

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

Кваліфікаційна робота
на здобуття освітнього рівня бакалавра
за спеціальністю 121 Інженерія програмного забезпечення

на тему:

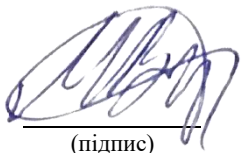
**РОЗРОБКА СИСТЕМИ ДЛЯ ОРГАНІЗАЦІЇ ДИСТАНЦІЙНОГО
НАВЧАННЯ**

Виконала студентка 4-го курсу
Аліна ГРИНЬ



(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Максим ВЕРЕС



(підпис)

Засвідчую, що в цій роботі немає запозичень з
праць інших авторів без відповідних
посилань.

Студент



(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри інтелектуальних
програмних систем
«28» травня 2021р.,
протокол №14
Завідувач кафедри
Олександр ПРОВОТАР

(підпис)

РЕФЕРАТ

Обсяг роботи – 57 сторінок, 22 рисунки, 6 таблиць, 17 джерел посилань.

ДИСТАНЦІЙНА ОСВІТА, СИСТЕМА ДИСТАНЦІЙНОГО НАВЧАННЯ,
DJANGO, MVC, PYTHON.

Об'єкт роботи являється процес створення системи для організації навчального процесу. Предмет роботи є система, призначена для автоматизації організаційних моментів навчального процесу.

В результаті роботи була розроблена система дистанційного навчання, основа системи – вебсайт, який можна буде використовувати для організації навчання.

Інструменти розроблення: інтегроване середовище розробки для мови програмування Python, мови програмування Python та JavaScript, фреймворки Django та Bootstrap, система контролю версій Git.

У першому розділі дипломної роботи було розглянуто поняття дистанційної освіти, історію розвитку, розглянуто переваги та недоліки дистанційної освіти загалом. У другому розділі розглянуто вимоги до систем дистанційної освіти, дві популярні системи Moodle та Classroom та їхній аналіз. У третьому розділі було описано вимоги до системи, її структурування та моделювання управління.

Програмний продукт може застосовуватися в навчальному процесі як у вищих навчальних закладах, так і шкільних та професійно-технічних закладах.

ЗМІСТ

ЗМІСТ.....	3
ВСТУП	4
РОЗДІЛ 1 ЗАГАЛЬНИЙ ОГЛЯД ДИСТАНЦІЙНОЇ ОСВІТИ.....	6
1.1 Поняття та сутність.....	6
1.2. Історія розвитку.	7
1.3. Переваги та недоліки.	10
РОЗДІЛ 2 ОГЛЯД СИСТЕМ ДИСТАНЦІЙНОГО НАВЧАННЯ	16
2.1. Вимоги до систем дистанційного навчання.....	16
2.2. Розгляд наявних рішень.	17
2.3. Аналіз отриманих результатів.	19
РОЗДІЛ 3. Розробка системи. Функціонал.....	23
3.1 Опис використаних технологій	23
3.1.1 Серверна частина	23
3.1.2 Клієнтська частина	27
3.1.3 Вибір середовища для розробки.....	29
3.2 Розробка системи	30
3.2.1 Модель виробничого циклу	30
3.2.2 Опис функціональних користувацьких вимог	32
3.2.3 Структурування системи.....	37
3.2.5 Модульна декомпозиція	37
3.2.6 Розробка бази даних	39
3.3 Робота користувача з системою	47
ВИСНОВКИ	55
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	56

ВСТУП

Актуальність роботи та підстави для її виконання. З кожним днем інновації все більше змінюють звичний для людей порядок речей. Внаслідок глибокої інтеграції цифрових технологій з нашим життям виникають ризики, як це зазвичай відбувається і з будь-якими іншими інноваціями. Однак щоб забезпечити впевнений розвиток галузі та суспільства, інновації стають необхідною умовою, і старі звички та ризики не повинні перешкоджати прогресу.

Враховуючи, що для сучасних людей інформаційні технології є невіддільною частиною їхнього повсякденного життя, очевидно, що освітня галузь повинна бути однією із перших у впровадженні новітніх цифрових технологій та практик. Використання сучасних технологій у дистанційній освіті посприяло відкриттю нових можливостей для полегшення навчання.

Дана робота націлена на допомогу в інтеграції цифрових технологій у навчання, а саме створення нових програмних засобів, пов'язаних з організацією та керуванням навчального процесу.

Зараз дистанційне навчання розвивається дуже швидкими темпами. Найбільший інтерес викликає використання дистанційного навчання у сфері вищої освіти. Багато вищих навчальних закладів пропонують свої послуги з використанням дистанційних технологій, сприяючи тим самим розширенню можливостей здобуття освіти та індивідуальних освітніх потреб учнів.

Мета й завдання роботи. Метою кваліфікаційної роботи є створення системи для організації та

Для досягнення цієї мети поставлено такі завдання.

- розглянути переваги та недоліки дистанційного навчання;
- проаналізувати наявні системи;
- визначити вимоги до власного проєкту;
- спроектувати базу даних;

– розробити систему організації дистанційного навчання на основі заданих вимог.

Об'єкт, методи й засоби розроблення. Об'єктом роботи являється процес розробки системи, яка допомагатиме організовувати навчальний процес, встановить засоби спілкування між викладачем та студентом.

Розробка проєкту поділяється на кілька етапів. Першим етапом є специфікація вимог до системи, яка розроблюється. Далі відбувається опис користувацьких вимог: формується опис усіх можливостей, які надаватиме система користувачам. Після цього обирається інструментарій. Було обрано наступні засоби: мову програмування Python з використанням вебфреймворку Django, JavaScript з фреймворком Bootstrap, інтегроване середовище розробки PyCharm.

Можливі сфери застосування. Програмний продукт може застосовуватися в навчальному процесі як у вищих навчальних закладах, так і шкільних та професійно-технічних закладах.

РОЗДІЛ 1 ЗАГАЛЬНИЙ ОГЛЯД ДИСТАНЦІЙНОЇ ОСВІТИ

1.1 Поняття та сутність.

Дистанційне навчання є формою навчання, основними елементами якого є фізичне розділення викладачів та студентів під час навчального процесу та використання різних технологій для полегшення спілкування студентів з викладачами. Заочне навчання традиційно орієнтоване на не традиційних студентів, таких як викладачі, військовослужбовці та нерезиденти, або осіб у віддалених районах, які не можуть відвідувати лекції в аудиторіях. Однак дистанційне навчання стало складовою частиною освітньої галузі, і ця тенденція вказує на стабільне зростання [1]. Для проведення дистанційного навчання використовуються Інтернет і вебтехнології.

E-learning – це поняття, яке часто використовується для опису дистанційного навчання. Слово E-learning – одне з численних слів з приставкою «Е-», що стали популярними в останні роки. Дедалі популярнішими стають послуги електронного банкінгу та електронної комерції, а також такі фрази, як електронний уряд, електронна охорона здоров'я та електронна судова система. В результаті метою дистанційної освіти є поступове перетворення суспільства в інформаційне.

Здійснення дистанційного навчання через установи, що відрізняється від самоосвіти. Так само як навчальні заклади надають можливість очної освіти, так можлива реалізація і дистанційної.

Характеристики дистанційного навчання:

- доступність та зручність є важливими перевагами дистанційного навчання. Добре розроблені програми дозволяють стати менш помітними відмінностям між студентами в інтелектуальному, культурному та соціальному планах;

- для взаємодії студентів з групою та викладачем використовується інтерактивна телекомунікація. Найчастіше використовують сучасні технології, такі

як електронну пошту, проте не потрібно забувати й про традиційні форми комунікації між людьми (поштова система). В незалежності від оточення, зв'язок з іншими має важливе значення для дистанційної освіти. Взаємодія учнів, викладачів та навчальних ресурсів стають менш залежними від територіальної близькості, чому посприяли такі речі як мобільні телефони, електронна пошта та Інтернет, що й дали поштовх швидкому зросту дистанційного навчання;

– як і будь-яка освіта, дистанційна освіта утворює навчальну групу, до складу якої входять студенти, викладач та навчальні ресурси (книги, аудіо, відео). Такі групи можуть створюватися не тільки в рамках закладів, прикладом цього є соціальні мережі. Створюючи профілі на таких сайтах як YouTube, Telegram чи Reddit, користувачі утворюють свої власні спільноти однодумців. Під час пандемії у режимі лише дистанційного навчання така взаємодія допомагає побороти відчуття ізоляції.

1.2. Історія розвитку.

Як організаційна форма навчального процесу в українській системі вищої освіти, проблема дистанційного навчання останнім часом привертає особливу увагу викладачів. Особливістю методу дослідження є те, що, хоча вона є провідною технологією 21 століття, та має глибокі історичні корені. Вивчення історії та аспектів навчання дистанційного навчання є важливим теоретичним завданням, і вирішення дозволить чітко визначити сучасні характеристики використання. Метою статті [2] є дослідити витоки дистанційного навчання в обширному огляді історії, географії та соціальної педагогіки та визначити внутрішній взаємозв'язок між сучасними перспективами дистанційного навчання та їх історичними та педагогічними колегами.

У 1999 р. фразу «Електронне навчання» було вперше згадано в професійному контексті Елліоттом Масі під час конференції TechLearn. Однак найперші спроби електронного навчання простежуються в 1924 році, коли професор університету

штату Огайо Сідні Прессі створив електронну навчальну систему «The Automatic Teacher».

Мабуть, найзначнішим винаходом в історії технологій онлайн-навчання є PLATO.



Рисунок 1.1 – Комп'ютерна система PLATO

PLATO – це комп'ютерна система, розроблена Дональдом Л. Бітцером у 1960 році, і вона була новаторською для свого часу (див. рис.1.1) [3].

За два десятиліття до винаходу Всесвітньої павутини мережа PLATO могла приймати тисячі користувачів, і на PLATO було створено велику частину найбільшого навчального програмного забезпечення століття для багатьох навчальних дисциплін. Термінали PLATO були дуже дорогими – від 5000 до 7000 доларів кожен, але вони коштували кожної копійки.

Слід зазначити, однак, що не всі винаходи в галузі онлайн-навчання та електронного навчання отримали таку високу оцінку, як PLATO, який пройшов численні успішні повторні випуски та чотири покоління. Історії успіху, як правило,

висвітлюються частіше, ніж невдачі, але так само важливо виділити ті нововведення, які не змогли охопити маси. Найдавніші винаходи в освітніх технологіях, можливо, були незграбними та базовими за сучасними мірками, але вони відіграли вирішальну роль у розвитку історії електронного навчання.

У статті [4] розглянуто розвиток технологій і зміна визначень для них, які були описані Ериолом у 2009 році:

- навчання на базі Web-технологій. Навчання полягає в тому, що освітній контент можна переглядати за допомогою веббраузера. Завантаження ресурсу з мережі не входить в цей контекст, доступ до нього може здійснюватися через інтернет, локальну мережу або CD/DVD. Особливо це може бути корисно в місцях, де немає повної мультимедійної інформації або обмежено підключення до мережі;

- онлайн навчання. Таке навчання охоплює матеріали, які доступні на комп'ютері, такі як онлайн-підтримка, онлайн-документи та онлайн-сервіси. Офісний помічник і майстер створення вебсторінок також включені в онлайн-навчання. При цьому немає потреби користуватися мережею;

- e-Learning. Це використання комп'ютерних і інтернет-технологій в будь-який час і в будь-якому місці, де учні мають потребу в навчанні;

- дистанційна освіта. Дистанційна освіта заснована на принципі спілкування між студентами і їхніми викладачами на великій відстані. Для цього необов'язково використовувати мережу. Навчання може бути більш ефективним, якщо використовується сумісно з електронним навчанням. Типовими прикладами дистанційної освіти з електронним навчанням є прямі освітні трансляції, відеоконференції, електронна пошта і дискусійні групи.

Лише час покаже, в яких напрямках еволюції відтепер буде прогресувати онлайн навчання. Але точно можна сказати, що воно нікуди не дінеться. Минула історія показала, що в цей час воно йде шляхом експоненціального зростання, і аналітики сходяться на думці, що з часом це зростання буде лише прискорюватися.

Невідомо наскільки цей зріст матиме вплив на традиційні навчальні заклади, але цілком ймовірно, що їм потрібно буде адаптувати себе до нової практики, щоб залишатися конкурентоспроможними в освітній галузі.

Вважається, що найближчі роки найбільшими нововведеннями в онлайн-навчанні буде використання адаптивного електронного навчання, яке здатне адаптувати себе та свої навчальні матеріали відповідно до набору навичок та рівня знань кожного студенту. Персоналізоване навчання – це шлях у майбутнє, а електронне навчання, безсумнівно, є одним із найкращих способів досягнення справді персоналізованих результатів навчання.

1.3. Переваги та недоліки.

Попри те, що попит на дистанційну освіту достатньо зріс, така форма навчання має свої переваги та недоліки:

– технологічність. Використовуючи сучасні програмні й технічні засоби, електронна освіта стає більш ефективною, а інформація, що подається у візуальному вигляді, сприймається більш яскраво та динамічно.

На рис.1.2 можна побачити результат опитування щодо навчання після пандемії. Наразі 44 відсотки респондентів сподіваються на те, що навчальні заклади введуть більшу кількість онлайн-навчання, ніж раніше. Ще 17 відсотків вважають, що школа буде зовсім іншою: навчання в Інтернеті стане неодмінною частиною шкільної практики. З іншого боку, 39 відсотків вважали, що зміниться дуже мало, 32 відсотки погодились з тим, що школа повернеться до своєї початкової практики з незначними змінами, а 7 відсотків вважали, що школа повернеться до своєї початкової практики.

