МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ САДІВНИЦТВА

Факультет економіки і підприємництва

Кафедра інформаційних технологій

Допущено до захисту
"_____" _____ 2023 р.
Зав. кафедри, к.т.н., доцент
_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

РОЗРОБКА ВЕБ-МОДУЛЮ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ ПРИ УПРАВЛІННІ ВИКОНАННЯМ ПРОЕКТУ

Виконав – студент 2 курсу, групи 21к-кн
освітньої програми "Комп'ютерні науки"
спеціальності 122 Комп'ютерні науки
Віхренко Назар Віталійович

Науковий керівник – к.е.н., доцент
Концеба Сергій Михайлович

Умань – 2023

ЗМІСТ

ВСТУП 4

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 6

1.1 Огляд останніх досліджень і публікацій 6

1.2 Аналіз програмних продуктів – аналогів 8

1.3 Постановка задачі 16

РОЗДІЛ 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ 21

2.1 Архітектура додатку 21

2.2 Структурно-функціональне моделювання роботи додатку 22

2.3 Проектування бази даних 26

2.4 Моделювання варіантів використання 27

2.5 Моделювання діаграм діяльності 30

2.6 Моделювання діаграми класів 34

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ДОДАТКУ 36

3.1 Структура програмного додатку 36

3.2 Програмна реалізація 37

3.3 Використання програмного додатку 51

ВИСНОВКИ 60

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 61

ДОДАТКИ 63

ВСТУП

Управління проектами – діяльність, яка поєднує сучасні технології та знання з практичними навичками. Управління проектом є процес, протягом якого очікується ефективна робота з метою отримання максимального ефекту від реалізації. Управління проектами – це методологія планування та використання ресурсів протягом його життєвого циклу.

Для вдалого завершення проекту необхідна не тільки команда кваліфікованих співробітників, але і інструменти, за допомогою яких можна фіксувати виконані задачі та вдало управляти ресурсами. Серед чітко орієнтованих є рішення, які підходять для управління проектами різних видів. Проте, зазвичай, вони є дороговартісними і для компаній, які тільки знаходяться в стані становлення, це суттєві витрати.[14]

Тому метою даної роботи є розробка власної інформаційної системи для управління виконанням проектів.

Для досягнення поставленої мети потрібно виконати такі задачі.

1. Провести аналіз предметної області, а саме:

— проаналізувати теперішній стан використання інформаційних технологій у області проектного менеджменту;

— провести аналіз існуючих аналогів інформаційних систем для допомоги в управлінні проектів.

2. Визначити методи дослідження інструментів для швидкої та надійної роботи при створенні системи та спланувати роботу над проектом.

3. Провести проектування системи.

— проаналізувати та обрати найкращий тип архітектури;

— побудувати концептуальну діаграму бізнес логіки проекту та провести декомпозицію процесу роботи системи;

— побудувати діаграму варіантів використання для деталізації функціональних етапів та залежностей;

— побудувати моделі діяльності основних модулів проекту;

— побудувати діаграму послідовності для деталізації передачі сигналів з одного процесу до іншого;

— спроектувати базу даних;

— побудувати діаграму класів для майбутнього проектування.

4. Виконати розробку інформаційної системи.

Практичне значення роботи полягає в тому, що отримано зручний продукт для автоматизації процесу управління процесом вокнання проектів.

Результати роботи доповідалися на щорічній науковій конференції «IMA-2020» [22].

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Предметна область - сукупність послуг та інструментів, створення яких має бути забезпечене в рамках здійснюваного проекту.

Питанням про те, чи варто застосовувати в компанії проектний менеджмент, задаються багато керівників. Цей інструмент вже впроваджено у багатьох сферах бізнесу, і зараз необхідно розуміти, як застосовувати технологію виходячи з планів, цінностей та завдань компанії, які інструменти використовувати для цього [12,13].

На даний момент є великий вибір методологій управління виконанням проектів.

Всі доступні типи менеджменту можна розподілити на такі групи:

— традиційні послідовні методології (CPM, Waterfall);

— змінні методології(ECM, XPM);

— класичні методології (PMBOK);

— гнучкі методології (Agile, Scrum, Kanban і ін.);

— процесні методології (Lean, Process-Based Project Management);

— інші(Prince2, PRiSM).

Після вдало обраної методології, можна перейти до вибору інструментів для управління виконанням проекту. Обираючи такий інструмент користувач шукає продукт,який міг би складати діаграми Ганта, будувати графіки, розподіляти повноваження і завдання виконавцям з різними ролями.[9][10]

Ефективність проектного менеджменту була підтверджена багаторічним досвідом світових бізнесгігантів. Отже, необхідність обрання цієї форми управління стає все більш очевидною. Серед тих хто використовують цей підхід є передові компанії в самих різних напрямках:

— УКРСИББАНК BNP PARIBAS;

— ВОЛЯ КАБЕЛЬ;

— РАЙФФАЙЗЕН БАНК АВАЛЬ;

— МЕДІАХОЛДИНГ «ВІСТІ»;

— CORUM GROUP;

— COMFY TRADE;

— TERRA FOOD;

— МТС.

Практичне значення роботи полягає у розробці інструментарію для ефективного управління проектами. Розроблена веб-система є повноцінний інструмент для планування проектів та управління виконанням.[15]

Провівши аналіз існуючих інструментів для контролю управлінням виконання проектів та загальної предметної області було визначено такі характеристики:

— Взаємодія користувача з системою;

— Контроль процесу виконання проектів за допомогою графіків;

— Можливість управління командами проекту;

— Можливість зміни користувачем власних даних;

— Можливість завантаження файлів до інформації про проект;

— Наявність Kanban дошки як особистої для користувача, так і загальної для проекту;

— Можливість планування майбутніх задач;

— Можливість контролю виконанням користувачів відповідних задач.

На даний момент більшість крупних компаній використовують проектний менеджмент як засіб для ефективного планування, управління проектами та їх виконанням [16].

Тому для допомоги менеджерам, керівникам проектів з роботою створюється велика кількість інструментів, але всі ці засоби або не мають центрального модулю для поєднання в одну систему, або мають певні недоліки у використанні [11,18].

З цього можна зробити висновок, що створення такої системи має перспективи у розвитку та масштабуванні у майбутньому. Метою даної

роботи є, розробка інформаційної системи, яка буде повністю відповідати предметній області.

Вирішувати всі необхідні проблеми та відповідати поставленим вимогам. Система буде слугувати універсальним інструментом для управління виконанням проектів, які створювались з використанням різних методологій та відносяться до різних галузей [20,17].

1.2 Аналіз програмних продуктів – аналогів

Системи для управління проектами поділяються на спеціалізовані та неспеціалізовані, в залежності від призначень та набору інструментів [1].

Неспеціалізовані системи легші у освоєнні для базового користувача, тому що мають базовий інтерфейс с базовим функціоналом, але якщо потрібні більш широкі функції, то потрібно дивитись у бік спеціалізованих сервісів. [3,19].

Список систем управління проектами та їх короткий опис:

— Microsoft Project (або MSP),

— Spider Project,

— Trello,

— Jira,

— Bitrix24,

— ProjectMate,

— OpenPlan.

Microsoft Project – це доступний комплекс управління цілісними проектами та окремими внутрішніми процесами, сервіс пропонує оптимізувати бізнес-процеси з метою автоматизації та створення деяких видів діяльності. Реалізовані інструменти вже не одне десятиріччя привертають увагу як великих міжнародних організацій, так і представників місцевого малого та середнього бізнесу.

MS Project дуже простий в ідейному плані. Він оперує трьома сутностями завдання, ресурси, календар і зв'язки між ними. По суті - це база даних, призначена

для користувача, інтерфейс для створення і редагування сутностей і мінімальна, досить проста автоматизація [1].

Trello – зручна система управління проектами, простий і універсальний онлайн-інструмент у вигляді веб-версії trello.com , мобільних додатків для Android і IOS, а також розширення для браузера Google Chrome. Сервіс реалізований за принципом канбан-дошки: користувач створює дошку, на ній списки, всередині яких картки. Простота, універсальність і багатофункціональність. За це його обирають професійні керівники проектів і продукт-менеджери. У цій системі управління проектами за замовчуванням немає діаграми Ганта, неможливо оцінити прогрес в часі[2].

Тарифи Trello:

1 Безкоштовний:

— необмежену кількість персональних дошок, списків і карток в дошці;

— підключення 1 плагіна до дошки;

— вкладення в картку - до 10 Мб

— 10 групових дошок.

2 Business Class - $ 9.99 в місяць за 1 користувача:

— необмежену кількість персональних дошок, списків і карток в дошці;

— необмежену кількість плагінів до дошки;

— вкладення в картку - до 250Мб;

— необмежену кількість групових дошок;

— додатковий рівень доступу для користувачів без можливості редагувати, тільки коментувати;

— колекції дошок;

3 Enterprise - $ 20.83 в місяць за 1 користувача:

— необмежену кількість персональних дошок, списків і карток в дошці;

— необмежену кількість плагінів до дошки;

— вкладення в картку - до 250Мб;

— необмежену кількість групових дошок;

— додатковий рівень доступу для користувачів без можливості редагувати,

тільки коментувати;

— колекції дошок;

— додаткові можливості адміністрування дошок, прав доступу і індивідуальна адаптація співробітників.

Jira – це онлайн-сервіс або система управління проектами з веб-версією і мобільними додатками для Android і iOS. Заснований на методології Agile, являє собою дошку з картками і є найпопулярнішим серед команд розробки. Jira належить компанії Atlassian, яка також володіє Trello. [2]

Плюси Jira:

— простий і зрозумілий інтерфейс;

— канбан або scrum-дошки;

— можливість створення звітів;

Недоліки Jira:

— спочатку ця система управління проектами розрахована на спеціалізовані

— команди розробників, включає в себе можливості прив'язки до коду і не завжди необхідні для інших проектів опції.

Тарифи Jira:

Вартість залежить від кількості користувачів.

— Фіксований тариф для команди до 10 чоловік - $ 10 в місяць;

Гнучка модель:

— перші 11-100 користувачів - $ 7 на місяць за людину;

— наступні 150 користувачів - $ 5 в місяць за людину;

— після 250 чоловік - $ 1.10 в місяць за кожного наступного користувача.

Бітрікс24 – це найпопулярніший на пострадянському просторі CPM-сервіс, схожий з Мегаплан, дуже потужна система управління проектами.

Сповідує принцип одного вікна, включає в себе практично все, що може знадобиться бізнесу[2].

Переваги Бітрікс24:

— можливість налаштування наскрізний аналітики;

— можливість налаштування воронки продажів;

— комфортна інтеграція корпоративного порталу;

— месенджер і чат-боти;

— інтеграція з 1С і 1С-Бітрікс.

Недоліки Бітрікс24:

— багато в чому надлишковий інтерфейс;

— вимагає ресурсів на навчання та впровадження[2].

ProjectMate – це Російська PSA-система автоматизованої роботи. Дослідження модуля управління має максимум функцій, необхідних у компаніях сфери консультаційних служб (юристи, адвокати та інші) [3].

На рисунку 1.1 зображено приклад інтерфейсу програми Spider Project.

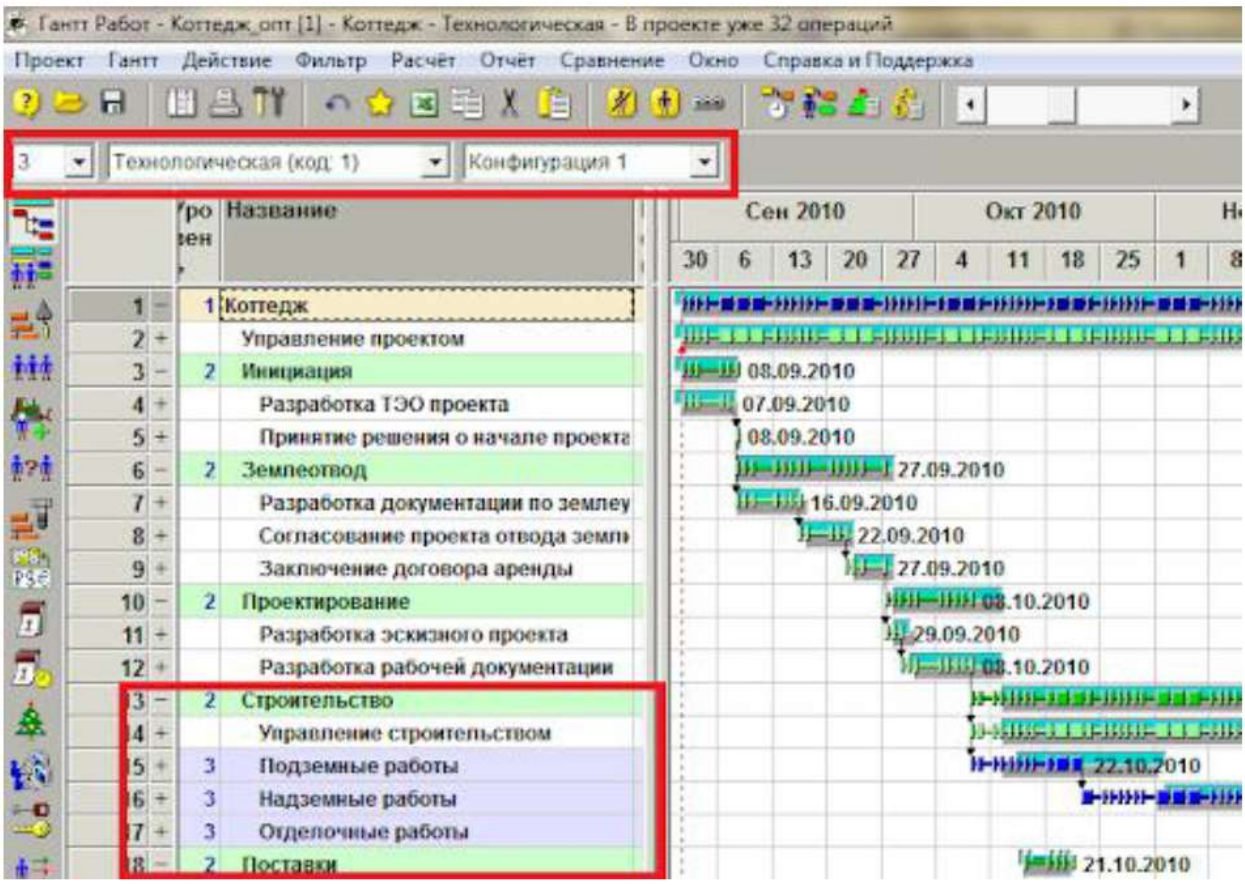


Рис. 1.1 Інтерфейс Spider Project

На рисунку 1.2 зображено приклад інтерфейсу програми Trello.

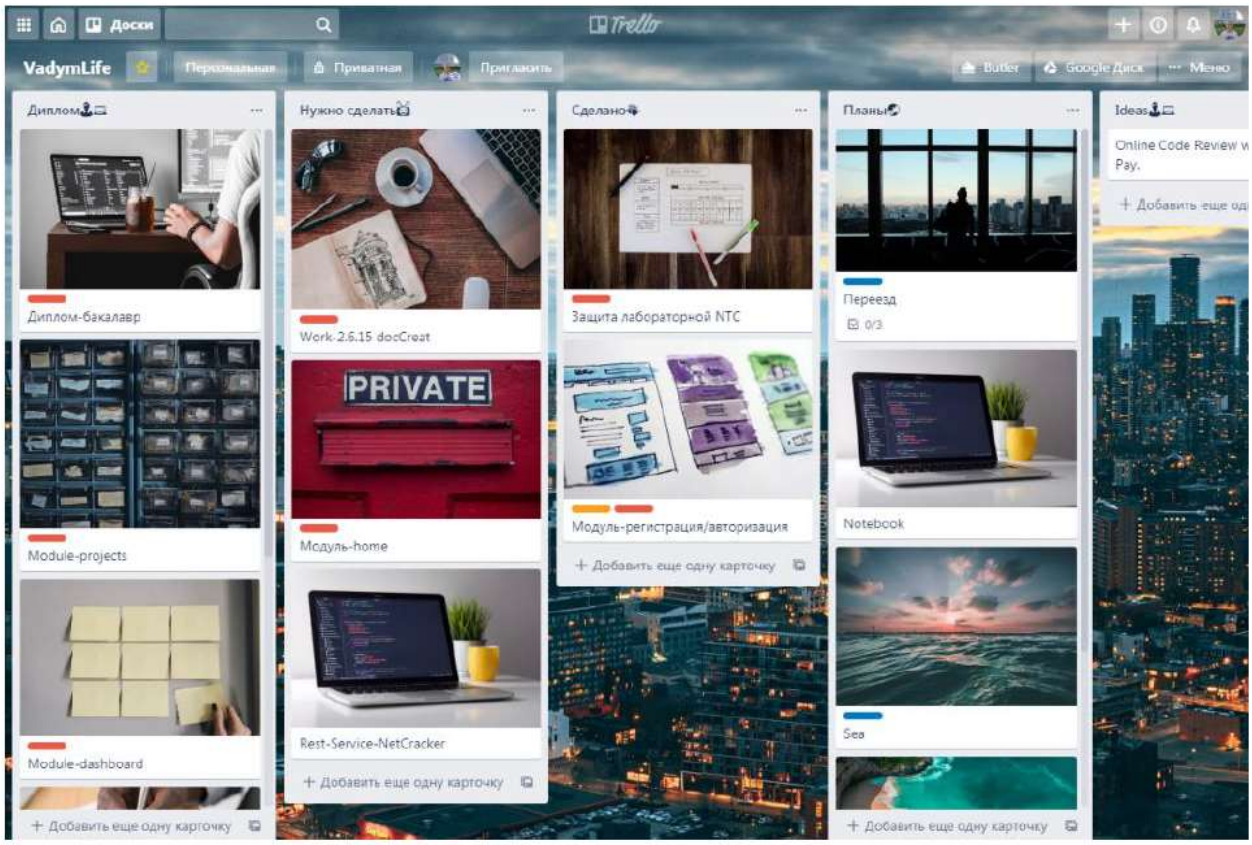


Рис. 1.2 Інтерфейс Trello

На рисунку 1.3 зображено приклад інтерфейсу програми Bitrix24.

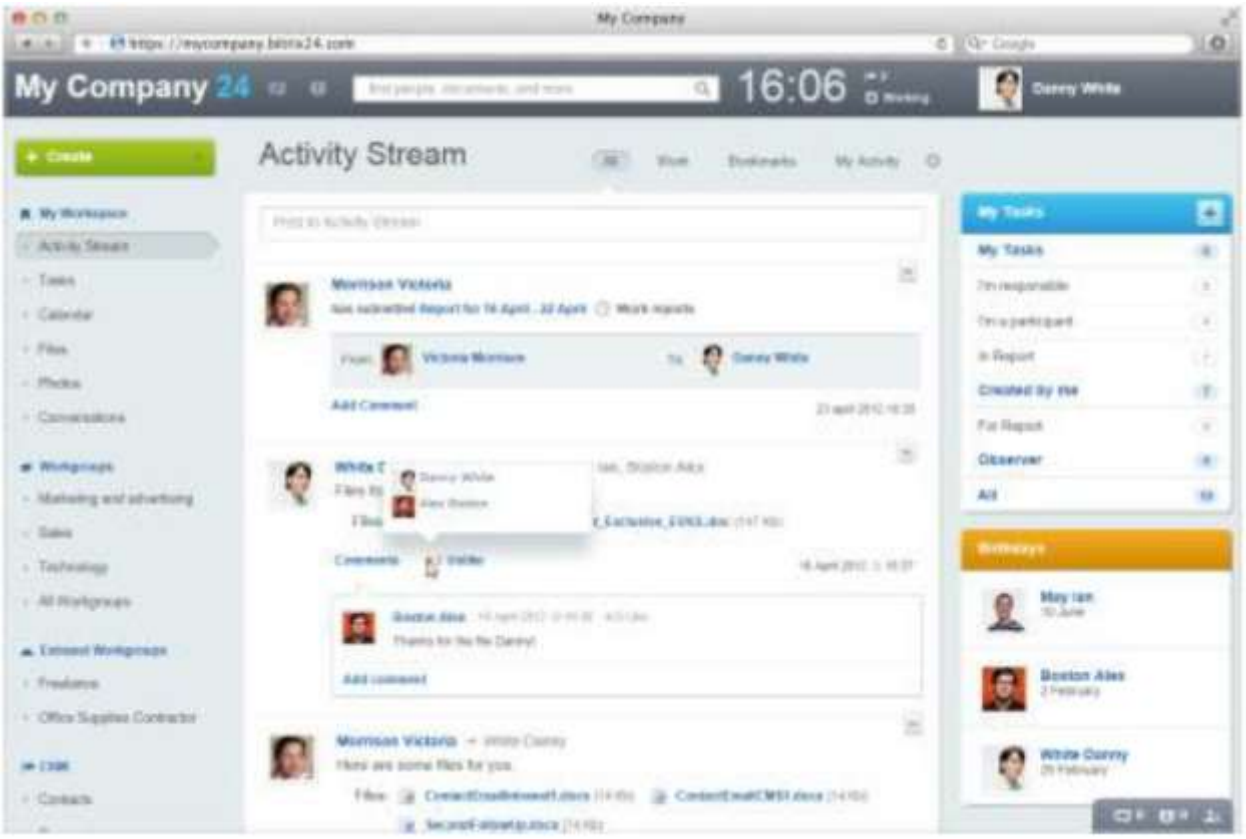


Рис. 1.3 Інтерфейс Bitrix24

На рисунку 1.4 зображено приклад інтерфейсу програми OpenPlan.

