

Національний технічний університет України
Київський політехнічний інститут ім. Ігоря Сікорського
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

ЗВІТ
ПРО ПРОХОДЖЕННЯ ПЕРЕДДИПЛОМНОЇ ПРАКТИКИ

Студента(ки) 4 курсу ІТ-01 групи

Спеціальності 121 «Інженерія програмного забезпечення»

Цуканова Марина Сергіївна

(прізвище, ім'я, по батькові студента)

Термін практики з «15» квітня 20 24р. по «19» травня 20 24р.

База практики: ПАТ «Карлсберг Україна»

(назва підприємства)

Керівник від підприємства

Провідний фахівець з ІТ

(науковий ступінь, вчене звання, посада)

Зобнін Вячеслав Анатолійович

(прізвище, ім'я, по батькові, підпис)

Керівник практики від
кафедри ІІІ

старший викладач

(науковий ступінь, вчене звання, посада,)

Вітковська Ірина Іванівна

(прізвище, ім'я, по батькові, підпис)

Київ-2024р.

Пояснювальна записка
до дипломного проєкту

на тему: **Портал служби підтримки ІТ відділу**

КПІ.ІТ-0324.045440.02.81

Зміст

1 ПЕРЕДПРОЄКТНЕ ОБСТЕЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1 Аналіз предметної області	6
1.2 Аналіз існуючих рішень	7
1.2.1 Аналіз відомих програмних продуктів	7
1.3 Опис бізнес-процесів	9
1.4 Постановка задачі	13
Висновки до розділу	14
2 РОЗРОБЛЕННЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	15
2.1 Варіанти використання програмного забезпечення	15
2.2 Аналіз системних вимог	38
2.3 Розроблення функціональних вимог	38
2.4 Розроблення нефункціональних вимог	43
Висновки до розділу	43
3 КОНСТРУЮВАННЯ ТА РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	45
3.1 Архітектура програмного забезпечення	45
3.2 Обґрунтування вибору засобів розробки	46
3.3 Конструювання програмного забезпечення	47
3.4 Аналіз безпеки даних	62
Висновки до розділу	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	64
ДОДАТКИ	65

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IDE	– Integrated Development Environment – інтегроване середовище розробки.
API	– Application programming interface, прикладний програмний Інтерфейс
IT	– Інформаційні технології
ER	– Entity-Relation diagram
БД	– База даних.

Вступ

Сучасні технології створили нові можливості для організацій у сфері обслуговування користувачів, зокрема, в області служб підтримки ІТ відділів. Розробка та впровадження порталів служб підтримки стали важливим кроком для компаній у забезпеченні якісного обслуговування своїх клієнтів.

У цьому контексті, одним із головних завдань є забезпечення швидкого реагування на запити користувачів, надання кваліфікованої допомоги та використання зручних інструментів для обслуговування. Дослідження показують, що відповідно до сучасних тенденцій, більшість користувачів віддають перевагу самообслуговуванню, використовуючи сторінки з відповідями на поширені запитання, чат-ботів та віртуальних помічників[1].

Позитивний досвід роботи зі службою підтримки може сприяти гарній продуктивності під час робочого дня, а негативний досвід спілкування може сприяти незадоволенню користувача, що може результувати у зниженій ефективності роботи [2].

В ході виконання дипломного проекту буде розроблено вебзастосунок portalу служби підтримки ІТ відділу, що надає користувачам можливість отримувати відповіді на поширені запитання, залишати заявки для виконання служби підтримки. Розроблений застосунок надаватиме можливість швидшої обробки заявок для покращення ефективності роботи служби підтримки. Розроблений застосунок матиме зручний та зрозумілий інтерфейс для створення, обробки та відстеження прогресу виконання заявок.

Тема проекту є важливою, оскільки все більше бізнесів мають на меті покращити якість обслуговування користувачів та оптимізувати процес обробки заявок, що сприятиме підвищенню ефективності та задоволеності клієнтів.

1 ПЕРЕДПРОЄКТНЕ ОБСТЕЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області

До головних атрибутів, що є важливими для користувачів служби підтримки, відносяться швидкий час реагування на запит, кваліфікований персонал, кілька контактних осіб та прості у використанні інструменти для обслуговування [3].

Близько 67% клієнтів надають перевагу самообслуговуванню, а не спілкуванню з працівниками служби підтримки [1]. Користувачі можуть перейти на сторінку порталу, побачити контактні номери служби підтримки, переглянути відповіді на поширені запитання. Такий формат порталів є зручним для бізнесу та користувачів, оскільки не потребує великих витрат для створення системи обліку заявок, а користувачі можуть одразу побачити відповіді на питання без необхідності контакту з працівниками служби підтримки.

