

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій, обліку та фінансів  
Кафедра комп'ютерних технологій  
і моделювання систем

Кваліфікаційна робота  
на правах рукопису

Чернюк Максим Юрійович

УДК 004.732.161.1

## **КВАЛІФІКАЦІЙНА РОБОТА**

**WEB-розробка для управління послугами в сфері ІТ**

122 «Комп'ютерні науки»

Подається на здобуття освітнього ступеня бакалавр

Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання на  
відповідне джерело

---

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи  
Ковальчук Майя Олегівна,  
доцент кафедри КТіМС, к.п.н.

Житомир – 2024

**Висновок кафедри**

---

за результатами попереднього захисту:

---

**Протокол засідання кафедри**

---

№ \_\_\_\_\_ від «\_\_\_\_\_» \_\_\_\_\_ 20\_\_\_\_ р.

Завідувач кафедри \_\_\_\_\_

---

(науковий ступінь, вчене звання)      (підпис)      (прізвище, ім'я, по батькові)  
«\_\_\_\_\_» \_\_\_\_\_ 20\_\_\_\_ р.

**Результати захисту кваліфікаційної роботи**

Здобувач вищої освіти \_\_\_\_\_ захистив (ла)  
(прізвище ,ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою \_\_\_\_\_  
за шкалою ECTS \_\_\_\_\_  
за національною шкалою \_\_\_\_\_

**Секретар ЕК**

---

(науковий ступінь, вчене звання)      (підпис)      (прізвище, ім'я, по батькові)

## АНОТАЦІЯ

Чернюк М. Ю. Розробка WEB-застосунку для управління послугами в сфері ІТ.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра за спеціальністю 122 – Комп’ютерні науки. – Поліський національний університет, Житомир, 2024.

Необхідність впровадження ефективних засобів управління завданнями та координація поведінки команди в онлайн-середовищі. Особливо в Україні, де ринок праці стає все більш глобальним і віртуальним.

Мета кваліфікаційної роботи: Покращення ефективності роботи компанії шляхом забезпечення оптимального розподілу завдань між співробітниками в ІТ-компаніях.

Об’єкт дослідження: процес розробки WEB-застосунку для управління послугами в сфері ІТ.

Предмет дослідження: засоби та методи, які використовуються при розробці WEB-застосунку для управління послугами в сфері ІТ.

Ключові слова: WEB-застосунок, управління проектами, розподілення завдань, автоматизація завдань, управління послугами, React, Node.js.

## ABSTRACT

Chernyuk M.Yu. Development of a WEB application for managing services in the field of IT.

Qualification work for obtaining a bachelor's degree in specialty 122 - Computer science. – Polisska National University, Zhytomyr, 2024.

The need to implement effective task management tools and coordinate team behavior in an online environment. Especially in Ukraine, where the labor market is becoming increasingly global and virtual.

The purpose of the qualification work: Improving the efficiency of the company's work by ensuring the optimal distribution of tasks between employees in IT companies.

The object of the study: the process of developing a WEB application for managing services in the field of IT.

The subject of the study: tools and methods used in the development of a WEB application for managing services in the field of IT.

Keywords: WEB application, project management, task allocation, task automation, service management, React, Node.js.

## Зміст

ВСТУП.....	6
СКРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ .....	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ ДОСЛІДЖЕННЯ .....	9
1.1. Аналіз інформаційних потреб і визначення предметної області дослідження.....	9
1.2. Аналіз сучасних середовищ для web-розробки.....	14
1.3. Моделювання бізнес-процесів предметної області.....	17
Висновок до першого розділу .....	20
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ WEB-РОЗРОБКИ.....	21
2.1. Моделювання web-розробки .....	21
2.2. Проектування структури бази даних та інтерфейсу програмного продукту.....	23
2.3. Процес реалізації web-застосунку управління послугами в сфері ІТ .....	26
РОЗДІЛ 3. ТЕСТУВАННЯ ГРИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ .....	29
3.1. Інструкція користувачу web-застосунку управління послугами в сфері ІТ.....	29
3.2. Тестування web-розробки та аналіз результатів.....	31
Висновки до третього розділу .....	32
ЗАГАЛЬНІ ВИСНОВКИ.....	34
ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	36

## ВСТУП

ІТ-сферу часто пов'язують із діяльністю суб'єктів, причетних до процесів зберігання, опрацювання даних, а також управління даними. Розроблені цією сферою продукти характеризуються цілісністю і результатом поєднання програмно-технічних засобів, апаратного забезпечення, інформації та баз даних. Пропозицію на ринку ІТ-продуктів формують спеціалізовані підприємства ІТ-сфер, ІТ-служби неспеціалізованих підприємств, ІТ-консультанти, фахівці, що самостійно розробляють ІТ-продукти. У сучасних умовах ІТ-сфера показує стрімкий розвиток, внаслідок чого зростає попит на інформаційні системи, що забезпечують управління послугами в ІТ-компаніях. Саме тому, розвиток технологій та впровадження веб-застосунків стає ключовим фактором в оптимізації бізнес-процесів та підвищення продуктивності праці.

Актуальність обраної тематики визначається необхідністю впровадження ефективних засобів управління завданнями та координації поведінки команди в онлайн-середовищі. Особливо в Україні, де ринок праці стає все більш глобальним і віртуальним, важливі зручні та надійні веб-додатки, які можуть розподіляти завдання, контролювати продуктивність і забезпечувати ефективну комунікацію між співробітниками. Забезпечення ефективної взаємодії між співробітниками, розташованими в різних географічних регіонах, вимагає не тільки технологічних рішень, але дослідження оптимальних методів управління та спілкування в онлайн-середовищі. Важливо продумати деталі віртуального спілкування, взаємодії та адаптувати інструментарій до конкретних умов та потреб вашої команди. Вибір даної теми є виправданим, оскільки проблема ефективного управління завданнями за допомогою мережевих засобів в умовах віртуального робочого середовища вивчені недостатньо.

**Метою кваліфікаційної роботи** є забезпечення оптимального розподілу завдань між співробітниками в ІТ-компаніях. Для досягнення поставленої мети необхідно вирішити наступні **завдання**: проаналізувати інформаційні потреби і визначити предметну область дослідження; змоделювати бізнес-процеси WEB-застосунку для управління послугами в сфері ІТ; спроектувати інтерфейс та реалізувати функціонал; протестувати розроблений WEB-застосунок.

**Об'єктом дослідження** є процес розробки WEB-застосунку для управління послугами в сфері ІТ.

**Предметом дослідження** є засоби та методи, які використовуються при розробці WEB-застосунку для управління послугами в сфері ІТ.

Для досягнення поставлених цілей було використано наступні **методи дослідження**: *аналізу* (аналіз наукової та науково-методичної літератури з області дослідження; розробки рекомендацій щодо вдосконалення процесу розподілу завдань у віртуальних командах); *порівняння* (визначення переваг і недоліків існуючих рішень); *моделювання* (застосовували під час моделювання бізнес процесів предметної області); *тестування й аналітики* (дослідження працездатності розробки та оцінки її продуктивності).

**Практичне значення отриманих результатів.** Результати дослідження можуть бути використані як інноваційний засіб для забезпечення оптимального розподілу завдань між співробітниками в ІТ-компаніях.

**Наукова новизна:** полягає у застосуванні системного підходу до прийняття рішень в сфері управління ІТ-послугами, що ґрунтується на використанні веб-застосунку, який поєднує в собі два ключових елементи: ієрархічну композицію завдання, що розбиває складну проблему на більш дрібні, керовані завдання, для полегшення аналізу та прийняття рішень на кожному рівні; рейтингування альтернативних рішень, використовуючи різні критерії та методи оцінювання, веб-застосунок ранжирує можливі рішення, допомагаючи користувачеві вибрати оптимальний варіант.

Апробація результатів дослідження здійснювалася під час Міжфакультетської науково-практичної інтернет-конференції здобувачів вищої освіти і молодих вчених «Безпека, технології, інновації: нові горизонти» (м. Житомир, 2023 р.) та II Міжнародної науково-практичної конференції «Дизайн, візуальне мистецтво та творчість: сучасні тенденції та технології» (м. Запоріжжя, 2024 р.), за результатами яких опубліковано тези у збірниках конференцій.

Структура та обсяг роботи. Кваліфікаційна робота складається з вступу, трьох розділів, висновку та списку використаних джерел. Загальний обсяг роботи становить 37 сторінок.

## **СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ**

CRM – Customer Relationship Management

WEyB – World Wide Web

IT – Information Technology

WM – Work Management

SM – Service Management

CRM – Customer Relationship Management

SaaS – Software as a service

SQL – Structured Query Language

MySQL – My Structured Query Language

UML – Unified Modeling Language

DFD – Data Flow Diagram

IDEF0 – Integrated Definition for Function Modeling

IDEF3 – Integrated Definition for Function Modeling Three

PM – Project Manager



## **РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ ДОСЛІДЖЕННЯ**

### **1.1. Аналіз інформаційних потреб і визначення предметної області дослідження**

Предметною областю інформаційної системи для управління послугами в сфері ІТ є система управління задачами та проектами, що спрямована на покращення ефективності командної роботи, полегшення виконання поставлених завдань, оптимізації роботи компанії та забезпечення комунікації під час виконання роботи.

Існування застосунку для управління послугами в сфері ІТ в сучасному інформаційному середовищі, коли ІТ-компанії стикаються з високою динамікою та великим обсягом роботи, застосування системи розподілу завдань визначається необхідністю оптимізації та поліпшення ефективності робочих місць, підтримання комунікації, що особливо важливо під час дистанційної роботи [1]. Таким чином, виникає потреба у застосунках які вирішують проблеми розподілу ресурсів між співробітниками залежно від їх навичок, вирішують координацію великих команд та поліпшення роботи в команді [2]. Також він має відстежувати процес виконання завдань, забезпечувати аналітику та звітність для керівництва. Систематизуючи велику кількість завдань та проектів в ІТ-компанії, система для розподілу завдань стає необхідним інструментом для продуктивності, зменшення помилок, економії часу для реалізації проекту, забезпечення якості та ефективності її роботи в умовах постійних змін та вимог ринку [3].

Основними особливостями інформаційного забезпечення системи розподілу завдань в ІТ-компанії визначають необхідність оптимізації робочих процесів і підвищення ефективності. Система має обробляти великий обсяг інформації, здебільшого проектів, завдань, пріоритетів, ресурсів та термінів виконання, забезпечуючи ефективний потік інформації між співробітниками та керівництвом. Кожен користувач має мати різний рівень доступу, що зробить систему більш безпечною та забезпечує конфіденційність даних. Система має взаємодіяти з різними джерелами інформації, такими як бази даних та інструментами інтеграції, щоб надавати актуальну інформацію. Автоматизація процесів, таких як розсилки

повідомлень та нагадувань, спрямовані на оптимізацію робочого процесу, а можливість аналітики та звітність дозволяють досліджувати та аналізувати продуктивність та прогрес проектів, сприяючи прийняттю оптимальних рішень. У цілому, інформаційне забезпечення системи розподілу завдань має бути розробленим для оптимальної організації робочих процесів і досягнення максимальної ефективності [4].

До інформаційних потоків системи для управління послугами в сфері ІТ співробітники вносять завдання користуючись WEB-інтерфейсом системи, для цього потрібно ввести їх назву, опис, терміни виконання та іншу необхідну інформацію. Система зберігає завдання, які зберігаються в базі даних системи. Project Manager розподіляє завдання між співробітниками, розподіл відбувається завдяки інформації про навички, досвід та доступність співробітників, на основі цього створюється команда, якій надається відповідна інструкція до виконання завдань, та інша додаткова інформація. Співробітники виконують поставлені завдання та звітуються про вчасність та якість роботи використовуючи функціонал системи. Система відстежує виконання завдань зберігаючи інформацію про статус, прогрес та дату завершення кожного завдання, для кращого моніторингу процесів, виявлення проблем та забезпечення вчасного виконання завдань. Система надсилає повідомлення про стан виконання, затримки, проблеми чи інші важливі події завдання користувачам, співробітникам та РМ. За допомогою цих інформаційних потоків забезпечується ефективність розподілу завдань між співробітниками в ІТ-компаніях, застосунок може допомагати РМ краще контролювати робочий процес співробітників [5].

Користувачів інформаційної системи для управління послугами в сфері ІТ можна поділити на такі групи: користувачі, співробітники, РМ, адміністратори. Користувачі можуть створювати замовлення та вести моніторинг виконання замовлення, також можуть ознайомлюватись з додатковою інформацією про підприємство. Співробітники можуть використовувати систему для створення, звітування та відстеження завдань по замовленню. РМ може використовувати систему для розподілу завдань між співробітниками та контролю за виконанням

завдань. Адміністратори можуть мати доступ до всіх даних системи і можуть використовувати систему для управління послугами в компанії, тобто створення та управління обліковими записами користувачів та співробітників, налаштуванням системи та моніторингу системи.

Джерела інформаційної системи для управління послугами в сфері ІТ поділяються на внутрішні та зовнішні. Внутрішні джерела – знаходяться всередині організації, яка використовує систему, до неї належать дані про співробітника, замовника, систему. Зовнішні джерела знаходяться поза організацією, яка використовує систему; до неї належать дані про клієнта, зовнішні системи. Система для управління послугами в сфері ІТ має отримувати інформацію з цих джерел, щоб забезпечити користувачам доступ до необхідних даних [6].

Для глибшого розуміння проблематики дослідження проаналізуємо існуючі системи з управління послугами в сфері ІТ.

Atlassian Jira – це система управління проектами, яка дозволяє закривати майже всі завдання РМ в рамках одного інструмента: від планування до контролю процесів та результатів. Є комплексом з ІТ-рішень від компанії Atlassian, які об'єднані в Jira Family Products: Jira WM для роботи з бізнес-процесами, Jira SM для побудови сервіс-диску, Atlassian Jira Software для проектів з розробки програмного забезпечення [7]. Основними плюсами Atlassian Jira є широкий набір функцій, великий інтеграційний потенціал, можливість розширення функцій, можливість роботи за методологіями Scrum або Kanban. Основним з мінусів Atlassian Jira часто виділяють тривалий процес налаштування під конкретні робочі процеси та складний в розумінні інтерфейс [8], (див. на рис. 1.1 дод. А) зображено головну сторінку Atlassian Jira.

В ASANA зручно розділяти великі плани на маленькі зрозумілі таски. Вона допомагає побачити наскільки завантажений ваш день та відсортувати завдання у зручному порядку. Вона буде корисною командам або особам з великою кількістю різних проектів та різнопланових завдань, коли записник вже не допомагає, а звичайні таск-менеджери виглядають малофункціональними. Хоча ASANA створювалася як таск-менеджер для команд, її зручно використовувати і як

персональний планер [9]. Переваги Asana: широкий вибір візуалізації проектів і задач, звітність та перевірка ходу роботи, зв'язок між роботою різних команд та відділів, управління завантаженості для запобігання вигорання, фідбеки, теги, коментарі до задач. Однак в Asana є ряд таких недоліків, з якими слід бути уважним, дуже велика вартість застосунку, нестабільність сервісу, низька продуктивність користувацької продуктивності [10], (див. на рис. 1.2 дод. А) зображено головну сторінку Asana.