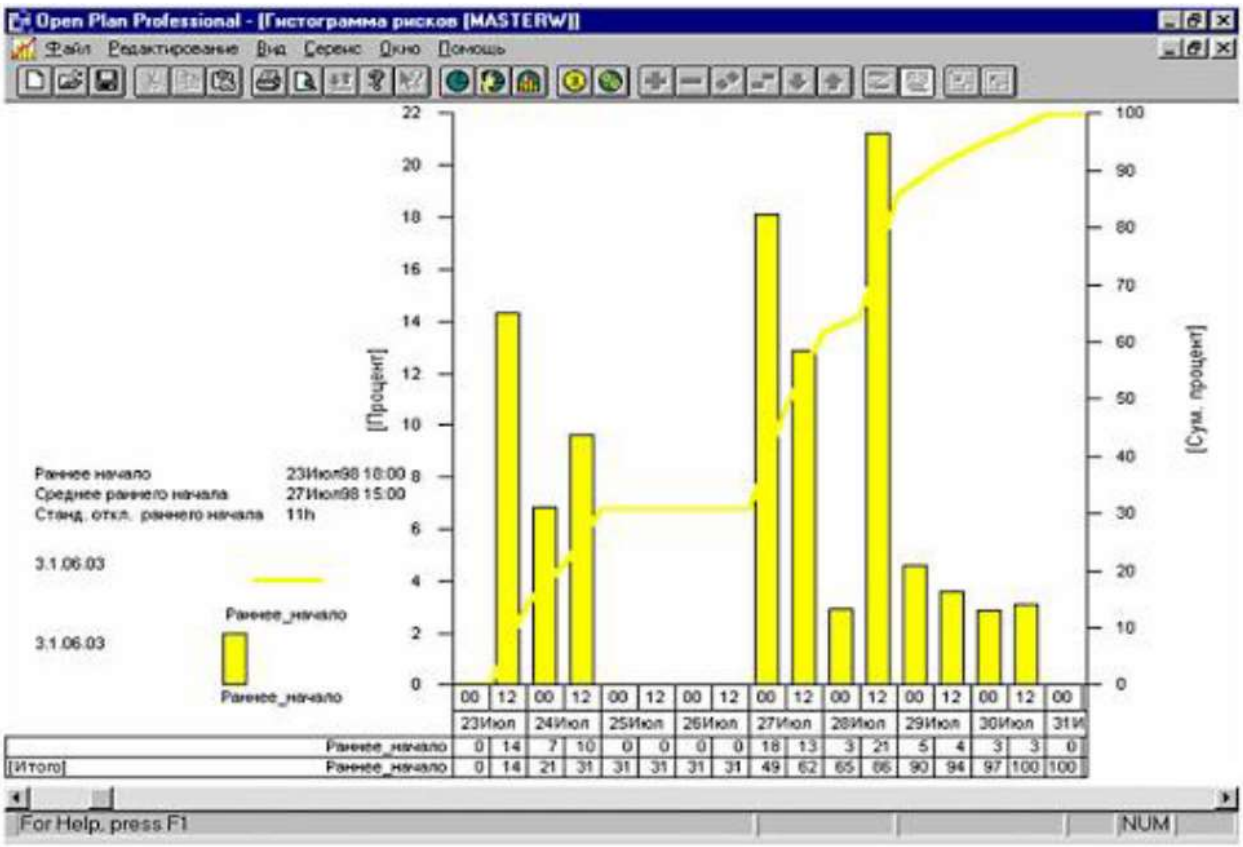


Рис. 1.4 Інтерфейс OpenPlan

На рисунку 1.5 зображено приклад інтерфейсу програми Jira.

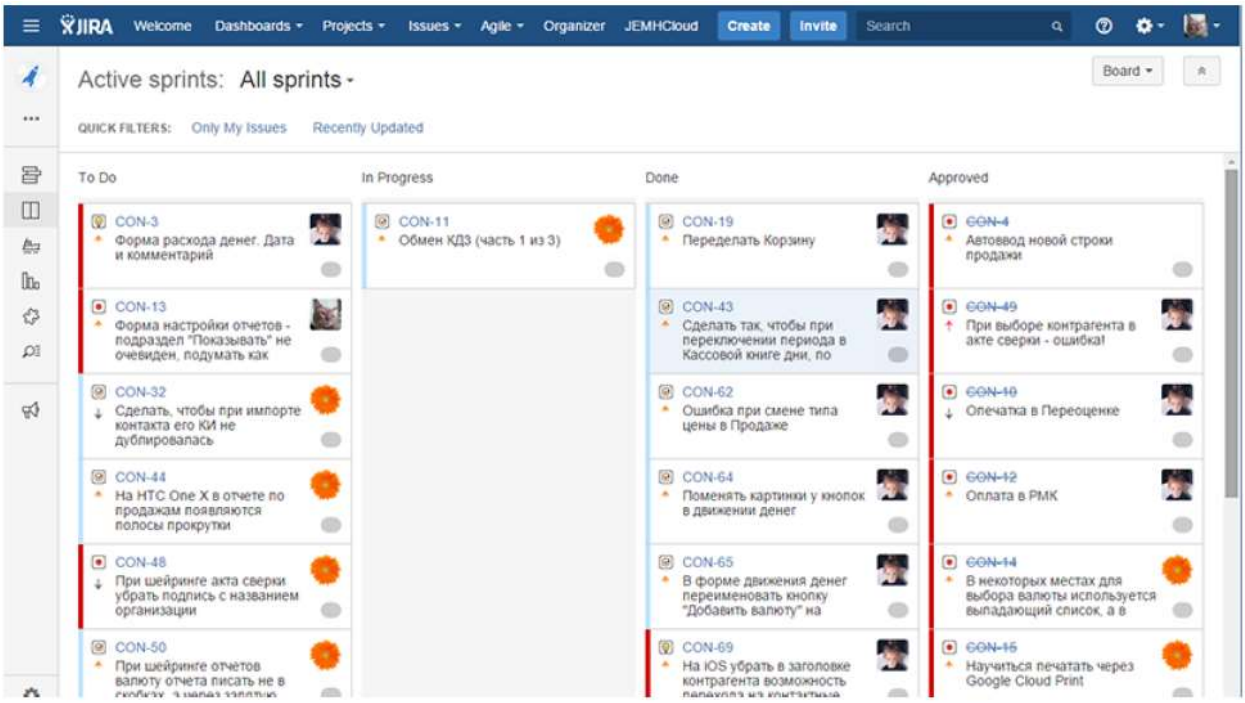


Рис. 1.5 Інтерфейс Jira

Проаналізувавши веб-сервіси управління проектами, було проведено порівняльний аналіз (табл.1.1), де було виявлено позитивні та негативні

сторони, які при створені дипломного проекту треба уникати або обернути на користь системі.

Недоліки:

— наявність непотрібної реклами;

— недопрацьований контент;

— необхідність навчання для використання системи;

— важкий у розумінні інтерфейс;

— надмірний інтерфейс;

— платний функціонал;

— обмежений доступ до пробного періоду;

— онлайн платформа у платному тарифі.

До переваг сайтів-аналогів можна віднести:

— зручна панель навігації;

— дизайн платформ;

— наявність хмарного середовища;

— інтеграція зовнішніх модулів;

— функціональний набір.

Таблиця 1.1

Аналіз декількох комерційних платформ для управління проектами

Критерії	**ManLuck**	Spider Project	Битрикс24	OpenPlan	Jira	Trello	Microsoft Project	Asana
Відсутність непотрібної реклами	+	+	+	+	+	+	+	+
Інструменти для побудови діаграм	-	+	+	-(лише у комерційній версії)	+	-	+	-
Можливість управління декількома проектами	+	-	-	-	+	+	-	+
Тарифи	-	45 000 руб. За одну ліцензію	Проект - безкоштовно 12 користувачів; 5Гб в хмарі. Проект + - 999 руб /	$ 9.99 в мі-сяць за	Фіксований для команди до 10 чоловік - $ 10 в місяць; Гнучка модель: - перші	Business Class - $ 9.99 в місяць за 1 користувача Enterprise - $ 20.83 в місяць за 1 користува	Project (план 1) $10.00 за користувача на місяць (річне зобов’язання)	Premium - $ 9.99 per month per user

			місяць за користувача при оплаті річної передплати 24 користувача; 24Гб в хмарі. Команда - 4990 руб / місяць за користувача при оплаті річної передплати 50 користувачів; 100 ГБ в хмарі.		11-100 користувачів - $ 7 на місяць за людину; - наступні 150 користувачів - $ 5 в місяць за людину; - після 250 чоловік - $ 1.10 в місяць за кожного наступного користувача.	ча	Project (план 3) USD$30.00 за користувача на місяць (річне зобов'язання) Project (план 5) USD$55.00 за користувача на місяць (річне зобов'язання)	
Зручна панель навігації	+	-	+	-	+	+	-	-
Дошка задач для власного користування	+	-	+	-	+	+	-	-
Дизайн платформи	+	-	-	-	+	+	-	+
Хмарне середовище	-	-	-	-	-	-	+	-
Безкоштовна	+	-	+/-	Безкоштовна версія з обмеженим функціоналом	-	+ є безкоштовний доступ - ширший функціонал платний	-	Безкоштовна версія Комерційна версія
Онлайн доступ	+	-	+	-	+	+	+	+
Інтуітивний дизайн	+	-	+	-	+	-	-	-
Розподілення доступу між командою	+	-	-	-	+	+	-	+
Засоби побудови звітів	+	+	+	+	+	-	+	-
Перегляд статистики	+	+	+	+	+	-	+	-
Кроссплатформна	+	-	+	-	+	+	-	+

1.3 Постановка задачі

Метою дипломної роботи є розробка web-додатка для управління виконанням проектів. Даний проєкт повинен забезпечити автоматизацію роботи, виконувану при управлінні виконанням проектів.

Для досягнення мети було визначено наступні задачі:

— провести аналіз існуючих додатків-аналогів;

— визначити послідовність дій створення веб-додатку та розмежувати процес

створення додатку на етапи;

— проаналізувати та обрати тип архітектури веб-додатку та інструменти для швидкого та оптимального процесу побудови додатку;

— змоделювати функції додатку засобами uml;

— спроектувати та створити базу даних веб-додатку;

— розробка веб-додатку;

— тестування додатку.

Функціональні вимоги до веб-додатку.

— *Аутентифікація у системі*. Користувачі матимуть змогу пройти авторизацію чи реєстрацію у системі.

— *Зміна особистих даних користувача.* Користувачі матимуть змогу змінити особисті дані.

— *Створення проектів.* Користувач матиме змогу створювати проекти у системі.

— *Управління командою проекту.* Користувач матиме змогу додавати або видаляти учасників проекту.

— *Створення списків задач.* Користувач матиме змогу формувати списки задач для окремого проекту чи для власного використання.

— *Формування та завантаження звітів.* Користувач матиме змогу обрати на панелі проекту завантаження звіту у форматі PDF.

— *Завантаження файлів до проекту.* Користувач матиме змогу завантажувати файли до обраного проекту.

— *Перегляд статистики проекту.* Користувач матиме змогу переглядати статистику обраного проекту.

Інтерфейс веб-додатку повинен бути інтуїтивно зрозумілим навіть тим користувачам, які не мають ніякого відношення до предметної області проекту.

Даний проєкт буде у майбутньому буде розширений за допомогою створення мобільного додатку, формуванні модулю сповіщення та додавання нейронної мережі для розподілення задач у команді.

Для розробки веб-додатку було використано середовище розробки Intellij IDEA, DataGrip JetBrains, Apache Tomcat, Java SE/EE.

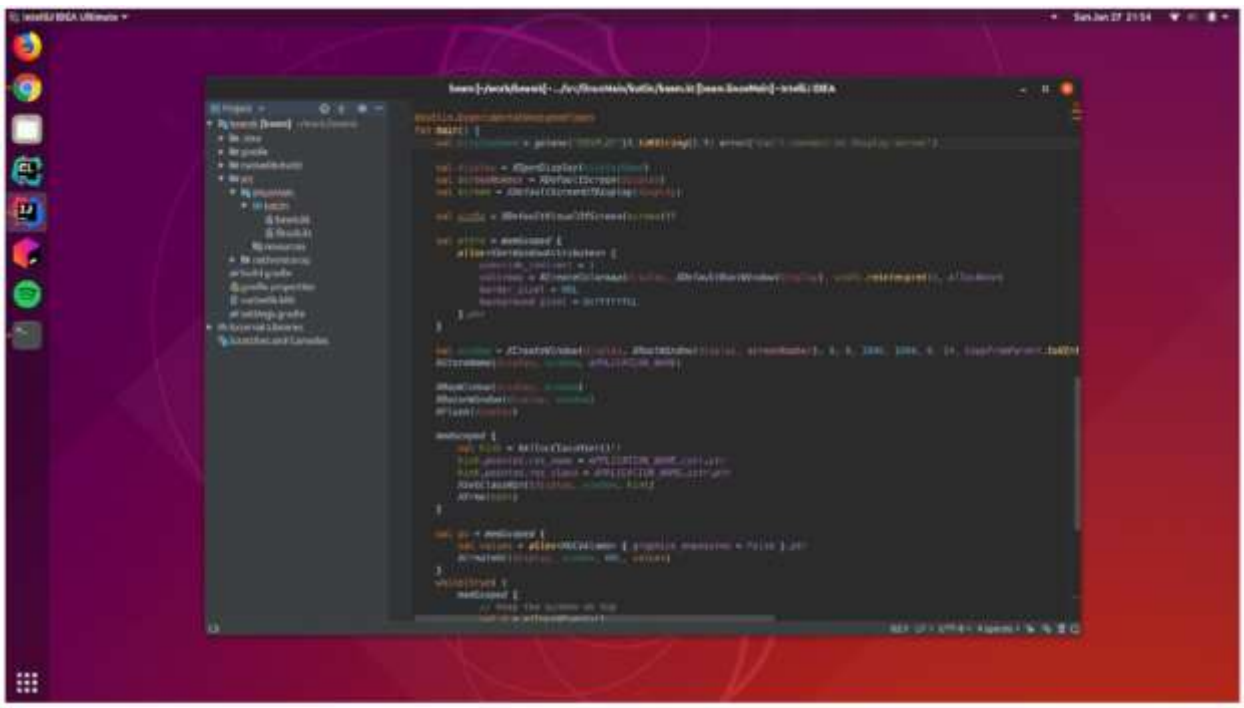


Рис. 1.6 Вікно редактора Intellij IDEA

JetBrains Intellij IDEA – інтегроване середовище розробки (IDE), в основному призначена для Java. Це середовище використовується спеціально для розробки програм. Він розроблений компанією під назвою JetBrains, яка формально називалася IntelliJ. Він доступний в двох редакціях: Community Edition, ліцензованої Apache 2.0, і комерційною версією, відомої як Ultimate Edition. Обидва вони можуть бути використані для створення програмного забезпечення, яке може бути продане. Що відрізняє IntelliJ IDEA від аналогів, так це простота використання, гнучкість і продуманий дизайн, також

середовище розробки має розширені функції перевірки помилок, що дозволяє швидше і простіше перевірити помилки.

Переваги Intellij IDEA:

— Розумне завершення;

— Завершення ланцюга;

— Статичне завершення членів;

— Аналіз потоку даних;

— Мовна ін'єкція;

— Багатомовне переробництво;

— Виявлення дублікатів;

— Перевірки та швидкі виправлення;

— Контроль версій;

DataGrip JetBrains – це вдосконалене середовище розробки інтегрованих баз даних багатомоторних систем, створене для професійних розробників SQL, щоб вони могли ефективно виконувати запити та виконувати навігацію за схемою. Розроблена JetBrains, система виділяє вбудовані драйвери для підтримки різних двигунів, таких як Exasol, Derby, AWS Redshift, DB2, HSQLDB, Microsoft Azure, H2, MySQL, Oracle, SQL Server, Sybase, SQLite та PostgreSQL. DataGrip пропонує безліч введення даних та деякі розширені функціональні можливості для полегшення перетворення та розвитку об'єктів, щоб відповідати підтримуваній мові.

Інтегрований інтерфейс передбачений для систем управління версіями, щоб запропонувати кращий досвід користувача з Perforce, Git, CVS, Subversion, GitHub, TFS, Perforce та Mercurial. Деякі основні функції включають редактор даних, об'єкти бази даних, розумний текстовий редактор, навігацію, заповнення коду, локальну історію, аналіз коду та швидкі виправлення, консоль запитів, підтримка VCS та переглядач різниці.

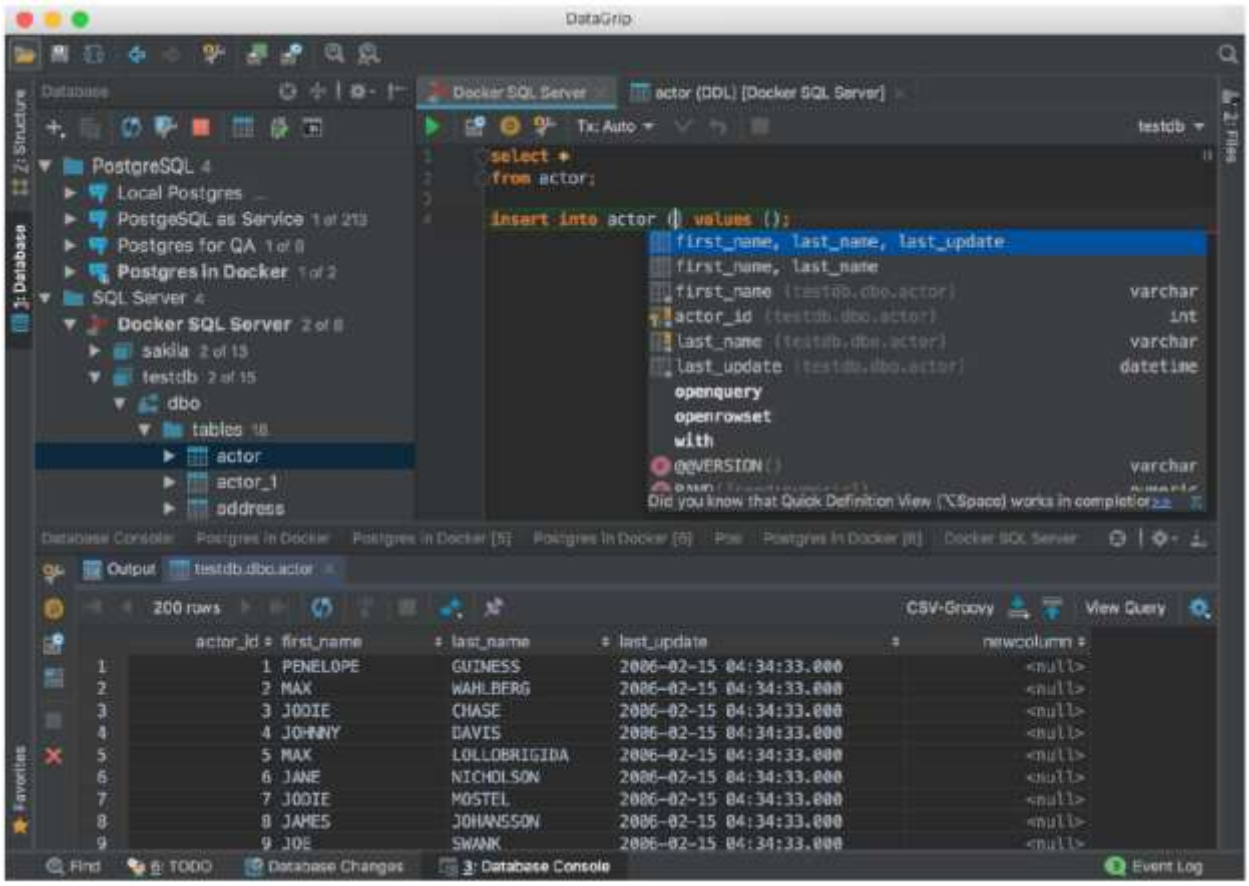


Рис. 1.7 Інтерфейс DataGrip

Apache Tomcat – веб-контейнер з відкритим кодом. Це не сервер прикладних програм, як JBoss, Glassfish тощо, які в основному необхідні для корпоративних веб-додатків. Зазвичай складається з EJB та інших важких компонентів J2EE. Tomcat – один з найбільш широко використовуваних веб-серверів розробниками Java. Його поточна версія - 9. Він використовується для запуску веб-додатків. Я використовував Tomcat для запуску таких програм:

— Програми JSP / Servlet.