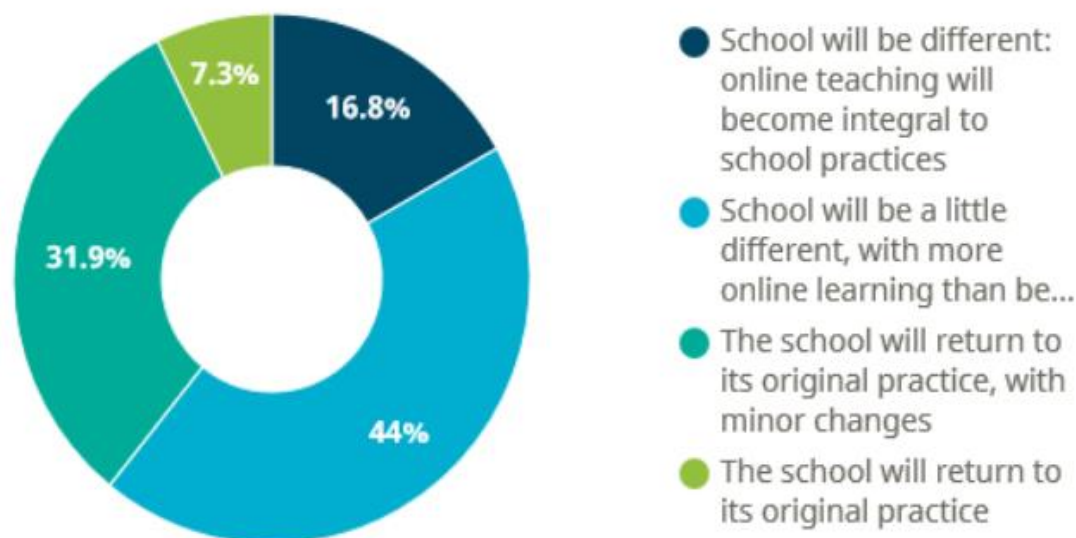


Рисунок 1.2 – Опитування студентів під час COVID-19 [5]

Майже кожен респондент повідомив про проблеми, серед яких найчастішим є доступ до технологій як для учнів, так і для вчителів, а також про збільшення навантаження та стресу, що виникає вдома. Було визначено кілька викликів, пов'язаних із підтримкою учнів, разом з їх цифровою компетентністю – та компетенцією вчителів.

Респонденти вважають, що підтримка з точки зору більшої кількості освітніх ресурсів допоможе їм розв'язати проблеми, а також чіткі вказівки Міністерства освіти разом із професійним розвитком, такими як короткі курси з викладання в Інтернеті та можливості для вчителів ділитися ресурсами, ідеями та виклики.

Ці висновки дають деяким раннім вказівкам для політиків та керівників шкіл, що досвід дистанційного/онлайн-викладання та навчання, хоча і є складним, може мати довготривалі позитивні наслідки, відкриваючи цікаві можливості для інновацій та нових способів роботи, особливо якщо їх підтримує відповідна та своєчасна професійний розвиток. Потрібна певна обережність при інтерпретації результатів, враховуючи, що вибірка респондентів базується на добровільній участі в Інтернеті та порівняно невелика у порівнянні з кількістю викладачів та інших зацікавлених сторін у Європі. Хоча в цьому опитуванні представлено понад 40

країн, більшість респондентів походять з Франції (36 відсотків від загальної кількості), Болгарії (19 відсотків) та Португалії (14 відсотків) (дослідження [5]).

Було проведено опитування, де брали участь викладачі американських вузів, яке полягало у порівнянні ефективності дистанційного й аудиторного навчання та охоплювало університети, що як забезпечують умови для дистанційного навчання, так і не надають такої можливості. Результат опитування показав, що 57 відсотків викладачів вважають, що дистанційне навчання не поступається або навіть перевершує традиційне навчання в класах, а також 33,3 відсотка респондентів вважають, що в найближчі кілька років результати дистанційного навчання перевершать результати аудиторного;

- доступність. Зі змогою навчатися з територіально віддалених частин люди можуть продовжувати отримувати знання протягом всього життя. Доступ через Інтернет дає можливість на поєднання різних діяльностей та доступ з будь-якого місця;

- нижчі витрати на навчання з боку учня. Дистанційне навчання дозволяє економити через те, що студент не повинен оплачувати дорогу, проживання, купівлю навчально-методичної літератури тощо;

- розбиття курсу на окремі блоки. Така можливість використання модулів дає змогу спростити пошук необхідної літератури та зробити вивчення більш гнучким;

гнучкість та доступність освіти. З появою можливості вибору відкритих курсів на різних ресурсах, можна одночасно навчатися за різними напрямками з використанням найефективніших технологій та найкваліфікованіших експертів. Студенти мають змогу обирати час, тривалість, послідовність вивчення, увесь процес навчання стає повністю адаптованим до їхніх здібностей та потреб. Застосовуючи дистанційне навчання, навчальні заклади можуть брати на навчання людей з особливими потребами та обмеженими можливостями, що дозволяє збільшити кількість студентів з інших міст та країн;

– індивідуальність систем дистанційного навчання – студент має змогу визначати хід навчання самостійно, кілька разів повертатися на окремий курс, пропускати поодинокі частини тощо. Студенти можуть мати доступ до навчальних матеріалів протягом усього навчального процесу, що дозволяє забезпечити глибші залишкові знання. Така система освіти змушує учнів брати участь самостійно та здобувати навички самоосвіти;

– можливість визначати критерії оцінки знань – в електронному навчанні є можливість виставляти чіткі критерії, за якими оцінюються знання, отримані студентом в процесі навчання. Це дозволяє виключити необ'єктивність і упередженість.

Недоліки дистанційного навчання:

– проблема автентифікації. Оскільки досі не існує оптимального рішення для цієї проблеми, то досить велика кількість навчальних закладів пропонує традиційну очну перевірку знань навіть під час пандемії. Навіть з використанням камер, що частково розв'язують проблему, немає змоги досконально перевірити навіть особистість, а не тільки присутність та самостійність виконання тих чи інших завдань. Ця проблема вимагає від викладачів вжиття спеціальних заходів, умінь та навичок;

– відсутність безпосереднього особистого спілкування студентів з викладачами. Щобільше, емоційно знебарвлене подання знань є великим недоліком в навчальному процесі;

– необхідність індивідуальних психологічних умов. Дистанційне навчання вимагає від студентів свідомості та суворої самодисципліни, від яких напряму залежить результат роботи;

– необхідність використання персонального комп'ютера та доступ до інтернету;

– висока складність розробки курсів. Одним зі способів вирішення цієї проблеми є пошук та використання існуючих відео- та аудіоматеріалів за допомогою методів, які дозволяють поступово ускладнювати курси дистанційного

навчання. Також є важливим те, що навчальні матеріали, завдання та інструкції для дистанційної освіти слід формулювати більш ретельно, ніж для очного навчання. Досвід дистанційного навчання на практиці показав, що метод командної роботи для розробки та організації навчального процесу в дистанційному навчанні є найбільш годящим. Розробка курсів вимагає досить велику роботу у створенні курсів, підборі навчальних матеріалів та використанні різних технологій, саме тому це потребує участі експертів, таких як: редактори, адміністратори, технічний персонал тощо [6].

Ще одне опитування (див.рис.1.3) показує, як необхідно запроваджувати дистанційну освіту не лише для студентів, а також для викладачів. З восьми варіантів найчастіше обирають додаткові безплатні ресурси від компаній, що працюють у сфері освітніх технологій (45 відсотків відповідей). Іншими корисними типами викладання та навчання стали вебсайти зі списками корисних ресурсів (29 відсотків) та освітні телепрограми національних ЗМІ (10 відсотків).

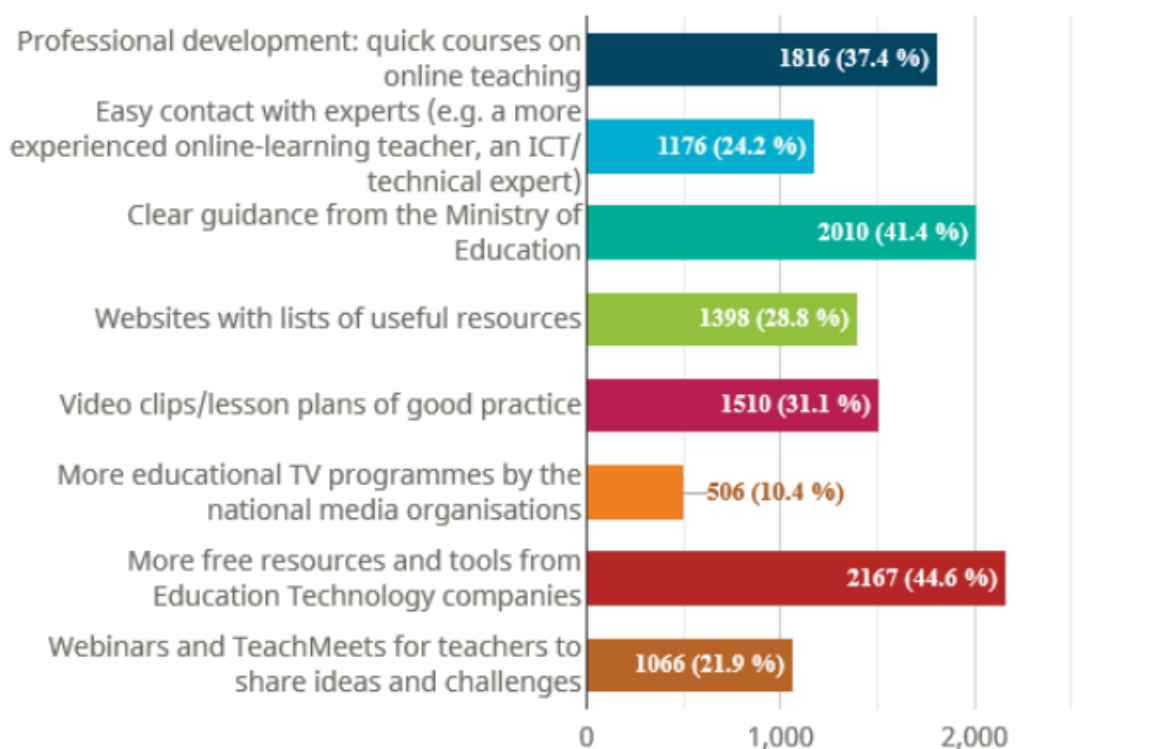


Рисунок 1.3 – Опитування викладачів шкіл під час COVID-19 [5]

Другим за частотою згадування про підтримку стали чіткі вказівки Міністерств внутрішніх справ Освіта (41 відсоток).

Часто обирали різні типи професійного розвитку, зокрема швидкі курси з онлайн викладання(37 відсотків), вебінари та TeachMeets для вчителів, де вони можуть обмінюватися ідеями та проблемами (22 відсотки), відеокліпами/планами уроків, заснованими на передовій практиці (31 відсоток) та контакт з експертами, такими як досвідчений викладач або експерт з ІТ (24 відсотки).

РОЗДІЛ 2 ОГЛЯД СИСТЕМ ДИСТАНЦІЙНОГО НАВЧАННЯ

2.1. Вимоги до систем дистанційного навчання.

Згідно з роботою [7] рішення для дистанційного навчання повинне задовольняти подальшим вимогам:

- комплексність. Система повинна мати можливість охоплення усіх етапів дистанційного навчання й усіх його учасників;
- здатність підлаштовуватися. Рішення має підходити вимогам та потребам замовника, підтримувати видозміни налаштувань та інтеграцію додаткових компонентів;
- простота та зручність інтерфейсу. Вимога до простого і виразного інтерфейсу користувача зумовлюється тим, що роботи з застосунком студенти й викладачі не повинні бути спеціалістами в області інформаційних технологій, тому для полегшення потрібна інтуїтивна зрозумілість;
- варіативність комунікації. Система повинна мати певний різновид варіантів для полегшення зв'язку між учасниками навчального процесу. Прикладами комунікації можуть бути форуми, окремі діалоги, аудіо та відеозв'язок;
- контроль успішності. Для зручного відстеження прогресу навчання студента, користувачі різного доступу повинні мати змогу переглядати, додавати або редагувати оцінки.

2.2. Розгляд наявних рішень.

2.2.1 Вебсервіс Classroom

Google Classroom – це освітній інструмент, який допомагає викладачам швидко створювати та управляти навчальними завданнями, забезпечувати зворотний зв'язок і спілкуватися зі своїми класами.

Компанія Google збрала в одному сервісі декілька своїх інструментів. Серед них: диск для зберігання файлів, Google Docs для публікації текстових лекцій, презентації, опитування, сервіс для відеозустрічей Meet.

Переваги використання Classroom:

- усі файли зберігаються на диску. За допомогою Google Drive є можливість перегляду теки усіх присутніх класів, курсів, модулів, керувати вмістом тек, надавати доступи. Багато хто відразу відмічає проблему з тим, що Classroom забирає багато місця та створення тек відбувається дуже повільно;

- створення завдань. При створенні завдань у формі Google Docs система створює та поширює особисті копії для студентів, якщо є така вимога. Також для завдань можна задавати терміни виконання. Якщо студент повертає готове завдання викладачеві, у цього документа відображається статус перегляду. Учитель має можливість повернути завдання на доопрацювання або ж відразу виставити бали;

- контроль за виконанням завдань. Можна спостерігати окремо за кожним завданням або відразу всіма завданнями в усіх класах;

- комунікація у класі. Спілкування реалізується шляхом оголошень, які можна створювати та надсилати всьому курсу. Також є можливість залишати коментарі до конкретних завдань конкретного студента.