Trello – це візуальний інструмент, що дає змогу команді керувати різноманітними проектами й робочими процесами та відстежувати виконання завдань. Додавати файли, контрольні списки або правила для автоматизації: налаштування Trello допомагають оптимізувати командну роботу, завдяки створенню дошки й зручному виконанню поставлених завдань [11]. Перевагами Trello є безкоштовний доступ до функціоналу, простий інтерфейс, зручна взаємодія з платформою, інтеграція в кілька кліків з програмою, широкий вибір мов. Основними недоліками Trello є урізаний функціонал, у функціоналі можна додати лише 3 розширення, дозволено завантажувати до 10мб [12], (див. на рис. 1.3 дод. А) наведено головну сторінку Trello.

CRM-система Firmao призначена для малих та середніх виробничих, сервісних та торгових компаній. Впроваджується в компаніях, де потрібен контакт із лідами або постійними підрядниками. Це найкраща система для компаній, які планують динамічний розвиток. Основними плюсами Firmao є управління взаємовідносин з клієнтами, планування ресурсів підприємства, виставлення рахунків, забезпечення для котирування та замовлень, забезпечення для проектів та завдань [13]. Основними недоліками Firmao є інтеграція з Zapier доступна лише в найвищому пакеті ліцензій. Складність у використанні – попри наявність багатьох функцій, деякі з них приховані або розміщені не інтуїтивно [14], (див. на рис. 1.4 дод. А) зображено головну сторінку Firmao.

Wrike – це інструмент SaaS для командної співпраці та керування проектами, який може похвалитися багатопанельним інтерфейсом користувача, а також допомагає у командній співпраці та управлінні проектами. Вони вирізняються

широким набором функцій керування проектами, зокрема командною комунікацією, відстеженням доступності співробітників і переглядом проектів із ризиками. Основними перевагами Wrike є інтеграція з широким набором інструментів, включаючи програми CRM, легке встановлення пріоритетів, має надійні протоколи безпеки, розширена аналітика та аналіз даних завдяки інтеграції з Tableau. Основними недоліками є складний в розумінні інтерфейсу, велика ціна, не достатня надійність інструментів бюджетування та виставлення рахунків, іноді має низьку продуктивність для деяких програм [15], (див. на рис. 1.5 дод. А) зображено сторінку PM Wrike.

Виходячи з описаних раніше аналогів WEB-застосунку для управління послугами в сфері ІТ, можна зробити таку таблицю порівнянь яка наведена на табл. 1.1 дод. А.

Виходячи з результатів порівнянь WEB-застосунку для управління послугами в сфері ІТ в наведеній табл. 1.1 дод. А, можна зробити такі висновки, що Asana є найкращим аналогом WEB-застосунку для управління послугами в сфері ІТ. Asana представляє великий спектр функцій, інтеграцію з іншими програмними продуктами та конкурентоспроможність цін. Трохи гірші результати показує Atlassian Jira, Trello, Firmao які є також хорошим вибором для управління послугами та проектами. А от Wrike має найнижчі характеристики в порівнянні з іншими аналогами, але все ж виконує хороші додаткові вміння, такі як управління ризиками та управління знаннями.

Реалізація проекту, передбачає також залучення нових клієнтів і координації команди в проекті. Застосунок дозволить простіше визначити пріоритети завдань, структурування завдань в рамках проекту, забезпечити ефективність комунікації та спільної роботи над проектом, ефективно розподіляти ресурси та завдання між співробітниками відповідно до їх навичок й призначати відповідні обов'язки кожному співробітнику. Він має враховувати специфіку галузі та інтегруватися з іншими інструментами ІТ-сфери для оптимальної продуктивності та моніторингу завдань, забезпечуючи зручний та ефективний інструментарій для спільної реалізації проектів.

## 1.2. Аналіз сучасних середовищ для web-розробки

Сучасне виробництво веб-додатків та веб-сайтів стало складнішим і різноманітнішим завдяки постійному розвитку технологій та зростанню вимог користувачів. У таких умовах важливо мати належні інструменти та середовища розробки, які сприяють продуктивності та якості роботи.

Середовище розробки мов програмування (IDE) відіграє ключову роль у роботі веб-розробників, надаючи їм зручні та потужні інструменти для писання, тестування та відлагодження коду. З моменту свого виникнення IDE вдосконалюються та адаптуються до потреб сучасної веб-розробки, надаючи широкі можливості для роботи з різними мовами програмування та технологіями.

Тож розглянемо найпопулярніші та найбільш широко використовувані середовища розробки за критеріями: *функціональність* (можливість виконувати основні операції перем; встановлення та редагування ключових елементів тощо); *інтерфейс* (простота використання; можливість налаштування інтерфейсу; автоматизація та розширення можливостей); *інтеграція* (сумісність; підключення плагінів для розширення можливостей; інтеграція з інструментами для редагування); *продуктивність* (швидка та чутлива робота; мінімізація часу; підтримка багатоядерних процесорів тощо); *вартість*.

Крім цих критеріїв, важливо також враховувати свої особисті вподобання та досвід роботи з програмним забезпеченням.

### 1. Visual Studio Code

Visual Studio Code є одним з найпопулярніших та найбільш широко використовуваних середовищ розробки на сьогодні. Це безкоштовне інтегроване середовище розробки, яке підтримує багато мов програмування, включаючи JavaScript, TypeScript, Python та багато інших. Він має розширення, що дозволяють розширити його функціональність, що забезпечує широкі можливості для розробки.

Visual Studio Code має багато переваг, серед яких безкоштовність та відкритий код, широкі можливості розширення та інтеграції з різними інструментами. Він підтримує багато мов програмування, інтегрована підтримка Git дозволяє зручно

керувати версіями, а зручний інтерфейс та швидкість роботи роблять його привабливим для багатьох користувачів.

Однак Visual Studio Code має і свої недоліки. Він вимагає налаштувань та додаткових плагінів для повного функціоналу, що може бути складним для деяких користувачів. Крім того, він не має вбудованої підтримки певних технологій, як-от ASP.NET MVC, що може вимагати додаткової настройки.[16]

## **2. PyCharm**

PyCharm є одним з найкращих середовищ розробки для Python. Це інтегроване середовище розробки з підтримкою Django та Flask, що дозволяє розробникам Python створювати веб-додатки та інші проекти.

PyCharm має багато переваг, включаючи розширений набір функцій, який включає автозаповнення, перевірку синтаксису, рефакторинг та інші інструменти, що полегшують розробку. Потужна система автоматичного завершення та інтелектуальний аналіз коду допомагають підвищити ефективність роботи. Крім того, PyCharm оснащений вбудованим налагоджувачем, який підтримує покрокове виконання та відображення значень змінних, а також підтримує віртуальне середовище для розділення залежностей та проектів.

Але PyCharm має свої недоліки. Його високі вимоги до комп'ютерних ресурсів можуть вплинути на продуктивність менш потужної системи, а велика кількість функцій може виявитися складним для новачків. Крім того, платна версія Professional Edition має обмежений доступ до деяких розширених функцій.[17]

## **3. Atom**

Atom є безкоштовним відкритим середовищем розробки, який був розроблений компанією GitHub. Воно має широкі можливості для налаштування та підтримує багато мов програмування, включаючи JavaScript, HTML, CSS, Python та інші.

Atom має багато переваг, включаючи безкоштовне розповсюдження з відкритим кодом та широкі можливості налаштування робочих областей, шрифтів та дизайну. Він підтримує різні мови програмування та фреймворки, різні кодування, а також пропонує функції швидкого завантаження, закладки, пошук та

заміну, розділення екрана, завершення, масштабування коду, підсвічування синтаксису, перевірку орфографії та закриваючі дужки. Інтеграція з Git та GitHub, наявність функцій teletype для спільної розробки та відстеження помилок, вбудовані пошукові системи, менеджери пакетів та проектів роблять Atom корисним інструментом для розробників.

Однак Atom також має свої недоліки. Відсутність підтримки української мови може бути незручною для україномовних користувачів. Крім того, швидкість роботи atom може бути проблематичною для великих проектів, оскільки вона може бути нижчою порівняно з іншими редакторами коду.[18]

#### **4. Sublime Text**

Sublime Text є дуже швидким та легким середовищем розробки з великою кількістю плагінів та розширень. Це дуже популярне середовище розробки для JavaScript, а також підтримує Python, HTML, CSS та інші мови програмування.

Sublime Text відомий своєю швидкістю завантаження та чуйністю користувачів, а також простим та інтуїтивно зрозумілим інтерфейсом. Він забезпечує універсальність завдяки безлічі модулів і можливостей настройки і гнучкий для вирішення різних завдань розробки. Крім того, Sublime Text підтримує крос-платформну роботу в Windows, macOS та Linux.

Однак sublime text має свої недоліки. У ньому відсутні вбудовані інструменти для роботи з великими проектами, що може погіршити його функціональність порівняно з іншими IDE. Крім того, повна версія програми вимагає придбання ліцензії, що може бути дорогим для деяких користувачів. Воно підтримує безліч мов програмування, але для деяких з них його функціональність може бути обмежена.[19]

#### **5. NetBeans**

NetBeans є інтегрованим середовищем розробки для Java, що дозволяє створювати веб-додатки, мобільні додатки та інші проекти. Воно підтримує також PHP, C/C++, HTML та інші мови програмування.

NetBeans надає широкий спектр можливостей для розробки, включаючи редактор коду з підсвічуванням синтаксису, автозаповнення, вбудований



налагоджувач та підтримку контролю версій. Він підтримує такі мови програмування, як Java, PHP, C/C++, HTML та JavaScript, а також інтегрується із серверами додатків та системами управління базами даних. Крім того, NetBeans безкоштовний.

Однак у NetBeans є свої недоліки. Це вимога до системних ресурсів, яка впливає на продуктивність на менш потужному комп'ютері, а інтерфейс не інтуїтивно зрозумілий для початківців. Крім того, NetBeans може працювати повільніше порівняно з іншими IDE, особливо при роботі з великими проектами. [20]

Після визначення середовищ для web-розробки та виявлення їхніх переваг та недоліків можна провести порівняльний аналіз середовища web-розробки за такими критеріями функціональність, простота використання, швидкість роботи, можливість розширення зображено на див. табл. 1.2 дод. А.

Зважаючи на цільову задачу з web-розробки управління послугами в сфері ІТ, найкращими середовищами, які ви можете розглянути, є Visual Studio Code, Sublime Text та PyCharm. Ось кілька причин для такого вибору:

1. Функціональність: У всіх трьох середовищах є розширений набір функціональності, який дозволить ефективно розробляти складні веб-додатки, включаючи управління послугами в сфері ІТ.

2. Простота використання: Visual Studio Code, Sublime Text та PyCharm відомі своєю зручністю використання та дружнім інтерфейсом, що сприяє швидкому освоєнню та комфортній роботі.

4. Можливості розширення: Visual Studio Code, Sublime Text та PyCharm мають розвинуту систему плагінів та розширень, яка дозволяє розширювати функціональність згідно з потребами вашого проєкту. [21]

Враховуючи ці фактори, Visual Studio Code, Sublime Text та PyCharm можуть бути найбільш оптимальними виборами для розробки веб-додатків для управління послугами в сфері ІТ.

### **1.3. Моделювання бізнес-процесів предметної області**

Щоб краще зрозуміти всі процеси інформаційної системи для управління послугами в сфері ІТ розглянемо діаграму IDEF0 (див. на рис. 1.6 додаток А), в якій вказані всі дані на вхід для повноцінного функціонування системи, розписані всі необхідні механізми для управління проектом, розписані всі методи управління проектом та визначені результати даної системи.[22]

У декомпозиції було розроблено послідовність дій виконання управління послуг в сфері ІТ та розписані всі необхідні дані, документації, механізмів управління та описи результату кожної з дій. Для початку користувач створює власний акаунт або авторизується надаючи свої дані для компанії. Після авторизації відбувається аналіз запиту, який визначає всі вимоги до проекту. Після аналізу відбувається прийняття замовлення де визначаються завдання для розробки проекту, ресурси які необхідні для виконання даного проекту та терміни роботи. Коли цілі та ресурси проекту біли визначені створюється команда розробників та відбувається розподілення завдань між розробниками. Після розподілу завдань починається процес виконання проекту та звітність про стан проекту. Наведена декомпозиція IDEF0 зображена на рис. 1.7 дод. А.

Розбір декомпозиції IDEF0 аналізу запиту, в якій описується сам процес аналізу запиту. Спочатку відбувається збір даних про замовлення, на основі цих даних можна визначити цілі та мету проекту. Після визначення мети проекту, визначається і збирається інформація про проект, на основі збору даних про проект визначаються вимога та завдання проекту. Далі виконується оцінка всіх ризиків проекту, після цього виконується планування виконання проекту. Наведена декомпозиція IDEF0 «Аналізу запиту» зображена на рис. 1.8 дод. А.

В наступній декомпозиції описується розподіл завдань між співробітниками. Щоб виконати розподіл завдань потрібно визначити задачі проекту, на основі яких проводиться оцінка навичок та вмінь співробітників, що залежить від їхнього досвіду. Коли визначені вміннями усіх працівників, тоді створюється список працівників компанії які підходять для виконання замовлення, далі відбувається перевірка завантаженості співробітників, по завантаженості співробітників створюється список вільних співробітників які підходять для розробки проекту. На основі списку вільних співробітників створюється команда розробників, в якій

визначається керівник проекту, після створення команди та визначення керівника проекту відбувається розподіл завдань між співробітниками. Наведена декомпозиція IDEF0 «Розподілення завдань» зображена на рис. 1.9 дод. А.

В останній декомпозиції описується процес виконання замовлення. Спочатку розробник приймає завдання та визначає вимоги до завдання. Далі розробник розробляє план виконання замовлення, коли план створено починається розробка проекту, під час якого відбувається комунікація між командою розробників та керівництвом. Під час розробки проекту Project Manager виконує постійний моніторинг виконання проекту, що дозволяє йому контролювати виконання проекту, далі розробники інформують про готовність проекту до тестування. Після успішного тестування замовлення розробники надають звітність про успішне виконання по проекту. Наведена декомпозиція IDEF0 «Виконання замовлення» зображена на рис. 1.10 дод. А.

Щоб детальніше розібрати виконання дій проекту було розроблено IDEF3 в якій описується детальна послідовність дій від прийняття замовлення й до виконання замовлення. Спочатку відбувається прийом замовлення, потім Project Manager виконує обробку замовлення, при незадоволенні умов замовлення Project Manager відхиляє замовлення, в результаті задоволення умов замовлення Project Manager приймає замовлення, коли замовлення буде прийняте відбувається визначення завдань, термінів та ресурсів необхідних для проекту. Коли все буде визначено, тоді починається виконання замовлення, далі потрібно підібрати спеціалістів під замовлення, коли всі спеціалісти будуть відібрані для проекту, то кожному спеціалісту буде надано завдання. Під час виконання замовлення буде відбуватись розробка замовлення, створення документації та проведення тестування замовлення. Після успішного тестування проекту та створення документації про проект, проект буде здано замовнику, після перевірки замовником проекту, якщо йому не сподобається виконання даного замовлення замовник може відправити замовлення знову на переробку замовлення або ж відмовитись від замовлення. В разі схвалення замовлення замовником, замовник оплачує замовлення. Наведена IDEF3 зображена на рис. 1.11 дод. А.[23]

## **Висновок до першого розділу**

У цьому розділі було розглянуто теоретичні основи дослідження, включаючи концепції управління ІТ-послугами, моделювання IDEF0 та IDEF3, а також сучасні середовища веб-розробки. Також визначили моделювання бізнес-процесів предметної області.

