рішення.	Scenario: Перегляд глобальної аналітики. Given: Користувач на головній сторінці HR-порталу. When: Відкриває вкладку «Здоров'я компанії». Then: Відображаються графіки середнього навантаження та динаміка за останній місяць.
US11	*Фіксація роботи у вихідні дні.* Як HR-менеджер, я хочу, щоб система пасивно фіксувала активність у репозиторіях на вихідних, щоб виявляти приховані перепрацювання.	Scenario: Робота в неділю. Given: Розробник проводить коміт коду в неділю. When: Система синхронізує логи з GitHub. Then: Ця активність додається як тригер до предиктивної моделі ризику вигорання цього працівника.
US12	*Затвердження складу команди.* Як проєктний менеджер, я хочу фінально затвердити підібрану команду, щоб система автоматично розіслала запрошення учасникам доєднатись до проєктного середовища.	Scenario: Успішне затвердження. Given: Склад команди відповідає вимогам і не порушує 85 % ліміт. When: Користувач натискає «Затвердити команду». Then: Статус проєкту змінюється на «Активний» And: Розробникам надсилається запрошення на електронну пошту.
US13	*Рекомендація відпустки.* Як HR, я хочу мати можливість надати працівнику додаткові вихідні.	Scenario: Відправка пропозиції PTO. Given: Працівнику необхідно надати додаткові вихідні. When: Натискає кнопку «Запропонувати відпустку». Then: Працівнику в особистий кабінет автоматично надсилається пропозиція з доступними датами на відпустку
US14	*Експорт аналітичних звітів.* Як топ-менеджер, я хочу експортувати звіт про утилізацію ресурсів у форматі Excel щоб презентувати його на стратегічній сесії.	Scenario: Завантаження звіту в Excel. Given: Користувач знаходиться на сторінці глобальної ресурсної аналітики. When: Обирає потрібний квартал і натискає «Експорт у Excel». Then: Файл зі зведеними діаграмами формується та завантажується на мій пристрій.
US15	*Розмежування прав доступу.* Як адміністратор, я хочу налаштовувати ролі користувачів, щоб обмежити доступ до певної інформації	Scenario: Перевірка прав доступу. Given: Розробник має посилання на HR-дашборд. When: Намагається перейти за цим посиланням. Then: Система перевіряє роль і видає повідомлення «Доступ заборонено» And: Залишає користувача на головній сторінці.
US16	*Особистий дашборд розробника.* Як розробник, я хочу бачити список своїх поточних та майбутніх проєктів, щоб ефективно планувати свій час.	Scenario: Перегляд завдань та планового навантаження. Given: Користувач авторизувався в системі. When: Відкриває вкладку «Мій розклад». Then: Бачить таймлайн зі своїми поточними проєктами та індикатор загального завантаження з відповідним кольором.