У випадках, коли зв'язок з службою підтримки є необхідним, користувачі надають перевагу можливості самостійного створення заявки, де вони описують власну проблему, і в разі необхідності працівники зв'язуються з користувачем для уточнення запиту. В разі створення заявок користувачем, заявка оброблюється наступним чином: користувач створює заявку, менеджер отримує заявку, визначає до його типу завдань відноситься ця заявка та призначає відповідного працівника. Цей процес потребує більшого часу для виконання, оскільки менеджеру потрібно особисто прочитати кожную заявку, визначити тип та працівника. Однак, більшість заявок до служби підтримки можна віднести до певних категорій, оскільки вони повторюються.

Для покращення ефективності роботи служби підтримки та зменшення навантаження на менеджерів буде розроблено алгоритм для визначення типу заявки за допомогою методів аналізу тесту. Далі за визначеним типом буде отримано список можливих виконавців та призначено виконавця з доступних.

Таким чином, заявки будуть автоматично направлятись до виконавців без необхідності обробки заявки менеджером. Менеджер матиме змогу самостійно змінити тип заявки та виконавця у разі необхідності. Такий підхід дозволить зменшити час обробки заявки та надасть менеджеру можливість обробляти тільки ті заявки, де не було визначено тип.

1.2 Аналіз існуючих рішень

Проаналізуємо відоме на сьогодні алгоритмічне забезпечення у даній області та технічні рішення, що допоможуть у реалізації порталу служби обслуговування ІТ відділу. Далі будуть розглянуті готові програмні рішення, допоміжні програмні засоби та засоби розробки.

1.2.1 Аналіз відомих програмних продуктів

Більшість компаній, що працюють у сфері інформаційних технологій, мають власну службу підтримки. Зазвичай ці портали працюють у форматі відповідей на поширені запитання, де користувач може знайти контактну інформацію працівників, що можуть вирішити його питання. Також користувачі мають можливість самостійно залишити заявку або питання для вирішення. Для порівняння проекту з аналогом було використано портали служби підтримки “Карлсберг Україна” та “Unilever”.

Для порівняння проекту з аналогом можна скористатись таблицею 1.1.

Таблиця 1.1 – Порівняння з аналогом

Функціонал	Дипломний проект (IT-portal)	Портал служби підтримки “Карлсберг Україна”	Портал служби підтримки Unilever	Пояснення
Перегляд відповідей на запитання	+	+	+	Всі портали надають можливість перегляду відповідей на FAQ

Продовження таблиці 1.1

Необхідність авторизації	+	+	-	На порталі Unilever користувачі можуть знайти контактну інформацію без авторизації
Можливість залишати текстову заявку	+	+	-	На порталі Unilever користувачі можуть залишати питання до служби підтримки
Можливість залишати заявку на отримання пристроїв	+	+	-	Користувачі мають можливість робити запит на отримання, повернення та ремонт пристроїв
Наявність менеджера для обробки заявок	+	+	-	Менеджери мають можливість вручну призначати виконавців
Призначення типу завдань, який може виконувати працівник	+	-	-	В дипломному проекті виконавці мають тип задач, який вони можуть виконувати для покращення автоматизації
Автоматична обробка заявок	+	-	-	Виконавець заявки присвоюється автоматично

1.3 Опис бізнес-процесів

На рисунках 1.1 - 1.5 наведено опис найбільш важливих бізнес процесів за допомогою BPMN моделі.

Опис послідовності авторизації користувача (рисунок 1.1):

- користувач переходить на сторінку авторизації
- користувач заповнює логін та пароль у відповідних полях
- система виконує перевірку на наявність необхідних даних
- якщо всі поля заповнені, система надсилає логін користувача для перевірки наявності такого логіну у системі
- якщо логін не був знайдений, користувачу повертається помилка авторизації
- якщо логін був знайдений, система перевіряє відповідність хешу паролю до хешу, збереженого в базі даних
- якщо пароль не є правильним, користувачу повертається помилка авторизації
- якщо пароль та логін є правильними, система зберігає дані про користувача та переводить користувача на головну сторінку, відповідно до його ролі
- авторизація користувача завершена