— Прикладні програми Spring Framework.

— Struts 2 програми.

Maven та Spring Boot мають додатки для цього, які можна використовувати для запуску вбудованого Tomcat для розгортання додатків на локальних машинах.

Java – спочатку була мовою програмування створеною у компанії Sun Microsystems. Сьогодні Oracle володіє Sun, а тому і Java. Таким чином, Java є

торговою маркою Oracle. З часом Java перетворилася на більше, ніж просто мову. Це повна платформа з великою кількістю стандартних API, API з відкритим кодом, інструментами, велика спільнота розробників з мільйонами розробників тощо.

РОЗДІЛ 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Архітектура додатку

Перед початком розробки інформаційної системи необхідно правильно побудувати архітектуру майбутньої системи та розробити логічну структуру для детального проектування системи.

Для базової роботи інформаційної системи потрібен сервер, на якому буде розміщено модулі нашої системи, для цього нас задовольнить звичайний безкоштовний хостинг, але для майбутнього масштабування системи та підтримки швидкості роботи при великих навантаженнях необхідно обрати якісний хостинг з потужними апаратними характеристиками.

Архітектура ІС приведена на зображенні (рис. 2.1) Для роботи з ІС користувачам потрібно відкрити веб-інтерфейс системи використовуючи веб-браузер. Архітектура системи побудована за принципом Клієнт-Сервер. [4]

Завдяки шаблону проектування Модель-Вигляд-Контролер ми маємо змогу відокремити взаємодію користувача безпосередньо з моделлю та передати управління контролеру, а контролер у свою чергу буде повертати необхідні дані користувачу. У такий спосіб ми мінімізуємо можливості проникнення до бази даних на сервері та зменшимо ризик втрати даних.

Користувач через браузер буде надсилати запити до контролеру, контролер у свою чергу буде перевіряти дані користувача на доступ до функціонали та у випадку успішної перевірки повертати необхідну відповідь з серверу.

Архітектура системи складається за таких компонентів:

— Користувачі;

— Веб-інтерфейс;

— API системи(контролери та сервіси роботи з базами даних);

— Бази даних.

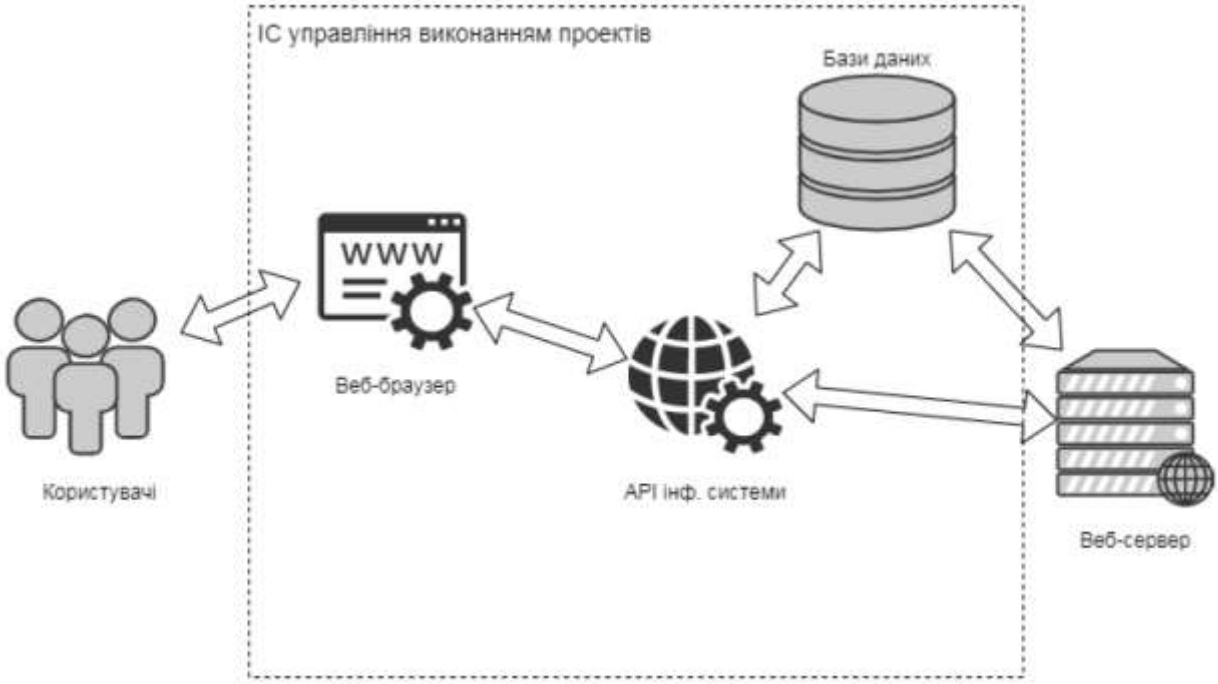


Рис. 2.1 Архітектура ІС

2.2 Структурно-функціональне моделювання роботи додатку

Для початку потрібно створити контекстну діаграму, IDEF0, функціональну модель, яка є центром побудови систем та поєднує бізнес процеси компанії [5].

Центральною моделлю є функція, яка відображена функціональним блоком та розкриває основну логіку роботи системи [5,12].

Вхідними даними до функції «Управління виконанням проектів» є:

— Інформація про користувача;

— Базова інформація про майбутній проект.

Вихідними даними є:

— Готовий проект як об’єкт для управління;

— Статистика по проекту;

— Звіт з обраного проекту.

Для контекстної діаграми є обов’язковим наявність хоча б одного елементу управління, який поступає згори моделі до функціонального блоку. На даній діаграмі елементом управління є:

— Функції системи.

Механізмами моделі є:

— ІС;

— Менеджери;

— Розробник проекту;

— Замовник проекту.

На рисунку 2.2 зображено IDEF0 модель, головного процесу інформаційної системи управління виконанням проектів.



Рис. 2.2 Діаграма IDEF0

Для більш детального опису роботи інформаційної системи необхідно провести декомпозицію контекстної діаграми [5].

В результаті аналізу системи мною було зроблено декомпозицію головного бізнес процесу.

Після виконаної роботи система була розділена на такі процеси:

— визначення доступу до системи;

— створення та управління проектом;

— управління дошкою завдань;

— аналіз виконаних задач;

— формування звіту з проекту;

— перегляд статистики проекту.

Процес «Визначення доступу до системи» має такі характеристики:

— Вхідні дані:

— інформація про користувача.

— Вихідні дані:

— надання послуг користувачу.

— Механізми:

— ІС;

— замовники;

— менеджери;

— розробники.

Процес «Створення та управління проектом» має такі характеристики.

— Вхідні дані:

— файли та інформація проекту;

— оновлені статистичні дані;

— обрання проекту.

— Вихідні дані:

— обраний проект;

— проект як об’єкт управління.

— Елементи управління:

— функції системи;

— надання послуг користувачу.

— Механізми:

— менеджери;

— ІС;

— розробник проекту.

Процес «Управління дошкою завдань» має такі характеристики.

— Вхідні дані:

— інформація про задачі проекту.

— Вихідні дані:

— інформація про задачі проекту.

— Механізми:

— менеджери;

— розробник проекту.

Процес «Аналіз виконаних задач» має такі характеристики.

— Вхідні дані:

— інформація про задачі проекту.

— Вихідні дані:

— оновлені статистичні дані.

Процес «Формування звіту з проекту» має такі характеристики.

— Вхідні дані:

— обраний проект.

— Вихідні дані:

— звіт з обраного проекту.

— Механізми:

— менеджери.

Процес «Перегляд статистики проекту» має такі характеристики.

— Вхідні дані:

— обраний проект.

— Вихідні дані:

— Статистика по проекту.

— Механізми:

— менеджери;

— замовник проекту.

На рис. 2.3 зображено IDEF діаграму декомпозиції головного процесу.

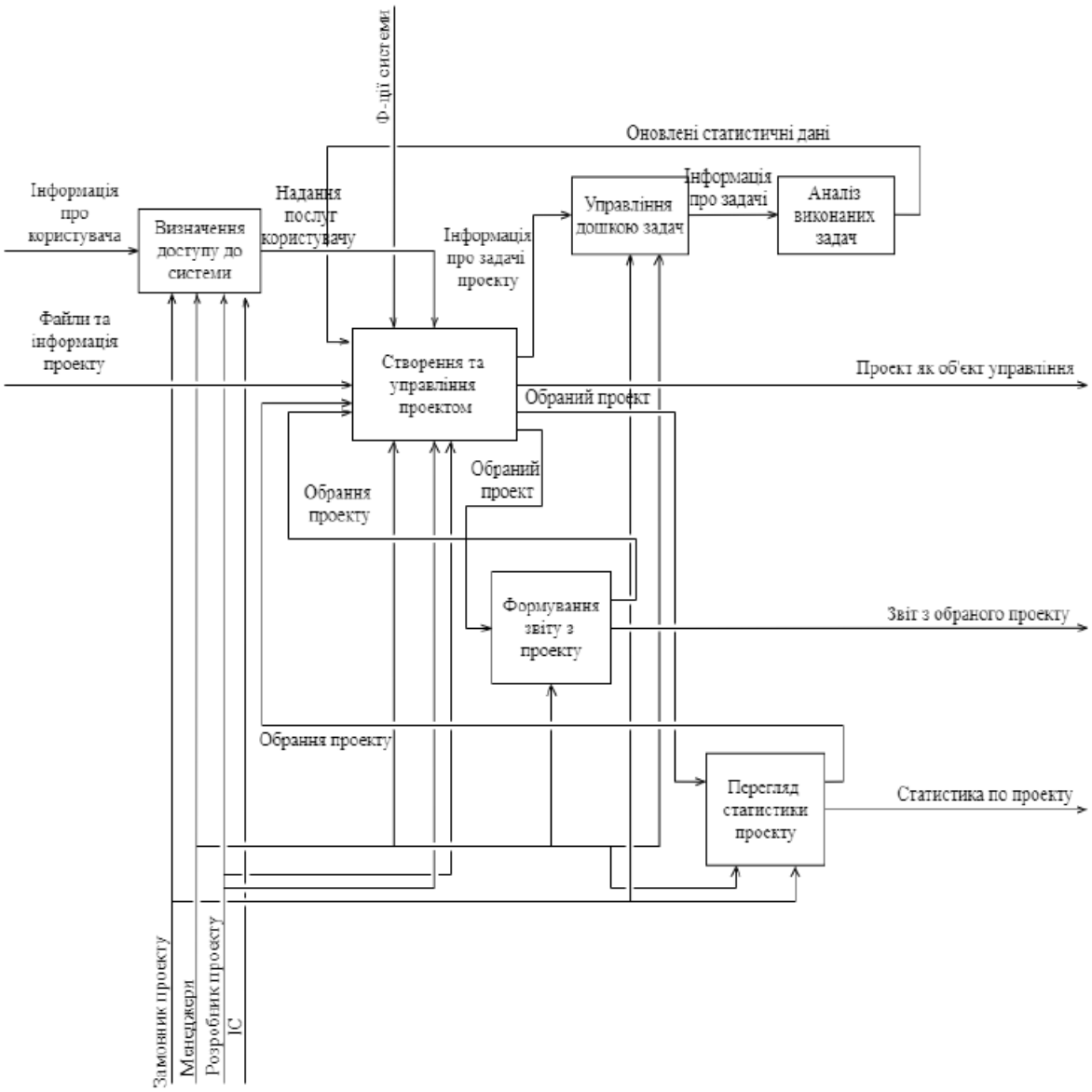


Рис. 2.3 Діаграма декомпозиції

2.3 Проектування бази даних

Для зберігання даних користувачів та оновлення чи видалення даних використовується база даних. Щоб продемонструвати структуру бази даних необхідно створити ER діаграму.

Діаграма ER показує взаємозв'язок між сутностями, де сутність - це таблиця або атрибут таблиці в базі даних, тому, показуючи взаємозв'язок між таблицями та їх атрибутами, ER-схема показує повну логічну структуру бази даних [8].

Сутностями системи виступають:

— user - сутність для моделювання базового об’єкту користувача;

— city -сутність для зберігання назв міст;

— country - сутність для зберігання назв країн;

— role - сутність, яка зберігає назви рівнів доступу у системі;

— team - сутність для зберігання команди як об’єкту;

— project - сутність, яка зберігає об’єкт проекту.

На рисунку 2.4 приведено схему бази даних інформаційної системи управління проектами.

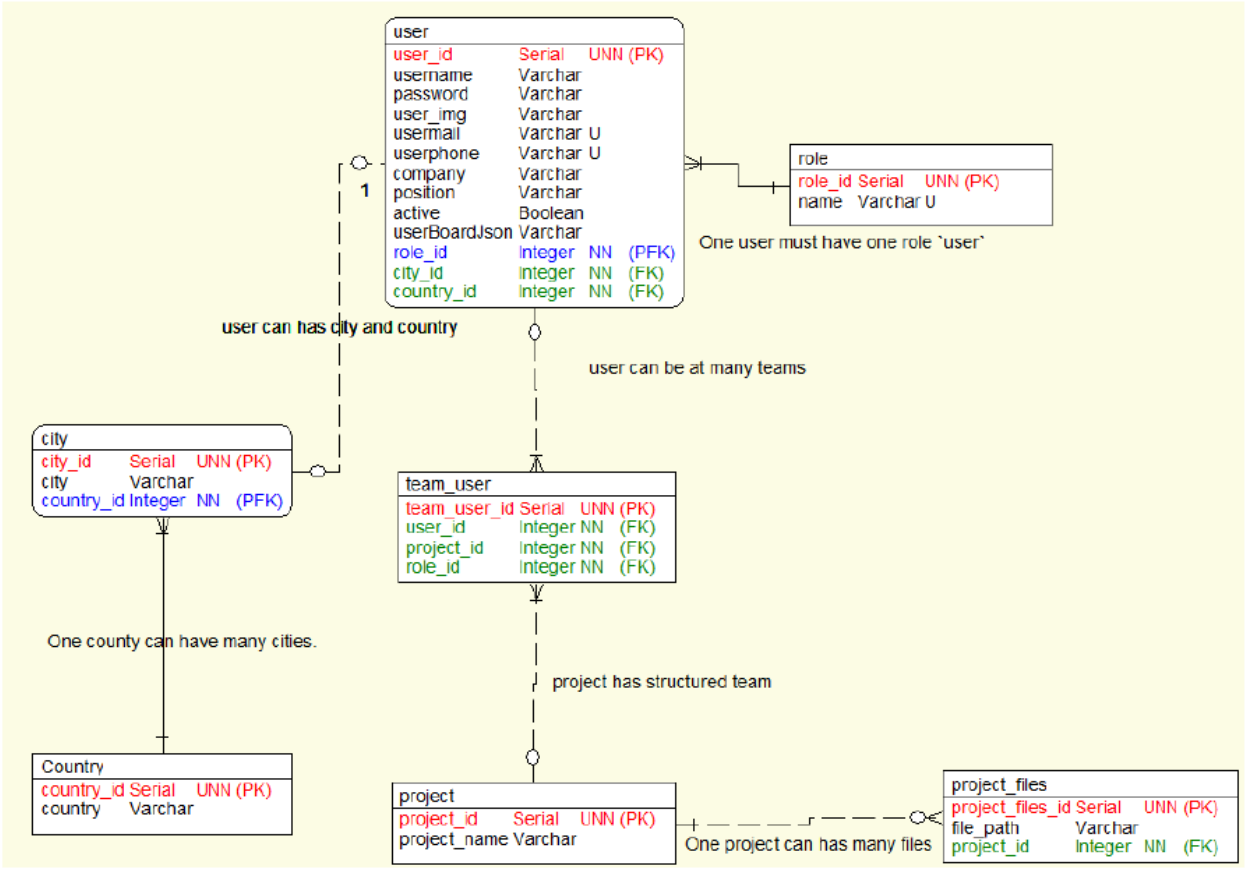


Рис. 2.4 ЕР діаграма бази даних додатку

2.4 Моделювання варіантів використання

Після моделювання базових бізнес процесів проекту я перейшов до створення діаграми варіантів використання, яка продемонструє послідовність дій при роботі з системою [6].

UML є стандартизованою мовою моделювання, що складається з набору діаграм, розроблених, щоб допомогти розробникам у визначенні та

візуалізації програмних систем. UML представляє графічні позначення для побудови архітектурного дизайну програмних продуктів.

Діаграму варіантів використання є сенс будувати під час вивчення технічного завдання, вона складається з графічної діаграми, яка описує дійові особи і прецеденти, а також специфікації, що представляє собою текстовий опис конкретних послідовностей дій (потоку подій), які виконує користувач при роботі з системою [6].

Елементами зовнішнього зв’язку на діаграмі використання (рис. 2.5) виділено сутність для зовнішнього зв’язку у вигляді Бази даних, зв’язок з нею виникає за необхідністю користувача. До бази даних заноситься особиста інформація користувача, інформація про проекти, статистичні дані.

Актором на діаграмі виступає користувач, який має можливість реєстрації або авторизації, якщо вже має акаунт, може створювати проекти, додавати інформацію про проект та змінювати учасників, в залежності від прав доступу на проекті. Користувач має змогу створювати задачі на панелі задач, видаляти та змінювати їх, обравши проект може створити звіт з обраного проекту маючи права, також є можливість перегляду статистики по кожному з проектів або особистої статистики користувача.