Таблиця А.2

Беклог задач проєкту

Код та назва US	Задача	Підзадача	Роботи, які потрібно виконати
US01: Авторизація	T.01.1 Налаштування SSO	ST.01.11 Конфігурація IdP	1.Реєстрація_додатку_в_Azure_AD.
			2.Обмін_SAML/OAuth_токенами.
			3.Перевірка_стану_2FA.
			4.Тестування_входу_через_SSO.
US01: Авторизація	T.01.2 Маршрутизація RBAC	ST.01.21 Налаштування редиректів	1.Читання_ролі_з_JWT-токена.
			2.Мапінг_ролей_до_сторінок.
			3.Створення_сесії_в_системі.
			4.Перевірка_редиректів.
US02: Профіль компетенцій	T.02.1 UI особистого кабінету	ST.02.11 Форма навичок	1.Макет_профілю_з_компетенціями.
			2.Компонент_вибору_навичок.
			3.Кнопка_збереження.
			4.Відображення_доданих_тегів.
US02: Профіль компетенцій	T.02.2 Оновлення індексів	ST.02.21 Логіка збереження	1.API_збереження_навичок.
			2.Синхронізація_з_ШІ-пошуком.
			3.Валідація_на_дублікати.
			4.Юніт-тест_доступності.
US03: Інтеграція з трекерами	T.03.1 Інтерфейс управління	ST.03.11 Сторінка налаштувань	1.UI_для_ключів_Jira/GitHub.
			2.Кнопка_синхронізації.
			3.Шифрування_ключів_на_клієнті.
			4.Валідація_порожніх_полів.
US03: Інтеграція з трекерами	T.03.2 Фоновий збір метрик	ST.03.21 Воркери імпорту	1.Мікросервіс_підключення_до_API.
			2.Черга_завантаження_історії.
			3.Мапінг_ID_користувачів.
			4.Стрес-тест_імпорту.
US04: Профіль нового проєкту	T.04.1 Форма ініціалізації	ST.04.11 UI панелі управління	1.Макет_списку_проєктів.
			2.Форма:Назва,Опис,Терміни.
			3.Блок_вибору_ролей.
			4.Валідація_дат_початку/кінця.
US04: Профіль нового проєкту	T.04.2 Реєстрація проєкту	ST.04.21 Збереження в БД	1.Таблиці_Projects_і_Roles.
			2.API_прийому_даних.
			3.Присвоєння_статусу_очікування.
			4.Інтеграційні_тести_зв'язків.
US05: Фільтри підбору	T.05.1 UI блоку фільтрації	ST.05.11 Інтерактивна панель	1.Вибір_тегів_Hard_Skills.
			2.Повзунки_Soft_Skills.
			3.Оновлення_стану_фільтрів.
			4.Скидання_налаштувань.
US05: Фільтри підбору	T.05.2 Збереження шаблону	ST.05.21 Фіксація критеріїв	1.Поля_вимог_у_таблиці.
			2.Серіалізація_фільтрів_у_JSON.
			3.API_оновлення_вимог.
			4.Перевірка_збереження.
US06: Генерація команд	T.06.1 Інтеграція з ШІ	ST.06.11 Запит до алгоритму	1.Формування_Payload.
			2.Відправка_запиту_до_ШІ.
			3.Обробка_тайм-аутів.
			4.Обробка_порожньої_видачі.
US06: Генерація команд	T.06.2 Візуалізація підбору	ST.06.21 UI карток кандидатів	1.Картка_працівника.
			2.Анімація_очікування(Скелетон).
			3.Повідомлення_відсутності.
			4.Рендеринг_списку.
US07: Навантаження	T.07.1 Аналітика завантаження	ST.07.11 Агрегація даних	1.SQL-запит_сумарної_утилізації.
			2.Розподіл_по_тижнях.
			3.Оптимізація_швидкодії.
			4.Юніт-тести_розрахунку.
US07: Навантаження			1.Інтеграція_бібліотеки_графіків.
			2.Створення_Tooltip_при_наведенні.

Код та назва US	Задача	Підзадача	Роботи, які потрібно виконати
	T.07.2 Візуалізація графіка	ST.07.21 Спливаюча підказка	3.Рендеринг_діаграми.
			4.Позиціонування_на_екрані.
US08: Блок перенавантаження	T.08.1 Перевірка лімітів	ST.08.11 Налаштування 85 %	1.Константа_ліміту_85%.
			2.Middleware_перехоплення.
			3.Розрахунок:Поточне+Запит> 85%.
			4.Блокування_транзакції.
US08: Блок перенавантаження	T.08.2 Обробка помилок	ST.08.21 Нотифікації UI	1.Обробка_HTTP_409_Conflict.
			2.Інтеграція_Snackbar.
			3.Виведення_тексту_помилки.
			4.Автозникнення_повідомлення.
US09: Ручне коригування	T.09.1 Drag-and-Drop	ST.09.11 Перетягування карток	1.Впровадження_бібліотеки_DnD.
			2.Зони_перетягування.
			3.Візуальна_заміна_картки.
			4.Тестування_Touch-подій.
US09: Ручне коригування	T.09.2 Перерахунок метрик	ST.09.21 Реактивне оновлення	1.Перерахунок_Match_Score.
			2.Виклик_API_ліміту.
			3.Функція_Undo_при_перевищенні.
			4.Зміна_кольору_індикатора.
US10: Здоров'я компанії	T.10.1 Глобальні метрики	ST.10.11 Аналітичні запити	1.Розрахунок_середньої_утилізації.
			2.Збір_тренду_за_30_днів.
			3.Кешування_Redis.
			4.Перевірка_розрахунків.
US10: Здоров'я компанії	T.10.2 Візуалізація HR	ST.10.21 Сторінка дашборду	1.Створення_вкладки_HR.
			2.Лінійні_графіки_динаміки.
			3.Віджети_з_KPI.
			4.Адаптивність_інтерфейсу.
US11: Робота у вихідні	T.11.1 Аналіз комітів	ST.11.11 Парсер логів	1.Аналіз_Timestamp_комітів.
			2.Визначення_вихідних_днів.
			3.Тегування_комітів.
			4.Врахування_Timezones.
US11: Робота у вихідні	T.11.2 Інтеграція вигорання	ST.11.21 Передача до ШІ	1.Додавання_балів_ризику.
			2.Передача_профілю_до_ШІ.
			3.Логування_подій.
			4.Інтеграційні_тести.
US12: Затвердження команди	T.12.1 Зміна статусу	ST.12.11 Фіксація команди	1.Кнопка_затвердження.
			2.Транзакція_фіксації_в_БД.
			3.Зміна_статусу_на_Активний.
			4.Блокування_при_лімітах.
US12: Затвердження команди	T.12.2 Розсилка запрошень	ST.12.21 Поштовий модуль	1.Інтеграція_SMTP-сервісу.
			2.HTML-шаблон_листа.
			3.Асинхронна_розсилка.
			4.Обробка_помилок_Bounces.
US13: Рекомендація відпустки	T.13.1 Інтерфейс пропозицій	ST.13.11 Кнопка та форма	1.Кнопка_пропозиції_відпустки.
			2.Модальне_вікно_вибору_дат.
			3.Відправка_на_бекенд.
			4.Перевірка_прав_HR.
US13: Рекомендація відпустки	T.13.2 Внутрішні сповіщення	ST.13.21 Доставка повідомлень	1.Таблиця_Notifications.
			2.Відображення_в_кабінеті.
			3.Кнопки_Прийняти/Відхилити.
			4.Перевірка_WebSockets.
US14: Експорт звітів	T.14.1 Генерація Excel	ST.14.11 Бібліотека конвертації	1.Встановлення_бібліотеки.
			2.Агрегація_даних_кварталу.
			3.Форматування_таблиці.
			4.Перевірка_експорту_чисел.
US14: Експорт звітів	T.14.2 Завантаження клієнтом	ST.14.21 Ендпоінт завантаження	1.Кнопка_Експорт.
			2.API_повернення_Stream.
			3.Обробка_MIME-типу.
			4.Тест_скачування_файлу.