Проте є також деякі мінуси системи, які розглянуто нижче:

- при дуже зручному створенні завдань виникають проблеми зі створенням тестів: їхня реалізація, задання типів та відсутність автоматизованої

перевірки для деяких типів питань. При такому комплексі проблем дуже важко організувати процес дистанційного навчання за цим принципом;

- до процесу оцінювання також виникає купа питань: результатом є таблиця з балами, в якій не відображаються усі потрібні дані. Відсутня можливість ведення журналів успішності, організації робочого процесу. Структура оцінок незручна, відсутнє поняття куратора;

- через певні обмеження панелі адміністратора, якоюсь мірою адекватно можна працювати до ста учнів, далі контроль процесу стає вже досить складним. LMS Moodle дає змогу організувати процес набагато простіше за допомогою плагінів;

- комунікація в реальному часі відсутня;

- Classroom не дозволяє приєднання учнів до вже початого курсу, неможливо розподіляти студентів у окремі групи в рамках курсу.

2.2.2 Система Moodle

Moodle – це гнучка система дистанційного навчання з відкритим початковим кодом, що підлаштовується під потреби користувача, допомагає задовольнити широкі потреби в організації навчання для освітніх установ і для бізнесу.

Moodle базується на модульній конструкції, яка дозволяє викладачам та адміністраторам створювати власну навчальну програму, використовуючи плагіни для різних робочих процесів, змісту та заходів. Користувачі можуть вибрати, встановити свій обліковий запис Moodle на своїх серверах або в хмарі. Хмарна платформа під назвою MoodleCloud надає кілька переваг, таких як доступність, масштабованість та дуже короткий процес реалізації. MoodleCloud поставляється у двох пакетах: Moodle for Free та Moodle for School.

Платформу з відкритим кодом Moodle можна безплатно завантажити та встановити. Однак його хмарне розгортання, MoodleCloud, має п'ять платних планів.

Moodle надає безліч функцій, таких як мобільний додаток, засоби створення вмісту та підтримку декількох мов. Деякі функції включають BigBlueButton для відео/вебконференцій, Quizventure для гейміфікації з тестами та тестами, інший модуль гейміфікації під назвою LevelUp, що включає прогресивні контрольні точки, кількість слів для написання завдань, редактор хімії для хімічних завдань, вибір групи для групових проєктів та контрольні списки та відстеження відвідуваності.

Moodle підтримує сайт спільноти, де користувачі можуть знаходити інформацію про оновлення продуктів, завантажувати останні випуски, отримувати доступ до документації щодо функцій програмного забезпечення та спілкуватися з іншими користувачами на форумі. Користувачі MoodleCloud, які потребують подальшої підтримки, можуть зв'язатися зі своїм призначеним партнером Moodle.

Moodle також проводить конференції користувачів під назвою MoodleMoots, які дозволяють користувачам спілкуватися з розробниками та партнерами Moodle, а також дізнаватися про новинки про Moodle.

2.3. Аналіз отриманих результатів.

Загалом, Google Classroom має простіший та зрозумілий користувацький інтерфейс, що пришвидшує роботу з цим сервісом. Створення тестів чи завдань в Moodle вимагає більше часу.

Створення та повторне використання минулих курсів з попередніх років також займає в Classroom менше часу та роботи з системою.

Оскільки Moodle являє собою повноважну LMS, то логічним є те, що вона має більший функціонал для педагогічної діяльності. Наприклад, представлення команд, банк питань з категоріями та випадкові питання з них, оцінку питань,

повторну оцінку, статистику тощо. Багато хто з цих речей або є відсутніми в Classroom, або є дуже поверхневими. У Moodle присутня велика кількість опцій, але відсутній оптимізований робочий процес. У Classroom, на відміну від Moodle, є свій канал, який можна використовувати для зв'язку, якщо немає можливості іншого спілкування.

Як видно на рисунку 2.1, більшість закладів освіти надають перевагу Classroom, на що вказують обраний колір та його інтенсивність [8].



Рисунок 2.1 – Вибір в різних частинах: а – України, б – світу

З таблиці 2.1 можна побачити, що Moodle як LMS має більше функціоналу, проте з рисунку 2.2 [9] видно, що Classroom має більший попит за свою простоту, інтегрованість з іншими продуктами Google та можливість безоплатного використання.

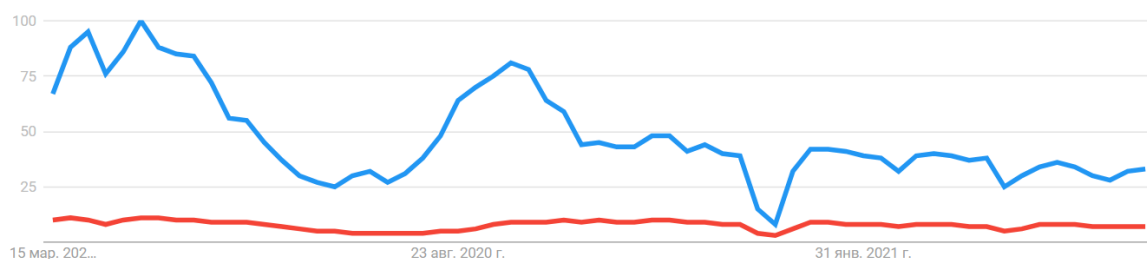


Рисунок 2.2 – Динаміка популярності Classroom та Moodle в світі

Таблиця 2.1 – Порівняльна таблиця Classroom та Moodle

	Classroom	Moodle
Можливості	Досить подібні. Але Moodle має більший функціонал для роботи з курсами.	
Ціна	Безкоштовна платформа.	Різноманітність тарифів, в тому числі і безкоштовна версія.
Інтегрованість	API Classroom	Moodle Core API
Легкість використання	4,6/10	4,2/10
Підтримувані мови	54 мови.	Понад 100.
Підтримувані операційні системи	Windows, Android, IOS, Web-Based	Windows, Android, Web-based

Проаналізувавши можливості, досвід користування (UX) та використовність Classroom та Moodle, можна визначити можливі варіанти функціоналу для власної майбутньої системи(таблиця 2.2).

Таблиця 2.2 – Таблиця з порівнянням можливостей аналогів з власним рішенням

Назва можливості	Classroom	Moodle	Власне рішення
Створення та редагування курсів.	+	+	+
Створення та редагування завдань.	+	+	+

Назва можливості	Classroom	Moodle	Власне рішення
Перегляд розкладу. Автоматичне створення пар.	-	+	+
Можливість спілкування викладача з курсом.	+	-	+
Автоматичний поділ варіантів.	-	-	+
Статистика	+	-	+

РОЗДІЛ 3. Розробка системи. Функціонал

3.1 Опис використаних технологій

3.1.1 Серверна частина

Python – інтерпретована об'єктноорієнтована мова програмування високого рівня зі строгою динамічною типізацією [10].

Ця мова є об'єктноорієнтованою загального призначення та була розроблена з метою підвищення продуктивності розробників. Переваги Python над іншими наступні:

- популярність. За індексом ТІОБЕ (індекс, який дає оцінку популярності мов програмування, беручи за основу кількість результатів пошукових запитів, які охоплюють назву мови) [11] зараз Python посідає друге місце після С та обирався мовою року чотири рази: у 2007, 2010, 2018 та 2020 роках. На рисунку 3.1 можна побачити прогрес мови за останні 18 років.

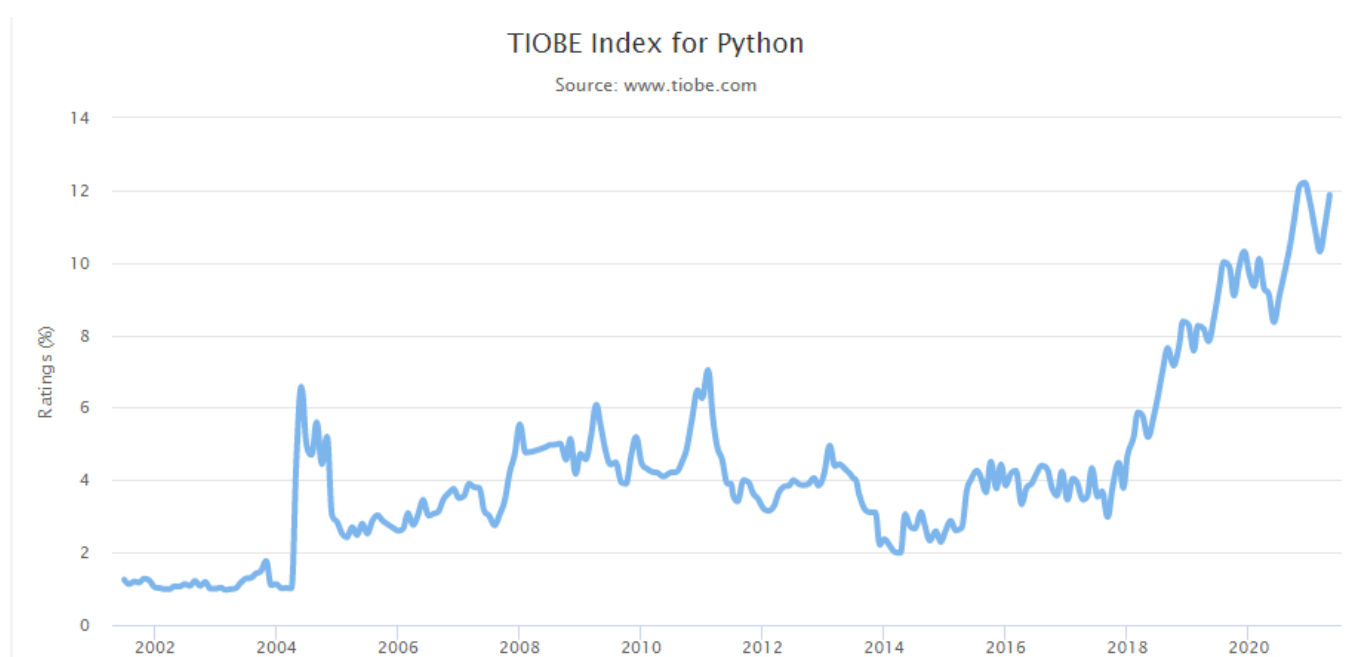


Рисунок 3.1 – Індекс мови Python за ТІОБЕ [11]

Популярність є важливою складовою, бо це є індикатором якості, а також велика аудиторія сприяє розвитку та покращенню мови;

- велика кількість зовнішніх бібліотек, чому сприяє популярність Python. Стандартна бібліотека спрощує виконання багатьох стандартних завдань, але також ще більше корисних зовнішніх Python пакетів, яких зараз вже понад двісті тисяч (це лише на офіційному PyPi – каталозі програмного забезпечення, яке написано на Python);

- кросплатформеність. За наявності інтерпретатора Python на Windows, програма буде функціонувати так само в незалежності від операційної системи. Відмінності виникають лише в окремих випадках, проте і вони можуть попереджені завчасно;

- простота та лаконічність. Python має простий синтаксис, який сприяє доступності читання коду;

- автоматичне збирання сміття, що дозволяє уникати витoku пам'яті;

- динамічна типізація. При розробці на Python непотрібно завчасно оголошувати тип змінної, що є дуже зручним. Проте це є також і мінусом, тому що тоді Python поступається у продуктивності таким мовам як C++;

- асинхронність. Під час очікування відповіді від сервера асинхронні додатки можуть продовжувати обробляти запити, що дозволяє уникати простоїв. Тому використовувати Python відповідає вимогам та дозволяється у використанні навіть для високонавантажених сайтів, де досить важлива швидкість обробки запитів.

Не дивлячись на деякі свої мінуси, для більшості задач Python є досить хорошим вибором для розробки системи.

Django – вебфреймворк, написаний на Python.

Він вважається одним з найкращих Python фреймворків. Основні переваги Django, які допомогли йому стати популярним:

- стандартизована структура. Фреймворк задає структуру проекту та дозволяє легко розуміти, де додавати нову функціональність. Така структура допомагає швидше знаходити вирішення до питань, що з'являються;
- додатки Django. Підхід, який дозволяє розділити проект на кілька частин, завдяки чому можна легко інтегрувати готові рішення;
- безпека. Django обробляє більшість відомих атак. Фреймворк надає захист від міжсайтового скриптингу (XSS), міжсайтових підробок запитів (CSRF), ін'єкцій SQL, клікджекінгу;
- адміністративна панель. Django автоматично генерує адміністративну панель під час створення додатку, що позбавляє необхідність створювати її вручну;
- розширюваність. Django дозволяє розширювати функціональність проекту та адаптувати його за допомогою плагінів, які можна швидко додавати, відключати та замінювати;
- висока швидкість роботи.

MySQL – це вільна система керування реляційними базами даних з відкритим кодом.

Переваги використання MySQL:

- простота у використанні. СУБД MySQL є досить простою у використанні. Для цього необхідно володіти лише базовими знаннями SQL. За допомогою всього лише декількох простих операторів SQL користувач може створювати та взаємодіяти з базою даних.
- безпека. MySQL включає в себе надійний рівень безпеки даних, який захищає конфіденційні дані від зловмисників. Крім того, паролі в MySQL зберігаються в зашифрованому вигляді.
- клієнт-серверна архітектура. MySQL використовує принцип роботи клієнт-серверної архітектури. На сервері бази даних (MySQL) знаходиться довільна

кількість клієнтів (прикладних програм), які взаємодіють з сервером, тобто можуть запитувати дані, зберігати зміни тощо.