Варіантами використання є:

— ВВ –

Реєстрація - дозволяє користувачу зареєструватись у системі та отримати доступ до функціоналу;

— ВВ - Авторизація - дозволяє користувачу авторизуватись у системі та подовжити роботу під існуючим акаунтом;

— ВВ - Створення проекту - дозволяє створити проект та почати роботу;

— ВВ - Управління проекту - дозволяє змінювати інформацію про проект, якщо на це є права;

— ВВ - Видалення проекту - дозволяє користувачу видалити проект з системи та видалити всі існуючі файли та інформацію, якщо на це є права доступу;

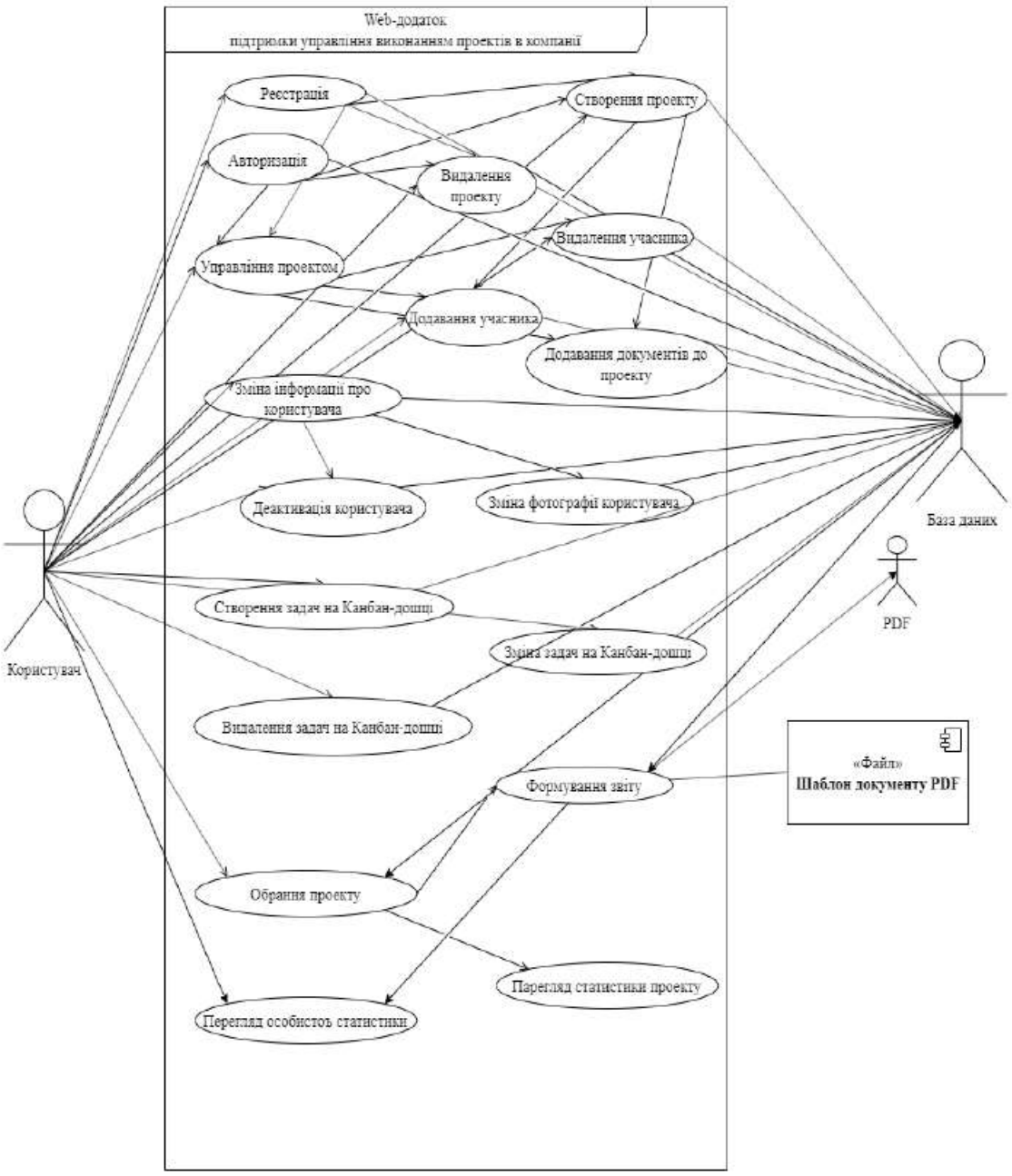


Рис. 2.5 Діаграма варіантів використання

— ВВ - Додавання учасника - дозволяє користувачу додавати учасників до команди, якщо є права на ці дії;

— ВВ - Видалення учасника - дозволяє видалити учасника з команди проекту та обмежити доступ до інформації проекту;

— ВВ - Додавання документів до проекту - користувач має змогу додавати файли та інформацію до проекту;

— ВВ - Зміна інформації про користувача - дозволяє користувачу оновлювати персональну інформацію про себе;

— ВВ - Зміна фотографії користувача - дозволяє користувачу оновлювати власну фотографію;

— ВВ - Деактивація користувача - функція дозволяє користувачу видалити власний акаунт;

— ВВ - Створення задач на Канбан-дошці - дозволяє створити задачу на дошці завдань;

— ВВ - Зміна задач на Канбан-дошці - дозволяє змінити інформацію задачі;

— ВВ - Видалення задач на Канбан-дошці - дозволяє видалити інформацію з дошки;

— ВВ - Обрання проекту - функція я виконується перед етапами формування звіту та статистики, дозволяє користувачу обрати проект для перегляду або формування необхідної інформації;

— ВВ - Формування звіту - формується звіт з обраного проекту;

— ВВ - Перегляд статистики проекту - дозволяє переглядати статистику з обраного проекту;

— ВВ - Перегляд особистої статистики - дозволяє переглядати власну статистику користувача.

2.5 Моделювання діаграм діяльності

Діаграма діяльності - це діаграма поведінки, яка показує процес роботи об’єкта з урахуванням умов потоку виконання. Дії у моделях діяльності можуть створюватись або через закінчення попередніх дій або через зовнішні чинники [7].

На рисунку 2.6 зображено діаграму діяльності модулю авторизації.

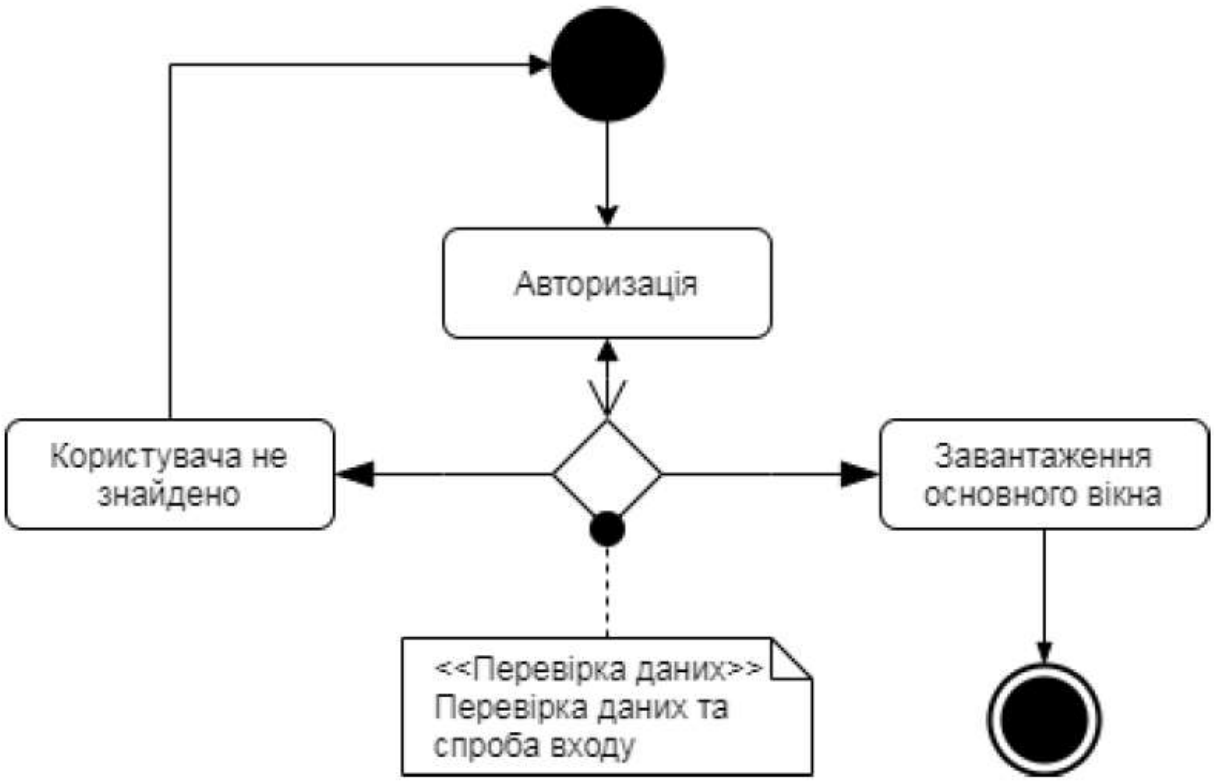


Рис. 2.6 Діаграма діяльності модулю авторизації

На рисунку 2.7 зображено діаграму діяльності модулю реєстрації.



Рис. 2.7 Діаграма діяльності модулю реєстрації

На рисунку 2.8 зображено діаграму діяльності модулю зміни інформації користувача.



Рис. 2.8 Діаграма діяльності модулю зміни інформації користувача

На рисунку 2.9 зображено діаграму діяльності модулю управління проектами.

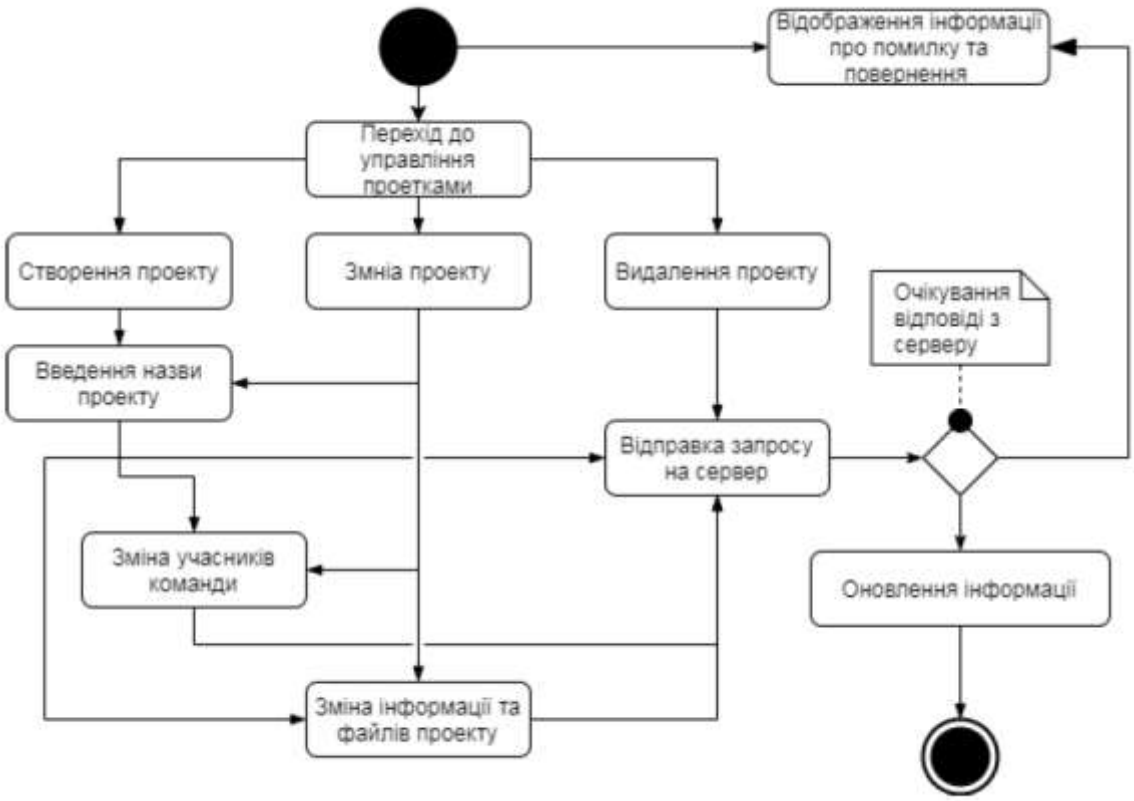


Рис. 2.9 Діаграма діяльності модулю управління проектами

На рисунку 2.10 зображено діаграму діяльності модулю створення звітів.

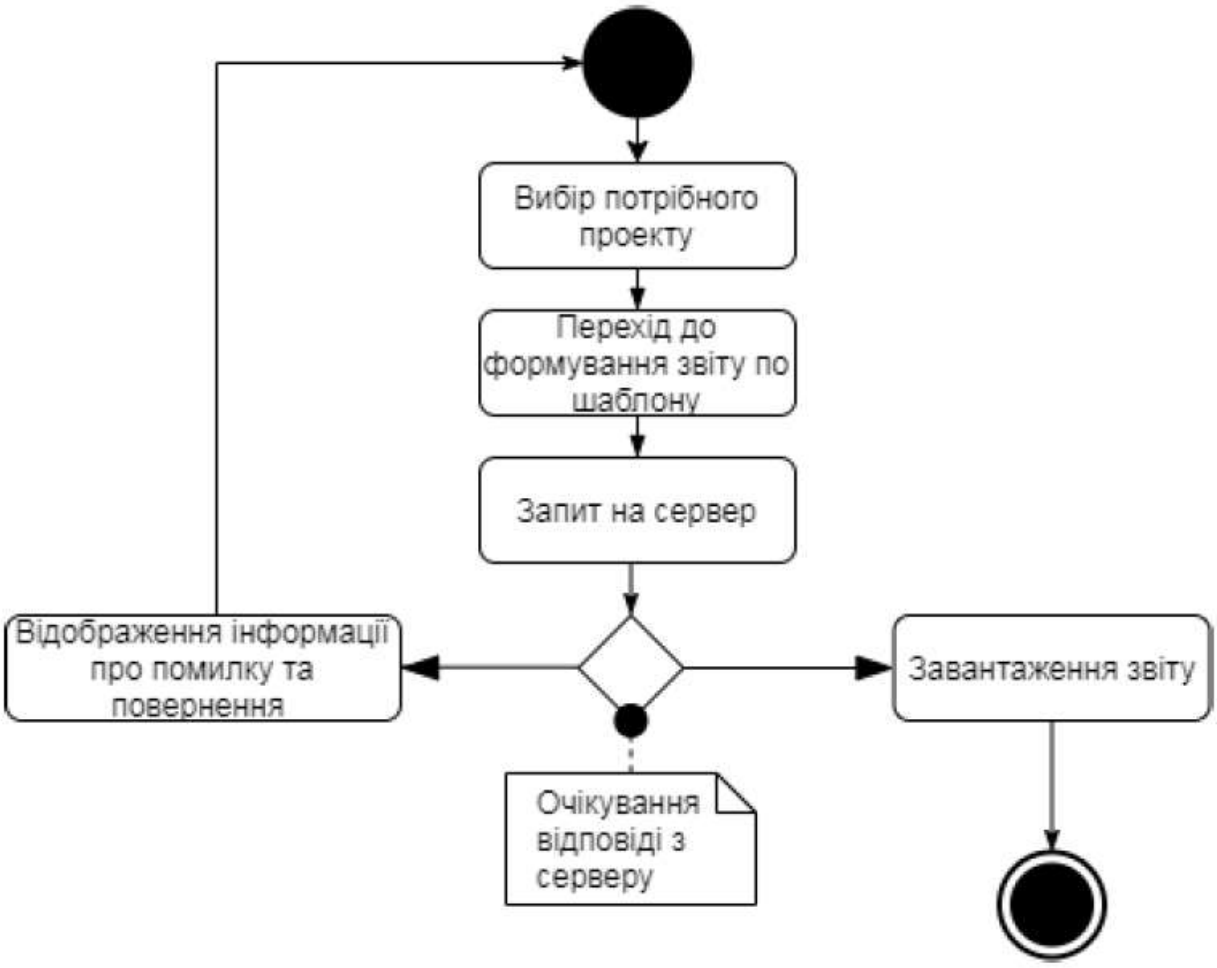


Рис. 2.10 Діаграма діяльності модулю створення звітів

На рисунку 2.11 зображено діаграму діяльності модулю перегляду статистики.



Рис. 2.11 Діаграма діяльності модулю перегляду статистики

2.6 Моделювання діаграми класів

Клас - це шаблон, який використовується для створення об’єкта. Клас визначає, що об’єкт може робити, дає огляд програмної системи шляхом відображення класів, атрибутів, операцій та їх взаємозв'язків. Ця діаграма включає в себе назву класу, атрибути та операції в окремих призначених відсіках.

Діаграма класів визначає типи об’єктів у системі та різні типи зв’язків, що існують серед них. Це дає перегляд програми на високому рівні [9].

На рисунку 2.12 та 2.13 представлено діаграму класів системи, з детальними прикладами зв’язків між ними.

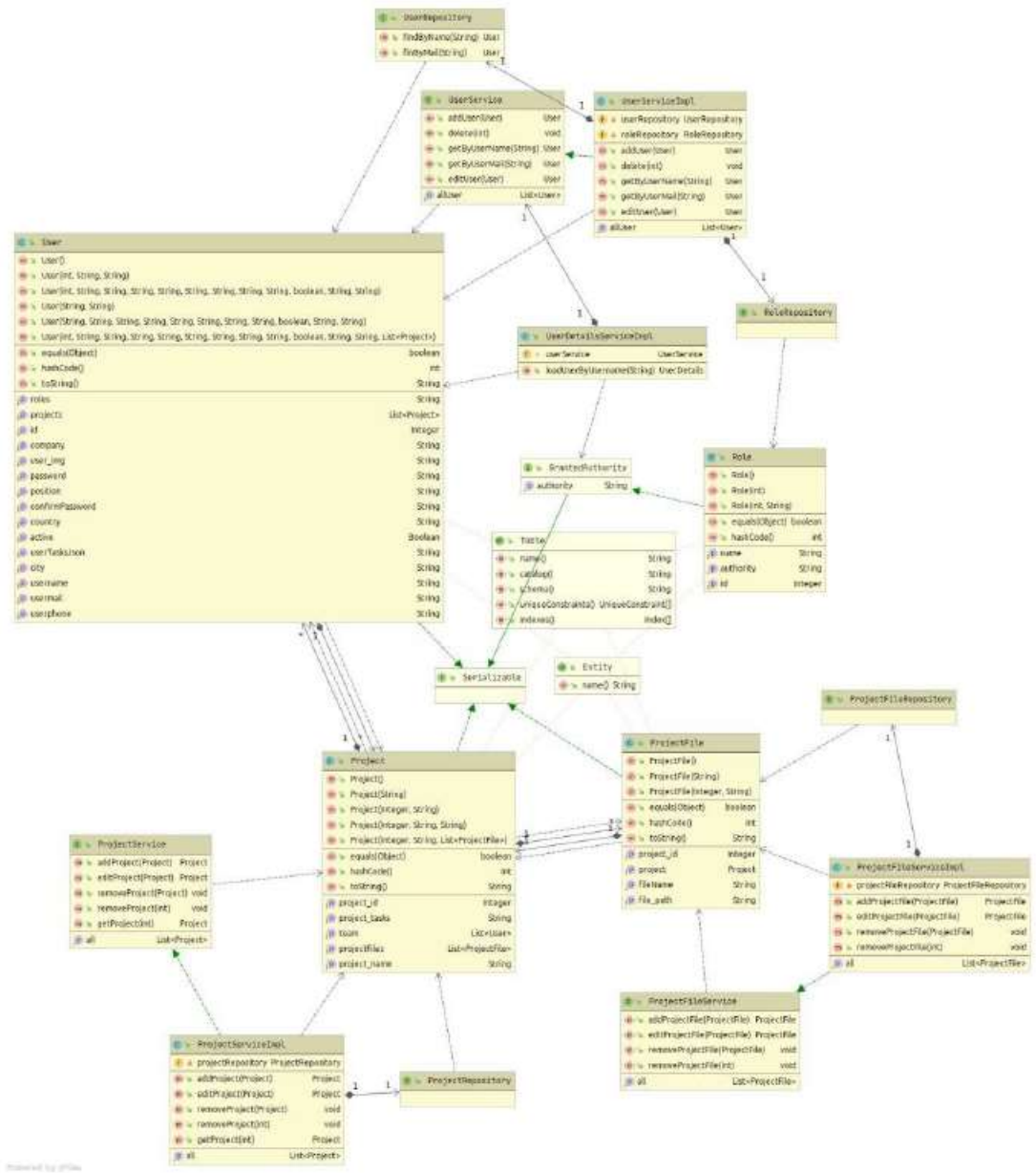


Рис. 2.12 Діаграма класів модулю роботи з базою даних

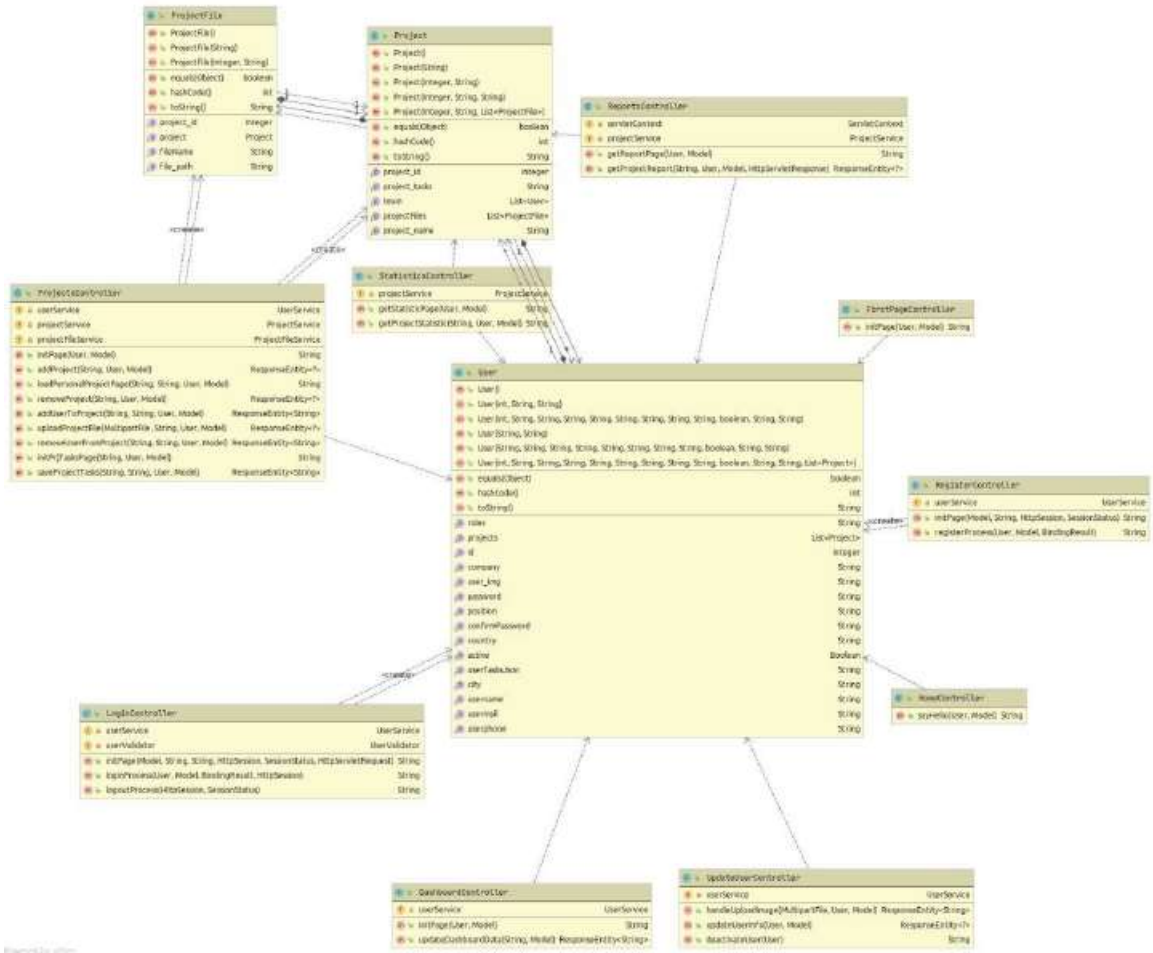


Рис. 2.13 Діаграма класів бізнес-логіки додатку

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ДОДАТКУ

3.1 Структура програмного додатку

Пройшовши етап моделювання та проектування додатку необхідно розпочати розробку структури програмного продукту.