Основні переваги управління ІТ-послугами включають забезпечення задоволення потреб бізнес-користувачів шляхом ефективного управління ІТ-ресурсами. Важливу роль у цьому процесі відіграють методи моделювання бізнес-процесів, такі як IDEF0 та IDEF3, які дозволяють детально відображати та аналізувати основні процеси. Сучасні середовища веб-розробки пропонують широкий спектр інструментів і функцій для створення веб-додатків та веб-сайтів, що сприяє підвищенню ефективності розробки та управління ІТ-послугами. Моделювання бізнес-процесів предметної області є корисним для визначення та розуміння ключових аспектів управління ІТ-послугами.

Щодо майбутніх досліджень, необхідно продовжувати роботу над розробкою більш детальної моделі процесу управління ІТ-послугами, щоб підвищити його ефективність та адаптивність. Важливо оцінити ефективність застосування методів IDEF0 та IDEF3 у контексті управління ІТ-послугами в ІТ-компаніях, що допоможе визначити їхню придатність та потенційні обмеження. Дослідження впливу сучасних середовищ веб-розробки на управління ІТ-послугами також є необхідним для розуміння їхнього внеску в покращення процесів та результатів у цій галузі.

Завдяки дослідженню теоретичних засад дослідження управління послугами в сфері ІТ стає можливо краще зрозуміти теоретичні основи дослідження, забезпечуючи знання для подальших досліджень та досліджень у сфері управління ІТ-послугами. Заглибившись у тонкощі методології IDEF0 та IDEF3 моделювання, сучасних середовищ веб-розробки та моделювання бізнес-процесів предметної області, ми отримали цінні ідеї, які допоможуть нашим майбутнім зусиллям.

## **РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ WEB-РОЗРОБКИ**

### **2.1. Моделювання web-розробки**

Щоб краще зрозуміти як працює дана технологія WEB-застосунку для управління послугами в сфері ІТ, необхідно розглянути UML-діаграму прецедентів (див. на рис. 2.1 дод. Б), в якій добре описується роль та функції кожного учасника даної системи. На якій можна побачити як відбувається взаємодія користувача, розробника та Project Manager з даною системою для управління послугами в сфері ІТ.

Для кращого розуміння розглянемо UML-діаграму активності, яка допомагає краще зрозуміти всі процеси надання послуг в сфері ІТ. Завдяки діаграмі активності можливо виконувати аналізи даної системи для її вдосконалення. В даній UML-діаграмі активності ми можемо прослідкувати за всіма процесами від авторизації користувача аж до виконання проекту, що дозволить при розробці робити менше помилок, та створювати більш логічну систему для управління послугами в сфері ІТ. (див. на рис. 2.2 дод. Б).[24]

З допомогою UML-діаграми станів можна розглянути всі стани в яких перебуває система для управління послугами в сфері ІТ. В діаграмі станів можна спостерігати взаємодію користувача з системою, на основі цих даних можна покращити та оптимізувати роботу системи так, щоб користувачам було простіше розібратись й працювати з даною системою. Діаграма станів є корисним інструментом для розуміння поведінки об'єктів у системі. Вона дозволяє зрозуміти, як об'єкти у системі можуть змінювати свій стан у залежності від події, це допомагає краще зрозуміти, як система працює, тим самим виявляти несправності системи, а потім усувати їх. UML-діаграма станів наведена на рис. 2.3 дод. Б.

Для більш детального розуміння функціонування системи, відслідкуємо дії учасників системи з використанням UML-діаграми послідовності (рис. 2.4 дод. Б). Яка описує життєвий цикл кожного з елементів системи, таких як замовник, сайт та база даних, на основі цих даних можна детально вивчити дану систему, взаємодію всіх елементів між собою. Також завдяки UML-діаграмі послідовності стає можливо краще зрозуміти та визначити вимоги проекту, розробляти

планування та виконання проекту з метою забезпечення ефективності і очікування вимог клієнтів до системи для управління послугами в сфері ІТ.

З метою кращого розуміння всіх сутностей системи та їх зв'язки в системі для управління послугами в сфері ІТ, для цього потрібно використовувати UML-діаграми класів (див. На рис. 2.5 дод. Б). В даній діаграмі описуються всі класи, їх атрибути та методи, а також відносини між ними. Основними перевагами цієї діаграми є те, що завдяки їй можна візуально представити структуру системи та взаємодії між її компонентами, це допомагає зрозуміти як система працює, і виявити потенційні проблеми або недоліки. З допомогою використання UML-діаграми класів можна створювати документування структури системи та взаємодії її між компонентами, виконувати аналіз ефективності та проблеми у системі, завдяки цьому розробляти нові функції або вдосконалювати уже існуючі функції у системі.

В загальному UML-діаграми потрібні для того, щоб можна було без готової системи представити всі процеси виконання даної системи, її структуру, поведінку та взаємодії всіх її елементів, зрозуміти вплив даної системи на клієнтів та розробників, допомагає визначити потреби клієнтів й вимоги розробників для виконання різного плану завдань.

З допомогою UML діаграм можна створювати різні документації, які необхідні для кращого розуміння структури системи. Описуючи структуру системи в документації ми зможемо краще визначити основні класи, їх структуру та методи, легше визначити взаємодію між класами. Описуючи поведінку системи в документації можливо краще визначити стани об'єктів, їх переходи між станами та діями. Описуючи взаємодію між системою та користувачем в документації можливо краще визначити інтерфейс користувача, а також сценарій використання системи.

Також з допомогою UML діаграм можна проводити різні аналізи над системою, щоб визначити ефективність системи в якій буде проведена оцінка відповідності системи до потреб клієнтів. На основі аналізу можна визначити, які функції системи потрібно вдосконалити або додавати. Аналіз визначення проблем

в системі допомагає виявляти помилки, які можуть призвести до затримок у обробці заявок на послуги, що допоможе вирішити помилки та покращити продуктивність системи. Аналіз вдосконалення системи дозволяє розробляти рекомендації щодо того, як покращити систему. Зазвичай це розробка нових функцій та вдосконалення уже існуючих функцій. З допомогою діаграм можна вдосконалювати систему.

Для кращого розуміння процесу розробки web-застосунку для управління послугами в сфері ІТ було розглянуто діаграму Ганта (рис. 2.6 дод. А).

Ця діаграма є важливим інструментом для візуалізації календарного плану проекту, що дозволяє ефективно управляти ресурсами та часом. Діаграма Ганта дозволяє наочно представити послідовність виконання завдань та їх тривалість у вигляді горизонтальних смуг на часовій шкалі. Це допомагає легко оцінити загальну тривалість проекту та визначити критичний шлях - послідовність задач, які визначають мінімальний час виконання всього проекту. На діаграмі також відзначено ключові віхи проекту, такі як завершення основних етапів або презентації для замовника. Це допомагає відстежувати прогрес та вчасно реагувати на можливі відхилення від плану. Крім того, для кожного етапу вказано відповідальних виконавців або команди, що полегшує координацію роботи та розподіл ресурсів. Використання діаграми Ганта в нашому проекті не лише допомагає ефективно управляти процесом розробки, але й слугує наочним інструментом для комунікації з командою та замовником. Це дозволяє всім учасникам проекту мати чітке уявлення про поточний стан розробки та очікувані терміни завершення кожного етапу.

## **2.2. Проектування структури бази даних та інтерфейсу програмного продукту**

Проектування структури бази даних інформаційної системи для управління послугами в сфері ІТ є важливим етапом для її побудови. База даних зберігає великі обсяги інформації про завдання, співробітників, користувачів, всі розроблені

проекти, дані про замовлення, які потрібні для повноцінного функціонування системи. Бази даних забезпечують ефективність зберігання та оброблення великого обсягу даних, безпеку даних та їх захищеність від несанкціонованого доступу, змін або знищення, забезпечують доступ до різних даних для користувачів, незалежно від їхнього місця знаходження.

За допомогою DFD можливо візуалізувати потоки даних між зовнішніми сутностями та процесами системи. Завдяки цьому можливо краще зрозуміти, як користувач взаємодіє з системою, як система отримує та обробляє інформацію з зовнішнього світу. Також можливо візуалізувати потоки даних між процесами системи, що дозволяє зрозуміти, як інформація передається між різними процесами системи.

Далі можна розглянути як відбувається поведінка системи при побудові DFD (див. на рис. 2.7 дод. Б) створення замовлення користувача. Можна розглянути як здійснюються процеси під час створення замовлення.

В даній діаграмі розписані всі етапи створення замовлення, під час яких потрібно використовувати такі бази даних, як базу даних користувача для того щоб користувач міг авторизуватись або створити нового користувача, потім для перегляду завдань потрібна база даних замовлень, щоб їх відобразити при створенні замовленні потрібна база даних статусу завдань, щоб призначити завданню статус, щоб в подальшому можна було спостерігати за всіма етапами розробки замовлення.

Як можна спостерігати на цьому етапі не застосовуються всі бази даних, саме тому потрібно ще розібрати DFD (див. на рис. 2.8 дод. Б) діаграму розподілення завдань, щоб краще зрозуміти всі процеси створення системи.

В DFD діаграмі описується процес розподілення завдань між співробітниками під час якого використовуються різні бази даних. Для того щоб створити команду розробників потрібно з бази даних співробітників витягнути дані про вміння та завантаженість співробітників. Під час прийому виконання завдання розробниками, завданню призначається статус замовлення «в розробці». Коли вже продукт повністю готовий до експлуатації, він міняє свій стан на виконаний та заноситься в базу даних до виконаних проектів, після чого передається замовнику.



Доцільно буде розглянути ER-діаграму яка описує зв'язки між всіма об'єктами системи. По ER-діаграмі яка зображена (див. на рис. 2.9 дод. Б) видно, що один клієнт може мати багато проектів але один проект не може мати двох замовників. Проект має зв'язки з завданнями, розробниками, РМ та статусом завдань. Зв'язок проекту з статусом завдань виглядає так, що один проект може мати багато різних статусів, так само як і один статус може бути у багатьох різних проектів. Зв'язок проекту з завданнями має такий вигляд, що один проект може містити в собі багато різних проектів, а от завдання може мати лише один проект. Зв'язок проекту з розробниками має такий вигляд, що один проект може виконувати декілька розробників, так як і один розробник може виконувати декілька проектів. Зв'язок проекту з РМ має такий вигляд, що одним проектом може керувати лише один РМ, а от один РМ може мати багато різних проектів. Ще РМ має такий зв'язок з командою, що один РМ може мати групу співробітників, а от співробітник під час виконання проекту може мати лише одного РМ. Також співробітники мають такий зв'язок з завданнями, що один співробітник може виконувати одночасно декілька завдань, так як і одне завдання може виконувати одночасно декілька співробітників. Зв'язок між завданнями та статусом завдань, який виглядає так, одне завдання може мати багато різних статусів, так само як один статус може бути у багатьох завданнях. Таким чином стає можливо краще ознайомитись з структурою бази даних та їх зв'язками з суб'єктами системи.

Для кожного WEB-застосунку є дуже важливим зручність користування та привабливий інтерфейс. Саме завдяки привабливому інтерфейсу та зручному використанні його користувачі можуть краще взаємодіяти з сайтом, краще та швидше налаштувати роботу свого проекту під себе та під всю команду, легше та швидше навчатись взаємодіяти з застосунком. А також привабливий інтерфейс сприяє зацікавленню потенційних клієнтів, що підвищує попит підприємства та поліпшує конкурентоспроможність. Зручність використання даного застосунку є одою з переваг в порівнянні з використанням аналогів WEB-застосунків для управління послугами в сфері ІТ.

Визначившись з дизайном та оформлення WEB-застосунку (див. на рис. 2.10 дод. Б) можна почати створення інтерфейсу застосунку. Всього в сайті буде декілька сторінок таких, як сторінка авторизації, головна сторінка, сторінка інформації про компанію, сторінка співробітників, сторінка управління послугами, сторінка проектів, сторінка створення замовлення та інші сторінки додаткової інформації.

Для того щоб керувати проектами потрібна спеціальна ділянка, на сайті це все замінюють дошки послуг (див. на рис. 2.11 дод. Б), де розписуються всі проекти, які є на підприємстві. На самих дошка коротко описана назва самого проекту, дата створення проекту та кінцевий термін виконання проекту, знизу дошки описаний статус проекту та кількість людей які входять в проект. Щоб дізнатись більше інформації потрібно натиснути на кнопку. Також на цій сторінці відображаються уже всі розроблені проекти.

Також важливим елементом для керування проектами та зручності використовується список завдань (див. на рис. 2.12 дод. Б), де можливо зручно та ефективно відслідковувати всі дії стосовно проектів та керувати їх станом. В списках можна відслідковувати хто, ще працює над проектом та зручно переглядати всі завдання.

Також досить ефективним є використання календаря де можна буде прослідковувати в зручному режимі терміни виконання завдань. Календар (див. на рис. 2.13 дод. Б).

### **2.3. Процес реалізації web-застосунку управління послугами в сфері ІТ**

Розробка та впровадження вашого програмного продукту включає кілька етапів. На фронтенді використовувався React.js для створення інтерактивного користувацького інтерфейсу. Було розроблено компоненти TaskBoard, ListTask і Calendar для управління завданнями та інтеграції з календарем. Стилзацію компонентів виконано за допомогою CSS, а для створення адаптивного дизайну

використано Bootstrap, зокрема компонент Accordion у ListTask. Дані зберігаються локально за допомогою localStorage, що дозволяє зберігати завдання між сесіями.

На серверному боці створено сервер на Node.js з використанням фреймворку Express, налаштовано маршрутизацію для обробки API запитів та інтегровано Sequelize ORM для роботи з базою даних. Реалізовано аутентифікацію та авторизацію за допомогою OAuth2 для інтеграції з Google Calendar. Було реалізовано інтеграцію з Google Calendar API для отримання та відображення подій календаря, а також налаштовано OAuth2 потік для авторизації користувачів через Google.

Використані технології включають React.js, JavaScript, HTML, CSS, Bootstrap на фронтенді та Node.js, Express.js, Sequelize ORM на бекенді. Серед функцій веб-застосунку – дошка завдань для створення та управління завданнями, список завдань у форматі акордеону, календар для інтеграції з Google Calendar, система статусів завдань та можливість призначати користувачів до завдань. Процес впровадження передбачає налаштування серверного середовища, конфігурацію бази даних, збірку React застосунку, інтеграцію з Google Calendar, тестування та розгортання на продакшн-сервері.