- безкоштовна для завантаження. MySQL є безплатною для використання, тому є можливість завантажити з офіційного сайту MySQL без будь-яких витрат;

- масштабованість. MySQL підтримує багатопоточність, що робить її легко масштабованою, може обробляти практично будь-який обсяг даних до 50 мільйонів рядків і навіть більше. Граничний розмір файлу за замовчуванням становить близько 4 ГБ. Однак існує можливість збільшити це число до теоретичної межі в 8 ТБ даних;

- швидкість. MySQL вважається однією з мов швидкої дії баз даних, що підтверджується численними еталонними тестами;

- висока гнучкість. СУБД підтримує велику кількість вбудованих додатків;

- сумісність з безліччю операційних систем. MySQL підтримує роботу на багатьох операційних системах, таких як Novell NetWare, Windows, Linux, багато різновидів UNIX та інші. MySQL також надає можливість клієнтам працювати на тому ж комп'ютері, що і сервер;

- можливість відкату. MySQL дозволяє виконувати відкат транзакцій, фіксацію і відновлення після збою;

- висока ефективність використання пам'яті. Ефективність MySQL досить висока, оскільки проблема витоку пам'яті дуже низька;

- висока працездатність. MySQL є швидкою, надійною і дешевою завдяки своїй унікальній архітектурі механізму зберігання даних. СУБД забезпечує дуже високу продуктивність у порівнянні з іншими базами даних без втрати суттєвої функціональності програмного забезпечення;

- незалежність від платформи. Можна завантажити, встановити та виконати на більшості доступних операційних систем;

- поділ на розділи. Ця функція підвищує продуктивність і забезпечує швидке управління великою базою даних;
- підтримка графічного інтерфейсу користувача. MySQL надає уніфікований візуальний інструмент графічного призначеного для користувача інтерфейсу бази даних під назвою MySQL Workbench [12].

Також однією з переваг вибору фреймворку Django є те, що він використовує ORM для праці з базами даних, що є досить практичним для розробників, оскільки це сприяє не вдаватися в структуру самих баз даних та дозволяє працювати з різними БД. Тому даний фреймворк залишається достатньо гнучким для цілей системи.

3.1.2 Клієнтська частина

HTML (HyperText Markup Language) – мова розмітки гіпертексту, за допомогою якої створюються вебсторінки. Вона інтерпретується браузером і відображається у зручному читабельному вигляді. HTML визначає структуру та зміст вебконтенту.

HTML – це мова тегів розмітки документів. Будь-який HTML-документ являє собою набір елементів, де спочатку та вкінці кожного ставлять спеціальні позначки, які називають тегами. Можливі випадки пустих тегів, коли відсутній будь-який текст чи інша інформація, тоді зазвичай не ставлять тегу вкінці елемента.

CSS (Cascading Style Sheets) – це формальна мова опису стилю вебсторінок, яку використовують для опису зовнішнього вигляду HTML-документу. Також ця мова може бути застосована до XHTML та XML документів.

Використовують CSS для забарвлення шрифтів, стилів, розташування окремих елементів та блоків сторінок. Головна ціль використання CSS – це відокремлення опису зовнішнього вигляду сторінки від опису її логічної структури.

JavaScript – мова, яка використовується Програми, написані цією мовою, називають скриптами. Їх можна вбудовувати в HTML-документи, і код буде

звершуватися автоматично при запуску сторінки. Скрипти не потребують спеціальної підготовки чи компіляції, щоб їх запустити.

Переваги JavaScript:

- швидкість. Клієнтська сторона JavaScript є досить швидкою, оскільки може бути запущеною безпосередньо в браузері на стороні клієнта. При відсутності потреби у зовнішніх ресурсах, JavaScript не звертає уваги на мережеві звернення до внутрішнього сервера;
- простота. JavaScript є порівняно простою мовою у вивченні та застосуванні;
- популярність. Дана мова програмування широко використовується в мережі інтернет;
- сумісність. Легка взаємодія з іншими мовами й може бути використана у більшості додатків;
- навантаження на сервер. Робота з боку клієнта знижує навантаження на сервер сайту;
- різноманітність та насиченість інтерфейсів. Можливість створення різноманітних інтерфейсів [13].

Bootstrap 4 – це CSS фреймворк, який є безплатним для використання та призначений для створення сайтів. Він має безкоштовний набір CSS та JavaScript файлів, які не потрібно завантажувати, а досить під'єднати до сторінки, щоб почати використання. Після підключення стають доступними такі складові Bootstrap як:

- інструменти для створення макету сайту: адаптивні утилітні класи, наперед задані, готові до використання колонки (сітки);
- класи для стилізації базового вебконтенту: зображення, текст, таблиці, код та фігури;
- готові компоненти такі як форми, навігаційні панелі, випадні списки, слайдери тощо;
- утилітні класи для розв'язання найпоширеніших задач: відображення елементів HTML, вирівнювання та форматування тексту.

Цей фреймворк має наступні переваги у використанні:

- висока швидкість у створенні якісної адаптивної верстки, яку можуть забезпечити навіть початківці;
- кросплатформеність;
- наявність відкритого коду, що дозволяє використання у комерційних та некомерційних проєктах;
- велика кількість готових рішень, які вже є протестованими на різних пристроях різними розробниками;
- можливість налаштування під власні проєкти.

3.1.3 Вибір середовища для розробки

Для розробки було обрано PyCharm – інтегроване середовище розробки для мови програмування Python, яке має необхідний набір засобів для ефективного програмування.

Переваги використання PyCharm:

- сумісність з операційними системами Windows, Linux та Mac OS;
- легка інтеграція з git, Mercurial і SVN;
- розумний редактор коду: виділення ключових слів, класів і функцій, надання функції автозаповнення та додаткові інструкції. Завдяки цьому можна легко визначати помилки, покращується читабельність коду, програмування стає швидшим та легшим;
- навігація по коду: на додачу до легкого переходу по коду також є режим об'єктива. Він дозволяє розробнику ретельно перевіряти і налагоджувати весь свій вихідний код;
- ефективний та швидкий рефакторинг, який дозволяє вдосконалювати внутрішню структуру, не змінюючи зовнішньої продуктивності коду, та розділяти класи та функції, використовуючи метод вилучення;

– підтримка популярних фреймворків Python: web2py, Pyramid, Django та інші популярні фреймворки. Оскільки в проєкті використовується фреймворк Django, то досить важливим фактом є те, що середовище розробки має досить тісну інтеграцію з цим фреймворком. PyCharm надає функцію відлагодження шаблонів Django, автозаповнення щодо параметрів Django.

GitHub – це вебсервіс для хостингу та спільної розробки проєктів, який базується на системі контролю версій під назвою Git. GitHub є безплатним для проєктів з відкритим кодом та надає різноманітні платні пакети послуг для окремих індивідуальних проєктів [14].

Особливості вебсервісу GitHub:

- вбудована система відстеження помилок;
- функція візуалізації гілок у проєкті;
- створення необмеженої кількості приватних репозиторіїв в безплатному тарифному плані;
- можливість спільної роботи до трьох розробників у приватних репозиторіях безплатно;
- можливість пропонувати власні виправлення до коду інших розробників;
- легка публікація та завантаження коду. Користувач має можливість завантажити код до репозиторію як за допомогою Git, так і з вебінтерфейсу сервісу.

3.2 Розробка системи

Розробка системи є досить довгим шляхом та включає в себе багато процесів, які будуть розглянуті в наступних підрозділах.

3.2.1 Модель виробничого циклу

Розробка програмного забезпечення є структурованим процесом, який має наступні фази життєвого циклу:

- аналіз вимог;
- проектування;
- програмування;
- тестування;
- експлуатація.

Аналіз вимог вимагає дослідження предметної області, збір та аналіз інформації про аналогічні продукти та рішення. На основі отриманих даних відбувається специфікація вимог до програмного забезпечення.

Наступним кроком є проектування, що вимагає від розробника створення архітектури майбутнього програмного забезпечення, опис користувацьких та системних вимог, створення діаграм для візуалізації вимог.

Після цього починається програмування. На основі попередніх етапів розробки розроблюється програмне забезпечення.

Під тестуванням програмного забезпечення мають на увазі процес перевірки відповідності описаних вимог до реалізованого функціоналу системи.

Останнім етапом є експлуатація, що є процесом впровадження та подальша підтримка програмного забезпечення.

Для розробки системи обрано інкрементну модель виробничого циклу.

Принципом інкрементної моделі є розширення можливостей, добудовування модулів і функцій програми. Буквальний переклад слова інкремент: «збільшення на один». Такий інкремент використовують в тому числі для позначення версій продукту.

В той час, коли каскадна модель має лише початковий та кінцевий стани, то з появою ітераційних моделей стало застосовуватися версіонування продукту.

Кожна ітерація позначається цифрою і тоді продукт після кожної ітерації, яка відбувається в ядрі, має версію з наступним номером: v1, v2, v3.

Проте коли версія в експлуатації, наступна ще в розробці, а вдосконалення замовнику і користувачеві необхідні якнайшвидше, тоді з'являються мінорні версії. Туди потрапляють зміни, що не впливають на ядро розробки і представлені як під версії 1.1,1.2,1.3 або релізи 1.1.1, 1.1.2, приклад схеми інкрементної моделі можна розглянути на рисунку 3.2.

У сукупності такі поетапні релізи призводять до повноцінної версії 2.0.

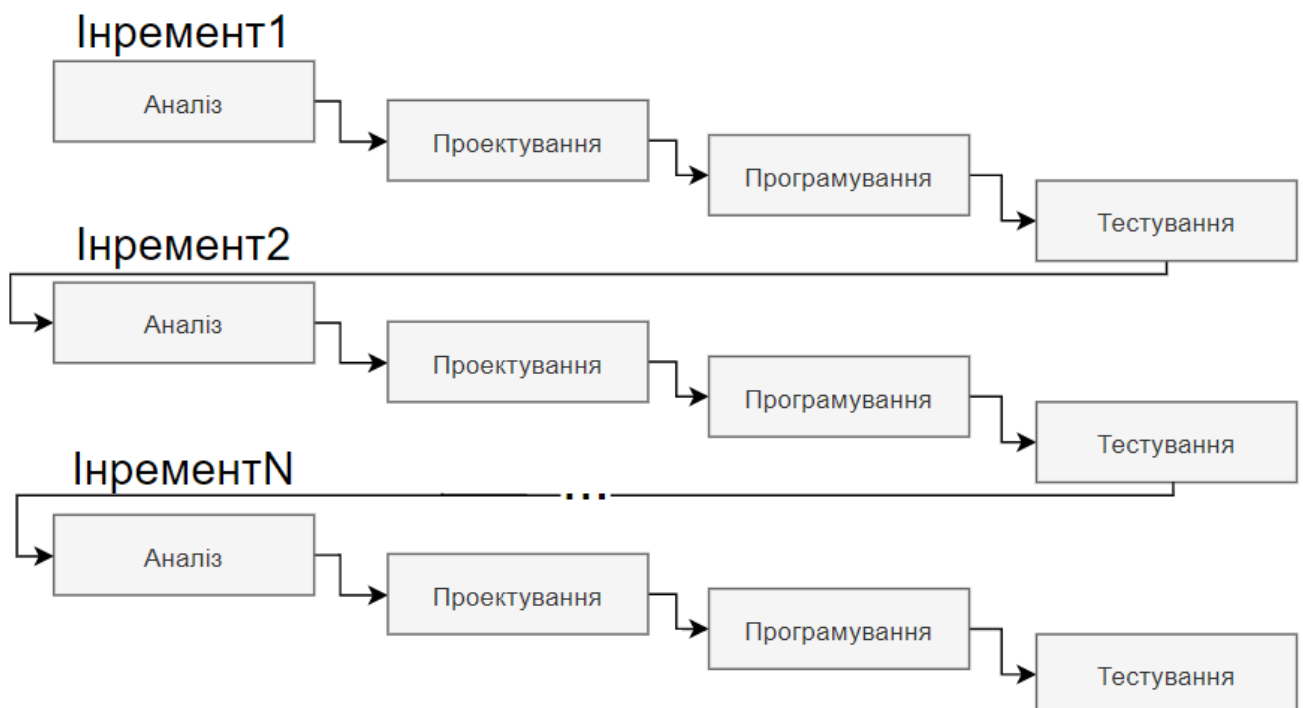


Рисунок 3.2 – Інкрементна модель

Було обрано таку модель оскільки вона дозволяє швидко реагувати на ризики та зміни у проєкті, і на кожній ітерації додається нова частина функціональності з урахуванням цих змін. Вона підходить для розробки даного проєкту, оскільки спочатку будується каркас з базових функцій, який потім дає змогу доповнювати. Така гнучкість та поділ на окремі модулі дає можливість вносити зміни на різних стадіях розробки та є досить зручною.

3.2.2 Опис функціональних користувацьких вимог

Формулювання вимог є одним з найважливіших етапів проєктування та розробки проєкту. На даній стадії необхідно чітко визначити засади на яких буде будуватися архітектура системи. Детально сформулювати всі можливості, що вона надаватиме користувачам та зобов'язання акторів для її використання та підтримки.

Дійова особа системи – це суб'єкт з реального світу, який взаємодіє з системою через сценарій використання. Акторами можуть бути ролі, виконувані людьми, такі як клієнти, видавці, продавці, оператори тощо. Дійовими особами також можуть бути зовнішні системи, тобто системи, що знаходяться поза сферою дії розроблюваного проєкту.

Актори взаємодіють з цільовою системою, надсилаючи та приймаючи інформацію через інтерфейс. У випадку з людськими суб'єктами обмін інформацією зазвичай здійснюється через пристрої введення даних, такі як клавіатури, миші або інші спеціальні пристрої. Ці суб'єкти отримують інформацію від системи через інтерфейси, такі як екрани, принтери або інші спеціальні пристрої. Зв'язок з суб'єктами, якими є зовнішні системи, зазвичай відбувається через комп'ютерну мережу. В цьому випадку комунікаційний інтерфейс складається з мережі і її протоколів [15].

Система матиме наступних акторів:

- користувач – може переглядати наступні сторінки: головну, реєстрації та входу.
- студент – може переглядати сторінки, що і користувач, а також сторінки: профіль, завдання, список викладачів, розклад.
- викладач – може переглядати ті сторінки, що і користувач, а також сторінки: профіль, завдання, скриньку, список студентів, розклад.
- адміністратор – має доступ до всіх сторінок та панелі адміністратора.