Для побудови структури додатку було виділено два базові етапи, а саме: обрання найбільш вдалих технологій та побудова прототипу.

Згідно вибору, описаному в пункті 2.2 розробка продукту виконується мовою Java.

Для створення та обробки веб-запитів користувача був обраний Spring Framework для Java. Spring за допомогою модулів Spring Data, Spring MVC та Spring Security забезпечує можливість швидкої побудови веб-додатків, налаштування роботи з базою даних, та забезпечення перевірки вхідних даних.

Для зберігання даних було обрано PostgreSQL, ключовими показниками для обирання цієї бази даних були:

— безкоштовність;

— якість збереження даних;

— можливість гнучкого налаштування сховища.

Для розробки за локальний контейнер було обрано Apache Tomcat, використання якого створить імітацію роботи додатку на хостингу від Heroku Cloud.

На другому етапі було розроблено прототип майбутнього веб-додатку, або навігація по системі. При першому відвідуванні системи, користувач потрапляє на сторінку привітання, на сторінці привітання розташовується назва проекту, текст для залучення користувачів та дві кнопки для авторизації чи реєстрації. При натисканні на кнопку авторизації, користувач переходить на сторінку авторизації у системі, де знаходиться форма для введення електронної адреси та паролю користувача вже зареєстрованого у системі. При натисканні на кнопку реєстрації користувач перенаправляєть на

сторінку реєстрації, де знаходиться форма для створення акаунту користувача. Після обрання одного зі способів початку роботи з системою, користувач перенаправляєть до базової сторінки веб-додатку, на якій знаходиться частина статистичних даних користувача, та панель навігації по системі.

На панелі навігації розташовані панель меню для навігації та відображення зображення користувача та його ім’я у системі. Панель меню складається з посилань на ключові модулі системи:

— домашня сторінка;

— сторінка проектів;

— сторінка дошки завдань;

— сторінка формування звітів;

— сторінка перегляду статистики;

— вихід з акаунту користувача системи.

Домашня сторінка складається з форм, які зберігають у собі особисту інформацію користувача, ввівши нові дані, користувач матиме змогу змінити їх.

Сторінка проектів складається з кнопки для створення проектів та дошки вже існуючих проектів. Натиснувши на проект користувач перехожить до особистої сторінки проекту, на якій знаходить три блоки, а саме: дошка завдань, формування звіту з проекту, переглад статисти та блок додавання користувача чи файлу, чи управління командою проекту, у правій частині знаходиться панель з існуючими файлами.

На сторінках формування звіту чи перегляду статистики, користувач має змогу обрати завантаження звіту чи перегляд статистики по обраному проекту на панелі проектів.

3.2 Програмна реалізація

Програмна реалізація – це процес розробки додатку використовуючи обрані технології та засоби реалізації, по вже розробленому плану та структурі проекту.

На етапі програмної реалізації виконується процес підключення модулів додатку до бази даних, створення обробників запитів користувача до серверу та навігації по системі. На даному етапі буде описано ключові складові процесу реалізації веб-додатку, а саме:

— пакети для структурування тарозмежування модулів програми;

— класи, як основні складові розробки;

— проміжні файли для доповнення функціоналу системи;

— підключення та робота з базою даних;

— робота зі сховищем серверу.

Код додатку можна переглянути в додатку В.

Середовище Intellij IDEA використовувалося як основний текстовий редактор для розробки системи.

Для початку неоюхідно відобразити основні складові проекту, структуровані за допомогою пакетів та каталогів середовища розробки, у свою чергу це допоможе швидко знаходити компоненти програми та полегшить навігацію по проекту, дерево проекту зображено на рисунку 3.1. Основні частини додатку розташовані у каталогах та пакетах середовища, пакети – це каталог для зберігання .java файлів.

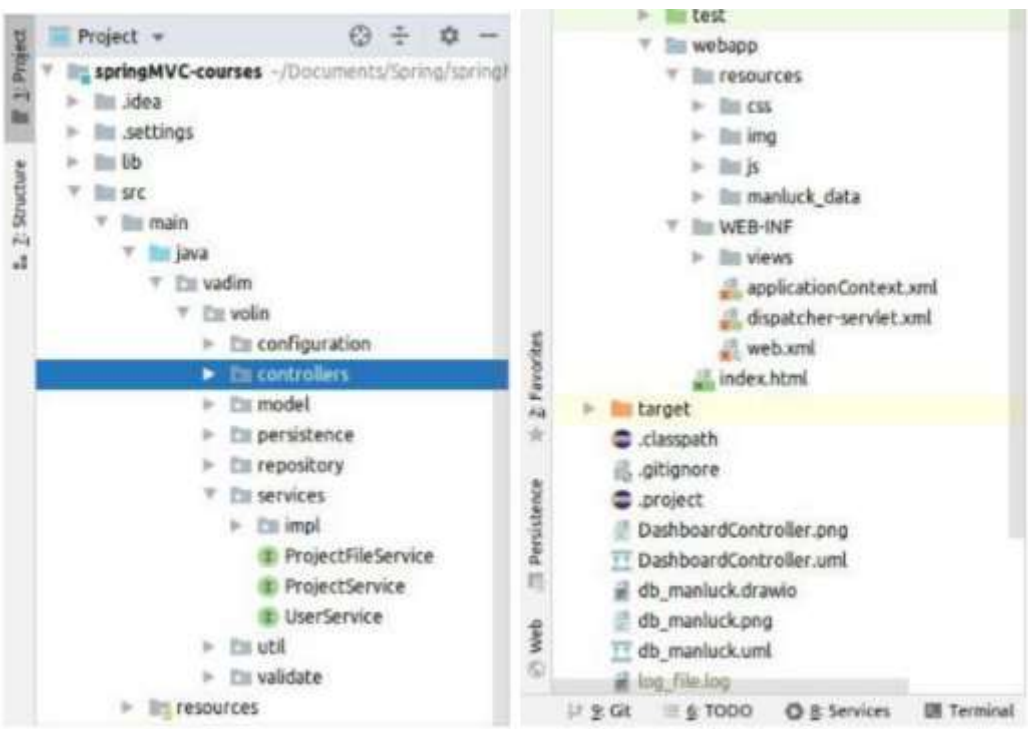


Рис. 3.1 Структура проекту

Пакет ***configuration*** – це пакет в якому знаходяться файли конфігурації додатку у інтернет мережі та розмежування рівню доступу.

Пакет ***controllers*** – цей пакет містить класи, які виконують роль обробників запитів користувача та містять основну логіку навігації та завантаження сторінок.

Пакет ***model*** – в цьому пакеті знаходяться класи, які відображають ключові логічні сутності системи.

Пакети ***persistence*** – в пакеті розміщені файли для підключення до бази даних.

Пакет ***repository*** – в пакеті знаходяться файли, які виконують базові функції роботи з базою даних та логічними сутностями.

Пакет ***services.impl*** – в пакетах розташовані інтерфейси та їх реалізації для виконання основної бізнес-логіки системи.

Пакет ***util*** – в даному пакеті знаходяться файли для виконання допоміжних процесів.

Пакет ***validate*** – в цьому пакеті розташовані файли, які використовуються для валідації даних.

Пакет ***webapp*** - в цьому каталозі розміщені каталоги та файли відображень.

Пакет ***resources.js*** – у цьому каталозі розташовані файли javascript, які виконують побічні бізнес процеси.

Пакет ***resources.css*** – в цьому пакеті розташовані файли, використовуються для стилізації відображень.

Пакет ***WEB-INF.views*** – в цьому пакеті розташовані файли відображень.

Для більшого розуміння компонентів програми необхідно детально описати файли каталогу проекту. На рисунку 3.2 зображено детальну структуру каталогу проекту з повним обсягом пакетів та каталогів описаних вище та файлів, які там зберігаються.

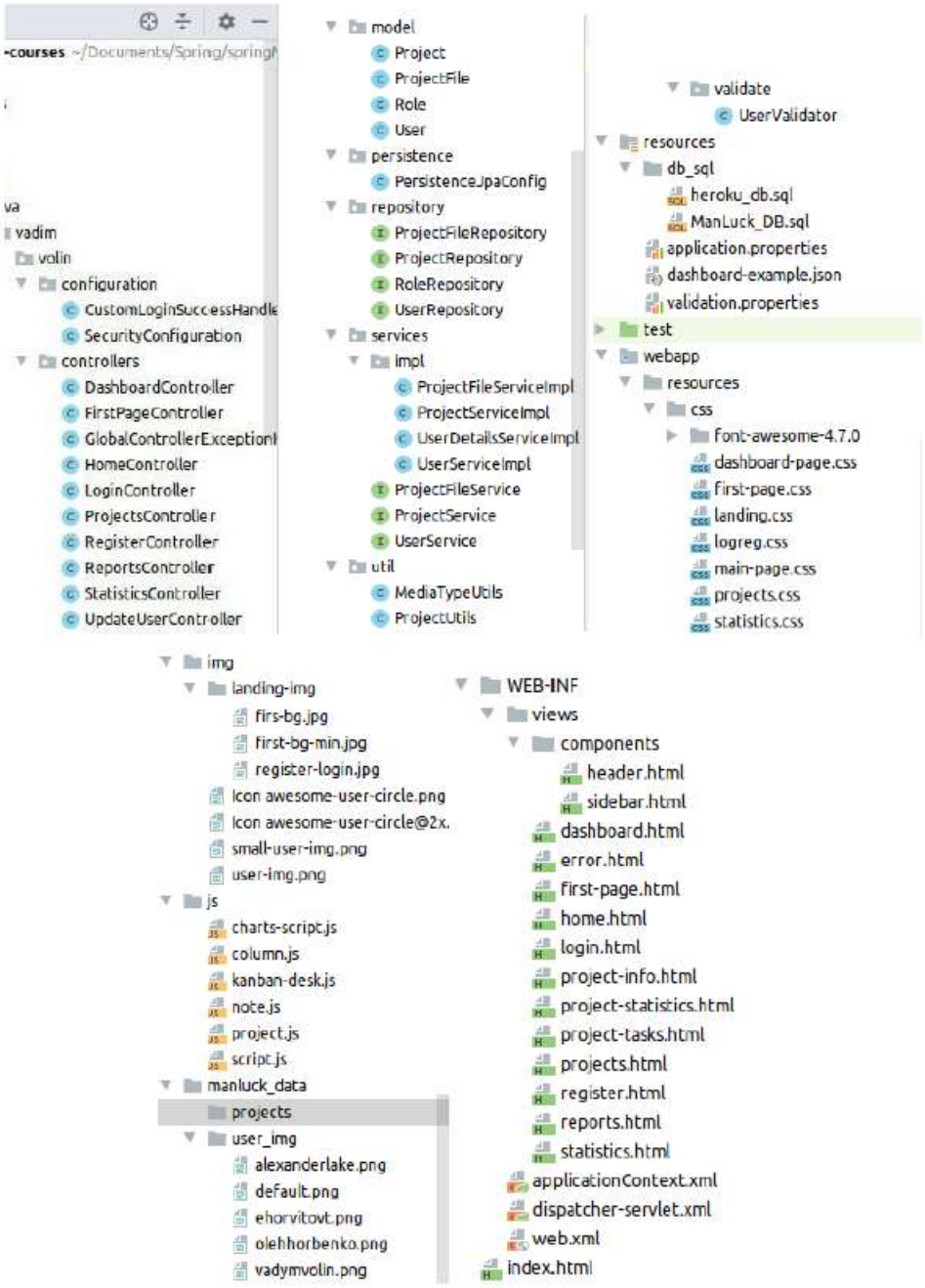


Рис. 3.2 Зображення файлів в пакетах

Таблиця 3.1

Опис файлів, які знаходяться в пакеті configuration

Пакет (и)	Клас (и), файл (и)	Метод (и)
configuration	***CustomLoginSuccessHandler*** – класс для реагування на процес авторизації у системі.	***handle ()*** – метод викликається під час авторизації для перевірки та переадресації. ***determineTargetUrl ()*** – виконує процес валідації користувача.
	SecurityConfiguration – класс налаштувань перевірок та процесів реагування на запити користувачів.	***daoAuthenticationProvider ()*** – встановлення базового провайдера авторизації. ***userDetailsService ()*** – створює компонент деталізації даних для авторизації. ***bCryptPasswordEncoder ()*** – створює компонент для шифрування та дешифрування паролю. ***persistentTokenRepository ()*** – зберігає дані у пам'ять машини. ***configure ()*** – налаштування перевірок та розмежування доступу.

Таблиця 3.2

Опис файлів, які знаходяться в пакеті controllers

Пакет (и)	Клас (и), файл (и)	Метод (и)
controllers	***DashboardController*** – клас контроллер для сторінки завдань.	***initPage(@ModelAttribute User user, Model model)***– метод ініціалізації сторінки. ***updateDashboardData(@RequestBody String jsonTasks, Model model)***– метод оновлення даних дошки задач.
	FirstPageController – клас контроллек для вхідної сторінки.	***initPage(@ModelAttribute User user, Model model)***– метод ініціалізації сторінки.

Продовження таблиці 3.2

controllers	***GlobalControllerExceptionhandler*** – клас обробник базових помилок.	***handleHttpSessionRequiredException() –*** метод обробки помилки сесії сторінки. ***handleNotFound()*** – метод обробки помилки сторінки. ***NotFoudPage(Model model)*** – метод обробки помилки сторінки.
	HomeController – клас для сторінки особистої інформації користувача.	***sayHello(@ModelAttribute User user, Model model)–*** метод ініціалізації сторінки.
	LoginController– клас контроллер для модулю авторизації.	***initPage(Model model, String error, String logout, HttpSession httpSession, SessionStatus sessionStatus, HttpServletRequest request)–*** метод ініціалізації сторінки. ***loginProcess(@ModelAttribute User user, Model model, BindingResult bindingResult, HttpSession httpSession)–*** метод авторизації у системі. ***sayHello(@ModelAttribute User user, Model model)–*** метод ініціалізації сторінки. ***logoutProcess(HttpSession httpSession, SessionStatus sessionStatus)–*** метод виходу з системи.
	ProjectsController– клас контроллер для сторінки проектів.	***initPage(@ModelAttribute User user, Model model) –*** метод ініціалізації сторінки. ***addProject(@RequestParam(name = "project_name", defaultValue = "") String project_name, @ModelAttribute User user, Model model) –*** метод додавання проекту. ***loadPersonalProjectPage(@PathVariable String user_id, @PathVariable String project_id, @ModelAttribute User user, Model model) –*** завантаження сторінки проекту. ***removeProject(@RequestParam(name = "projectId", defaultValue = "") String projectId, @ModelAttribute User user, Model model***) – видалення проекту. ***addUserToProject(@RequestParam(name***

		= "membermail", defaultValue = "") String membermail, @PathVariable String id, @ModelAttribute User user, Model model) – додавання користувача до проекту. ***uploadProjectFile(@RequestBody MultipartFile file, @PathVariable String id, @ModelAttribute User user, Model model)–*** додавання файлу до проекту проекту. ***removeUserFromProject(@RequestParam(name = "projectId", defaultValue = "") String projectId, @RequestParam(name = "usermail", defaultValue = "") String usermail, @ModelAttribute User user, Model model***) – видалення користувача з проекту. ***initPrjTasksPage(@PathVariable String project_id, @ModelAttribute User user, Model model) –*** завантаження дошки завдань проекту. ***saveProjectTasks(@RequestBody String jsonTasks, @PathVariable String project_id, @ModelAttribute User user, Model model) –*** збереження дошки завдань проекту.
controllers	***RegisterController*** – клас контроллер для модулю реєстрації.	***initPage(Model model, String error, HttpSession httpSession, SessionStatus sessionStatus)*** – метод , завантажує сторінку реєстрації. ***registerProcess(@ModelAttribute("user") User user, Model model, BindingResult bindingResult)*** – метод , реєстрації користувача у системі.
	ReportsController – клас контроллер для сторінок звітів.	***getProjectReport(@PathVariable String project_id, @ModelAttribute User user, Model model, HttpServletResponse response)*** – метод , котрий розрахований для генерації звіту з проекту.

Продовження таблиці 3.2

	StatisticsController– клас контроллер для сторінок статистики.	***getStatisticPage (@ModelAttribute User user, Model model)*** – метод , котрий розрахований для декількох кнопок, тобто анімація буде працювати для декількох кнопок. ***getProjectStatistic (@PathVariable String project_id, @ModelAttribute User user, Model model)*** – метод , котрий розрахований для декількох кнопок, тобто анімація буде працювати для декількох кнопок.
	UpdateUserController– клас контроллер для модулю зміни особистої інформації.	***handleUploadImage (@RequestBody MultipartFile file, @ModelAttribute User user, Model model)*** – метод , котрий розрахований для зміни зображення користувача. ***updateUserInfo (@ModelAttribute User user, Model model)*** – метод , котрий розрахований для зміни особистої інформації користувача. ***deactivateUser (@ModelAttribute User user)*** – метод , котрий розрахований для деактивації користувача у системі.

Таблиця 3.3

Опис файлів, які знаходяться в пакеті model

Пакет (и)	Клас (и), файл (и), інтерфейс (и)	Метод (и)
model	***Project –*** логічна сутність проекту.	Стандартні методи об’єкту Pojo.
	ProjectFile – логічна сутність проектного файлу.	Стандартні методи об’єкту Pojo.
	Role – логічна сутність проектної ролі.	Стандартні методи об’єкту Pojo.
	User – логічна сутність користувача системи	Стандартні методи об’єкту Pojo.