Код та назва US	Задача	Підзадача	Роботи, які потрібно виконати
US15: Права доступу	T.15.1 Захист маршрутів	ST.15.11 Перевірки фронтенду	1.HOC_перевірки_ролі.
			2.Захист_маршрутів_UI.
			3.Екран_403_Access_Denied.
			4.Блокування_прямих_переходів.
US15: Права доступу	T.15.2 Безпека API	ST.15.21 Валідація сервера	1.Перевірка_ролі_в_middleware.
			2.Повернення_403_Forbidden.
			3.Захист_чужих_профілів.
			4.Інтеграційні_тести_доступу.
US16: Дашборд розробника	T.16.1 Вкладка розкладу	ST.16.11 UI-таймлайн	1.Макет_часової_шкали.
			2.Рендеринг_проєктів.
			3.Функція_масштабування_часу.
			4.Відображення_перетинів.
US16: Дашборд розробника	T.16.2 Візуалізація зайнятості	ST.16.21 Індикатор навантаження	1.API_запит_сумарного_відсотка.
			2.Круговий_прогрес-бар.
			3.Динамічна_зміна_кольору.
			4.Оновлення_стану_індикатора.

Таблиця А.3

Ризики проєкту

Код ризику	Тип ризику	Ризикова подія	Сила впливу	Керованість
P01	Форс-мажор	Тривалі відключення електроенергії (блекаути), що зупиняють роботу команди	висока	середня
P02	Форс-мажор	Мобілізація ключових співробітників проєкту	висока	середня
P03	Форс-мажор	Фізична загроза життю команди через обстріли та повітряні тривоги	висока	низька
P04	Форс-мажор	Нестабільність інтернет-з'єднання у регіонах перебування команди	висока	середня
P05	Технічні	Складність реалізації технології миттєвих зрізів у реальному часі	висока	середня
P06	Технічні	Неточності в роботі алгоритмів ШІ	висока	середня
P07	Технічні	Висока затримка при обробці великих масивів даних	середня	висока
P08	Технічні	Уразливість вебплатформи до кібератак та витоку даних	висока	середня
P09	Технічні	Складність інтеграції платформи з існуючими HRM/ERP системами замовників	середня	середня
P10	Технічні	Вибір архітектури, що не масштабується під зростаюче навантаження	висока	висока
P11	Управлінські	Неконтрольоване розростання вимог до функціоналу	висока	середня
P12	Управлінські	Помилки в оцінці строків та бюджету розробки модулів машинного навчання	висока	середня
P13	Управлінські	Низький рівень комунікації між Data Science відділом та backend-розробниками	середня	висока
P14	Управлінські	Недостатня залученість стейкхолдерів на етапі тестування прототипу	середня	висока
P15	Управлінські	Відсутність або низька якість технічної документації API	середня	висока
P16	Кадрові	Професійне вигорання та стрес членів команди на тлі зовнішніх факторів (зокрема війни)	висока	середня
P17	Кадрові	Вплив когнітивних упереджень розробників на логіку оцінки ризиків у системі	середня	середня