Актори мають цілі, які можна описати за допомогою діаграми прецедентів(див.рис.3.3) як користувацькі функціональні вимоги.

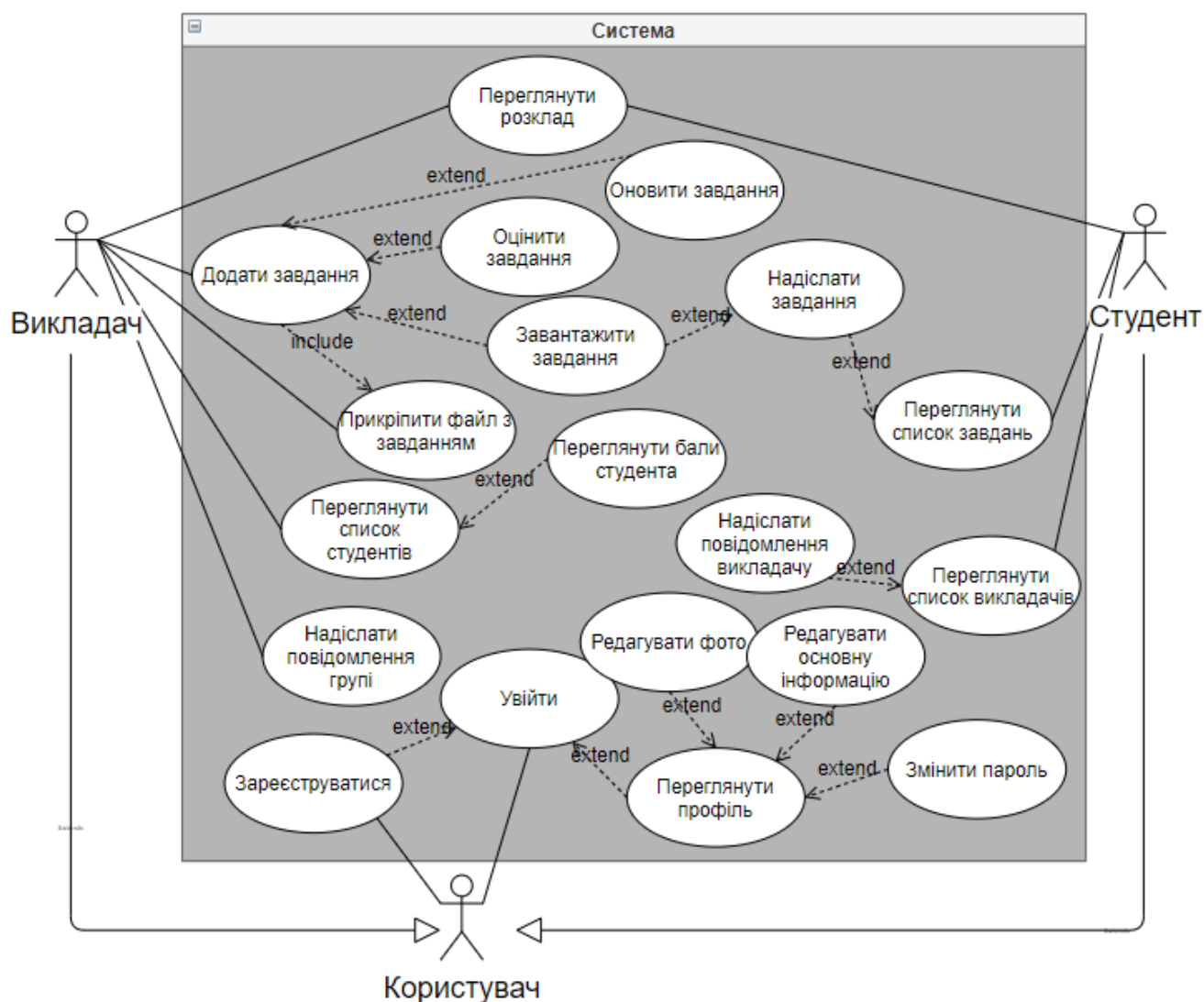


Рисунок 3.3 – Функціональні вимоги. Діаграма прецедентів

Важливим і складним етапом проектування програмного продукту є визначення реальних потреб користувача. Це пов'язано з тим, що користувач часто не в змозі передати всю повноту своїх потреб і бажань, а та інформація, яку він надає, може бути неповною, неточною та суперечливою. Важливо ретельно проаналізувати вимоги користувача і ретельно побудувати та задокументувати набір високоякісних системних вимог. У таблиці 3.1 задокументовано найважливіші вимоги до системи [16].

Таблиця 3.1 – Функціональні вимоги

№	Назва вимоги	Опис вимоги
K1	Зареєструватися	Користувач створює профіль у системі. При реєстрації надається ім'я, пошта, пароль.
K2	Увійти	Користувач проходить автентифікацію.
K3	Переглянути профіль	Користувач має можливість переглянути інформацію про себе.
K4	Редагувати фото	Користувач має змогу додати чи змінити фото.
K5	Редагувати основну інформацію	Користувач може редагувати інформацію про себе, надану при реєстрації.
K6	Змінити пароль	Користувач може змінити пароль, надавши старий.
B1	Додати завдання	Викладач може надати групі завдання для виконання, де вказує умову та прикріплює файли.
B2	Оновити завдання	Викладач може змінити інформацію в завданні.
B3	Завантажити завдання	Викладач може завантажити файл, який був прикріплений до завдання.
B4	Оцінити завдання	Викладач має можливість оцінити завдання, вказавши кількість балів за розв'язок студента та максимальну кількість балів.
B5	Переглянути список студентів	Викладач може переглядати список студентів у своїй групі й загальний список.

№	Назва вимоги	Опис вимоги
B6	Переглянути розклад	Викладач може переглядати розклад занять на наступний тиждень.
C1	Переглянути список завдань	Студент має змогу переглядати весь список завдань та виставлені бали.
C2	Завантажити завдання	Студент може завантажити файл, який був прикріплений до завдання.
C3	Надіслати завдання	Студент може прикріпити файл та надіслати викладачу.
C4	Переглянути розклад	Студент може переглядати розклад занять на наступний тиждень.

Опис нефункціональних вимог знаходиться у таблиці 3.2, наданій нижче:

Таблиця 3.2 – Нефункціональні вимоги до системи

Назва вимоги	Опис вимоги
Інтуїтивний дизайн	Користувач зможе легко та швидко користуватися функціональними можливостями сервісу.
Адаптування дизайну для мобільних пристроїв	Користувач зможе використовувати дану систему з різних девайсів. Адаптація підтримується за допомогою Bootstrap.
Конфіденційність особистої інформації	Надійне зберігання даних користувачів. Підтримується за допомогою Django.
Надійність	Правильна архітектура системи – код системи повинен бути написаний таким чином, щоб була можливість швидкого виявлення та усунення помилок

Назва вимоги	Опис вимоги
Доступність	Система повинна підтримуватись усіма популярними браузерами.

3.2.3 Структурування системи

Система має клієнт-серверну архітектуру. Для відокремлення частин застосунку, що мають різне призначення, використовуємо розбиття на відповідні підсистеми.

В попередньому підрозділі було вказано, що для вдалого структурування системи було обрано Python – фреймворк Django, оскільки він надає ряд засобів, які допомагають у швидкій розробці вебсайтів. Розробнику не потрібно створювати контролери та сторінки для адміністративної частини сайту тому, що Django має вбудований модуль для керування вмістом, який можна включити у будь-який сайт, зроблений на Django.

Архітектуру системи можна розділити на кілька відносно незалежних рівнів, завдяки чому модифікація одного із рівнів, не впливає на інші рівні, а відділення прикладних функцій від функцій управління БД спрощує розуміння всієї системи.

На серверній частині вся бізнес-логіка реалізована у вигляді окремих модулів, доступ до бази даних реалізований в моделях, а зв'язок із клієнтською частиною здійснюється через моделі.

3.2.5 Модульна декомпозиція

До появи фреймворків MVC вебпрограмування поєднувало код бази даних з серверним кодом сторінки. Фактично, для переважної більшості серверних мов код використовується як мінімум трьома мовами, наприклад Python, SQL, HTML.

Шаблон MVC був створений для відділення бізнес-логіки від представлення та є найпопулярнішою архітектурою, яка використовується сьогодні. Багато популярних фреймворків, таких як Ruby on Rails, Laravel, CodeIgniter використовують його.

Архітектура MVC ділить додаток на наступні три рівні:

- модель. Моделі представляють, як організовані дані в базі даних: шаблон MVC використовує моделі для визначення таблиць та зв'язку між ними;

- представлення. Представлення – це те, що бачить користувач, коли відвідує вебсайт. Представлення містять усю інформацію, яка у фінальному наслідку буде відправлена замовникові, тобто веббраузеру. Як правило, представленнями є HTML-документи;

контролери. Контролер контролює потік інформації. Коли користувач надсилає запит на відображення сторінки, то він передається контролеру, який використовує запрограмовану логіку, щоб вирішити, яку інформацію треба витягнути з бази даних і яку інформацію слід передати представленню. Контролер – це серце архітектури MVC, який склеює між собою моделі та представленнями (див. рис.3.4) [17].

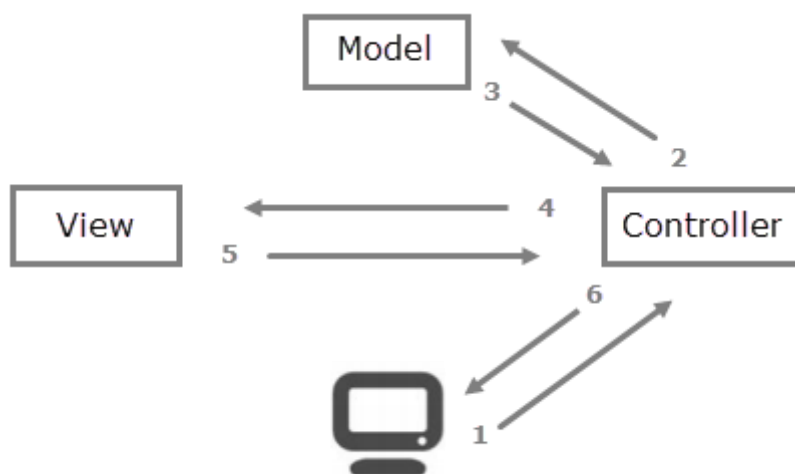


Рисунок 3.4 – Взаємодія компонентів MVC [17]

Шаблон MVC не лише допомагає створювати та підтримувати складні додатки, а також виручає, коли це прибуває у відділення турбот. Наприклад, В

компанії мережевого розвитку, існують дизайнери вебсторінок і розробники. Робота дизайнерів вебсторінок – створювати представлення. Розробники отримують представлення та об'єднують їх з моделями й контролерами.

3.2.6 Розробка бази даних

Особливість фреймворку Django полягає у тому, що для роботи з базами даних він використовує вбудовану об'єктно-реляційну проєкцію. Також фреймворк за замовчуванням створює таблиці користувачів, груп, дозволів та допоміжні для них таблиці за замовчуванням.

Призначення автоматично створених таблиць бази даних розглянуто в наступній таблиці 3.3.

Таблиця 3.3 – Призначення автоматично створених таблиць

Назва таблиці	Опис
user	Таблиця, яка зберігає наступні дані про користувача: ідентифікатор, пароль, дату останнього входу, чи є він адміністратором, викладачем чи студентом, псевдонім, ім'я, адреса електронної пошти, чи є він редактором, чи активовано його обліковий запис, дату його додавання, прізвище.
user groups	Єднальна таблиця, що створює зв'язок між таблицею користувачів та груп. Містить у собі власний ідентифікатор, ідентифікатори користувача та групи.

Назва таблиці	Опис
group	Таблиця, що зберігає дані про групи користувачів та має лише два ідентифікатор та назву групи.
group permissions	Єднальна таблиця, що створює зв'язок між таблицями груп і дозволів та вміщує у собі власний ідентифікатор, ідентифікатори групи та дозволу.
permissions	Таблиця, яка має записи про всі доступні дозволи. В ній знаходиться власний ідентифікатор, ідентифікатор типу вмісту, кодова назва та назва дозволу.
user user permissions	Єднальна таблиця, що зберігає дані про таблиці користувачів та дозволів: власний ідентифікатор, ідентифікатори користувача та дозволу.

Назва таблиці	Опис
content type	Таблиця, що зберігає дані про тип вмісту. В ній знаходиться власний ідентифікатор, позначка застосунку та модель.
admin log	Таблиця, яка зберігає записи про всі дії користувачів. Вона зберігає власний ідентифікатор, дату та час виконання дії, ідентифікатор об'єкту, представлення об'єкту, повідомлення про зроблену дію, ідентифікатори типів вмісту, користувача, який був залученим та флаг дії.
session	Таблиця, яка містить в собі інформацію про сесії користувачів. Вона зберігає ключ сесії, дані сесії користувача, дату та час закінчення терміну дії сесії.
migrations	Таблиця, в якій зберігаються дані про міграції проекту: містить у собі інформацію про команди для зміни таблиць в базі даних. Вона вміщує власний ідентифікатор міграції, назву міграції, дату та час її застосування.

При розробці бази даних таблиці мають таку ж структуру, як компоненти Model з рисунку 3.5.