Таблиця 3.4

Опис файлів, які знаходяться в пакеті persistence

Пакет (и)	Клас (и)	Метод (и)
persistence	***PersistenceJpaConfig*** – клас, який описує підключення та логіку роботи з базою даних при підключенні	***entityManagerFactory()*** – метод повертає менеджер сутностуй для обробки логічних компонентів. ***transactionManager(LocalContainerEntityManagerFactoryBean entityManagerFactoryBean)*** – метод, який повертає менеджер транзакцій фреймворку. ***exceptionTranslation()*** – метод, який помилки роботи з базою даних на фреймворк контекст. ***getHibernateProperties()*** – метод, який повертає властивості для сериалізування. ***dataSource()*** – метод, який повертає об’єкт підключення до бази даних.

Таблиця 3.5

Опис файлів, які знаходяться в пакеті repository

Пакет (и)	Клас (и)	Метод (и)
repository	***ProjectFileRepository*** – інтерфейс, який відповідає за роботу з об’єктами та базою даних.	Немає методів.
	ProjectRepository – інтерфейс, який відповідає за роботу з об’єктами та базою даних.	Немає методів.
	RoleRepository – інтерфейс, який відповідає за роботу з об’єктами та базою даних.	Немає методів.
	UserRepository – інтерфейс, який відповідає за роботу з об’єктами та базою даних.	Немає методів.

Таблиця 3.6

Опис файлів, які знаходяться в пакеті services

Пакет (и)	Клас (и), файл (и)	Метод (и)
services	***UserService*** – інтерфейс, який описує базові CRUD операції з логічною сутністю користувача.	***addUser(User user)*** – додати користувача до бази даних. ***delete(int id)*** – видалити користувача по ідентифікатору. ***getByUserName(String username)*** – знайти користувача по імені. ***getByUserMail(String usermail)*** – знайти користувача по електронній адресі. ***editUser(User user)*** – змінити користувача. ***getAllUser()*** – взяти список користувачів.
	ProjectService– інтерфейс, який описує базові CRUD операції з логічною сутністю проекту.	***addProject(Project project)*** – додати проект до бази даних. ***removeProject (int id)*** – видалити проект по ідентифікатору. ***removeProject(Project project)*** – видалити проект. ***editProject(Project project)*** – змінити проект. ***getProject(int id)*** – взяти проект по ідентифікатору. ***getAll()*** – взяти список проектів.
	ProjectFileService – інтерфейс, який описує базові CRUD операції з логічною сутністю проектного файлу.	***addProjectFile (ProjectFile projectFile)*** – додати проектний файл до бази даних. ***removeProjectFile (int id)*** – видалити проектний файл по ідентифікатору. ***removeProjectFile (ProjectFile projectFile)*** – видалити проектний файл. ***editProjectFile (ProjectFile projectFile)*** – змінити проектний файл. ***getProjectFile (int id)*** – взяти проектний файл по ідентифікатору. ***getAll()*** – взяти список проектів.

Таблиця 3.7

Опис файлів, які знаходяться в пакеті util

Пакет (и)	Клас (и), файл (и)	Метод (и)
util	***MediaTypeUtils*** – клас, який відповідає за конвертацію медіа-файлів до веб-контенту.	***getMediaTypeForFileName(ServletContext servletContext, String fileName)***–метод який відповідає за конвертацію медіа-файлів до веб-контенту.
	ProjectUtils – клас парсить рядок формату json та дістає потрібні дані.	***getTasksData(String JSON_TASKS)*** – метод повертає таблицю відповідностей кількості задач та назви колонки. ***getTaskCount(String JSON_TASKS)*** – метод повертає кількість задач у дошці.

Таблиця 3.8

Опис файлів, які знаходяться в пакеті validate

Пакет (и)	Клас (и), файл (и)	Метод (и)
validate	***UserValidator*** - клас, який відповідає валідацію даних користувача.	***validate(Object o, Errors errors)*** – метод перевірки та реакції на введення даних користувача.

Наступним етапом було створено базу даних на сервісі heroku, сервіс дає змогу користувачам підєднатися до віддаленої бази даних, за допомогою посилання та даних користувача, який є адміністратором бази даних. Приклад створення таблиці наведений на рисунках 3.3 та 3.4.

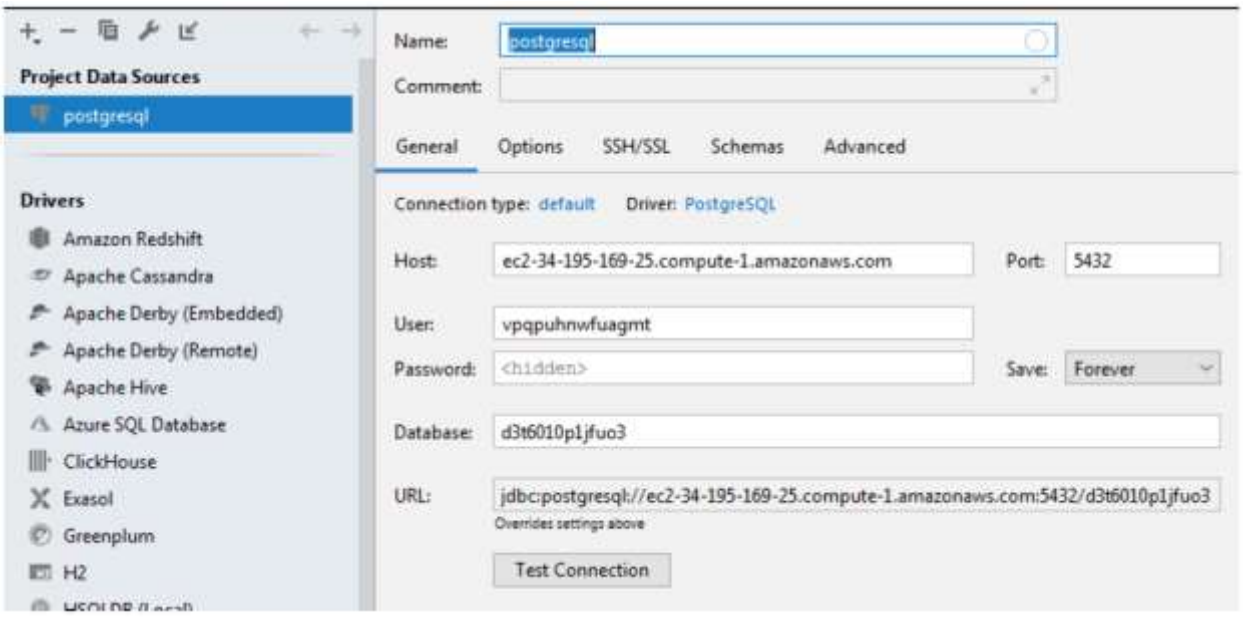


Рис. 3.3 Підключення до бази даних

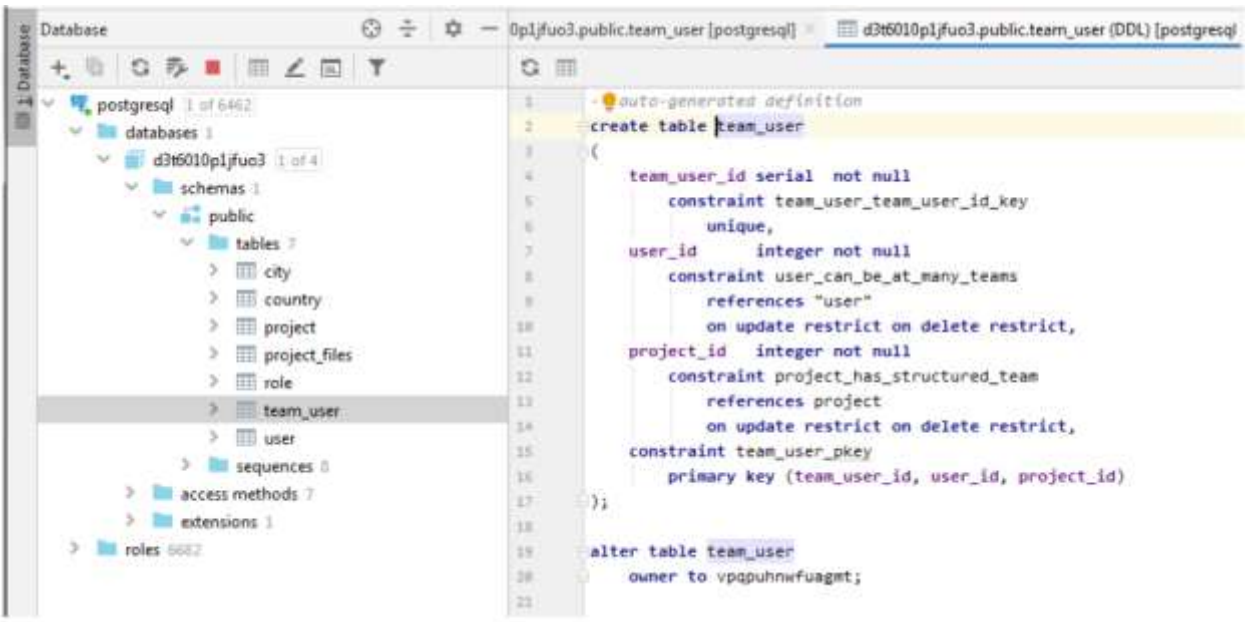


Рис. 3.4 Створення таблиці команди проекту

Після проектування та завантаження таблиць до сервісу, було створене підключення до віддаленої бази даних представленої на рисунку 3.5.

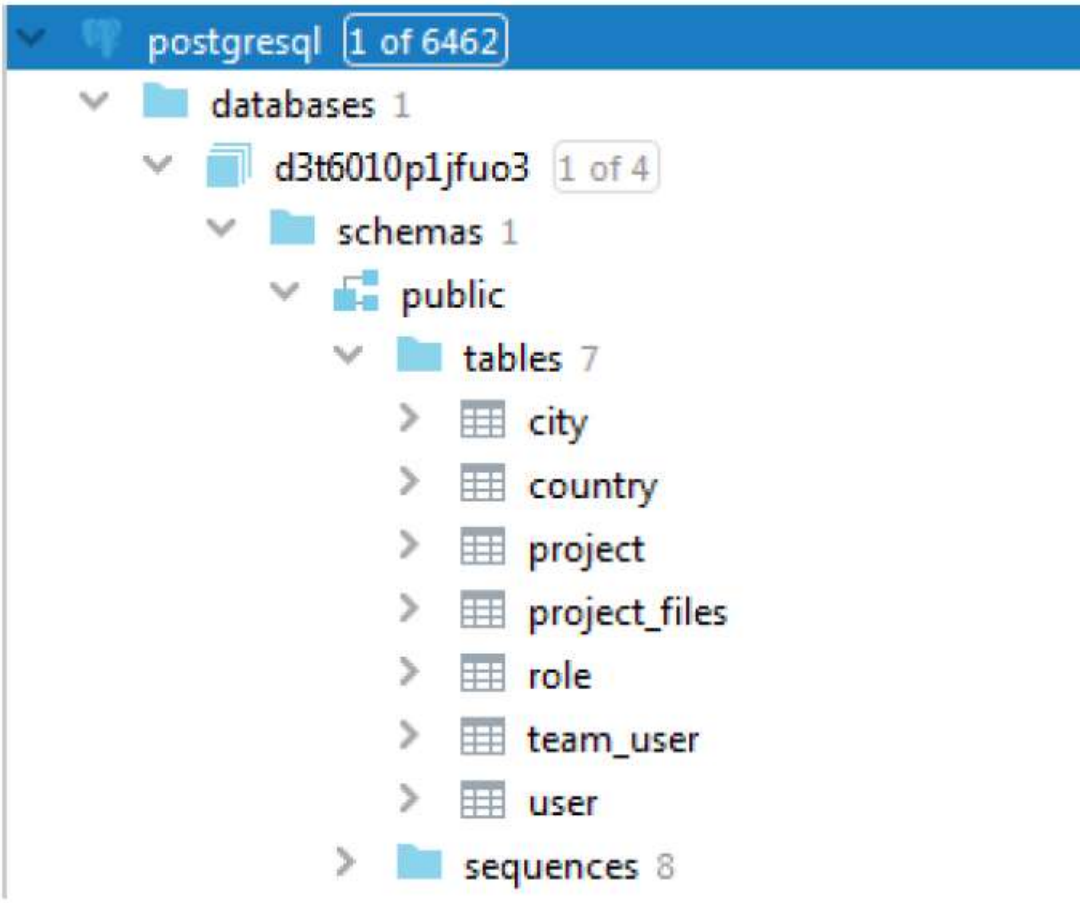


Рис. 3.5 База даних

Після шифрування даних сервісом heroku, отримано назву ***d3t6010p1jfuo3***. Таблиці мають назви: ***city, country, project, team_user, user, role, project_files***. Опишемо більш детально кожну із таблиць.

Таблиця 3.9

Опис таблиці user

Назва таблиці	Опис даних таблиці
user – сутність користувача.	***user_id*** – ідентифікатор користувача.
	username – ім'я користувача.
	password – пароль користувача.
	user_img – шлях до зображення користувача.
	usermail – електронна адреса користувача.
	userphone – телефон користувача.
	company – компанія.
	position – посада користувача.
	active – чи активний акаунт.
	userBoardJson – дошка завдань.

Таблиця 3.10

Опис таблиці team_user

Назва таблиці	Опис даних таблиці
team_user – збереження даних про запитання.	***team_user_id*** – ідентифікатор команди.
	user_id – ідентифікатор користувача.
	project_id – ідентифікатор проекту.
	role_id – ідентифікатор ролі.
	city_id – ідентифікатор міста.
	county_id – ідентифікатор країни.

Таблиця 3.11

Опис таблиці country

Назва таблиці	Опис даних таблиці
country – сутність країни.	***country_id*** – ідентифікатор країни.
	country – назва країни.

Таблиця 3.12

Опис таблиці role

Назва таблиці	Опис даних таблиці
***role* –** сутність ролі.	***role_id*** – ідентифікатор ролі.
	name – назва ролі.

Таблиця 3.13

Опис таблиці city

Назва таблиці	Опис даних таблиці
***city* –** сутність ролі.	***city_id*** – ідентифікатор міста.
	city – назва міста.
	country_id – ідентифікатор країни.

Таблиця 3.14

Опис таблиці project

Назва таблиці	Опис даних таблиці
project – таблиця, в якій зберігаються шляхи до відео фрагментів.	***project _id*** – ідентифікатор проекту.
	project_name– назва проекту.

Таблиця 3.15

Опис таблиці project_files

Назва таблиці	Опис даних таблиці
***project_files* –** сутність ролі.	***project_files _id*** – ідентифікатор файлу.
	file_path – шлях до файлу.
	project_id – ідентифікатор проекту.

Після реалізації бази даних необхідно завантажити створений веб-додаток на сервер. За хостинг та віддалену базу даних було обрано Service Heroku. Сервіс дозволяє безкоштовно на початку створити та використовувати базу даних PostgreSQL, та контейнер веб-додатку на базу Apache.

Знайти додаток можна за посиланням https://manluck.herokuapp.com/.

Завантажений веб-додаток на сервер представлений на рисунку 3.6.

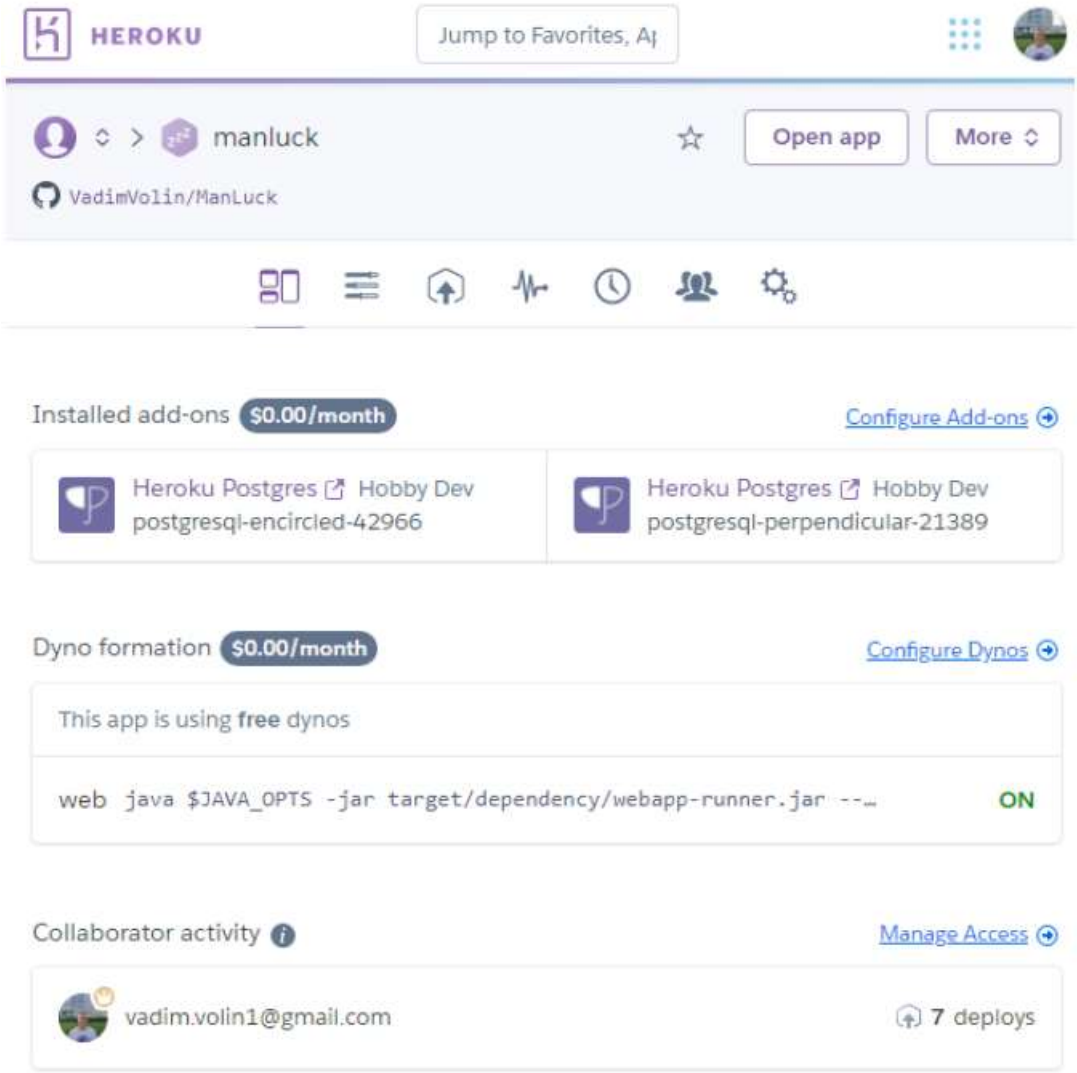


Рис. 3.6 Веб-додаток на сервері heroku

3.3 Використання програмного додатку

На даному етапі буде продемонстровано процес роботи програми, навігацію по системі та використання функціональних можливостей системи.

При першому відвідуванні системи користувач потрапить на сторінку привітання. На рисунку 3.7 можна спостерігати дану сторінку. В цьому вікні користувач обирає спосіб входу до системи.