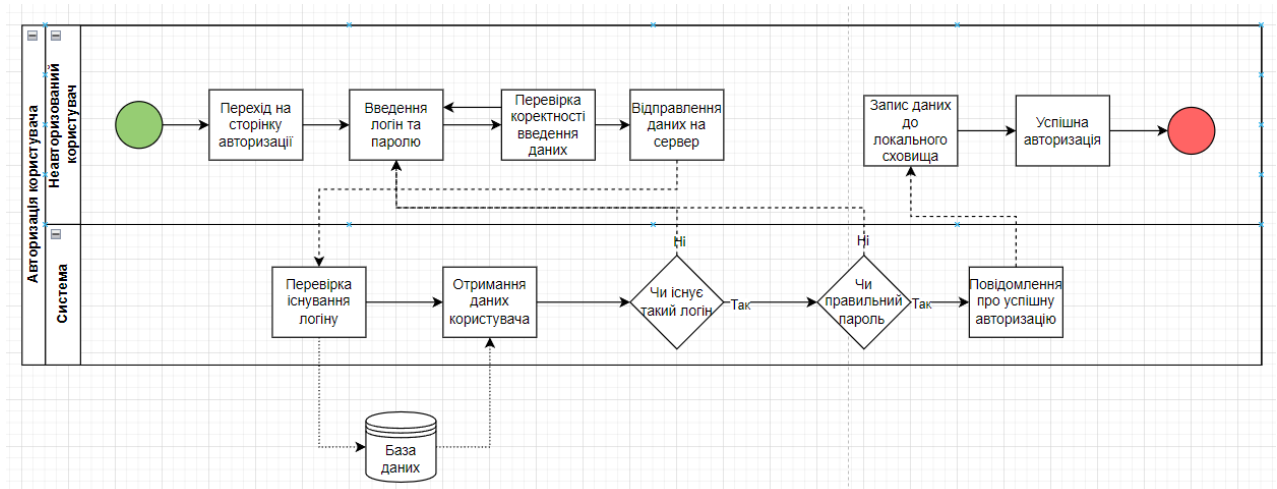


Рисунок 1.1 - BPMN модель послідовності авторизації користувача

Опис послідовності створення заявок (рисунок 1.2):

- користувач переходить на сторінку для створення заявки

- користувач вводить текст заявки у відповідне поле та натискає на кнопку для відправки заявки
- система отримує текст заявки та виконує задану функцію для визначення типу заявки
- після визначення типу заявки, система обирає працівників, що можуть виконувати заявки цього типу, та обирає одного з працівників
- після обрання працівника, система створює новий об'єкт класу заявок, де введена інформація про тип заявки, користувача, що створив заявку, час створення заявки, визначеного працівника та менеджера, обраного випадково
- система додає заявку до бази даних
- заявка створена, користувач може переглянути заявку у списку своїх заявок

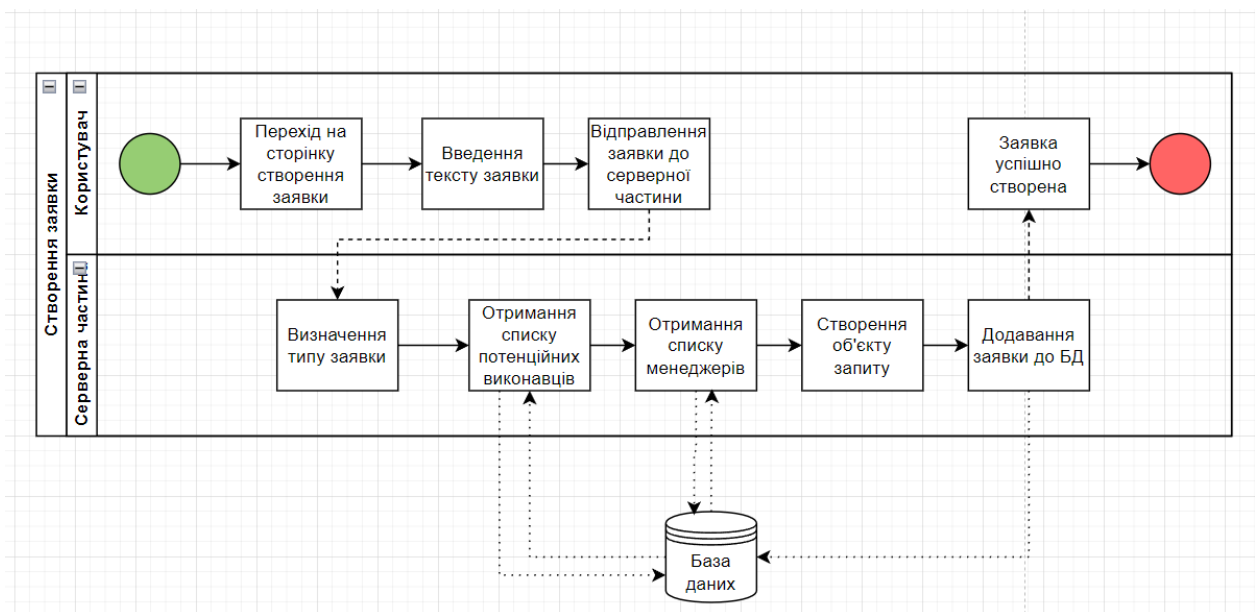


Рисунок 1.2 - BPMN модель послідовності створення заявки

Опис послідовності виконання заявок (рисунок 1.3):

- працівник переходить на сторінку з усіма своїми завданнями
- на сторінці завдання відображена інформація про завдання, отримана за допомогою запиту до бази даних
- на сторінці завдання працівник може змінити статус виконання завдання та підтвердити свій вибір, натиснувши на відповідну кнопку

- після обрання статусу завдання, система оновлює статус виконання завдання у відповідному записі бази даних
- якщо статус виконання завдання є “Completed”, то заявка вважається виконаною і будь-які зміни до статусу виконання стають неможливими
- користувач, що створив заявку, має можливість переглядати статус виконання своїх заявок на сторінці з усіма заявками; статус виконання завдання система отримує з відповідного запису бази даних