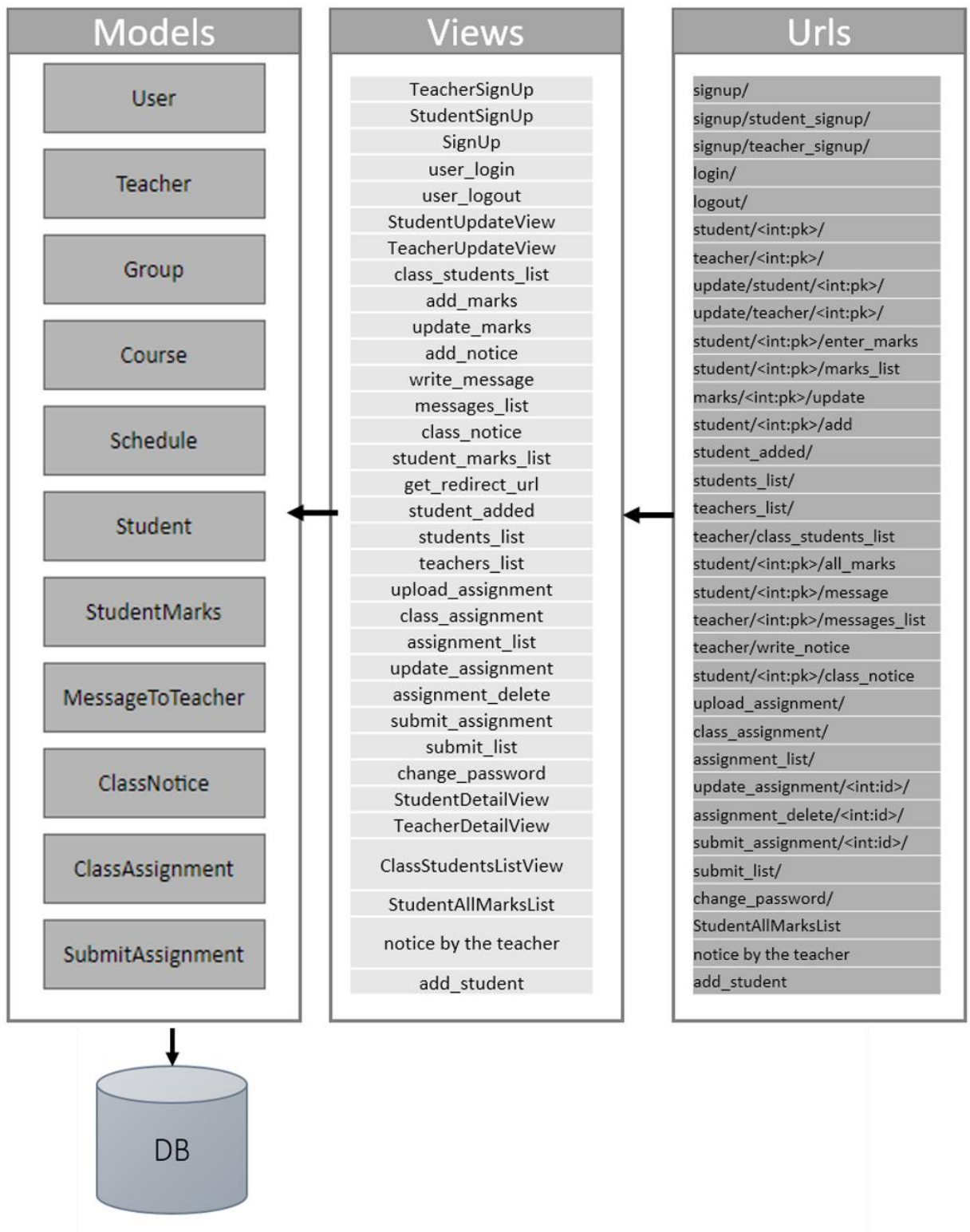


Рисунок 3.5 – Модульна декомпозиція

Отже, в базі даних системи знаходиться одинадцять таблиць, котрі відповідають моделям:

- User – зберігає дані про користувача.
- Teacher – зберігає дані про викладача.
- Student – зберігає дані про студента.
- Group – зберігає дані про групу студентів.
- Course – зберігає дані про викладача, групу студентів та предмет.
- Schedule – зберігає дані про розклад.
- StudentMarks – зберігає дані про бали студента.
- ClassNotice – зберігає дані про повідомлення від викладача курсам.
- ClassAssignment – зберігає дані про завдання з курсу.
- SubmitAssignment – зберігає дані про надіслані роботи студентів.

Розглянемо в таблиці 3.4 докладніше основні поля таблиць для роботи.

Таблиця 3.4 – Основні таблиці БД

Назва таблиці	Назва поля	Тип даних	Опис поля
User	id	INTEGER	Автоінкремент
	password	VARCHAR	Пароль користувача
	last_login	DATETIME	Остання дата входу користувача в систему
	is_superuser	BOOL	Флаг, який показує, чи є користувач адміністратором
	username	VARCHAR	Ім'я користувача

Назва таблиці	Назва поля	Тип даних	Опис поля
	last_name	VARCHAR	Прізвище користувача
	email	VARCHAR	Електронна пошта користувача
	date_joined	DATETIME	Дата реєстрації
	is_student	BOOL	Флаг, що показує, чи є користувач студентом
	is_teacher	BOOL	Флаг, що показує, чи є користувач викладачем
Teacher	user_id	INTEGER	Зовнішній ключ
	phone	INTEGER	Номер телефону викладача
	teacher_profile_pic	VARCHAR	Зображення викладача
Group	teacher_id	INTEGER	Зовнішній ключ
	name	VARCHAR	Назва групи
	speciality	VARCHAR	Спеціальність
	group_profile_pic	VARCHAR	Зображення групи
Course	teacher_id	INTEGER	Зовнішній ключ з таблиці викладача

Назва таблиці	Назва поля	Тип даних	Опис поля
	group_id	INTEGER	Зовнішній ключ таблиці групи
	name	VARCHAR	Назва курсу
	conference_programs	TEXT	Програми, що використовуютьс я для конференцій
	link	VARCHAR	Посилання на програми
Schedule	course_id	INTEGER	Зовнішній ключ таблиці курсу
	week_number	INTEGER	Номер тижня пари
	weekday	INTEGER	День тижня, коли відбувається пара
	date	DATETIME	Дата та час пари
Student	user_id	INTEGER	Зовнішній ключ користувача
	group_id	INTEGER	Зовнішній ключ таблиці групи
	phone	INTEGER	Номер телефону студента
	student_profile_pic	VARCHAR	Зображення студента
StudentMarks	course_id	INTEGER	Зовнішній ключ таблиці курсу

Назва таблиці	Назва поля	Тип даних	Опис поля
	student_id	INTEGER	Зовнішній ключ студента
	subject_name	VARCHAR	Назва теми
	marks_obtained	INTEGER	Отриманий бал студентом
	maximum_marks	INTEGER	Максимальний бал за завдання
MessageTo Teacher	student_id	INTEGER	Зовнішній ключ студента
	teacher_id	INTEGER	Зовнішній ключ викладача
	created_at	DATETIME	Дата та час створення повідомлення
	message	VARCHAR	Текст повідомлення викладачеві
	message_html	TEXT	Повідомлення в html розмітці
ClassNotice	teacher_id	INTEGER	Зовнішній ключ викладача
	group_id	INTEGER	Зовнішній ключ групи
	created_at	DATETIME	Дата створення повідомлення
	message	VARCHAR	Текст повідомлення

Назва таблиці	Назва поля	Тип даних	Опис поля
	message_html	TEXT	Повідомлення в html розмітці
Class Assignment	group_id	INTEGER	Зовнішній ключ групи
	teacher_id	INTEGER	Зовнішній ключ викладача
	created_at	DATETIME	Дата створення завдання
	assignment_name	VARCHAR	Назва завдання
	assignment	TEXT	Прикріплений файл для виконання
Submit Assignment	student_id	INTEGER	Зовнішній ключ студента
	teacher_id	INTEGER	Зовнішній ключ викладача
	created_at	DATETIME	Дата надсилання завдання студентом
	submitted_assignment_id	INTEGER	Зовнішній ключ завдання
	submit	VARCHAR	Прикріплений файл

На рисунку 3.6 представлено діаграму, що ілюструє відношення таблиць і взаємодію бази даних спроектованого сервісу для дистанційного навчання.

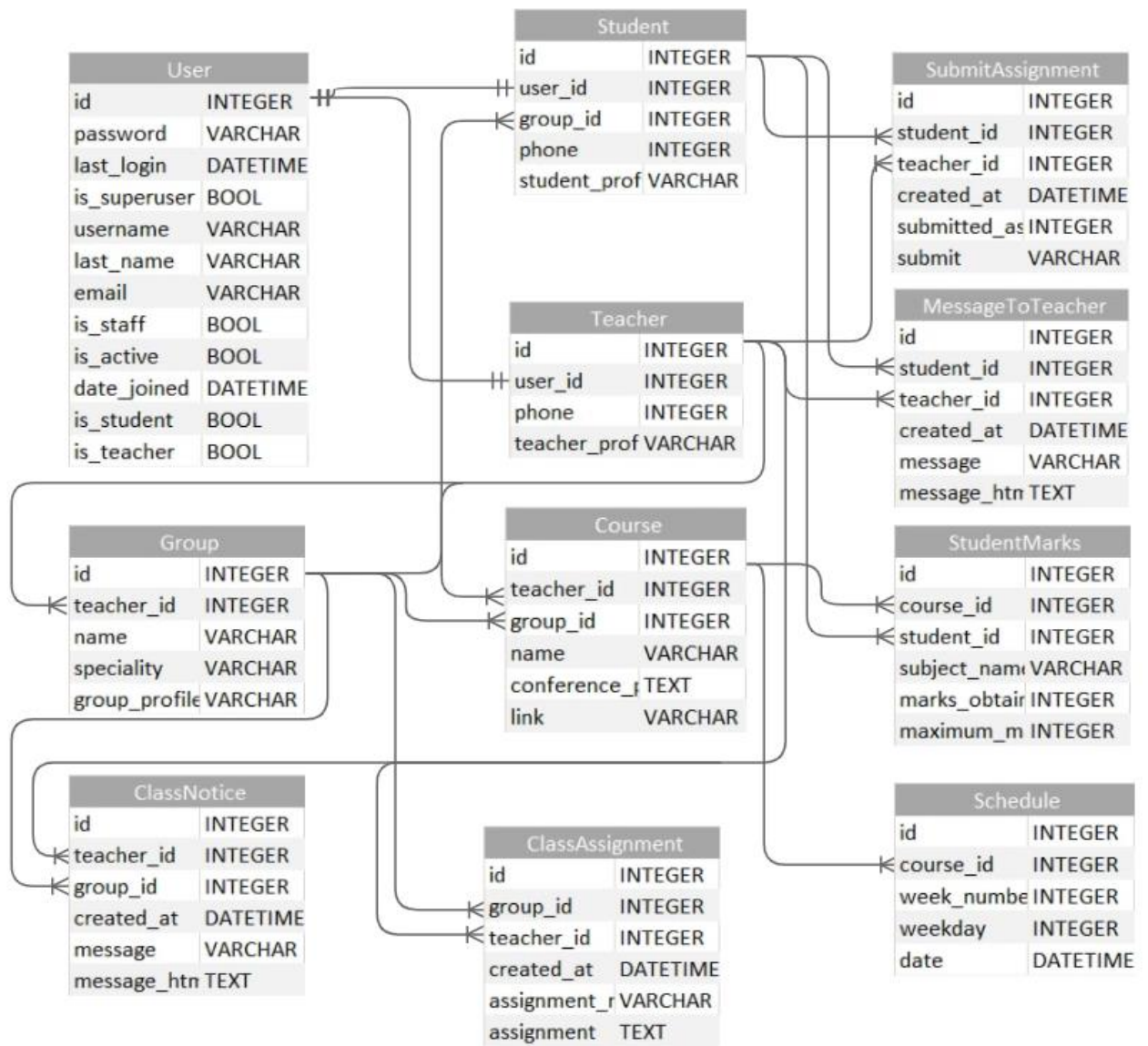


Рисунок 3.6 – ER-діаграма

3.3 Робота користувача з системою

Користувачі мають доступ до функціонала системи через сторінку у веббраузері. В загальному вигляді ресурс надає користувачам наступні можливості:

а. для всіх користувачів:

- перегляд та редагування профілю;
- перегляд розкладу.

б. для викладачів:

- надсилання та редагування своїх завдань;
- доступ до перегляду студентів на курсі та загалом;
- перегляд оцінок студентів;
- редагування оцінок курсу та окремих студентів;
- отримання та перегляд повідомлень від студентів;
- надсилання повідомлень групі або курсу.

с. для студентів:

- надсилання виконаних завдань;
- надсилання повідомлення викладачу;
- перегляд доступних завдань у викладачів;
- перегляд отриманих оцінок;
- отримання повідомлень курсу від викладача.

Головна сторінка, сторінки реєстрації та авторизації доступні всім незареєстрованим користувачам.

Вимога К1: Реєстрація. Користувач переходить з головної сторінки, на сторінку реєстрації. На сторінці обирає яким чином зареєструватися: як учень чи як викладач, на наступній сторінці користувач вводить особисту інформацію, що можна побачити на діаграмі послідовності (див.рис.3.7), яка описує реєстрацію користувача.