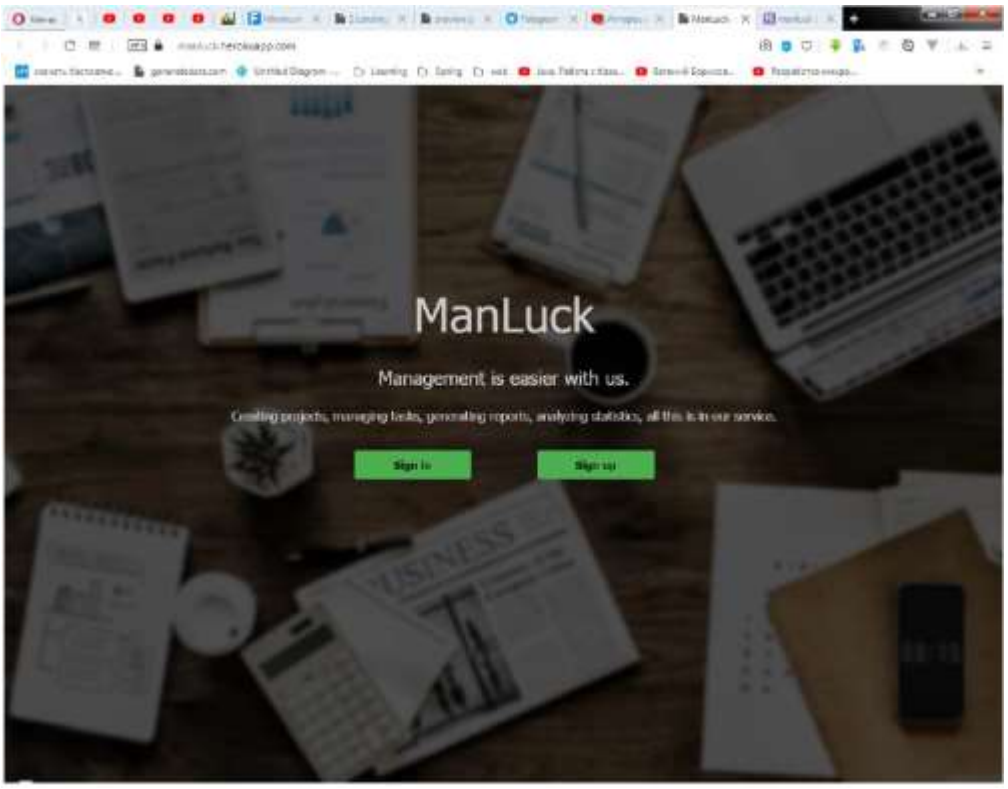


Рис. 3.7 Сторінка привітання

Якщо, користувач натиснув на кнопку авторизації, то він буде перенаправлений на сторінку авторизації. На рисунку 3.8 представлена ця сторінка.

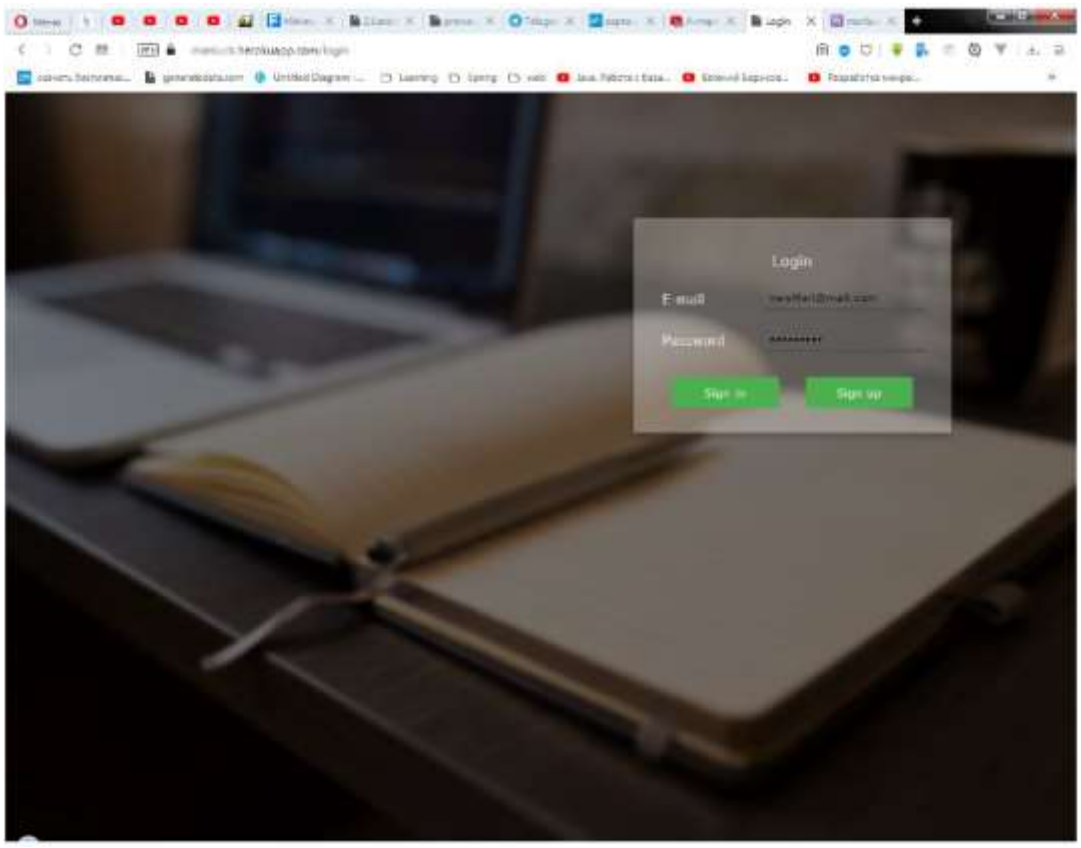


Рис. 3.8 Вікно авторизації

Якщо користувач ввів невірні дані або залишив обов’язкове поле пусти, то він отримає попередження. На рисунку 3.9 зображено вікно попередження.

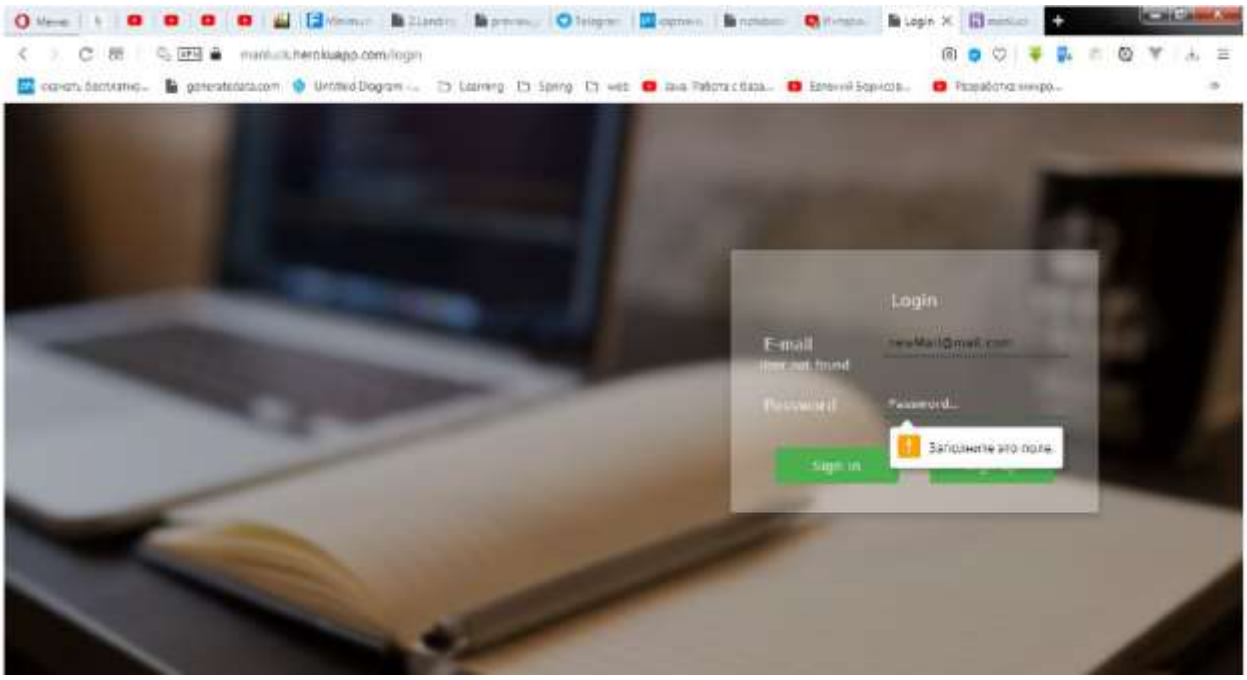


Рисунок 3.9 Вікно попередження

Якщо користувач обрав реєстрацію, то він буде перенаправлений на сторінку реєстрації, яку зображено на рисунку 3.10.

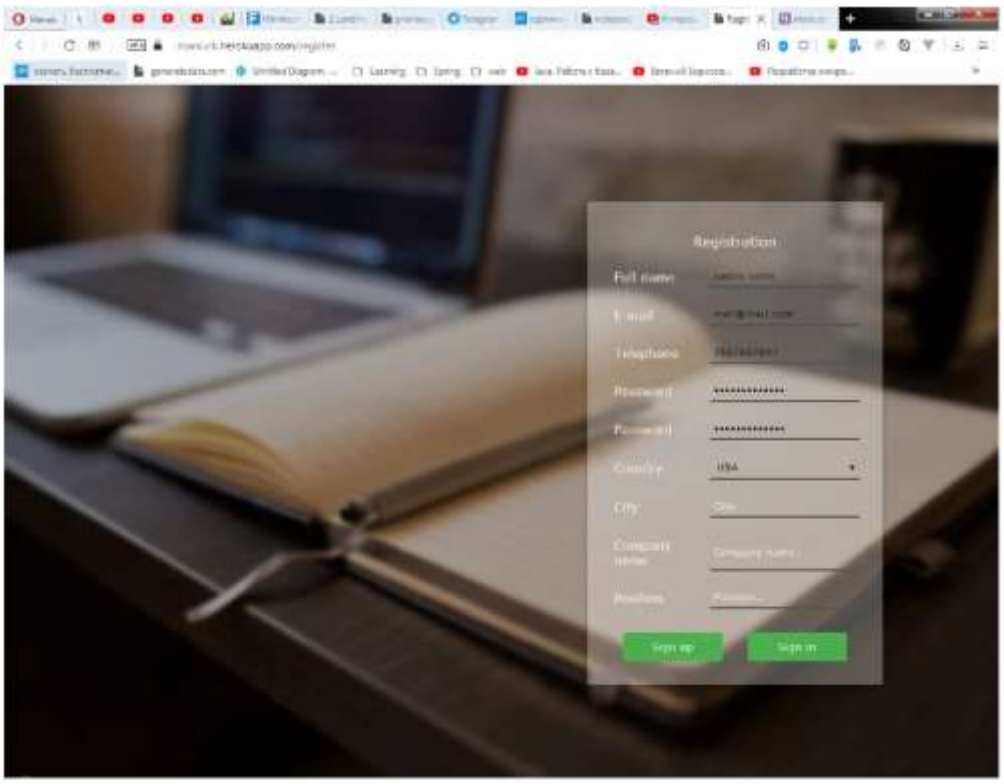


Рис. 3.10 Вікно реєстрації

Авторизувавшись чи зареєструвавши акаунт користувач буде перенаправлений на сторінку зі статистикою та базовою інформацією користувача, яку зображено на рисунку 3.11.

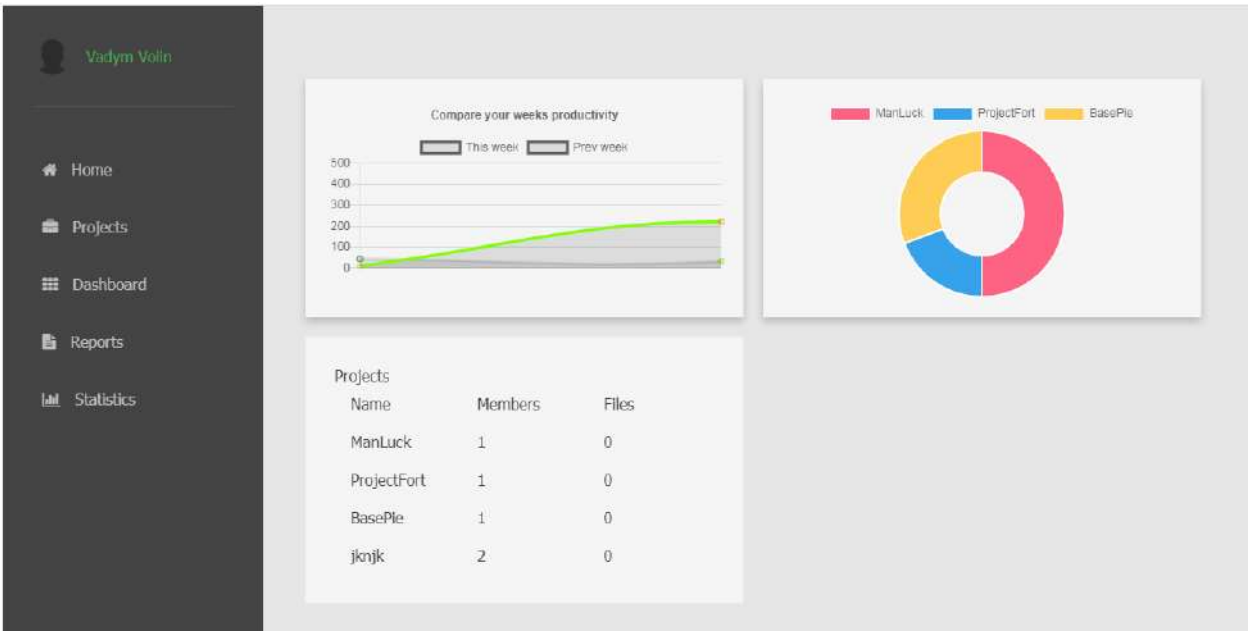


Рис. 3.11 Початкова сторінка

На домашній сторінці зображено форми для зміни даних користувача чи його фото. Змінивши дані у полях для введення, відразу на сервер буде відправлено запит, для швидкої зміни інформації. Сторінку зображено на рисунку 3.12.

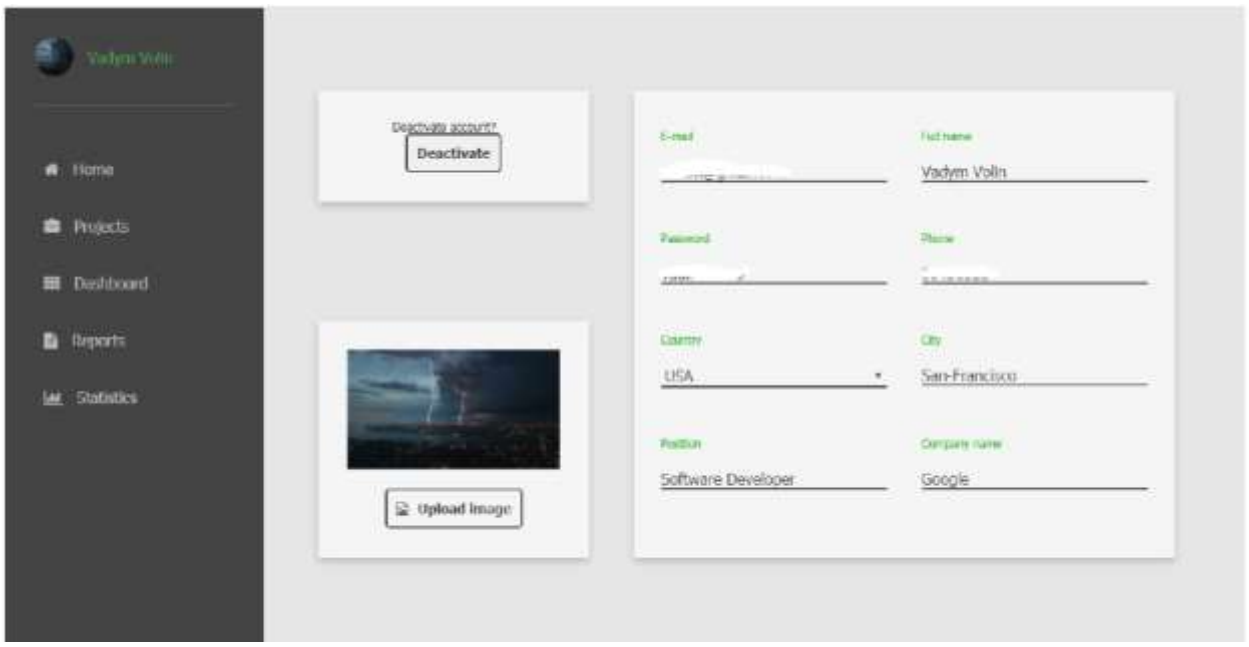


Рис 3.12 Домашня сторінка

Перейшовши на панелі навігації на сторінку проектів, користувача перенаправить на сторінку з плиткою проектів. На сторінці знаходить плитка проектів та кнопка додавання проекту до плитки у горі, видалити проект з власного акаунту можна натиснувши на кнопку у правому куті проекту, якщо проект не матиме учасників команди, то він буде видалений з системи, сторінку зображено на рисунку 3.13.

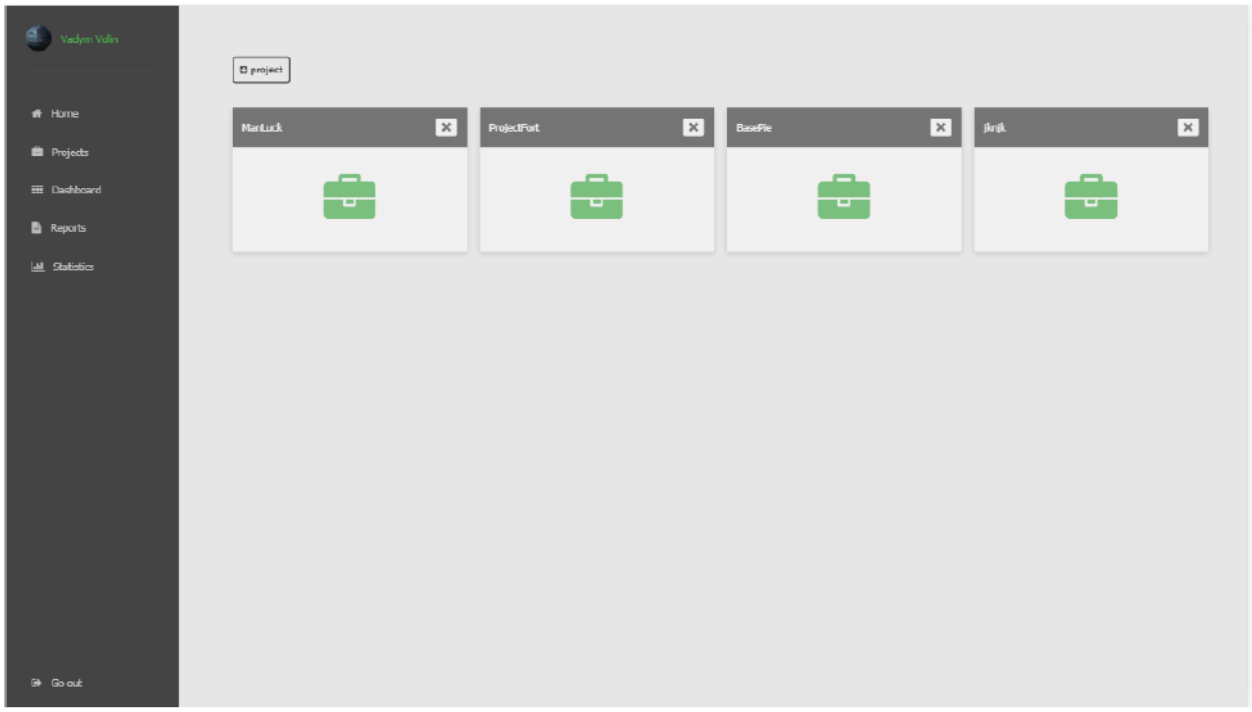


Рис. 3.13 Проектна сторінка

Обравши один з проектів, користувач буде перенаправлений на детальну сторінку проекту. На верхній панелі знаходяться три плитки, а саме панель задач проекту, формування звіту, перегляд статистики проекту відповідно. Додати користувача чи файл можна натиснуваши відповідну кнопку. Видалити користувача можна натиснувши кнопку хрест на відповідному користувачеві, сторінку зображено на рисунку 3.14.