Код ризику	Тип ризику	Ризикова подія	Сила впливу	Керованість
P18	Кадрові	Дефіцит вузькопрофільних фахівців (наприклад, експертів з AI risk management)	висока	середня
P19	Кадрові	Звільнення або релокація ключових фахівців на етапі активної розробки	висока	середня
P20	Операційні	Збої в роботі хмарного провайдера де розміщена платформа	висока	низька
P21	Операційні	Проблеми із сумісністю при оновленні сторонніх бібліотек та фреймворків	середня	висока
P22	Операційні	Падіння продуктивності системи при одночасних запитах від багатьох користувачів	висока	середня
P23	Фінансові	Непередбачуване зростання вартості хмарних обчислень для тренування ШІ-моделей	середня	середня
P24	Фінансові	Затримки у фінансуванні проєкту на проміжних етапах	висока	низька
P25	Юридичні	Невідповідність платформи вимогам GDPR при обробці персональних даних	висока	висока

Таблиця А.4

Якісна оцінка ризиків проєкту

№	Ризикова подія	Затримки у часі		Фінансові втрати		Ймовірність		Частота за проєкт		Важливість ризику
		ЯО	КО	ЯО	КО	ЯО	КО	ЯО	КО	
1	Тривалі відключення електроенергії (блекаути), що зупиняють роботу команди	ВВ	8	ВС	10	ВВ	8	ВВ	8	34
2	Мобілізація ключових співробітників проєкту	ВВ	9	ВВ	8	ВС	8	ВС	6	31
3	Фізична загроза життю команди через обстріли та повітряні тривоги	ВВ	10	ВВ	9	СВ	5	ВС	6	30
4	Нестабільність інтернет-з'єднання у регіонах перебування команди	ВС	6	СВ	5	ВС	7	ВС	7	25
5	Складність реалізації технології миттєвих зрізів у реальному часі	ВС	7	ВС	6	ВС	6	ВС	6	25
6	Неточності в роботі алгоритмів ШІ при предиктивному аналізі ризиків	ВС	6	СВ	5	СВ	5	СВ	5	21

№	Ризикова подія	Затримки у часі		Фінансові втрати		Ймовірність		Частота за проєкт		Важливість ризику
		ЯО	КО	ЯО	КО	ЯО	КО	ЯО	КО	
7	Висока затримка при обробці великих масивів даних	СС	4	СС	4	ВС	7	ВС	7	22
8	Уразливість вебплатформи до кібератак та витоку даних	ВВ	8	ВВ	9	СН	4	СС	4	25
9	Складність інтеграції платформи з існуючими HRM/ERP системами замовників	ВС	6	ВС	6	ВС	6	ВС	6	24
10	Вибір архітектури, що не масштабується під зростаюче навантаження	ВС	7	ВВ	8	СН	4	СН	4	23
11	Неконтрольоване розростання вимог до функціоналу	ВВ	8	ВС	7	ВС	7	ВС	7	29
12	Помилки в оцінці строків та бюджету розробки модулів машинного навчання	ВС	7	ВВ	8	ВС	6	ВС	6	27
13	Низький рівень комунікації між Data Science відділом та backend-розробниками	СВ	5	СС	4	ВС	6	СВ	5	20
14	Недостатня залученість стейкхолдерів на етапі тестування прототипу	СВ	5	СС	4	СВ	5	СВ	5	19
15	Відсутність або низька якість технічної документації API	СС	4	НС	3	ВС	6	СВ	5	18
16	Професійне вигорання та стрес членів команди на тлі зовнішніх факторів	ВС	7	СВ	5	ВВ	8	ВВ	8	28
17	Вплив когнітивних упереджень розробників на логіку оцінки ризиків у системі	СВ	5	ВС	6	СВ	5	СВ	5	21