Заходи з безпеки включають використання OAuth2 для авторизації, застосування HTTPS для захищеного з'єднання, використання змінних середовища для зберігання ключів API та реалізацію CORS для контролю доступу з різних джерел. Ці кроки забезпечують захищеність та конфіденційність даних користувачів.

## **Висновок до другого розділу**

Другий підрозділ присвячений проектуванню структури бази даних та інтерфейсу програмного продукту для управління послугами в сфері IT. У ньому детально розглядається важливість правильної організації бази даних для ефективного зберігання та обробки інформації про завдання, співробітників,

проекти та замовлення. Особлива увага приділяється використанню різних типів діаграм для візуалізації та аналізу структури системи.

Зокрема, у підрозділі представлені DFD-діаграми, які допомагають зрозуміти потоки даних при створенні замовлень та розподілі завдань. Ці діаграми наочно демонструють взаємодію користувача з системою та процеси обробки інформації. Також розглянуто ER-діаграму, яка описує зв'язки між різними об'єктами системи, такими як клієнти, проекти, завдання, розробники та менеджери проектів. Це дозволяє краще зрозуміти структуру бази даних та взаємозв'язки між її елементами.

Значна частина підрозділу присвячена проектуванню користувацького інтерфейсу. Підкреслюється важливість створення зручного та привабливого інтерфейсу для ефективної взаємодії користувачів із системою. Описуються основні сторінки веб-застосунку, включаючи головну сторінку, дошки послуг, список управління завданнями та календар. Особлива увага приділяється тому, як різні компоненти інтерфейсу сприяють ефективному управлінню проектами та завданнями, підвищуючи загальну продуктивність системи та її конкурентоспроможність на ринку.

## **РОЗДІЛ 3. ТЕСТУВАННЯ ГРИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ**

### **3.1. Інструкція користувачу web-застосунку управління послугами в сфері ІТ**

Для кожного WEB-застосунку є дуже важливим зручність користування та привабливий інтерфейс. Саме завдяки привабливому інтерфейсу та зручному використанні його користувачі можуть краще взаємодіяти з сайтом, краще та швидше налаштувати роботу свого проекту під себе та під всю команду, легше та швидше навчатись взаємодіяти з застосунком. А також привабливий інтерфейс сприяє зацікавленню потенційних клієнтів, що підвищує попит підприємства та поліпшує конкурентоспроможність. Зручність використання даного застосунку є одєю з переваг в порівнянні з використанням аналогів WEB-застосунків для управління послугами в сфері ІТ.

Інструкція користувача web-застосунку SMIT для управління послугами в сфері ІТ охоплює основні кроки для ефективного використання системи. Перший крок - авторизація через кнопку "Вхід" на головній сторінці. Після входу користувач отримує доступ до повного функціоналу. Для початку роботи слід натиснути "Перейти до роботи" або "Почати роботу", що відкриває інструменти управління завданнями та співпраці. Головна сторінка розміщена на рис. 3.1 дод. В.

Для входу в систему натисніть кнопку «Війти» або «Почати роботу» після цього відкриється сторінка реєстрації (див. на рис. 3.2 дод. В), де буде форма з двома полями введення: «Введіть ваш email» та «Введіть ваш пароль». Після заповнення полів натисніть на кнопку «Зареєструватись», якщо вже є створений акаунт, то перейдіть за посиланням війти вас перекине на сторінку авторизації див. на рис. 3.1.3 додаток В, де потрібно буде ввести email та пароль, після чого ви перейдете на сторінку розробки.

При переході в «мій проект» на вкладку «дошка» (див. на рис. 3.4 дод. В) у верхній частині дошки розташовані поля для створення нових завдань. Користувач може ввести назву завдання, опис завдання а також встановити дату створення та завершення завдання, завдяки кнопці «додати завдання» можна швидко додати

нове завдання. Основна частина дошки розділена на 4 статуси завдання: «Чернетка», «Заплановано», «В процесі» та «Готова», кожне завдання представляється у вигляді картки, яка містить ключову інформацію завдання: назву, опис, дату створення та закінчення, іконки призначеного користувача.

При редагуванні завдань (див. на рис. 3.5 дод. В), можна змінити: назву, опис, дату створення та завершення, призначити нового користувача, також можна зробити автоматичне призначення користувачів (див. на рис. 3.6 дод. В), де при алгоритмі підбору користувачів рекомендується кращий спеціаліст під завдання. Також при нажаті на випадаючий список можна керувати станом свого завдання (див. на рис. 3.7 дод. В).

Зверху на сторінці «мій проект» представлено групу кнопок див. на рис. 3.1.8 додаток В, де при натисканні на «Список» відображається список завдань, при натисканні на «Дошку» відображається дошка завдань, при натисканні на «Календар» відображається календар завдань та при натисканні «Файли» можна загрузити додаткові файли для завдання.

При відкритті списку завдань відображається список завдань в якому відображається: назва, опис, термін виконання та учасники з кожними категоріями завдань (див. на рис. 3.9 дод. В).

При відкритті Календаря відображається кнопка «Авторизуватися з google» (див. на рис. 3.10 дод. В), при нажаті на яку перекидає на сторінку авторизації google календаря (див. на рис. 3.11 дод. В), після авторизації перекидає на сторінку google календаря (див. на рис. 3.12 дод. В).

При відкритті вкладки «Файли» (див. на рис. 3.13 дод. В), посередині екрану з'являється надпис Завантажити файл, та нижче є кнопка з надписом «Вибрати файл», після того як файл завантажений його можна переглядати або видаляти (див. на рис. 3.14 дод. В).

### 3.2. Тестування web-розробки та аналіз результатів

Для забезпечення якості та надійності веб-розробки було проведено комплексне тестування, яке охоплювало різні аспекти функціонування системи. Основні типи проведених тестів включали функціональне тестування, навантажувальне тестування, тестування юзабіліті та тестування безпеки.

Функціональне тестування виконувалось за допомогою Postman та ручного тестування, перевіряючи коректність роботи всіх функцій системи, особливо зосереджуючись на авторизації користувачів, інтеграції з Google Calendar API та взаємодії з базою даних. Для автоматизації UI-тестів використовується Cypress, що дозволило ефективно перевірити коректність відображення та функціонування React-компонентів.

Навантажувальне тестування здійснювалось за допомогою Apache JMeter, симулюючи високе навантаження на сервер для оцінки продуктивності системи. Це допомогло виявити потенційні проблеми з масштабованістю та оптимізувати обробку запитів.

Тестування юзабіліті проводилось, ретельно аналізуючи інтерфейс з точки зору різних типів користувачів. Це дозволило виявити деякі неінтуїтивні елементи інтерфейсу та покращити загальний користувацький досвід. Для тестування безпеки використовується OWASP ZAP, що допомогло виявити потенційні вразливості, зокрема пов'язані з XSS-атаками та проблемами в обробці користувацьких даних.

У процесі тестування було виявлено ряд помилок різного ступеня критичності. Серед найбільш значущих можна відзначити проблеми з оновленням OAuth токенів, вразливості до XSS-атак у формі авторизації, та неоптимізовані запити до бази даних, що спричиняли затримки при збільшенні навантаження. Усі виявлених помилок були успішно виправлені. Зокрема, було реалізовано механізм автоматичного оновлення токенів OAuth, впроваджено санітизації введених користувачем даних для запобігання XSS-атакам, та оптимізовано критичні запити до бази даних.

Аналіз результатів тестування виявив, що система загалом відповідає основним вимогам функціональності та безпеки, але є простір для вдосконалення. Рекомендації щодо подальшого розвитку охоплюють кілька ключових напрямків: посилення безпеки через більш комплексне тестування, можливо із залученням експертів; оптимізацію фронтенду за допомогою сучасних технік, таких як lazy loading та code splitting; впровадження системи моніторингу продуктивності в реальному часі для оперативного виявлення та вирішення проблем. Реалізація цих заходів дозволить значно підвищити якість, надійність та ефективність web-розробки, забезпечуючи кращий користувацький досвід та створюючи міцну основу для майбутнього розвитку проекту.

### **Висновки до третього розділу**

У цьому розділі було детально розглянуто процес тестування та аналізу результатів web-застосунку для управління послугами в сфері ІТ. Було підкреслено важливість зручного та привабливого інтерфейсу для ефективної взаємодії користувачів із системою.

Інструкція користувача надала чіткий огляд основних функцій та можливостей застосунку, включаючи процес авторизації, управління завданнями, використання дошки проектів, календаря та роботи з файлами. Ця інструкція забезпечує користувачам швидке освоєння системи та ефективне використання її функціоналу.

Процес тестування охопив різні аспекти функціонування системи, включаючи функціональне, навантажувальне, юзабіліті та тестування безпеки. Використання сучасних інструментів, таких як Postman, Cypress, Apache JMeter та OWASP ZAP, дозволило провести комплексну перевірку якості розробки. В результаті було виявлено та усунуто ряд критичних помилок, що значно підвищило надійність та безпеку системи.

Аналіз результатів тестування показав, що система в цілому відповідає поставленим вимогам, але також виявив напрямки для подальшого вдосконалення. Рекомендації щодо покращення включають посилення безпеки, розширення



автоматизованого тестування, оптимізацію фронтенду та впровадження системи моніторингу продуктивності. Реалізація цих рекомендацій дозволить значно підвищити якість та ефективність web-розробки, забезпечуючи кращий користувацький досвід та створюючи міцну основу для майбутнього розвитку проекту.

## ЗАГАЛЬНІ ВИСНОВКИ

У кваліфікаційній роботі проаналізовано інформаційні потреби та визначено предметну область дослідження. Встановлено, що основною метою розробки веб-застосунку є забезпечення оптимального розподілу завдань між співробітниками в ІТ-компаніях. Визначено ключові інформаційні потоки та групи користувачів системи, включаючи клієнтів, співробітників, проектних менеджерів та адміністраторів. Це дозволило створити основу для розробки функціональних вимог до системи.

Здійснено порівняльний аналіз існуючих систем управління послугами в сфері ІТ та сучасних середовищ для веб-розробки. На основі чого визначено, що найбільш підходящими для розробки веб-застосунку є Visual Studio Code, Sublime Text та PyCharm.

Важливим етапом роботи стало моделювання бізнес-процесів предметної області з використанням методологій IDEF0 та IDEF3. Розроблено діаграми, які детально відображають процеси управління послугами в сфері ІТ, що дозволило глибше зрозуміти структуру та логіку роботи майбутнього веб-застосунку.

У процесі розробки було спроектовано структуру бази даних та користувацький інтерфейс веб-застосунку. Розроблено ER-діаграму, яка описує зв'язки між усіма об'єктами системи, та DFD-діаграми, що візуалізують потоки даних. Створено інтуїтивно зрозумілий інтерфейс, який включає дошки послуг, список управління завданнями та інтеграцію з Google Calendar, що забезпечило ефективну організацію даних та зручність користування системою.

Реалізація функціоналу веб-застосунку проводилася з використанням сучасних технологій, включаючи React.js на фронтенді та Node.js з Express на бекенді. Впроваджено систему авторизації, інтеграцію з Google Calendar API та механізми для ефективного управління завданнями. Це дозволило створити повнофункціональну систему, що відповідає поставленим вимогам.

Проведено комплексне тестування, що включає функціональне, навантажувальне, юзабіліті та тестування безпеки. Для цього було використано наступні інструменти – Postman, Cypress, Apache JMeter та OWASP ZAP.

Тестування інформаційної системи здійснювалося на кожному етапі розробки, що дозволило мінімізувати кількість помилок на фінальному тестуванні й отримати якісний продукт.

Розроблено детальну інструкцію користувача, яка охоплює всі аспекти роботи з веб-застосунком. Це забезпечує швидке освоєння системи та ефективне використання її функціоналу користувачами.

Таким чином, розроблений веб-застосунок для управління послугами в сфері ІТ відповідає поставленим цілям та вимогам. Він забезпечує оптимальний розподіл завдань між співробітниками (спочатку формується список працівників, після чого відбувається перевірка завантаженості працівників в результаті отримуємо список вільних працівників, після чого відбувається перевірка вмінь працівників, тобто проводиться перевірка над якими проектами більше працював працівник й на основі цих даних рекомендується кращий працівник під завдання), ефективну комунікацію в команді та інтеграцію з зовнішніми сервісами. Проведене тестування та аналіз результатів підтвердили надійність та ефективність системи, а також визначили напрямки для її подальшого вдосконалення. Впровадження цього веб-застосунку дозволяє значно підвищити продуктивність та ефективність управління послугами в ІТ-компаніях.

## ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Інформаційні системи для управління послугами: веб-сайт. URL: <http://surl.li/pjnrs> (Дата звернення: 5.12.2023)
2. Договір про надання ІТ послуг. Правила складання MSA/NDA/NCA: веб-сайт. URL: <http://surl.li/pkrpt> (Дата звернення: 5.12.2023)
3. Рішення управління послугами: веб-сайт. URL: <http://surl.li/pjpll> (Дата звернення: 5.12.2023)
4. Особливості управління послугами: веб-сайт. URL: <http://surl.li/pjoqw> (Дата звернення: 6.12.2023)
5. Інформаційні потоки: веб-сайт. URL: <http://surl.li/robu> (Дата звернення: 6.12.2023)
6. Кропивницька М. Е. Практика публічного управління запровадження системи надання послуги раннього втручання: методичні аспекти: Кропивницька М. Е. // Інвестиції: практика та досвід. – 2019. – № 24. – С. 149-155.
7. Atlassian Jira: веб-сайт. URL: <http://surl.li/lsmpr> (Дата звернення: 11.12.2023)
8. Переваги та недоліки Atlassian Jira: веб-сайт. URL: <http://surl.li/pjpro> (Дата звернення: 11.12.2023)
9. Asana: веб-сайт. URL: <http://surl.li/omxnr> (Дата звернення: 11.12.2023)
10. Переваги та недоліки Asana: веб-сайт. URL: <http://surl.li/omxog> (Дата звернення: 11.12.2023)
11. Trello: веб-сайт. URL: <http://surl.li/flxsg> (Дата звернення: 14.12.2023)
12. Переваги та недоліки Trello: веб-сайт. URL: <http://surl.li/pjptw> (Дата звернення: 14.12.2023)
13. Firmo: веб-сайт. URL: <http://surl.li/omxrp> (Дата звернення: 14.12.2023)
14. Переваги та недоліки Firmo: веб-сайт. URL: <http://surl.li/omxpr> (Дата звернення: 14.12.2023)

15. Wrike: веб-сайт. URL: <http://surl.li/omxrn> (Дата звернення: 14.12.2023)
16. Переваги та недоліки Visual Studio Code. URL: <http://surl.li/toxam> (Дата звернення: 14.02.2024)
17. Переваги та недоліки PyCharm. URL: <http://surl.li/toxqw> (Дата звернення: 14.02.2024)
18. Переваги та недоліки Atom. URL: <http://surl.li/toxyu> (Дата звернення: 14.02.2024)
19. Переваги та недоліки Sublime Text. URL: <http://surl.li/tpcbm> (Дата звернення: 14.02.2024)
20. Переваги та недоліки NetBeans. URL: <http://surl.li/tpbzl> (Дата звернення: 14.02.2024)
21. Порівняльний аналіз URL: <http://surl.li/ussgi> (Дата звернення: 15.02.2024)
22. IDF0 URL: <http://surl.li/ussse> (Дата звернення: 15.02.2024)
23. IDF3 URL: <http://surl.li/usspz> (Дата звернення: 15.02.2024)
24. UML-діаграма активності URL: <http://surl.li/ussus> (Дата звернення: 17.02.2024)
25. Ковальчук, М. О., & Войцехівський, В. Б. (2023, May). Web-орієнтована підсистема інтернет-магазину. In The 19th International scientific and practical conference “Innovative approaches to solving scientific problems”(May 16–19, 2023) Tokyo, Japan. International Science Group. 2023. 498 p. (p. 439).
26. Полтавець, М. М. (2017). Сучасні технології інтернет-маркетингу (Doctoral dissertation, РВЛ ЦНТУ).
27. Hnatchenko, D., Hnatchenko, T., & Maliar, V. (2020). Застосування сучасних інформаційних технологій для організації торгівлі у мережі інтернет. Управління розвитком складних систем, (41), 163-168.
28. Гладкий, І. С. (2013). Автоматизація процесів продажу товарів за допомогою Web-технологій. Збірник наукових праць, 31.