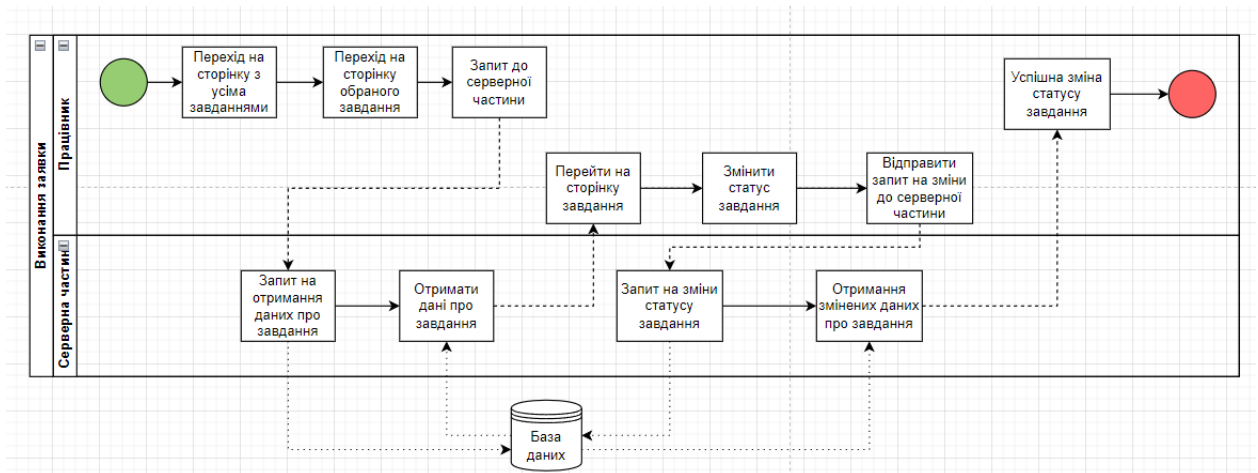


Рисунок 1.3 - BPMN модель послідовності виконання заявки

Опис послідовності отримання нового пристрою (рисунок 1.4):

- користувач знаходить на сторінці з формою для запиту нового пристрою
- користувач обирає тип пристрою та вводить характеристики пристрою у відповідне поле, натискає кнопку для відправки форми
- система обробляє запит користувача, створює нову заявку, яка автоматично має тип “Управління пристроями”, призначає виконавця зі списку виконавців, що можуть виконати цей тип заявок, призначає менеджера, час створення заявки та ким була створена заявка
- система додає запит до бази даних
- виконавець переглядає деталі заявки: для цього він натискає на відповідну кнопку
- система отримує запит на відкриття сторінки

- система отримує інформацію про заявку з бази даних, оскільки тип заявки “Управління пристроями”, то використовується шаблон сторінки для цього типу
- система отримує список пристроїв, що відповідають типу пристрою в заявці та є доступними до отримання, та виводить цей список на сторінку заявки
- виконавець обирає потрібний пристрій зі списку, запропонованих системою, та натискає “Підтвердити”
- система змінює статус заявки та статус пристрою
- користувач, після отримання пристрою, натискає на кнопку “Підтвердити отримання”
- система змінює статус заявки на “Виконаний”, а статус пристрою на “Зайнятий”

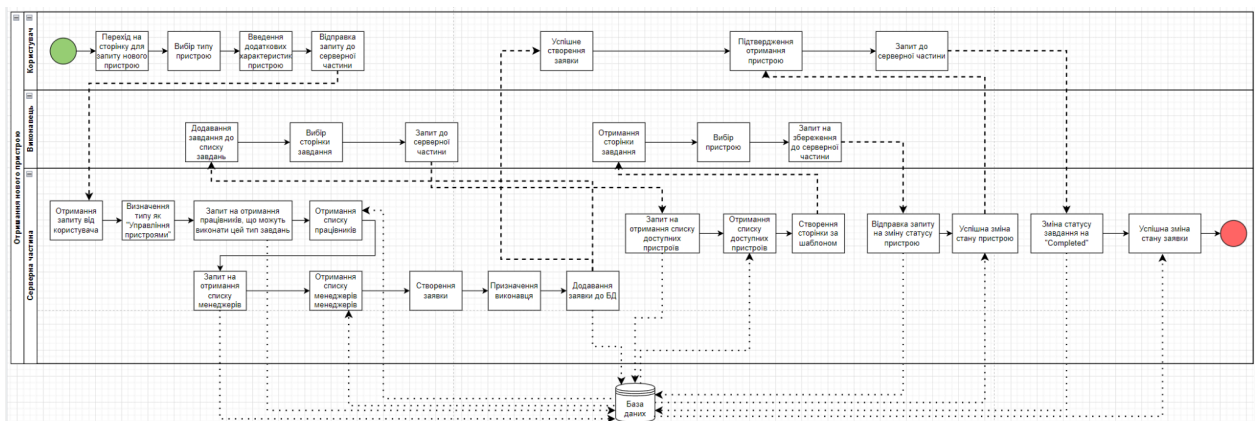


Рисунок 1.4 - BPMN модель послідовності отримання нового пристрою

Опис послідовності повернення пристрою (рисунок 1.5):