№	Ризикова подія	Затримки у часі		Фінансові втрати		Ймовірність		Частота за проєкт		Важливість ризику
		ЯО	КО	ЯО	КО	ЯО	КО	ЯО	КО	
18	Дефіцит вузькопрофільних фахівців (наприклад, експертів з AI risk management)	ВВ	8	ВВ	9	ВС	7	ВС	6	30
19	Звільнення або релокація ключових фахівців на етапі активної розробки	ВС	7	ВС	7	СВ	5	СС	4	23
20	Збої в роботі хмарного провайдера де розміщена платформа	ВС	6	ВС	7	НС	3	НС	3	19
21	Проблеми із сумісністю при оновленні сторонніх бібліотек та фреймворків	СС	4	НС	3	ВС	7	ВС	7	21
22	Падіння продуктивності системи при одночасних запитах від багатьох користувачів	СВ	5	ВС	6	ВС	6	ВС	6	**23**
23	Непередбачуване зростання вартості хмарних обчислень для тренування ШІ-моделей	НС	3	ВВ	8	ВС	7	ВС	6	**24**
24	Затримки у фінансуванні проєкту на проміжних етапах	ВВ	9	ВВ	8	СН	4	СС	4	**25**
25	Невідповідність платформи вимогам GDPR при обробці персональних даних	ВС	6	ВВ	9	НС	3	НС	2	**20**

Таблиця А.5

Карта протиризикових заходів проєкту

№	Ризикова подія	ПРЗ 1 (профілактика)	Симптом (рання ознака)	ПРЗ 2 (при симптомі)	ПРЗ 3 (при проблемі)
1	Тривалі відключення електроенергії (блекаути)	Забезпечення команд автономним живленням (EcoFlow, генератори)	Публікація графіків жорстких відключень	Перехід на гнучкий / асинхронний графік роботи	Робота в режимі offline-розробки, перенесення релізів
2	Мобілізація ключових	Бронювання фахівців,	Отримання повістки для	Термінова передача знань	Перерозподіл задач, екстрений

№	Ризикова подія	ПРЗ 1 (профілактика)	Симптом (рання ознака)	ПРЗ 2 (при симптомі)	ПРЗ 3 (при проблемі)
	співробітників проєкту	перехресне навчання	уточнення даних		пошук заміни на ринку
3	Фізична загроза життю через обстріли	Можливість релокації	Збільшення інтенсивності тривог у регіонах	Зупинка нарад, перехід в укриття	Надання оплачуваних відпусток для відновлення, зсув дедлайнів
4	Нестабільність інтернет-з'єднання	Оплата резервних провайдерів, Starlink	Часті відключення на зустрічах, затримки пушів коду	Перехід на мобільний інтернет, текстові звіти	Компенсація коворкінгів з гарантованим зв’язком
5	Складність реалізації технології миттєвих зрізів	Створення PoC на ранніх етапах, залучення Tech Lead	Відставання в оцінках часу на розробку модуля	Парне програмування, архітектурні консультації	Перегляд архітектури, спрощення частоти зрізів
6	Неточності в роботі алгоритмів ШІ при аналізі	Використання якісних датасетів, A/B тестування моделей	Хибнопозитивні результати на етапі тестування	Додаткове тренування (fine-tuning) моделі	Тимчасове відключення ШІ-модуля, ручний аналіз ризиків
7	Висока затримка при обробці масивів даних	Використання асинхронних черг (Kafka / RabbitMQ)	Відгук API перевищує SLA (затримки > 1-2 с)	Профілювання коду, оптимізація запитів до БД	Кешування через Redis, масштабування серверів
8	Уразливість до кібератак та витоку даних	Аудит коду, впровадження WAF, наскрізне шифрування	Аномальна активність у логах, скарги користувачів	Блокування IP, активація жорстких правил брандмауера	Відключення вузлів, патчінг помилок
9	Складність інтеграції з існуючими ERP системами	Розробка універсального API, детальне вивчення документації	Помилки мапінгу даних під час першої інтеграції	Створення кастомних адаптерів для специфічних ERP	Залучення інтеграторів клієнта, обмеження списку підтримуваних систем
10	Вибір архітектури, що не масштабується	Навантажувальне тестування, cloud-native підходи	Зависання платформи при пікових навантаженнях	Автоматичне горизонтальне масштабування	Екстрене збільшення потужностей інстансів, рефакторинг
11	Неконтрольоване розростання вимог («feature creep»)	Жорсткий процес Change Request, фіксація Scope	Поява нових «критичних» вимог посеред спринту	Перенесення нових ідей у беклог	Перегляд бюджету / термінів, заморожування нових фіч до релізу MVP
12	Помилки в оцінці строків розробки модулів ШІ	Залучення Senior Data Scientists, закладання буферу 25 %	Відставання від графіка розробки понад 15 %	Декомпозиція задач, ескалація проблеми на Project Manager	Використання готових pre-trained моделей/API замість кастомних
13	Низький рівень комунікації Data Science та backend	Спільне планування архітектури, жорсткі API Contracts	Несумісність форматів даних під час злиття гілок	Фасилітовані зустрічі, уточнення контрактів у Swagger	Призначення єдиного архітектора за точку інтеграції