# Додатки

## Додаток А

### Аналіз інформаційних систем аналогів

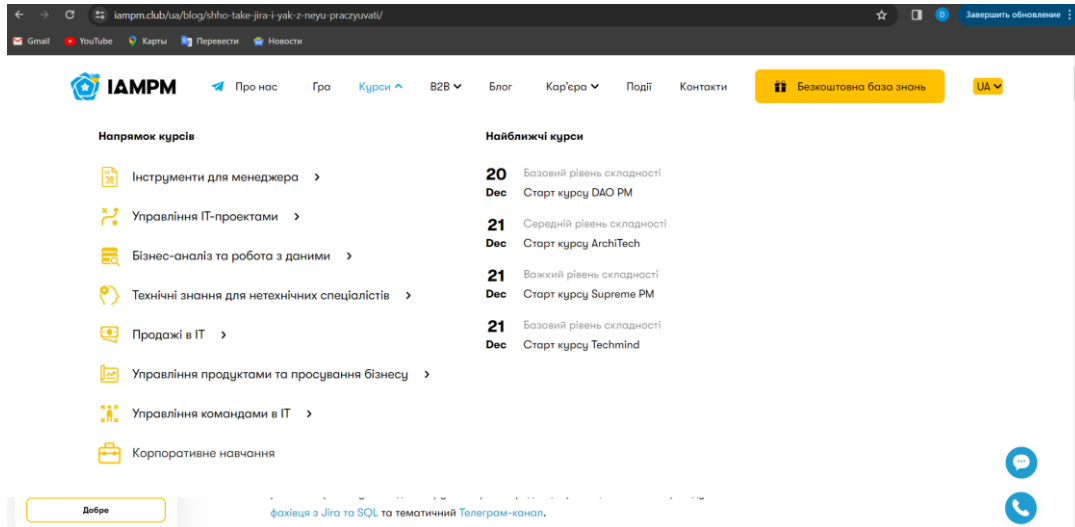


Рис. 1.1 – Головна сторінка Atlassian Jira

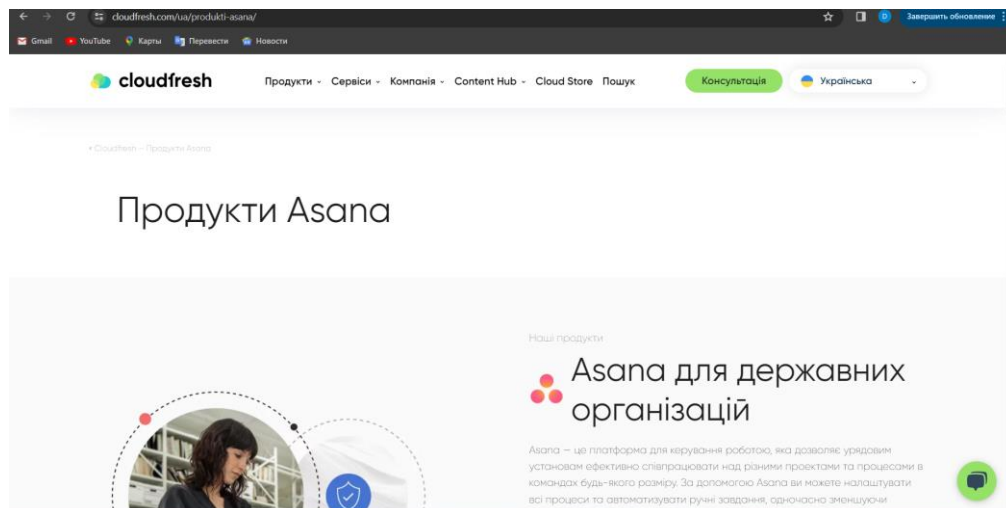


Рис. 1.2 – Головна сторінка Asana

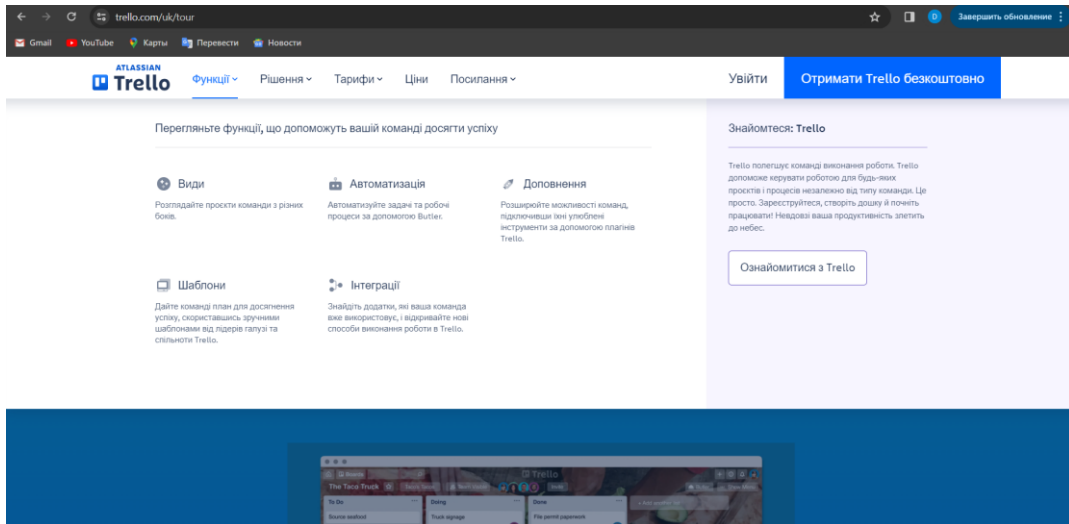


Рис. 1.3 – Головна сторінка Trello

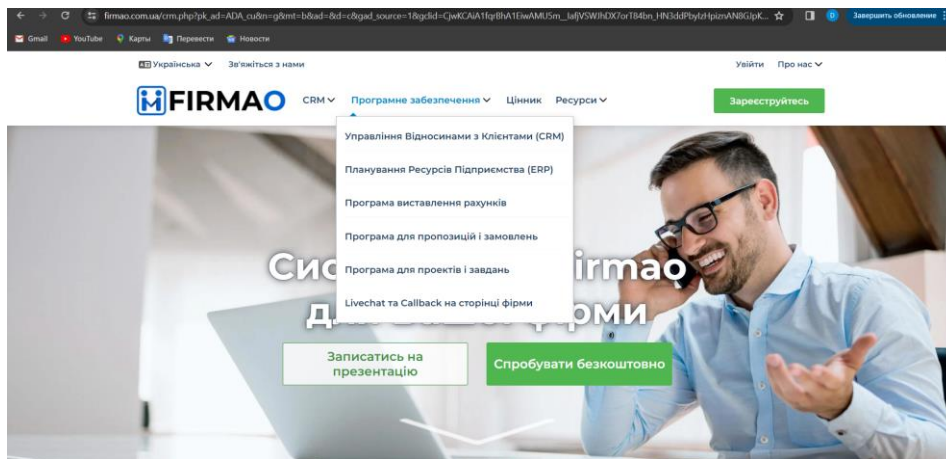


Рис. 1.4 – Головна сторінка Firmao

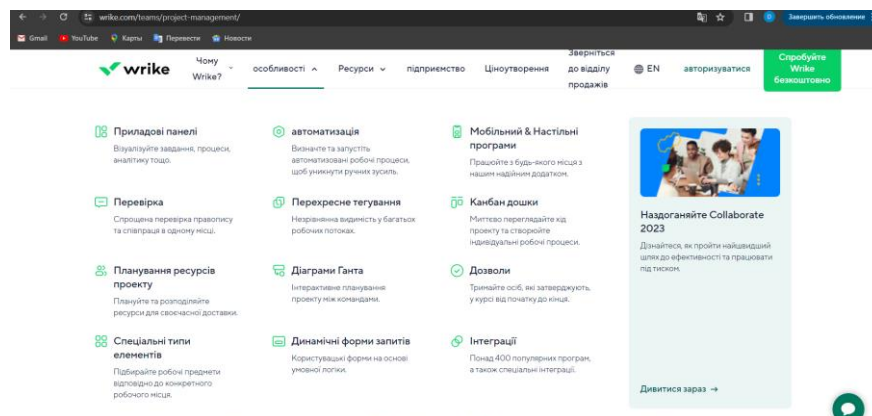


Рис. 1.5 – Сторінка PM Wrike

Назви сайтів	Функціональність	Інтеграція	Простота використання	Інтерфейс
Atlassian Jira	Висока	Висока	Складна	Складний

Asana	Висока	Висока	Легка	Простий
Trello	Висока	Низька	Легка	Простий
Firmao	Висока	Висока	Складна	Складний
Wrike	Низька	Висока	Складна	Простий

Табл. 1.1 – Порівняння аналогів WEB-застосунків для управління послугами в сфері IT

Критерії Середовища	Функціональність	Простота використання	Швидкість роботи	Можливості розширення
Visual Studio Code	5	4	4	5
PyCharm	5	4	4	5
Atom	3	3	3	3
Sublime Text	4	5	5	4
NetBeans	3	3	5	4

Табл. 1.2 – Порівняння сервісів для web-розробки

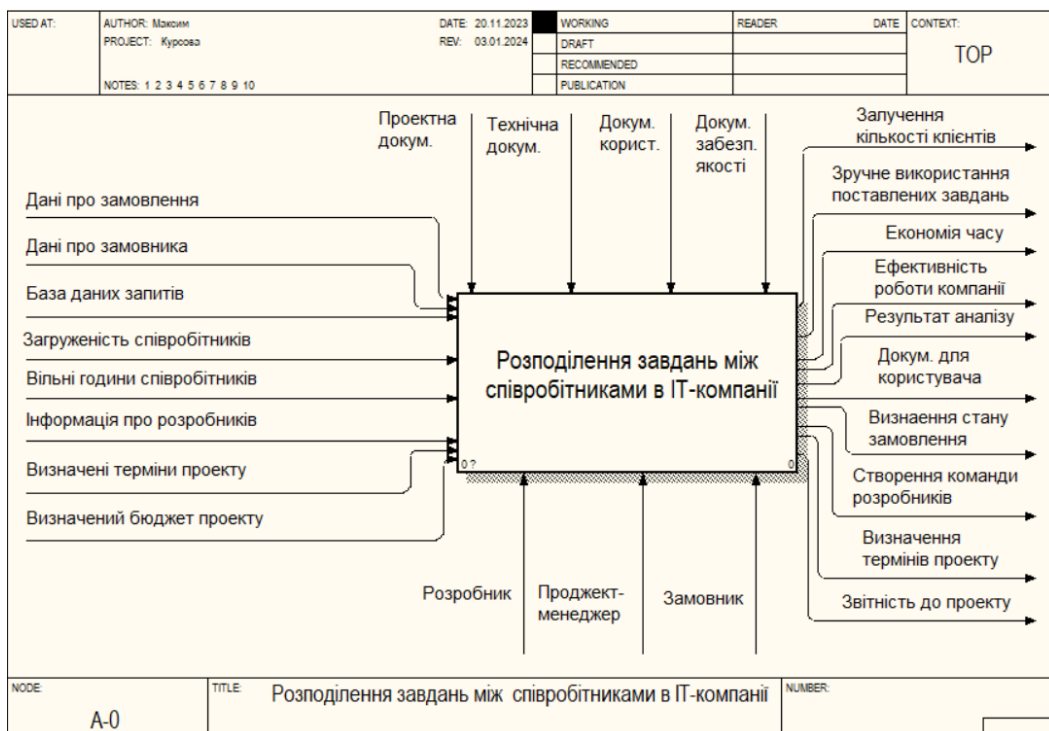




Рис. 1.6 – IDEF0 діаграма для управління послугами в сфері ІТ

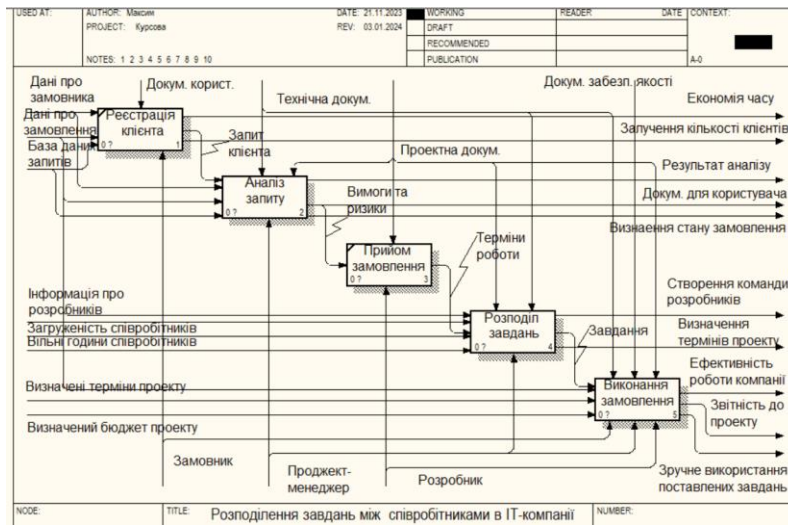


Рис. 1.7 – Декомпозиція IDEF0 Розподілення завдань між співробітниками в ІТ-компанії