- користувач обирає пристрій, який він хоче повернути, та натискає на кнопку “Повернути пристрій”
- система створює запит на повернення пристрою, визначає виконавця заявки зі списку працівників, що можуть виконувати цей тип завдань, призначає менеджера, час створення заявки, користувача, що створив заявку, та тип заявки “Повернення пристрою”
- система додає дані про заявку до бази даних

- призначений виконавець переглядає деталі заявки: для цього він натискає на відповідну кнопку
- система отримує запит на відкриття сторінки
- система отримує інформацію про заявку з бази даних, оскільки тип заявки “Повернення пристрою”, то використовується шаблон сторінки для цього типу
- виконавець, отримавши пристрій, обирає новий статус пристрою (“Доступний” або “На ремонті”) та натискає “Підтвердити”
- система змінює статус виконання заявки на “Виконаний”, статус пристрою змінює на заданий виконавцем
- якщо статус пристрою “На ремонті”, система створює заявку типу “Ремонт пристрою”, визначає виконавця пристрою, менеджера, хто створив заявку та час створення заявки

Вебзастосунок має надавати працівникам служби підтримки можливість обробляти заявки, визначати статус їх виконання, переглядати інформацію, пов'язану з заявкою. Вебзастосунок передбачатиме наявність зв'язків з базою даних для отримання даних про користувачів, заявки та пристрої, що були надані користувачам. Вебзастосунок має мати зрозумілий та зручний для використання інтерфейс.

Висновки до розділу

У цьому розділі було проаналізовано предметну область дипломного проекту, було зроблено висновки про необхідність використання автоматизованих процесів для роботи з користувачами, оскільки швидка роботи служби підтримки є необхідною для ефективного вирішення проблем користувачів та загального покращення роботи компанії.

Під час аналізу аналогічних порталів інших компаній було зроблено висновки, що більшість компаній мають сторінки з поширеними запитаннями, де користувачі можуть самостійно знайти відповіді на власні запитання. Використання алгоритмів обробки тексту заявок не є популярною технологією, проте результати використання таких алгоритмів сприяють більшій швидкості обробки заявок та зменшують навантаження на менеджерів, що самостійно визначають тип кожної заявки.

Найбільш важливі бізнес-процеси були описані за допомогою BPMN моделей. Було наведено опис бізнес-процесів створення та обробки текстових заявок та заявок на отримання пристроїв. Також було описано процес авторизації користувачів та визначення ролі користувача з використанням бази даних.

У цьому розділі було сформульовано та наведено постановку задачі дипломного проекту, визначено мету проекту та необхідні властивості майбутнього проекту.

2 РОЗРОБЛЕННЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Варіанти використання програмного забезпечення

Головною функцією вебзастосунку є обробка заявок, що надходять до служби підтримки ІТ- відділу , більше функцій можна побачити на рисунку 2.1.



Рисунок 2.1 – Діаграма варіантів використання

В таблицях 2.2 - 2.33 наведені варіанти використання програмного забезпечення.

Таблиця 2.2 - Варіант використання UC-1

Use case name	Авторизація користувача
Use case ID	UC-01
Goals	Авторизація користувача в системі
Actors	Гість (неzareєстрований користувач)
Trigger	Гість бажає увійти на свою сторінку
Pre-conditions	-
Flow of Events	Користувач переходить на сторінку авторизації. В поля для авторизації вводять логін та пароль до свого акаунту. Після заповнення полів користувач натискає на кнопку Login. Після цього користувач перенаправляється на головну сторінку свого акаунту згідно з заданою роллю.
Extension	В випадку введення не коректних даних, сторінка оновлюється та некоректні данні очищуються з полів. Над полями вводу з'являється повідомлення про те, яка помилка відбулась.
Post-Condition	Перехід на головну сторінку користувача

Таблиця 2.3 - Варіант використання UC-2

Use case name	Вихід з акаунту
Use case ID	UC-02
Goals	Користувач виходить з акаунту в системі
Actors	Авторизований користувач
Trigger	Гість бажає вийти зі своєї сторінки
Pre-conditions	-

Продовження таблиці 2.3

Flow of Events	Користувач натискає на кнопку Sign out на навігаційній панелі. Після натискання користувач перенаправляється на сторінку входу.
Extension	-
Post-Condition	Перехід на сторінку входу, користувач стає неавторизованим

Таблиця 2.4 - Варіант використання UC-3

Use case name	Додавання нового користувача
Use case ID	UC-03
Goals	Створити нового користувача
Actors	Авторизований користувач, що має роль Admin
Trigger	Адміністратор бажає додати нового користувача
Pre-conditions	-
Flow of Events	Адміністратор обирає вкладку Create a new user на навігаційній панелі та натискає на неї. Відкривається вікно Create a new user, де адміністратор вводять інформацію про користувача: ім'я та прізвище, адресу електронної пошти, номер телефону, назву посади, логін та пароль, відділ та роль користувача. Після вводу необхідної інформації адміністратор натискає кнопку Submit. Після успішного створення нового користувача, адміністратор перенаправляється до головної сторінки.
Extension	В випадку введення не коректних даних, сторінка оновлюється та некоректні данні очищуються з полів.