№	Ризикова подія	ПРЗ 1 (профілактика)	Симптом (рання ознака)	ПРЗ 2 (при симптомі)	ПРЗ 3 (при проблемі)
14	Недостатня залученість стейкхолдерів	Узгодження графіка демо-сесій, налаштування UAT середовища	Відсутність зворотного зв'язку після демо-релізів	Індивідуальні дзвінки, опитувальники для збору фідбеку	Блокування переходу на наступний етап до підписання актів UAT
15	Низька якість технічної документації API	Впровадження «Documentation-as-Code» (OpenAPI)	Питання від фронтенду щодо логіки роботи ендпоінтів	Включення оновлення документації у Definition of Done	Виділення окремого спринту на документування, залучення Tech Writer
16	Професійне вигорання членів команди	Регулярні 1-on-1, психологічна підтримка, контроль over-time	Зниження продуктивності, часті лікарняні, апатія	Зниження навантаження, позачергові короткі відпустки	Довгострокова відпустка, пошук тимчасової заміни
17	Когнітивні упередження при налаштуванні ШІ	Залучення незалежних експертів з ризик-менеджменту	Одностороння інтерпретація результатів, ігнорування факторів	Воркшопи з виявлення упереджень, перехресний рев'ю логіки	Зовнішній аудит алгоритмів оцінки ризиків
18	Дефіцит експертів з AI risk management	Моніторинг ринку, кадровий резерв	Затягування строків закриття вакансій (понад місяць)	Залучення рекрутингових агентств, розширення бюджету	Залучення зовнішніх консультантів на part-time
19	Звільнення ключових фахівців на етапі розробки	Підтримка високого Bus Factor, опціонні програми	Пасивність на зустрічах	Retention-мітинги, перегляд умов співпраці	Екстрена передача справ під час відпрацювання, найм аутсорсінгу
20	Збої в роботі хмарного провайдера	Вибір надійного провайдера, регулярні бекапи	Оповіщення моніторингу про деградацію сервісів провайдера	Перехід системи на «read-only» режим	Відновлення з бекапів на альтернативному хостингу
21	Проблеми із сумісністю сторонніх бібліотек	Фіксація версій залежностей, використання Docker	Падіння CI/CD пайплайнів після оновлень	Відкат до стабільної версії	Переписування коду під нове API, пошук аналогів бібліотеки
22	Падіння продуктивності при масових запитах	Налаштування Rate Limiting, Load Balancer	Зростання черги запитів, Timeout помилки у користувачів	Тимчасове відключення фонових некритичних процесів	Розширення серверних ресурсів, оптимізація обробки сесій
23	Зростання вартості хмарних обчислень для ШІ	Встановлення жорстких лімітів та алертів про бюджетування використання сервісів	Сповіщення про перевитрату бюджету до кінця місяця	Зупинка некритичних ресурсномістких експериментів	Оптимізація моделей ШІ, перехід на дешевші інстанси
24	Затримки у фінансуванні проєкту	Створення резервного фонду проєкту, чіткий графік траншів	Порушення строків узгодження актів з боку інвестора	Офіційна комунікація, скорочення другорядних витрат	Використання резервного фонду, заморожування робіт команди
25	Невідповідність платформи вимогам GDPR	Консультації з DPO на етапі архітектури бази даних	Виявлення юридичних прогалин під час внутрішнього аудиту	Призупинення випуску нових фіч для доопрацювання помилок	Тимчасове блокування сервісів в ЄС