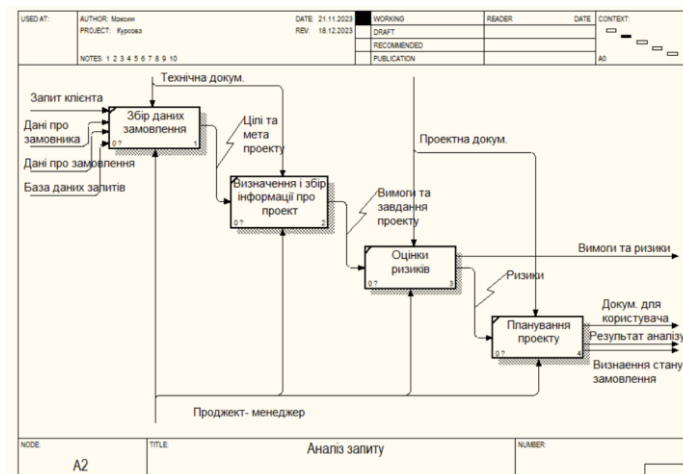


Рис. 1.8 – Декомпозиція IDEF0 Аналізу запиту

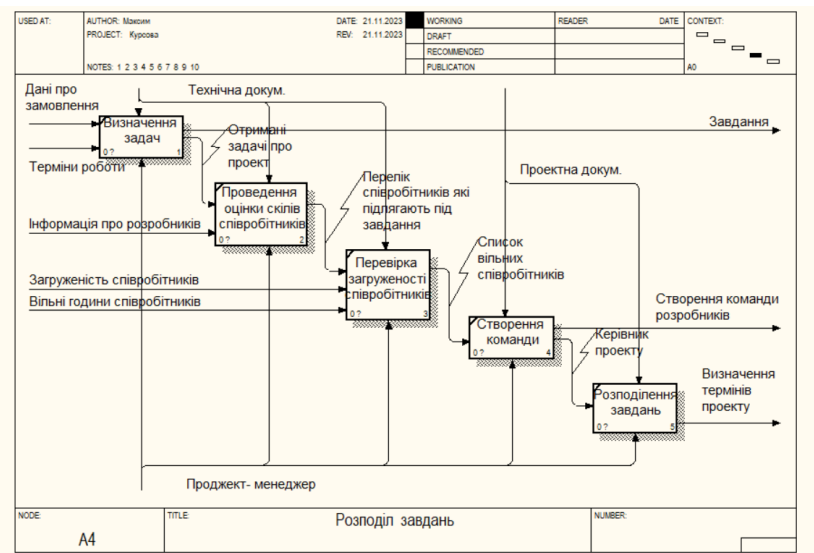


Рис. 1.9 – Декомпозиція IDEF0 Розподілення завдань

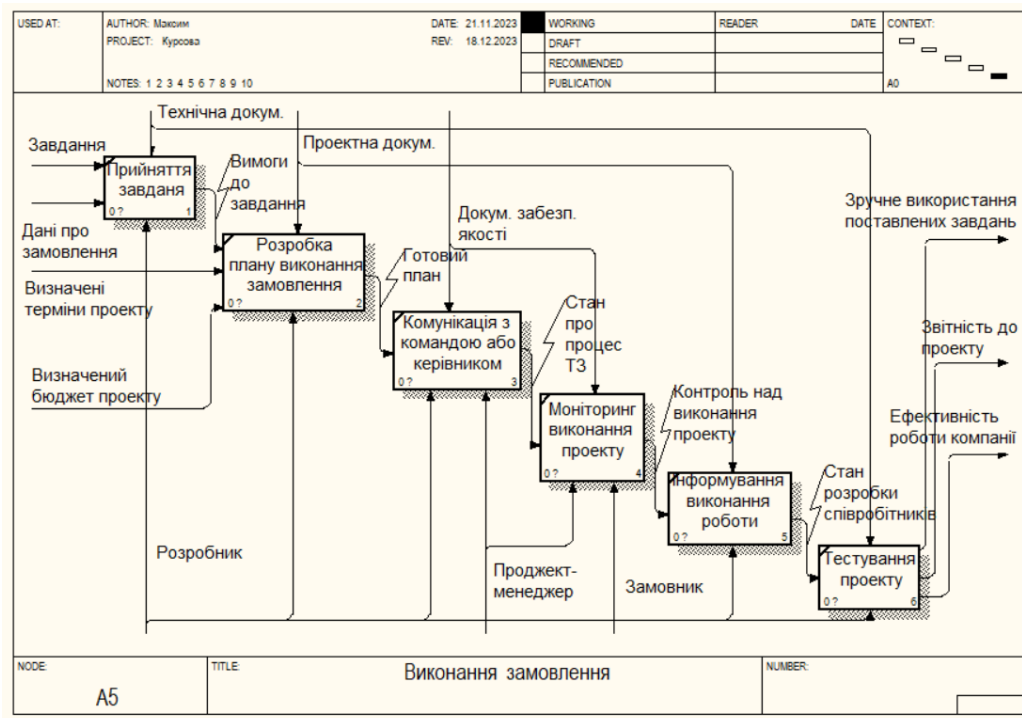


Рис. 1.10 – Декомпозиція IDEF0 Виконання замовлення

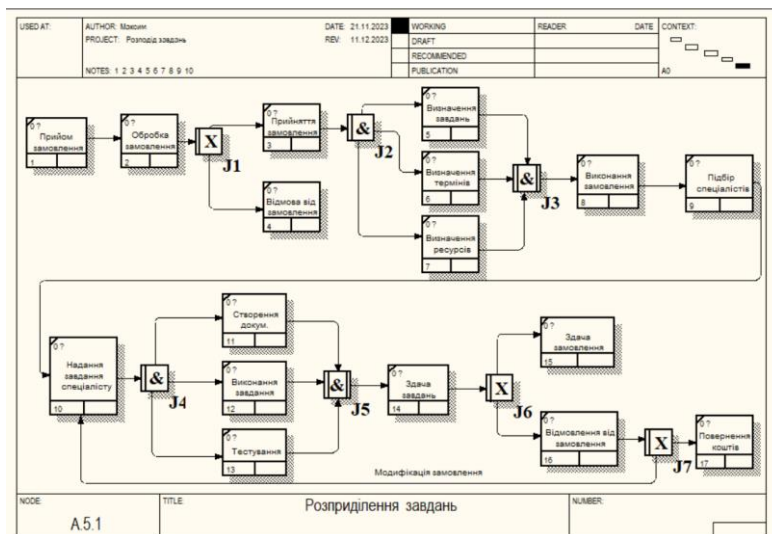


Рис. 1.11 – IDEF3 модель виконання замовлення

## Додаток Б

### Моделювання інформаційної системи

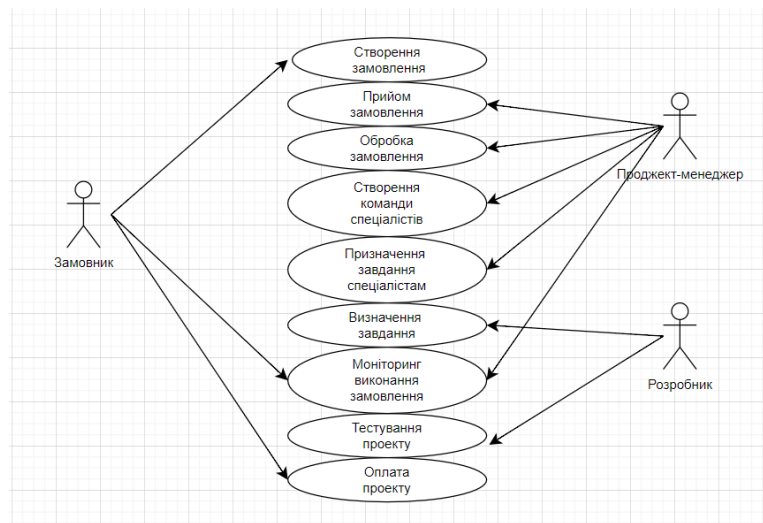


Рис. 2.1 – UML-діаграма прецедентів для управління послугами в сфері ІТ

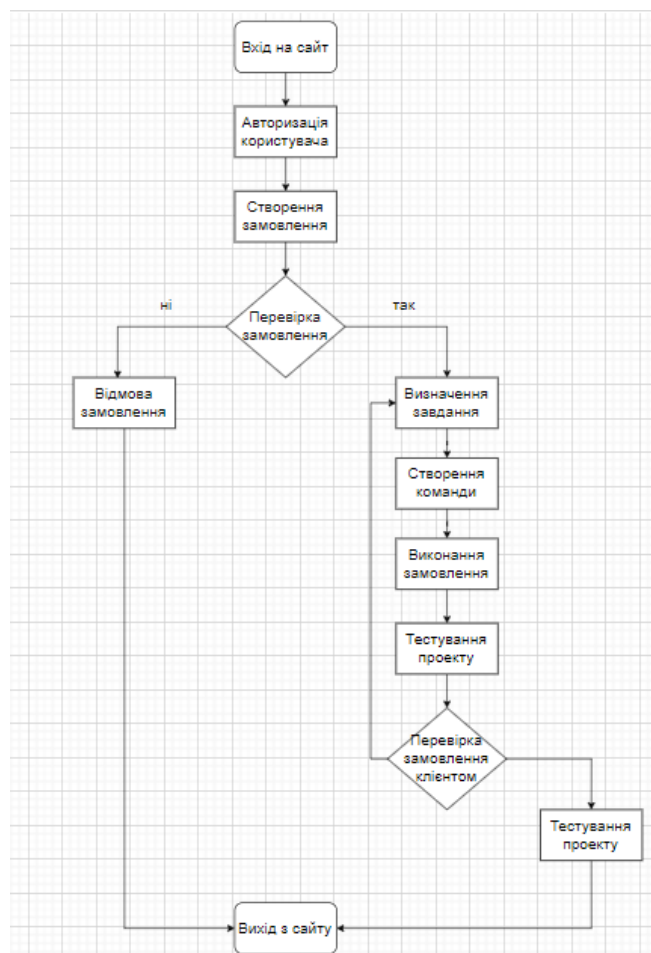


Рис. 2.2 – UML-діаграма активності для управління послугами в сфері ІТ

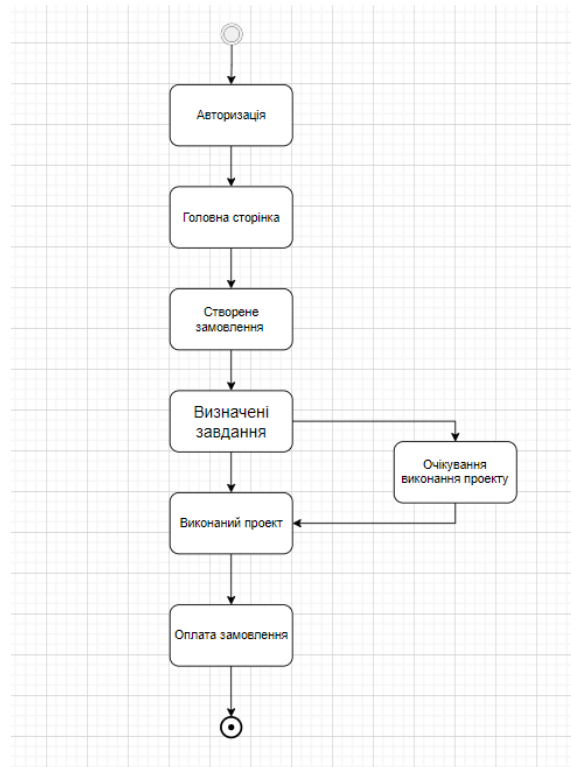


Рис. 2.3 – UML-діаграма станів для управління послугами в сфері ІТ

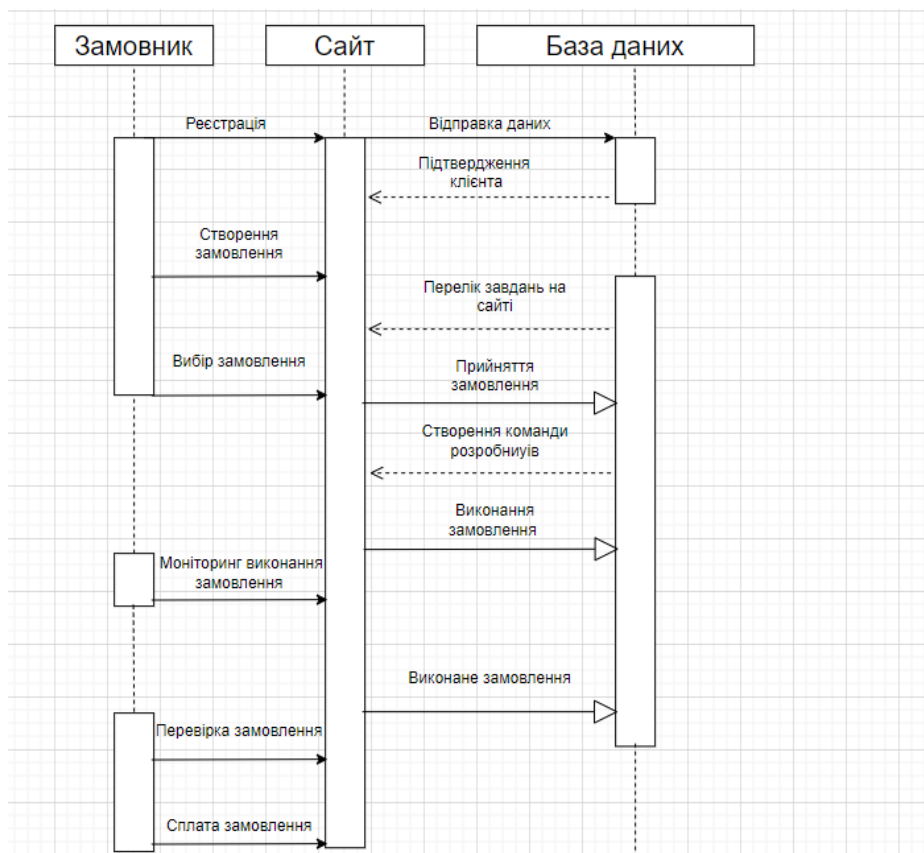


Рис. 2.4 – UML-діаграма послідовності для управління послугами в сфері ІТ

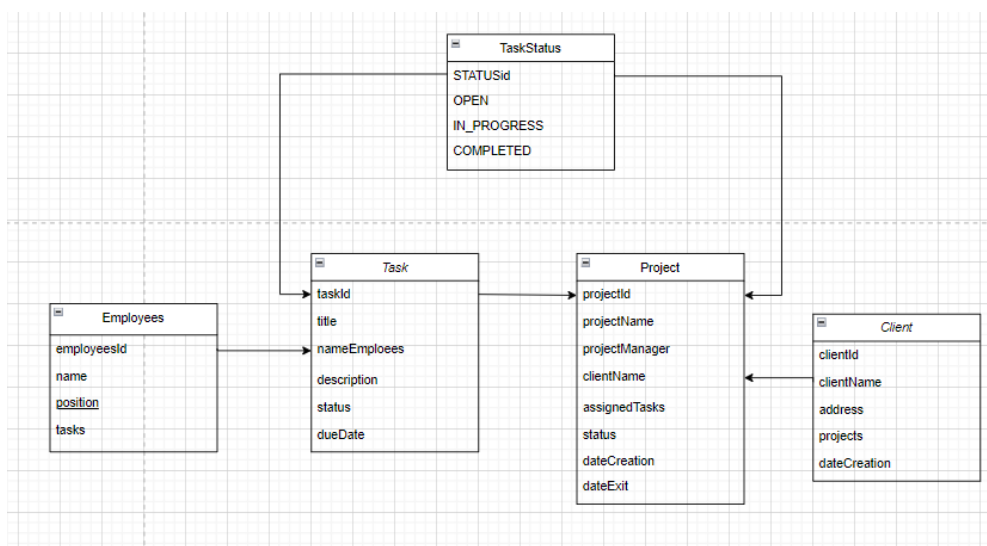


Рис. 2.5 – UML-діаграма класів для управління послугами в сфері ІТ