Продовження таблиці 2.4

Post-Condition	Перехід на головну сторінку адміністратора. Новий користувач був успішно створений.
----------------	---

Таблиця 2.5 - Варіант використання UC-4

Use case name	Видалення користувача
Use case ID	UC-04
Goals	Видалити користувача та дані про нього
Actors	Адміністратор
Trigger	Адміністратор бажає видалити інформацію про користувача
Pre-conditions	-
Flow of Events	Адміністратор знаходиться на головній сторінці. Адміністратор обирає користувача інформацію, про якого він бажає видалити та натискає на кнопку Delete у рядку з інформацією про користувача. Користувач видаляється, а сторінка оновлюється.
Extension	-
Post-Condition	Інформація про користувача видалена з системи

Таблиця 2.6 - Варіант використання UC-5

Use case name	Редагування інформації про користувача
Use case ID	UC-05
Goals	Зміна інформації про користувача
Actors	Адміністратор

Продовження таблиці 2.6

Trigger	Адміністратор бажає змінити дані про користувача
Pre-conditions	-
Flow of Events	Адміністратор знаходиться на головній сторінці. Адміністратор обирає користувача інформацію, про якого він бажає видалити та натискає на кнопку Edit у рядку з інформацією про користувача. Відкривається сторінка Edit user, що містить вже існуючі дані про користувача у полях: ім'я та прізвище, адресу електронної пошти, номер телефону, назву посади, логін та пароль, відділ та роль користувача. Після зміни необхідної інформації адміністратор натискає кнопку Submit. Після успішної зміни інформації про користувача, адміністратор перенаправляється до головної сторінки.
Extension	В випадку введення не коректних даних, сторінка оновлюється та у полях відображаються попередні дані.
Post-Condition	Перехід на головну сторінку адміністратора. Інформація про користувача була успішно змінена.

Таблиця 2.7 - Варіант використання UC-6

Use case name	Перегляд обраних користувачів
Use case ID	UC-06
Goals	Відобразити список користувачів, що відповідає заданим параметрам
Actors	Адміністратор

Продовження таблиці 2.7

Trigger	Адміністратор бажає переглянути список користувачів, обраних за параметрами
Pre-conditions	-
Flow of Events	Адміністратор знаходиться на головній сторінці. Адміністратор обирає параметри для користувачів з запропонованих. Таблиця, що відображає список користувачів, змінюється та відображає тільки користувачів, що відповідають заданим параметрам.
Extension	-
Post-Condition	Адміністратор може побачити список користувачів, що відповідають обраним параметрам

Таблиця 2.8 - Варіант використання UC-7

Use case name	Пошук заданого користувача
Use case ID	UC-07
Goals	Пошук заданого користувача
Actors	Адміністратор
Trigger	Адміністратор бажає знайти користувача за заданими ключовими словами
Pre-conditions	-
Flow of Events	Адміністратор знаходиться на головній сторінці. Адміністратор вводять ключове слово в строку пошуку. Таблиця, що відображає список користувачів, змінюється та відображає тільки користувачів, що містять задане ключове слово.

Продовження таблиці 2.8

Extension	В випадку введення ключового слова, якому не відповідає жодний користувач, відображається повідомлення “Searched user doesn’t exist. Try again”. При видаленні ключового слова таблиця повертається до відображення всіх записів.
Post-Condition	Адміністратор може переглянути список користувачів, що відповідають ключовому слову

Таблиця 2.9 - Варіант використання UC-8

Use case name	Перегляд списку всіх запитів
Use case ID	UC-08
Goals	Перегляд списку всіх запитів
Actors	Авторизований користувач з роллю Manager
Trigger	Менеджер бажає переглянути список всіх запитів, де він є менеджером
Pre-conditions	-
Flow of Events	Менеджер обирає вкладку “Manage requests” на навігаційній панелі. В результаті відображається список заявок, де користувач є менеджером. Користувач має змогу переглянути список заявок за заданими параметрами та ключовими словами.
Extension	-
Post-Condition	Перехід на головну сторінку менеджера, що містить список всіх заявок

Таблиця 2.10 - Варіант використання UC-9

Use case name	Перегляд інформації про заявку
Use case ID	UC-09
Goals	Перегляд інформації про обрану заявку
Actors	Менеджер
Trigger	Менеджер бажає переглянути інформацію про заявку
Pre-conditions	-
Flow of Events	Менеджер знаходиться на головній сторінці та обирає заявку, інформацію про яку він бажає переглянути. У відповідній строці натискає на кнопку Review. Відкривається сторінка Request info, де відображається інформація про заявку: текст заявки, ким була створена заявка, час створення заявки, статус заявки, виконавця заявки, час завершення виконання заявки, тип завдання.
Extension	-
Post-Condition	Перехід на сторінку заявки