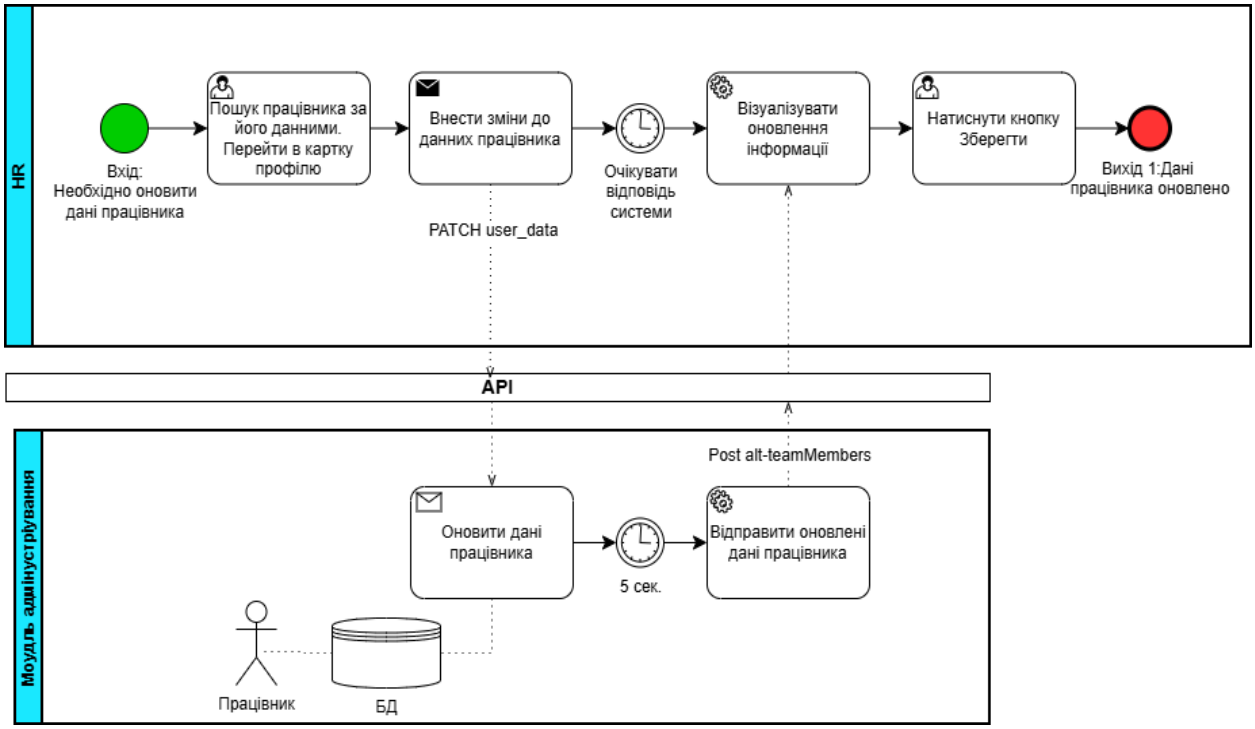


Рис. Б.1. Адміністрування інформації працівників

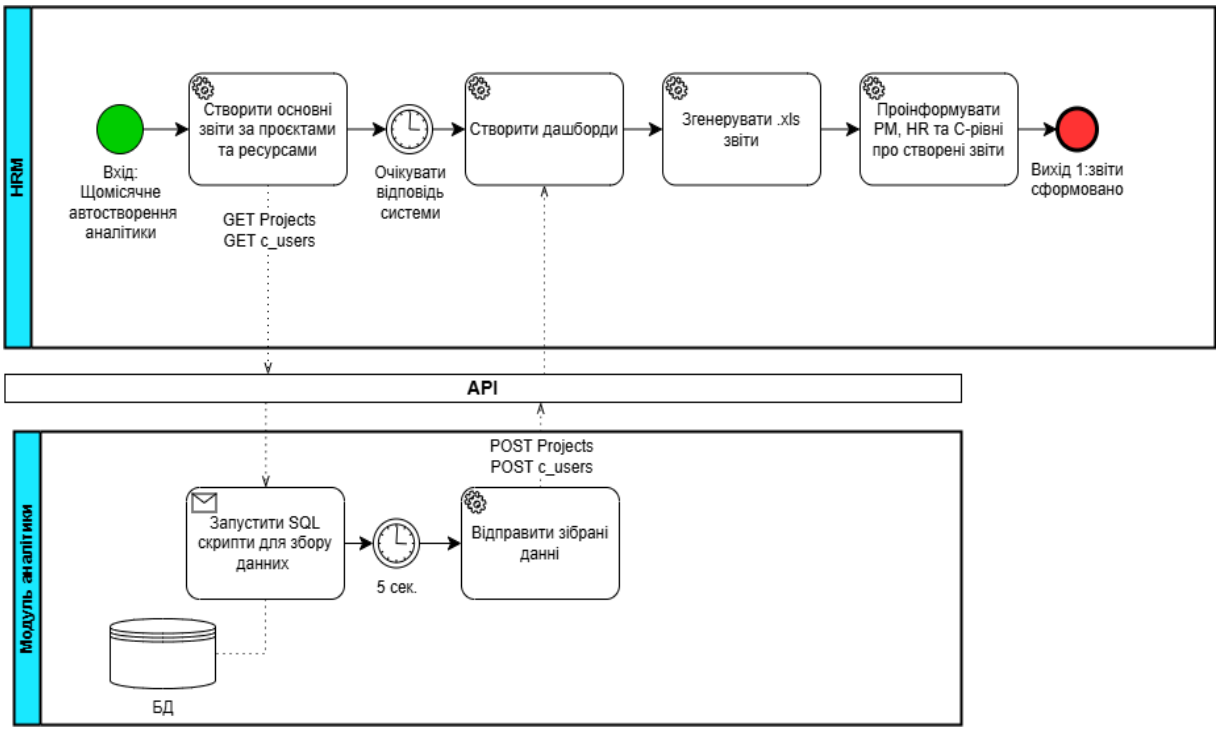