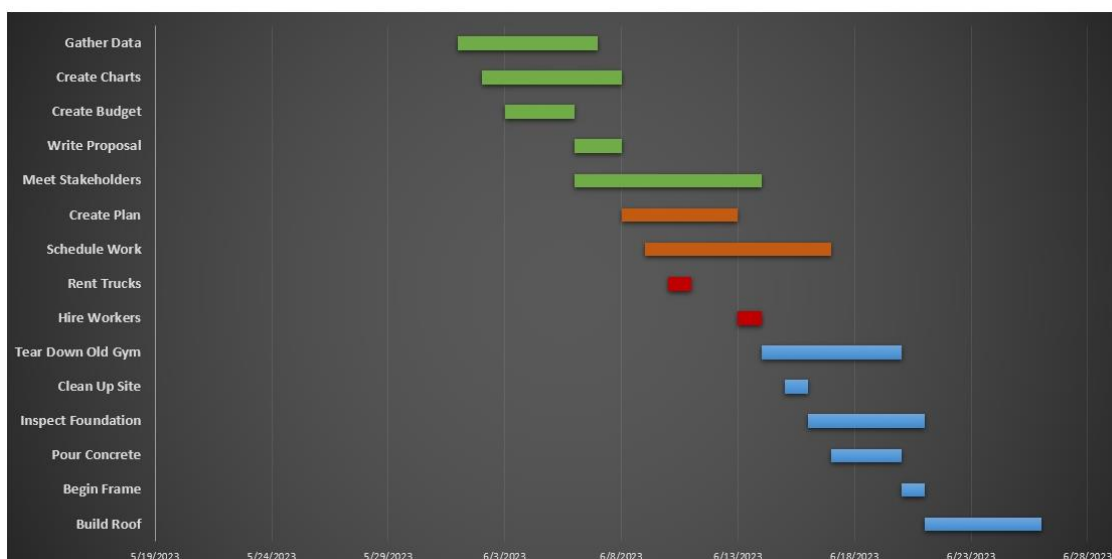


Рис. 2.6 – Діаграма ганта для управління послугами в сфері ІТ

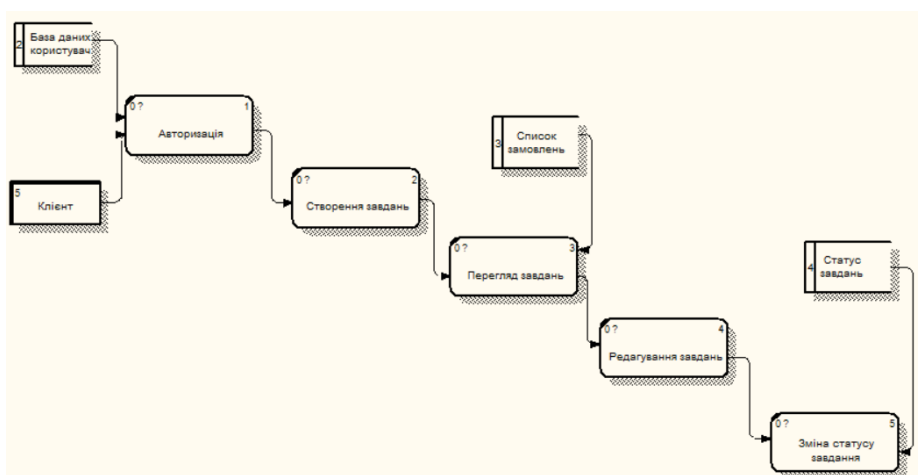


Рис. 2.7 – DFD діаграма створення замовлення

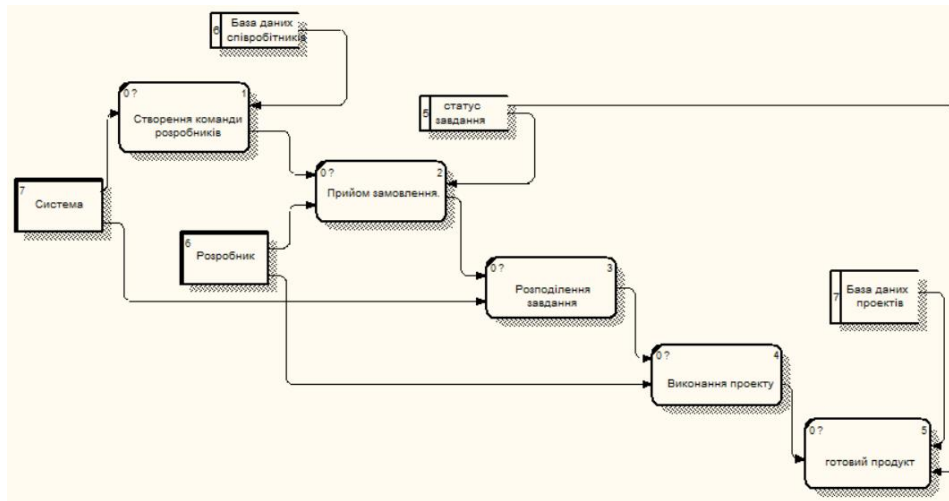


Рис. 2.8 – DFD діаграма розподілення завдань

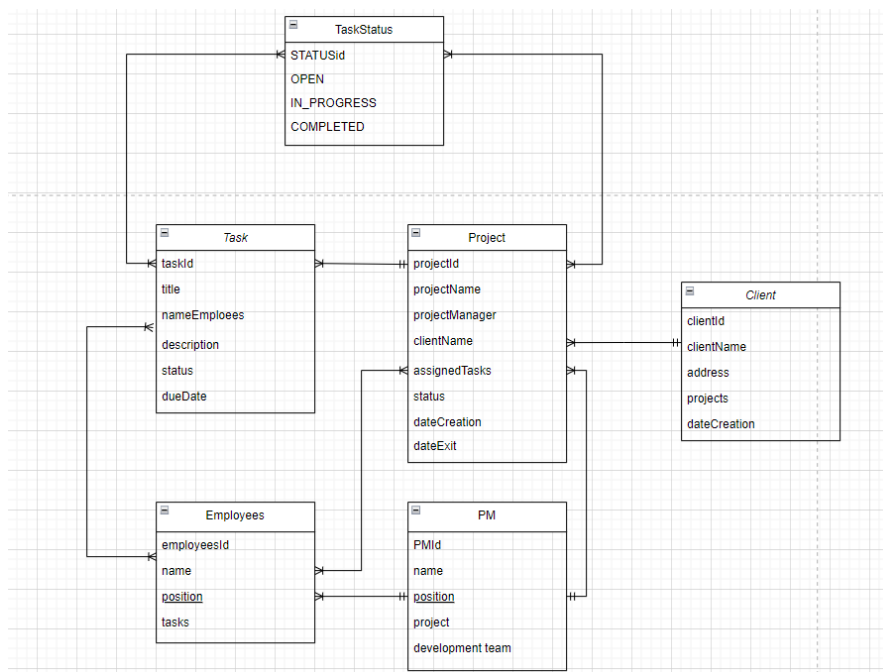


Рис. 2.9 – ER-діаграма для управління послугами в сфері ІТ

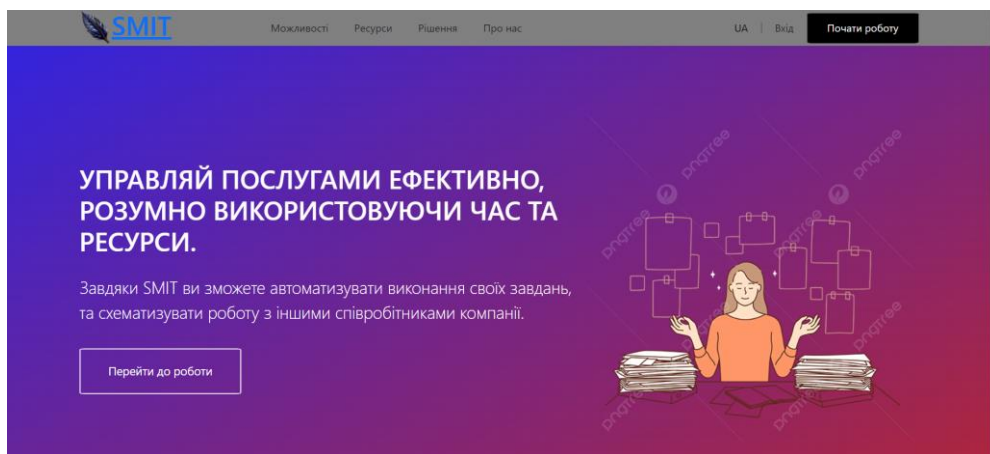


Рис. 2.10 – Головна сторінка WEB-застосунку для управління послугами в сфері ІТ

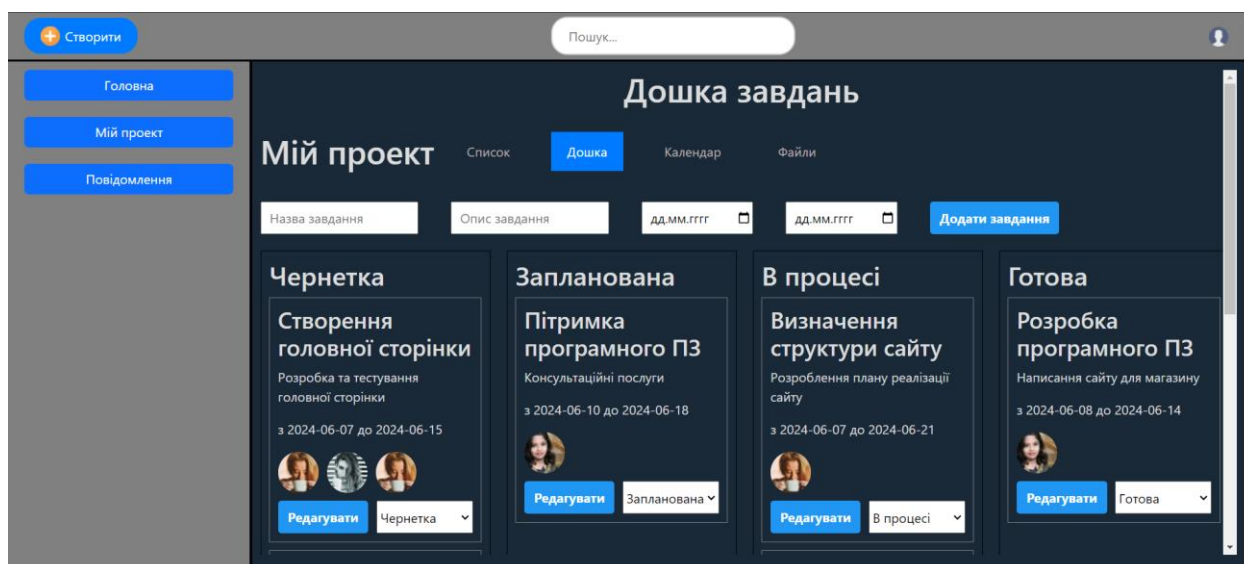


Рис 2.11 – Дошки управління продуктами

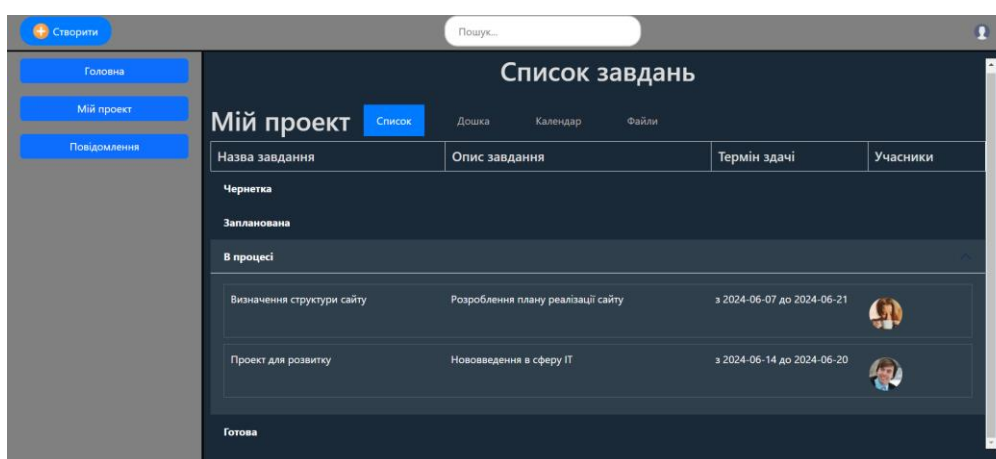


Рис 2.12 – Список управління завданнями

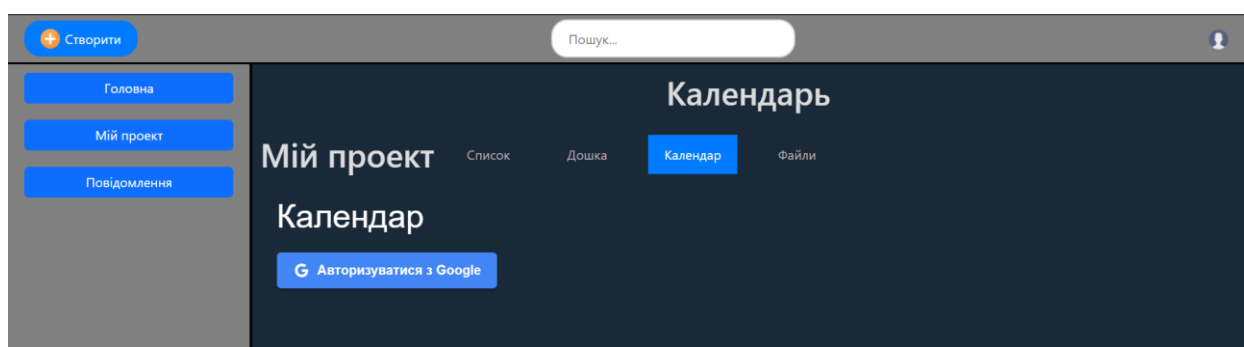


Рис 2.13 – Календар управління завданнями

## Додаток В

## Проектування інтерфейсу інформаційної системи

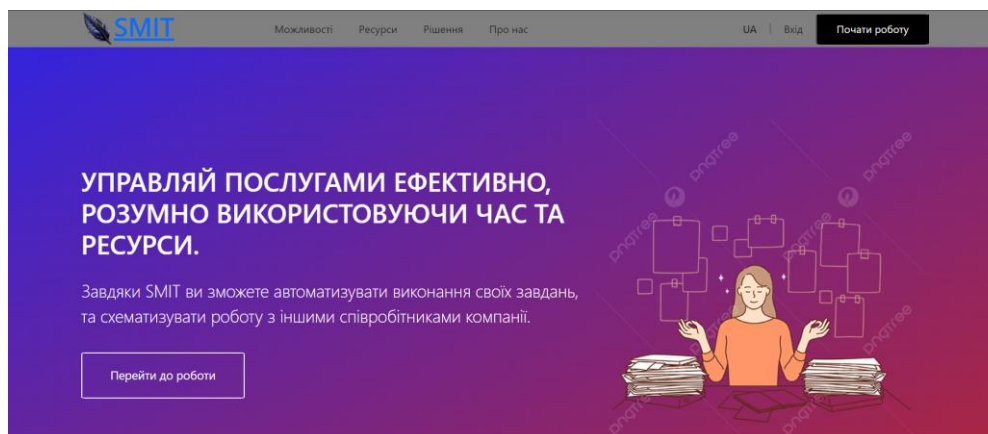


Рис 3.1 – Головна сторінка для користувача web-застосунку для управління послугами в сфері ІТ.