Таблиця 2.11 - Варіант використання UC-10

Use case name	Перегляд списку наданих пристроїв
Use case ID	UC-10
Goals	Перегляд списку всіх наданих пристроїв
Actors	Менеджер
Trigger	Менеджер бажає переглянути список всіх пристроїв, що були надані та статус цього зв'язку
Pre-conditions	-

Продовження таблиці 2.11

Flow of Events	Менеджер натискає на вкладку Manage devices на навігаційній панелі. Відкривається сторінка Manage devices, що містить інформацію про всі зв'язки між працівниками та пристроями. Може переглянути список обраних пристроїв за допомогою параметрів та ключових слів.
Extension	-
Post-Condition	Перехід на сторінку для перегляду статусу наданих пристроїв

Таблиця 2.12 - Варіант використання UC-11

Use case name	Додавання нового пристрою до системи
Use case ID	UC-11
Goals	Додавання нового пристрою до системи
Actors	Менеджер
Trigger	Менеджер бажає додати новий пристрій до системи
Pre-conditions	-
Flow of Events	Адміністратор знаходиться на сторінці Manage devices та натискає на кнопку “Add a new device”. Відкривається сторінка “Add a new device”, де менеджер вводять у відповідні поля необхідну інформацію: назва пристрою, серійний номер, тип пристрою, бренд пристрою. Менеджер натискає кнопку “Add device” та перенаправляється на сторінку зі списком всіх пристроїв.

Продовження таблиці 2.12

Extension	В випадку введення не коректних даних, сторінка оновлюється та некоректні данні очищуються з полів. Над полями вводу з'являється повідомлення про те, яка помилка відбулась. В випадку успішного створення користувача, на головній сторінці з'являється повідомлення "New device was successfully added".
Post-Condition	Перехід на сторінку з усіма пристроями. Новий пристрій додається до системи.

Таблиця 2.13 - Варіант використання UC-12

Use case name	Перегляд списку всіх пристроїв
Use case ID	UC-12
Goals	Перегляд списку всіх пристроїв
Actors	Менеджер
Trigger	Менеджер бажає переглянути список всіх пристроїв
Pre-conditions	-
Flow of Events	Менеджер знаходиться за сторінці Manage devices, що містить інформацію про зв'язки між користувачами та пристроями, та натискає на кнопку "View devices list". Далі менеджер перенаправляється на сторінку, де відображені всі пристрої. Менеджер має змогу переглянути пристрої за параметрами та ключовими словами.
Extension	-
Post-Condition	Перехід на сторінку з усіма пристроями

Таблиця 2.14 - Варіант використання UC-13

Use case name	Змінити статус пристрою на “Не доступний”
Use case ID	UC-13
Goals	Змінити статус пристрою на “Не доступний”
Actors	Менеджер
Trigger	Менеджер змінити статус пристрою на “Не доступний”
Pre-conditions	-
Extension	-
Flow of Events	Менеджер знаходиться за сторінці зі списком усіх пристроїв, обирає пристрій, статус якого він бажає змінити, та натискає на кнопку “Make not available” у відповідному рядку.
Post-Condition	Статус пристрою змінено

Таблиця 2.15 - Варіант використання UC-14

Use case name	Перегляд списку працівників відділу
Use case ID	UC-14
Goals	Перегляд списку працівників відділу
Actors	Менеджер
Trigger	Менеджер бажає переглянути список працівників його відділу
Pre-conditions	-

Продовження таблиці 2.15

Flow of Events	Менеджер обирає вкладку “Manage workers” на навігаційній панелі. Менеджер перенаправляється на сторінку, де відображені всі працівники його відділу та інформація про них: ім’я та прізвище, посада, адреса електронної пошти, номер телефону та обов’язки.
Extension	-
Post-Condition	Перехід на сторінку зі списком працівників відділу

Таблиця 2.16 - Варіант використання UC-15

Use case name	Редагування обов’язків працівника
Use case ID	UC-15
Goals	Редагування обов’язків обраного працівника
Actors	Менеджер
Trigger	Менеджер бажає змінити обов’язки обраного працівника
Pre-conditions	-
Flow of Events	Менеджер знаходиться за сторінці зі списком всіх робітників, обирає працівника, обов’язки якого він хоче відредагувати, та натискає на кнопку “Edit” у відповідному рядку. Далі менеджер натискає кнопку “Submit” для збереження результатів.
Extension	-
Post-Condition	Обов’язки працівника були оновлені

Таблиця 2.17 - Варіант використання UC-16

Use case name	Перегляд власних завдань
Use case ID	UC-16
Goals	Перегляд списку власних завдань працівника
Actors	Авторизований користувач з роллю “Worker”
Trigger	Працівник бажає переглянути список власних завдань
Pre-conditions	-
Flow of Events	Працівник натискає на вкладку “My tasks” на навігаційній панелі. Працівник перенаправляється на сторінку з власними завданнями, де можна побачити інформацію про завдання та виконати пошук за параметрами та ключовими словами.
Extension	-
Post-Condition	Перехід на сторінку із задачами для працівника