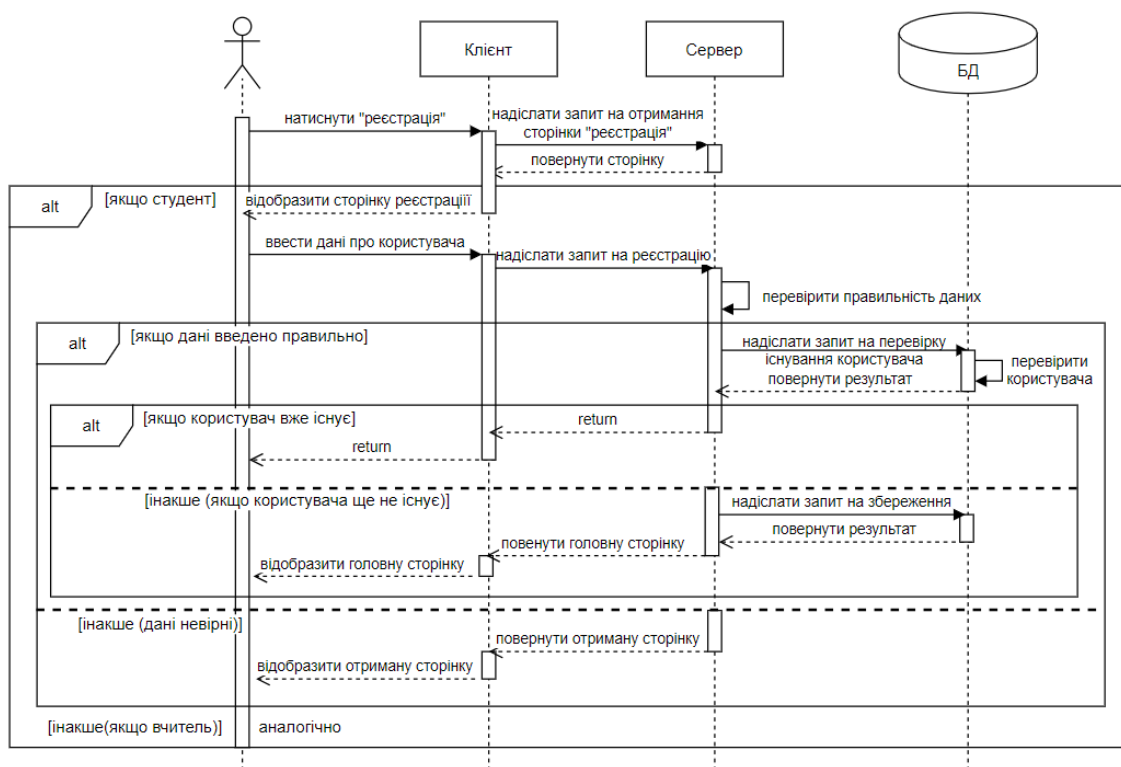


Рисунок 3.7 – Реєстрація. Діаграма послідовності

Вимога К2: На діаграмі послідовності (див.рис.3.8) можна побачити, як користувач проходить автентифікацію і повертається на головну сторінку. Також в залежності від ролі, користувач має власну навігаційну панель.

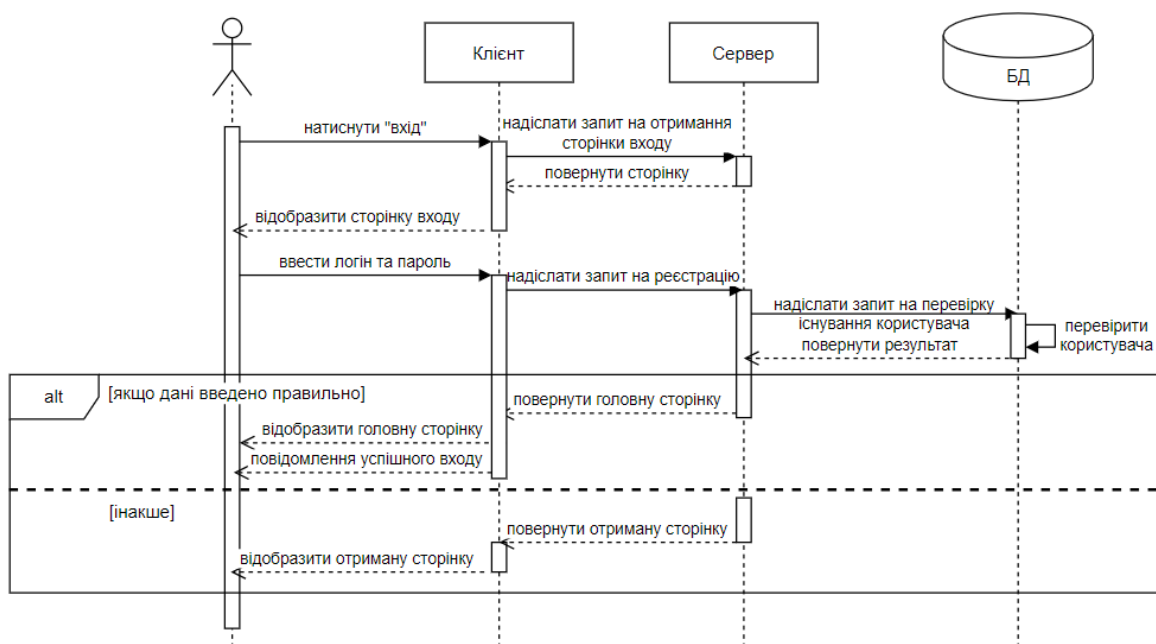


Рисунок 3.8 – Автентифікація. Діаграма послідовності

Вимога К3: Користувач заходить в особистий кабінет, який може переглядати та змінювати (Вимоги К4-К6), що зображено на наступній діаграмі(див.рис.3.9).

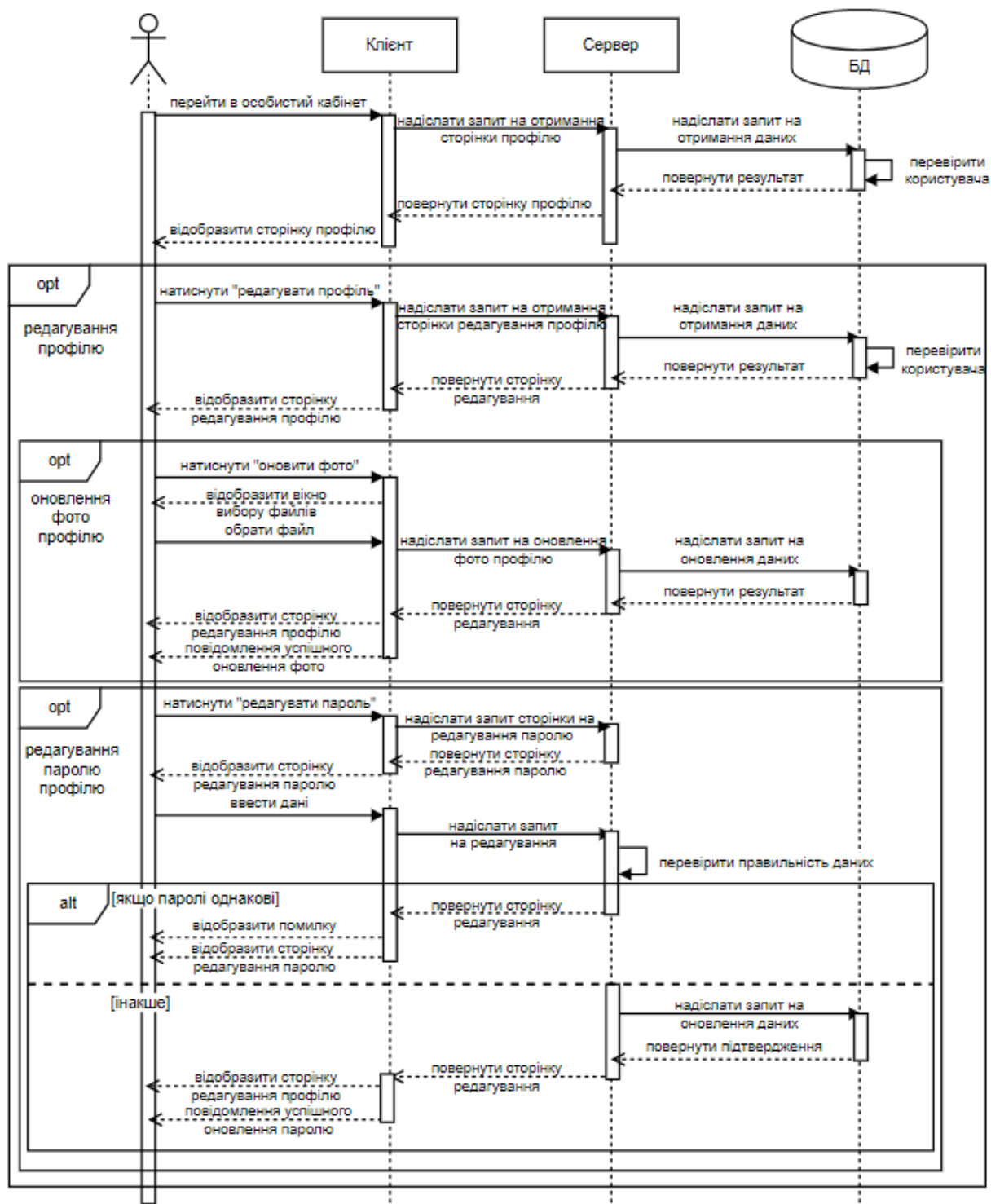
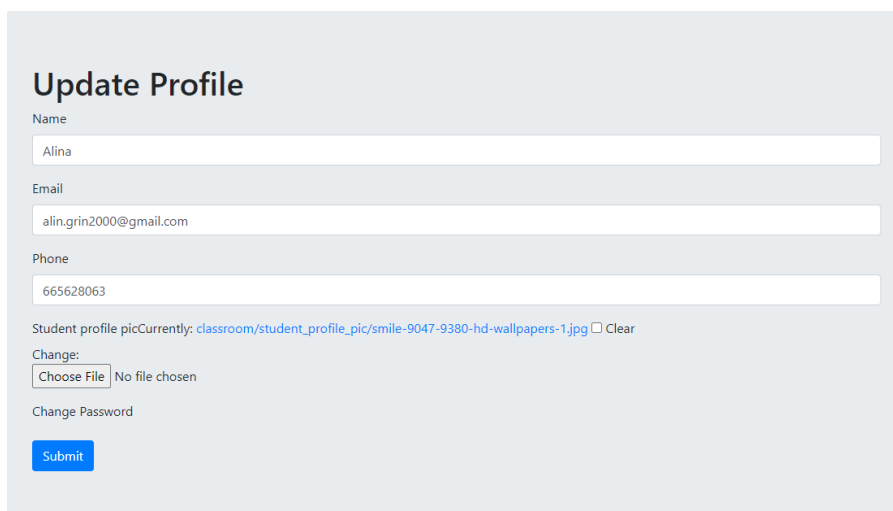


Рисунок 3.9 – Особистий кабінет. Діаграма послідовності

Вимоги К4, К5: Також на рисунку 3.9 можна побачити більш детально описані вимоги. Користувач переходить на сторінку редагування(див.рис.3.10), де він може редагувати основну інформацію про себе, додавати, змінювати та видаляти фото.



Update Profile

Name
Alina

Email
alin.grin2000@gmail.com

Phone
665628063

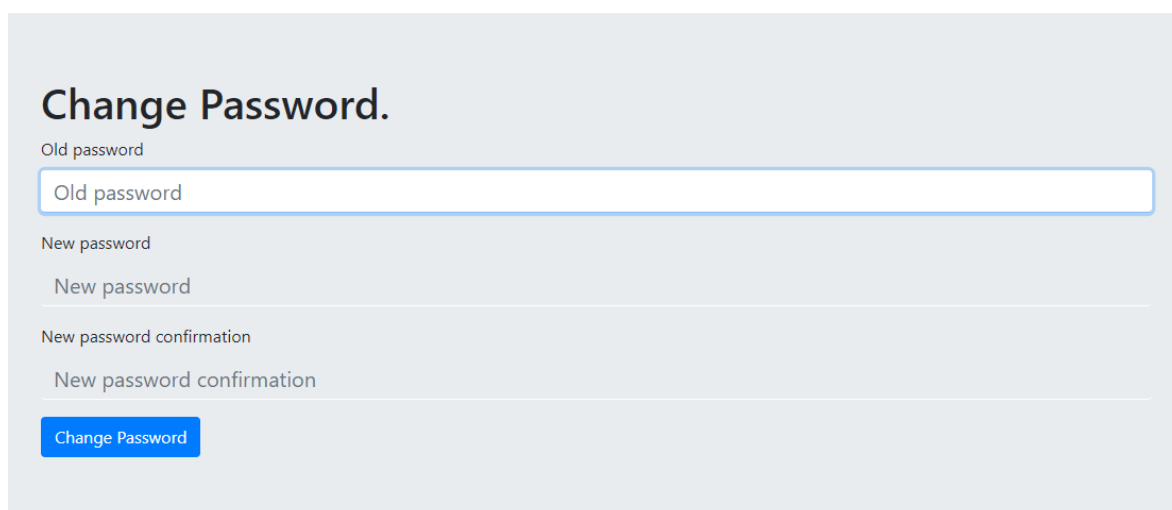
Student profile picCurrently: [classroom/student_profile_pic/smile-9047-9380-hd-wallpapers-1.jpg](#) Clear

Change:
 No file chosen

[Change Password](#)

Рисунок 3.10 – Редагування профілю

Вимога К6: Як було згадано та вказано на попередній діаграмі послідовності, користувач має змогу змінити пароль на новий, інтерфейс можна побачити на рисунку 3.11.



Change Password.

Old password
Old password

New password
New password

New password confirmation
New password confirmation

Рисунок 3.11 – Зміна паролю

Вимоги В1-В4: Викладач має змогу створювати завдання, оновлювати інформацію(див.рис.3.12) та завантажувати файл з завданням зі сторінки.

#	Assignment Name	Uploaded Date	Download	Update	Delete
1	task1	May 28, 2021, 12:48 p.m.	Download	Update	Delete

Рисунок 3.12 – Завантажені завдання

Вимога В5: За допомогою наступної діаграми послідовності (див.рис.3.13) можна зрозуміти, що викладач може переглядати список студентів та відстежувати бали, що показано на рисунку 3.14.