Рис. Б.2. Формування аналітичної звітності за використанням ресурсів

Додаток В

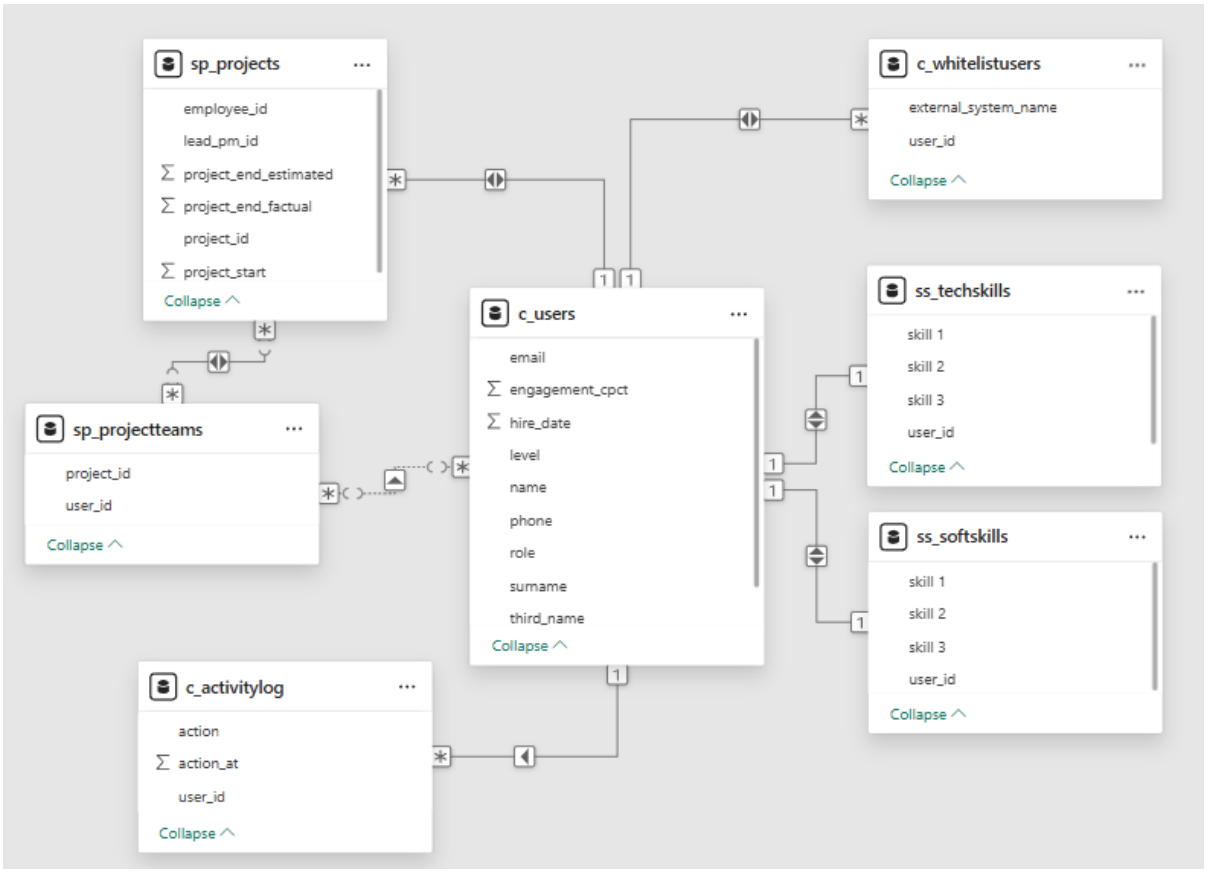


Рис. В.1. Концептуальна модель баз даних