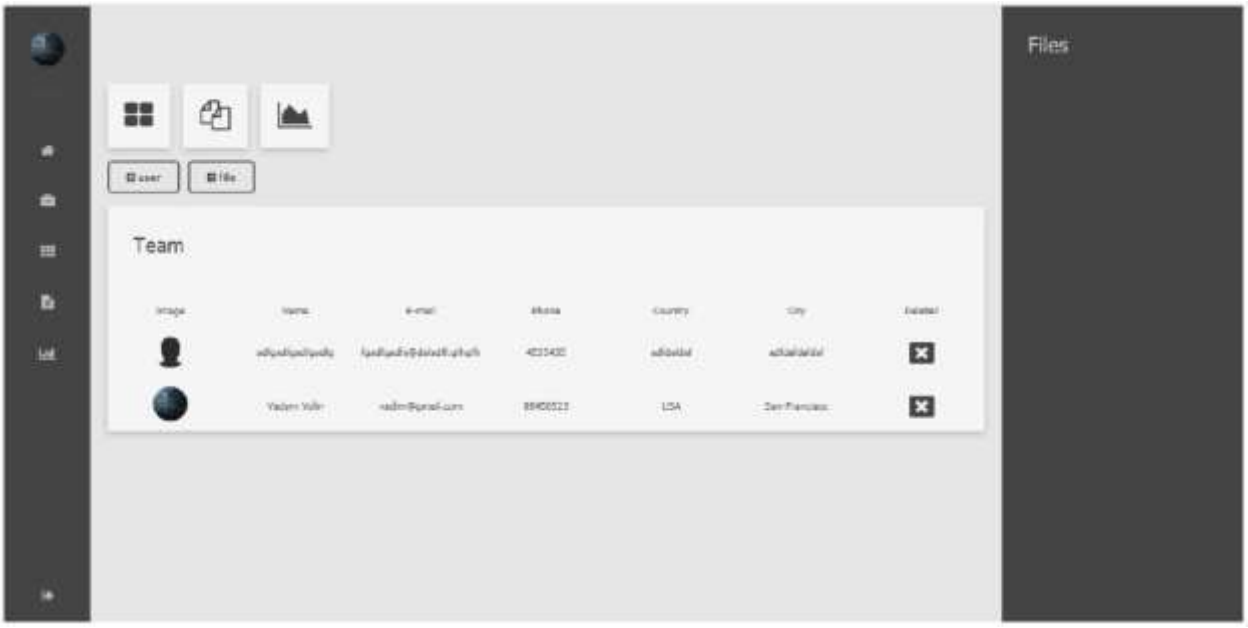


Рис. 3.14 Детальна сторінка проекту

Натиснувши на панель задач, користувача буде перенаправлено на сторінку з списками задач проекту, додавати списки можна натиснувши на відповідну кнопку, видалити натиснувши на кнопку хрест у правому верхньому куті списку, додати задачу можна натиснувши на кнопку додати запис, видалити можна просто перетягнувши запис у вільне місце. Запис можна перетягувати у інший список, списки також можна змінювати, сторінку задач зображено на рисунку 3.15.



Рис. 3.15 Сторінка задач проекту

Обравши панель статистики, користувач матиме змогу переглянути статистику проекту, з продуктивністю команди та процесом розвитку проекту за рахунок підрахунку к-сті задач ствоюравних за час проекту, та обігу файлів, сторінку зображено на рисунку 3.16.

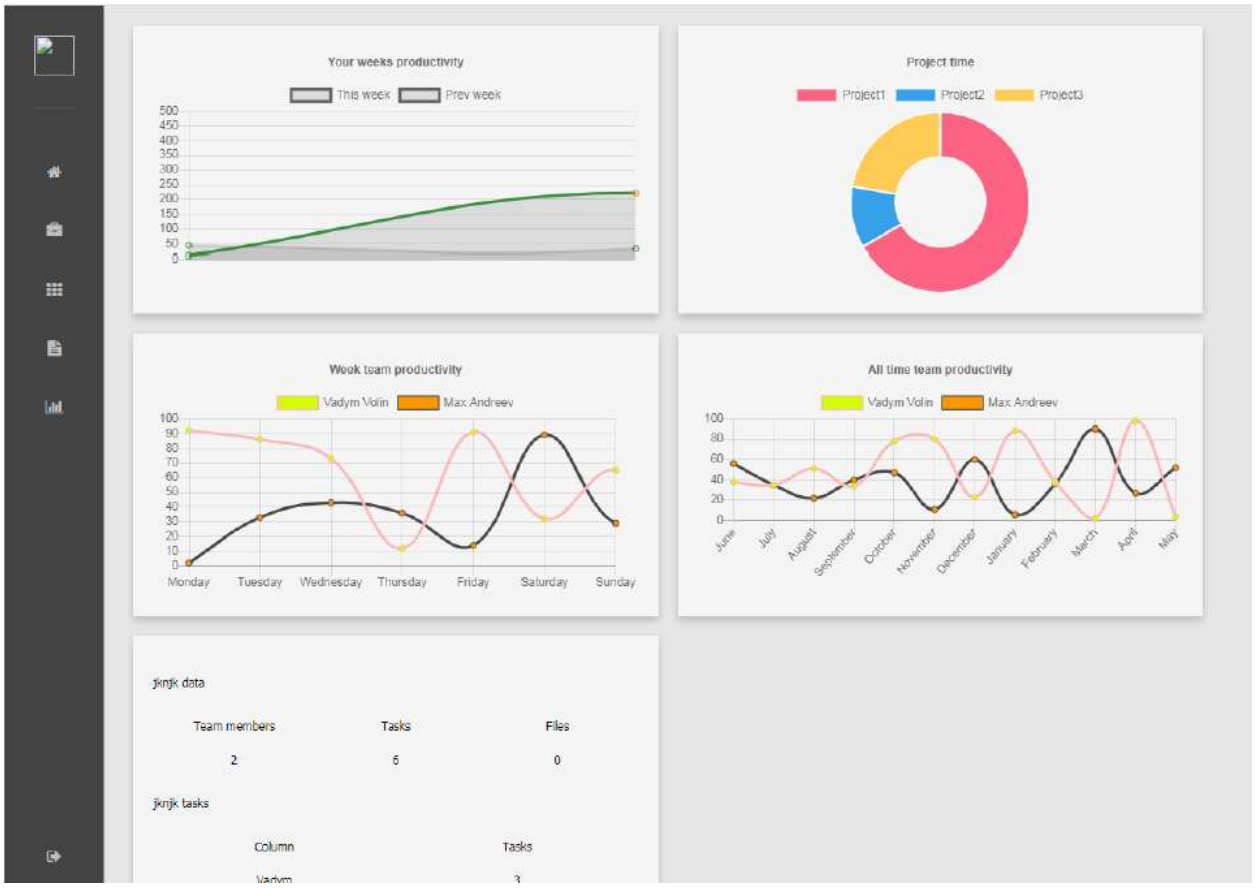


Рис. 3.16 Сторінка статистики проекту

Створення проекту проходить по натисненню на кнопку додати проект, з’являється форма, у якій користувач матиме змогу ввести назву проекту, натиснувши кнопку створити проект буде створений у системі, процес зображено на рисунку 3.17.



Рис. 3.17 Процес створення проекту

Якщо проект успішно створено, то з’являється повідомлення, приклад повідомлення зображено на рисунку 3.18 та 3.19.

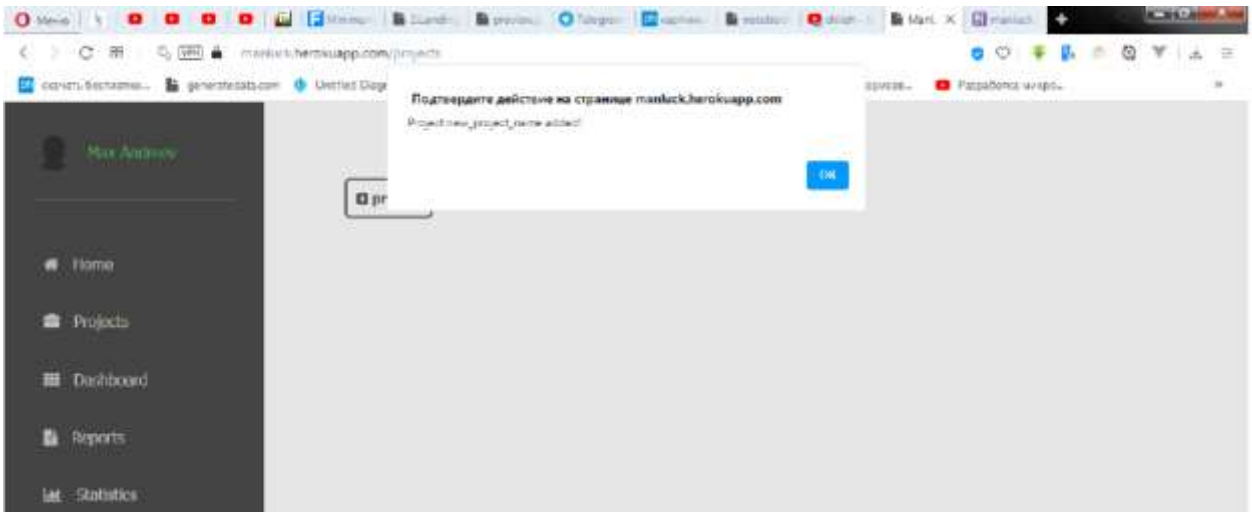


Рис. 3.18 Детальна сторінка проекту

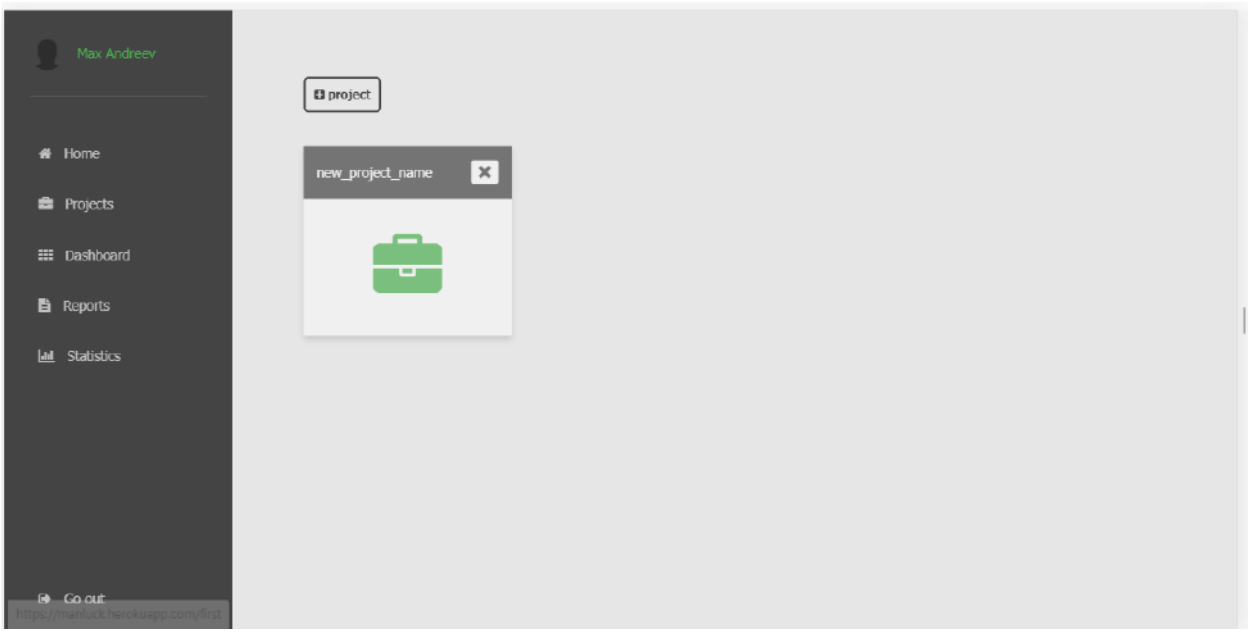


Рис. 3.19 Новоутворений проект

Обравши на панелі навігації сторінку звітів, користувач перенаправляється на список проектів, де має змогу обрати звіт для конкретного проекту, завантажити .pdf звіт можна натиснувши на файл у правій частині проекту, процес зображено на рисунку 3.20.

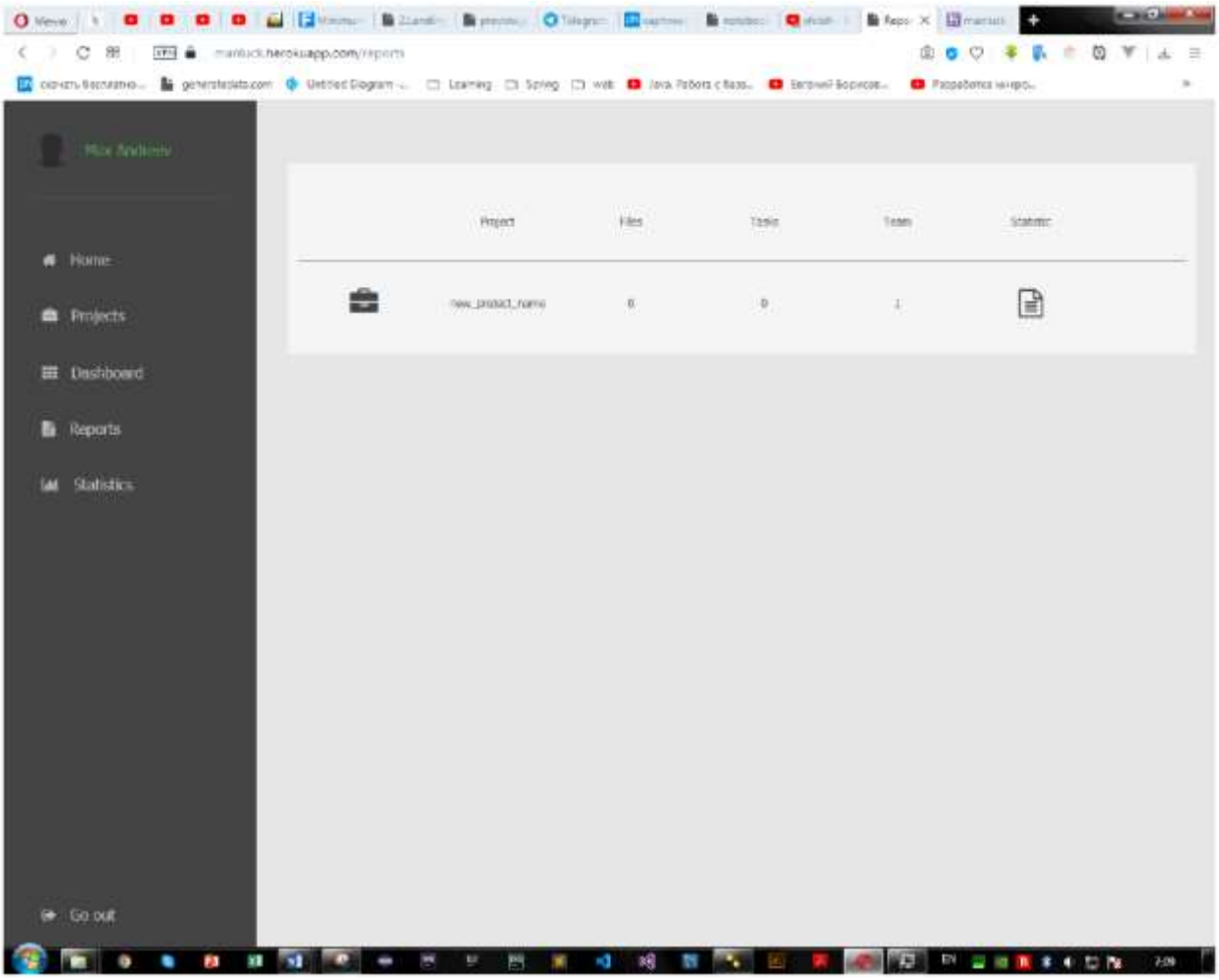


Рис. 3.20 Детальна сторінка проекту

ВИСНОВКИ

При виконанні дипломної роботи мною було проведено аналіз останніх досліджень в області проектного управління та проаналізовано предметну область проекту.

На наступному етапі було знайдено аналоги інформаційної системи та проведено дослідження їх функціоналу, знайдені переваги та недоліки цих сервісів перед нашою системою, в результаті цього етапу було доведено доцільність створення власної системи.

Після аналізу сервісів з аналогічними можливостями було виконано формування постановки задачі, описано базовий функціонал системи та обрано інструменти для реалізації системи.

Для налагодження процесу побудови системи управління виконанням проектів було детально описано процес проектування системи.

Для початку було проведено структурно-функціоналтне моделюваня та створено контекстну діаграму та діаграму декомпозиції з метою деталізованого огляду бізнес-логіки системи.

Для збереження інофрмації змоделбовано і розроблено власну базу даних.

Наступним етапом було створення діаграми варіантів використання, яка продемонструє послідовність дій при роботі з системою.

Після побудови діаграми варіантів використання було створено базові діаграми діяльності, які є діаграмами поведінки, що показують процес роботи об’єкта з урахуванням умов потоку виконання.

Наступним етапом була створена діаграма класів системи, яка визначає типи об’єктів у системі та різні типи зв’язків, що існують серед них. Це дозволяє переглядати систему на високому рівні.

За отриманими моделями виконано програмну реалізацію веб-додатку, який повністю відповідає технічному завданню та виконую уві висунуті вимоги.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Управління проектами та системи управління проектами. Поняття управління проектами: реферат. URL: https://ua-referat.com/Управління_проектами_та_системи_управління_проектами

2. Системи управління проектами. Кращі системи управління проектами: стаття. URL: https://www.fewskills.com/workflow-project-app/

3. Управління проектами та системи управління проектами. Системи управління проектами: реферат. URL: https://ukrbukva.net/page,3,1455-Upravlenie-proektami-i-sistemy-upravleniya-proektami.html

4. Архітектура клієнт-сервер на основі додатка відділу аспірантури та докторантури ОНАХТ. архітектура клієнт-сервер: реферат. URL: https://journals.onaft.edu.ua/index.php/atbp/article/view/974

5. Чи актуально на сьогодні моделювання в IDEF0? Projectimo. Моделювання. URL: http://projectimo.ru/biznes-processy/idef0.html

6. Основи UML — діаграми використання. Блог розробника. URL: https://pro-prof.com/archives/2594

7. Activity Diagrams. UML diagrams. URL: https://www.uml-diagrams.org/activity-diagrams.html

8. Entity Relationship Diagram – ER Diagram in DBMS. Be-ginnersBook. URL: https://beginnersbook.com/2015/04/e-r-model-in-dbms/

9. UML Class Diagram Tutorial with Examples. Guru99. URL: https://www.guru99.com/uml-class-diagram.html

10. Fan Q., Yang J., Hua G. A generic deep architecture for single image reflection removal and image smoothing 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017. P. 3258–3267.

11. Wei K., Yang J., Fu Y. Single image reflection removal exploiting misaligned training data and network enhancements 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2019. P. 8170–8179.

12. Адванта – онлайн система управління проектами URL: http://www.advanta-group.ru/

13. Андрєєва Т. Є. Проектний менеджмент як засіб досягнення мети підприємства. *Вісник економіки транспорту і промисловості.* 2011. № 34. С. 364-370.

14. Батенко Л. П.,О. А. Загородніх, В. В. Ліщинська Управління проектами: навч. посіб. К.: КНЕУ, 2003. 231 с.

15. Безверхнюк Т. М. Проектно-орієнтований підхід як нова філософія організації управління державними програмами і проектами. *Науковий вісник Академії муніципального управління : збірник наукових праць.* Вип. 3/2011. Київ: Академія муніципального управління, 2011. С. 17-24.

16. Безверхнюк Т. М. Публічна сфера як предметна область проектного менеджменту. *Збірник наукових праць Херсонського національного технічного університету.* №2(3). 2010. URL: http://www.nbuv.gov.ua/old_jrn/Soc_Gum/Apdup/2010_2/2-3-2.pdf

17. Yi R., Wang J., Tan P. Automatic fence segmentation in videos of dynamic scenes. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016. P. 705–713.

18. Застосування інформаційних систем для управління проектами на підприємствах малого та середнього бізнесу. URL: https://otherreferats.allbest.ru/management/00625670_0.html#text

19. YiChang Shih, Krishnan D., Durand F., Freeman W. T. Reflection removal using ghosting cues. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015. P. 3193–3201.

20. Бушуев С. Д.,С. Д. Бушуев, В. В. Морозов. Динамічне лідерство в управлінні проектами: Монографія. Українська асоціація управління проектами. К., 1999. 312 с.

21. Бушуев С. Д., С. Д. Бушуев, Н. С. Бушуева. Управління проектами: стан та перспективи: Матеріали IV міжнародної науково-технічної конференції. Миколаїв: НУК, 2008. С. 26-27.

22. Волін В.В., Ващенко С.М. Web-додаток підтримки управління виконанням проектів в компанії. / В.В. Волін, С.М. Ващенко С.М. // Інформатика, математика, автоматика: матеріали та програма науково-технічної конференції, м. Суми, 20 – 24 квітня 2020 р. Суми: Сумський державний університет, 2020. С. 108.