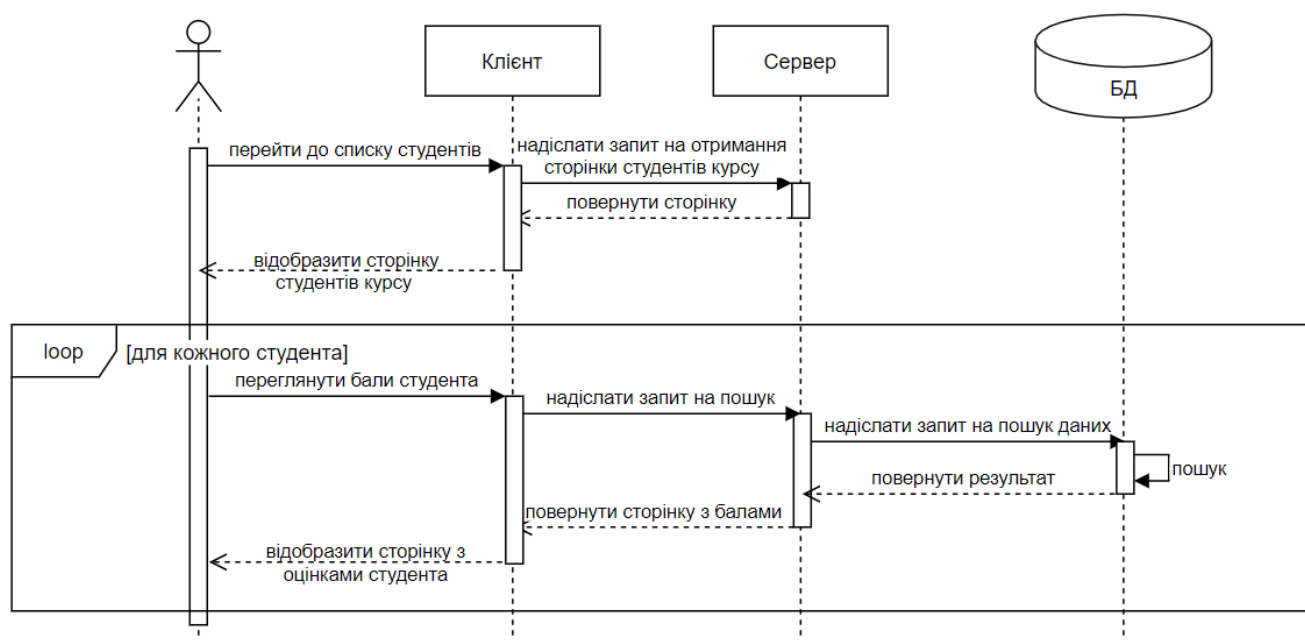


Рисунок 3.13 – Перегляд списку студентів. Діаграма послідовності

Sr. no.	Subject	Teacher Name	Marks Obtained	Max Marks
1	Maths	Zinoviya Sokolov	5	10
2	OOP	Qwerty Qw	3	5

Рисунок 3.14 – Перегляд балів

Вимоги С1-С2: Студент може переглядати список завдань, завантажувати файли та відстежувати бали, що зображено на рисунку 3.15.

#	Assignment Name	Uploaded By	Uploaded Date	Download	Status
1	task1	Qwerty Qw	May 28, 2021, 12:48 p.m.	Download	Submit

Рисунок 3.15 – Перегляд списку завдань

Вимога С3: Студент може надсилати завдання, під'єднуючи додаткові файли(див.рис.3.16).

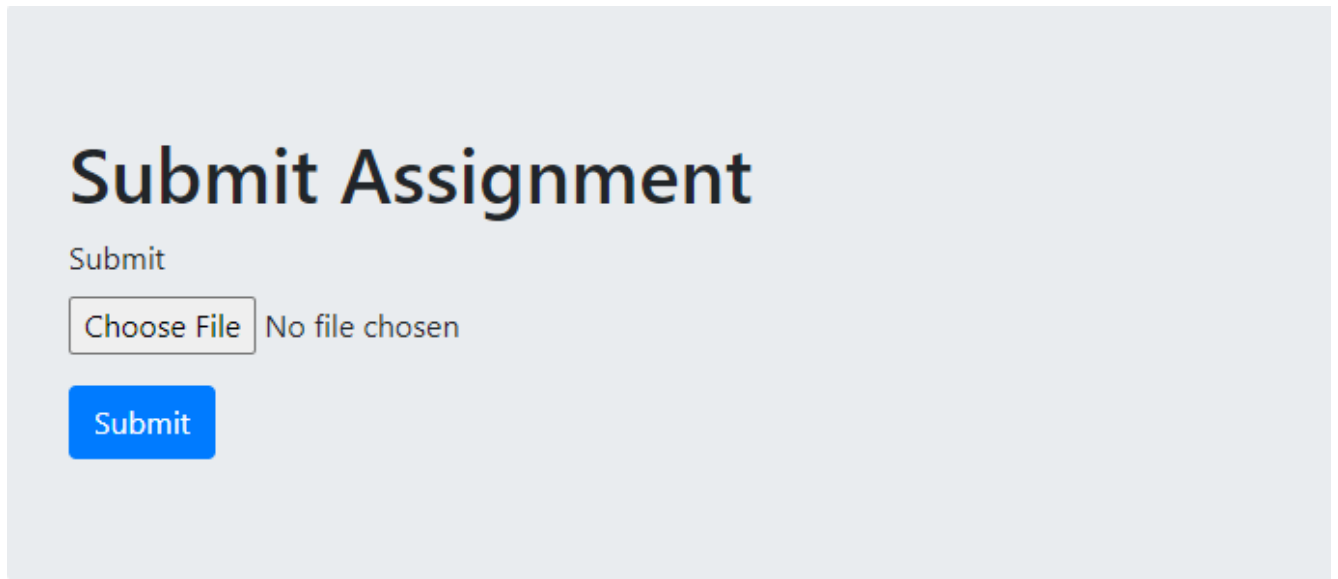


Рисунок 3.16 – Надсилання завдання викладачеві

Вимога С4: Студент може переглядати список викладачів, де також знаходиться додаткова інформація про кожного з них, та надсилати їм повідомлення. Процедура надсилання повідомлення зображена на рисунку 3.17.

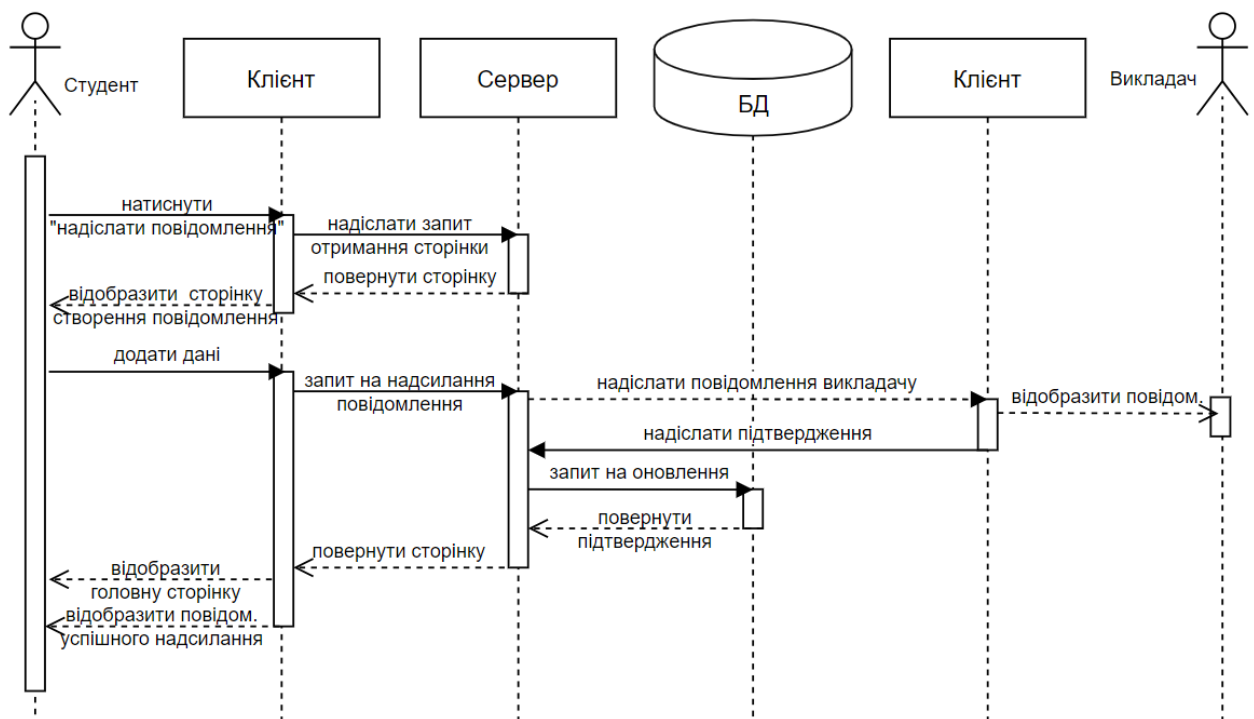


Рисунок 3.17 – Надсилання повідомлення викладачу

ВИСНОВКИ

При виконанні даної дипломної роботи було проведено огляд системи дистанційного навчання в цілому, розглянуто переваги та недоліки дистанційного навчання. Також було проведено розгляд, порівняння та аналіз окремих сучасних систем дистанційної освіти, принципи дистанційного навчання, а також методи та засоби його реалізації.

Було розроблено систему, яка допомагає організувати навчальний процес, встановлює засоби спілкування між викладачем та студентом. Перед цим проведено аналіз зібраних даних про існуючі рішення та створення вимог, на які можна було орієнтуватися при розробці системи.

Розроблене програмне забезпечення готове до застосування та відповідає усім визначеним вимогам. Також, існує можливість розширення функціоналу застосунку за допомогою модулів, які можна додати пізніше.

В ході роботи було вирішено усі поставлені задачі, а саме: розглянуто та проаналізовано необхідні теоретичні матеріали, розроблено вимоги та спроектовано базу даних. Пророблена праця дозволила створити власне рішення для організації навчального процесу, яке зможе використовуватися у навчальних закладах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Gary A. Berg. Distance learning: Additional Information [Електронний ресурс] / Gary A. Berg, Michael Simonson. – 2019. – Режим доступу до ресурсу: <https://www.britannica.com/topic/distance-learning>.
2. Жевакіна Н. З ІСТОРІЇ ДИСТАНЦІЙНОЇ ОСВІТИ. ВІСНИК ЛЬВІВ [Електронний ресурс] / Наталія Жевакіна // УН-ТУ VISNYK LVIV UNIV. Серія педагогічна.. Вип.17. – 2003. – Режим доступу до ресурсу: http://old.franko.lviv.ua/Pedagogika/periodic/visnyk/17/19_zhevakina.pdf.
3. Tamm S. The History of E-Learning [Електронний ресурс] / Sander Tamm. – 2019. – Режим доступу до ресурсу: <https://e-student.org/history-of-e-learning/>.
4. Tavukcu T. General overview on distance education concept [Електронний ресурс] / T. Tavukcu, I. Arapa, D. Özcan – Режим доступу до ресурсу: <https://www.sciencedirect.com/science/article/pii/S1877042811009505>.
5. Survey on online and distance learning [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.schooleducationgateway.eu/en/pub/viewpoints/surveys/survey-on-online-teaching.htm>.
6. МЕТОДИКА ПРИМЕНЕНИЯ ДИСТАНЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ ПРЕПОДАВАТЕЛЯМИ ВУЗА [Електронний ресурс] / Маматов А. В., Немцев А. Н., Клепикова А. Г., Штифано А. І.. – 2006. – Режим доступу до ресурсу: <https://core.ac.uk/download/pdf/187751783.pdf>.
7. Гультьяев А. К. Macromedia Authorware 6.0 Разработка мультимедийных учебных курсов [Електронний ресурс] / Гультьяев А. К. // СПб.: КОРОНА принт. – 2007. – Режим доступу до ресурсу: <http://www.padaread.com/?book=207515&pg=400>.
8. Порівняння Moodle та Google Classroom в Україні [Електронний ресурс] – Режим доступу до ресурсу:

- <https://trends.google.com/trends/explore?cat=74&date=2020-03-21%202021-04-30&geo=UA&q=Moodle,Google%20Classroom>.
9. Динаміка популярності Moodle та Google Classroom у світі [Електронний ресурс] – Режим доступу до ресурсу: <https://trends.google.com/trends/explore?cat=74&date=2020-03-21%202021-04-30&q=Moodle,Google%20Classroom>.
 10. Guido van Rossum. The Python Language Reference Manual [Електронний ресурс] / Guido van Rossum. – 2002. – Режим доступу до ресурсу: <http://ft-sipil.unila.ac.id/dbooks/Python%20Reference%20Manual.pdf>.
 11. TIOBE Index for May 2021 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tiobe.com/tiobe-index/>.
 12. Переваги MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javatpoint.com/mysql-features>.
 13. Переваги та недоліки JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://www.freecodecamp.org/news/the-advantages-and-disadvantages-of-javascript/>.
 14. Chacon S. ProGit [Електронний ресурс] / S. Chacon, B. Straub. – 2014. – Режим доступу до ресурсу: <https://git-scm.com/book/ru/v2>.
 15. Raul Sidnei Wazlawick. Object-Oriented Analysis and Design for Information Systems Modeling with UML, OCL, and IFML [Електронний ресурс] / Raul Sidnei Wazlawick. – 2014. – Режим доступу до ресурсу: https://ebrary.net/73266/computer_science/object-oriented_analysis_and_design_for_information_systems_modeling_with_uml_ocl_and_ifml..
 16. Parker J. Business, User, and System Requirements [Електронний ресурс] / John Parker. – 2012. – Режим доступу до ресурсу: <https://enfocussolutions.com/business-user-and-system-requirements/>.
 17. MVC Pattern and Django [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://overiq.com/django-1-10/mvc-pattern-and-django/>.