## Реєстрація

Введіть ваш email

Введіть ваш пароль

Вже зареєстровані? [Війти!](#)

Зареєструватись

Рис 3.2 – Сторінка Реєстрації web-застосунку для управління послугами в сфері ІТ.

## Авторизація

Введіть ваш email

Введіть ваш пароль

Не зареєстровані? [Зареєструватись!](#)

Війти



Рис 3.3 – Сторінка Авторизації web-застосунку для управління послугами в сфері ІТ.

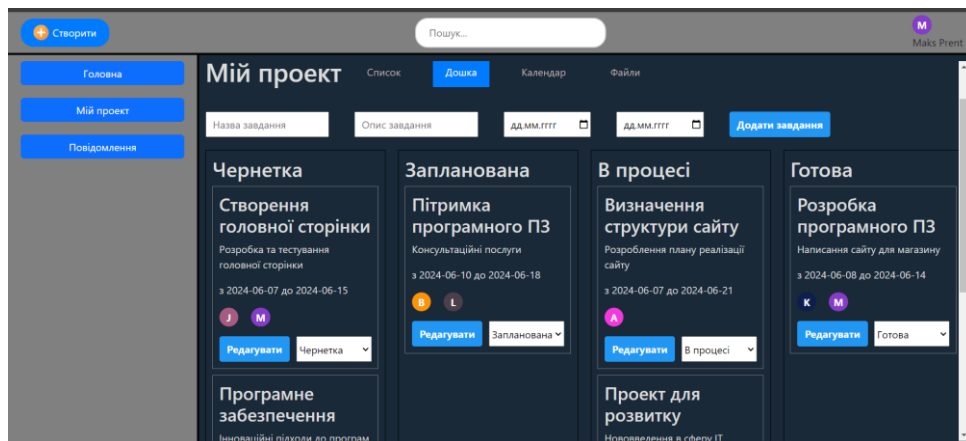


Рис 3.4 – Дошка управління задачами.

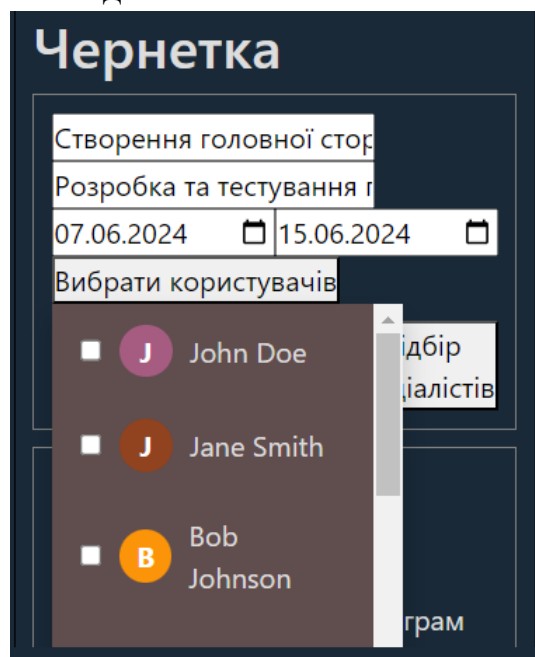


Рис 3.5 – Редагування картки з завданням.

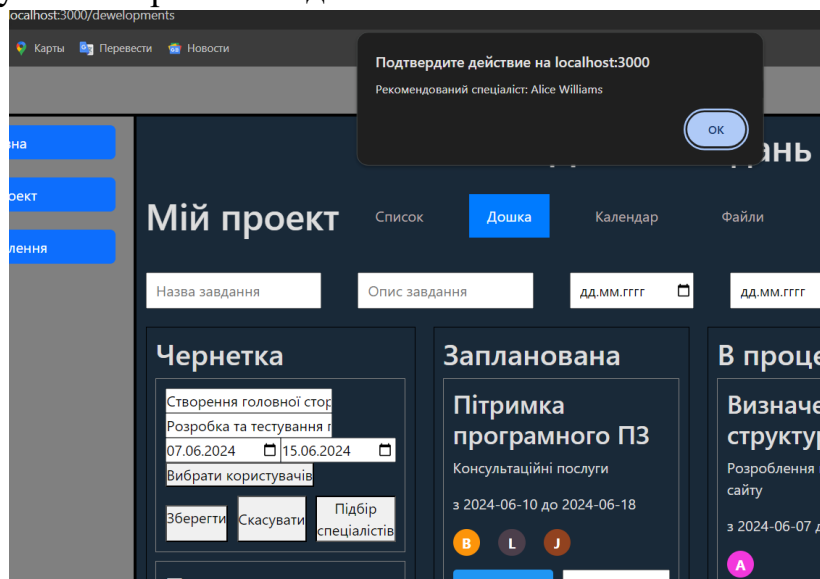


Рис 3.6 – Підбір спеціалістів.

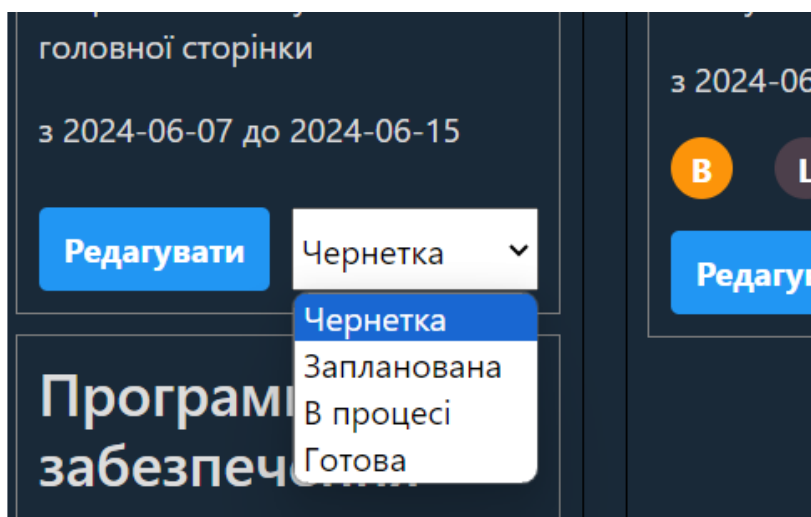


Рис 3.7 – Підбір спеціалістів.



Рис 3.8 – Навігація по завданням в «Мій проект».

Назва завдання	Опис завдання	Термін здачі	Учасники
<b>Чернетка</b>			
Запланована			
<b>В процесі</b>			
Визначення структури сайту	Розроблення плану реалізації сайту	з 2024-06-07 до 2024-06-21	J
Проект для розвитку	Нововведення в сферу ІТ	з 2024-06-14 до 2024-06-20	J
<b>Готова</b>			

Рис 3.9 – Список завдань.

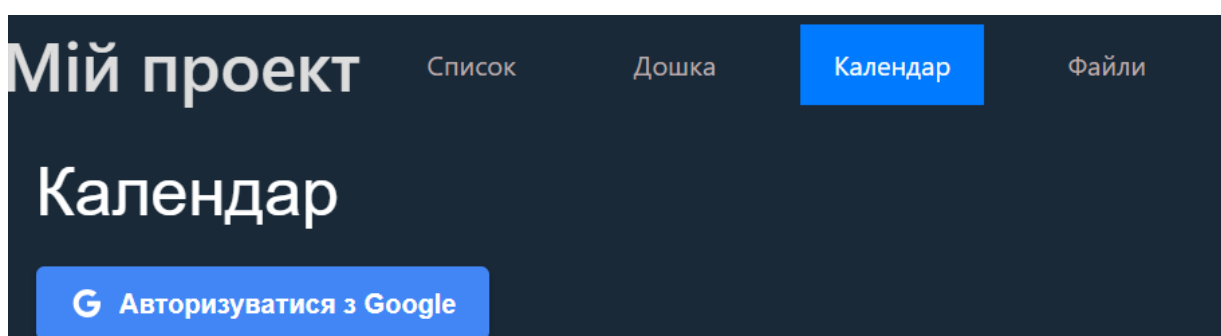


Рис 3.10 – Календар завдань.

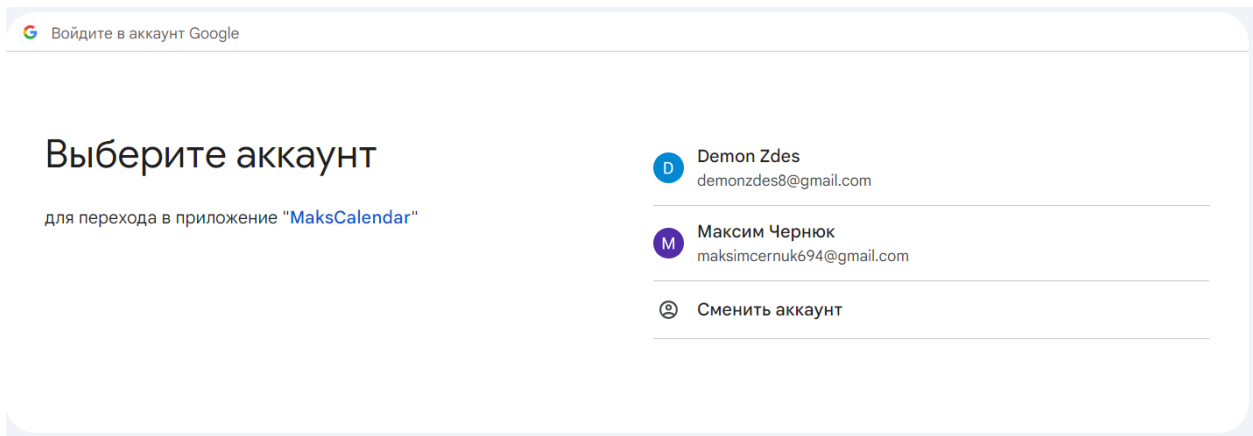


Рис 3.11 – Сторінка авторизації google календаря.

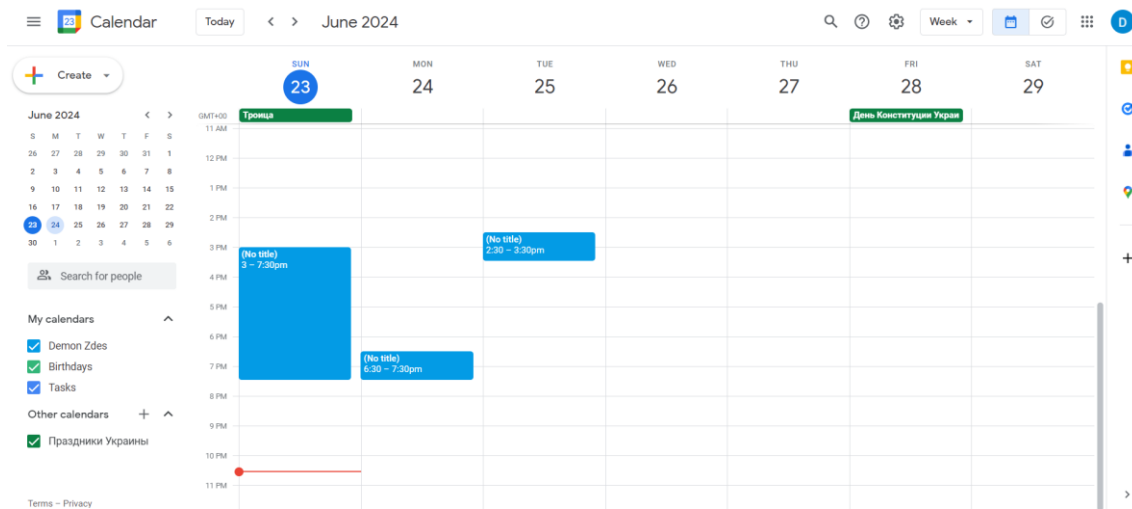


Рис 3.12 – Сторінка google календаря.

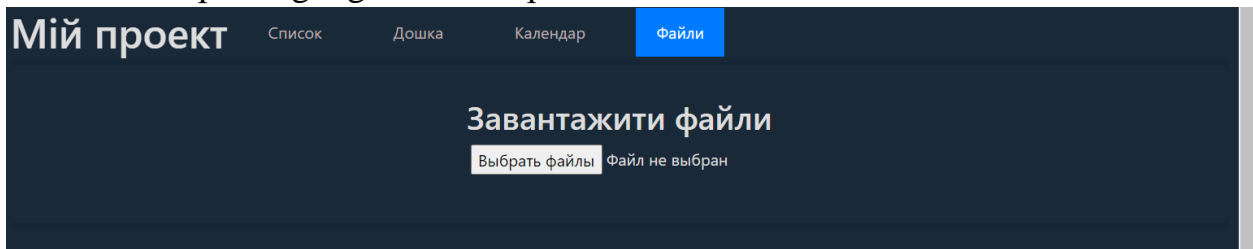


Рис 3.13 – Сторінка завантаження файлів.

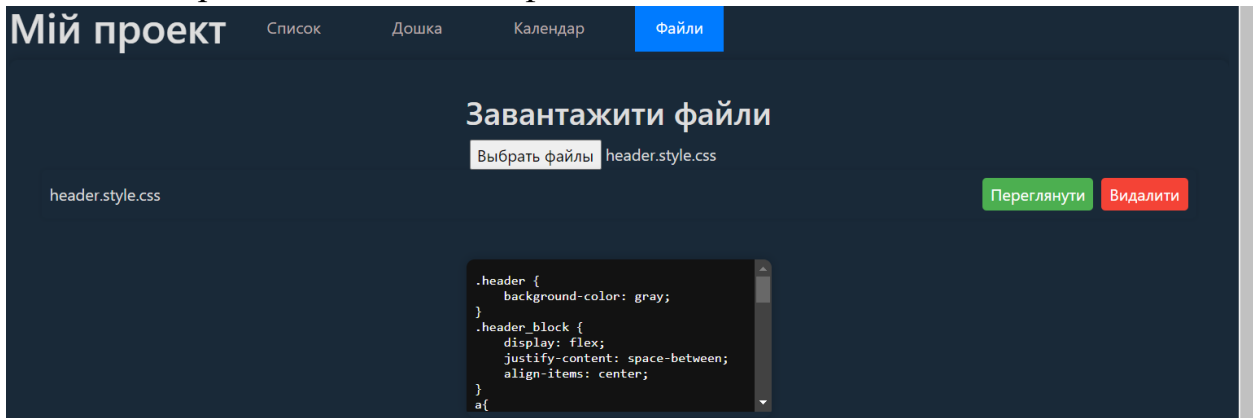


Рис 3.14 – Перегляд та видалення файлів.