Таблиця 2.18 - Варіант використання UC-17

Use case name	Відмовитись від завдання
Use case ID	UC-16
Goals	Відмовитись від завдання з переліку завдань
Actors	Працівник
Trigger	Працівник бажає відмовитись від виконання завдання
Pre-conditions	Кнопка “Reassign” є активною

Продовження таблиці 2.18

Flow of Events	Працівник знаходиться на сторінці з усіма завданнями, обирає завдання, від якого він хоче відмовитись, та натискає кнопку “Reassign” у відповідному рядку. Сторінка оновлюється, а обране завдання більше не призначено.
Extension	-
Post-Condition	Завдання більше не є призначеним до працівника

Таблиця 2.19 - Варіант використання UC-18

Use case name	Перегляд інформації про завдання
Use case ID	UC-18
Goals	Переглянути інформацію про завдання
Actors	Працівник
Trigger	Працівник бажає переглянути інформацію про завдання
Pre-conditions	-
Flow of Events	Працівник знаходиться на сторінці з усіма завданнями, обирає завдання, інформацію про яке він бажає переглянути, та натискає кнопку “Review” у відповідному рядку. Працівник перенаправляється на сторінку задачі, що містить інформацію про текст задачі, тип задачі, статус, ким була створена задача, час створення, інформація про менеджера та інше.
Extension	-
Post-Condition	Перехід на сторінку завдання

Таблиця 2.20 - Варіант використання UC-19

Use case name	Обрати пристрій для надання користувачу
Use case ID	UC-19
Goals	Обрати пристрій для надання користувачу
Actors	Працівник
Trigger	Працівник бажає обрати пристрій для надання користувачу
Pre-conditions	Завдання має тип “Device request”. Завдання має статус “Not started”
Flow of Events	Працівник знаходиться на сторінці завдання. У полі “Choose device”, що є активним, працівник обирає пристрій для призначення. Працівник натискає на кнопку “Assign device” та перенаправляється на сторінку з усіма завданнями. Статус завдання змінюється на “In progress”.
Extension	-
Post-Condition	Пристрій надається заданому користувачу. Статус завдання змінюється на “In progress”. Перехід на головну сторінку.

Таблиця 2.21 - Варіант використання UC-20

Use case name	Зміна статусу виконання завдання
Use case ID	UC-20
Goals	Зміна статусу виконання завдання
Actors	Працівник
Trigger	Працівник бажає змінити статус виконання завдання
Pre-conditions	Поле для зміни статусу є активним

Продовження таблиці 2.21

Flow of Events	Працівник знаходиться на сторінці завдання та обирає потрібний статус у полі “Status”. Працівник натискає на кнопку “Submit”, статус завдання змінюється та працівник перенаправляється на головну сторінку
Extension	-
Post-Condition	Статус пристрою змінено. Перехід на головну сторінку.

Таблиця 2.22 - Варіант використання UC-21

Use case name	Зміна статусу пристрою при поверненні пристрою
Use case ID	UC-21
Goals	Зміна статусу пристрою при поверненні пристрою
Actors	Працівник
Trigger	Працівник бажає змінити статус поверненого пристрою
Pre-conditions	Тип завдання є “Device return”
Flow of Events	Працівник знаходиться на сторінці завдання. У полі “Device status” обирає статус пристрою. За необхідності, додає коментарі до повернення. Далі працівник натискає кнопку “Confirm return” та працівник перенаправляється на головну сторінку.
Extension	-
Post-Condition	Перехід на головну сторінку. Статус пристрою змінюється на заданий, статус задачі змінюється на завершений.

Таблиця 2.23 - Варіант використання UC-22

Use case name	Перегляд власних пристроїв
Use case ID	UC-22
Goals	Перегляд списку пристроїв, що були надані працівнику
Actors	Працівник
Trigger	Працівник бажає переглянути список пристроїв, що були йому надані
Pre-conditions	-
Flow of Events	Працівник натискає на вкладку “My devices” на навігаційній панелі. Працівник перенаправляється на сторінку з власними пристроями, де можна побачити інформацію про пристрої та виконати пошук за параметрами та ключовими словами.
Extension	-
Post-Condition	Перехід на сторінку із пристроями для працівника

Таблиця 2.24 - Варіант використання UC-23

Use case name	Замовити новий пристрій
Use case ID	UC-23
Goals	Замовити новий пристрій
Actors	Працівник
Trigger	Працівник бажає замовити новий пристрій
Pre-conditions	-

Продовження таблиці 2.24

Flow of Events	Працівник знаходиться на сторінці “My devices” та натискає на кнопку “Request a new device”. Працівник перенаправляється на сторінку “Request a new device”, де потрібно обрати тип пристрою та написати коментар до пристрою. Після натискання кнопки “Submit” працівник перенаправляється на сторінку зі своїми пристроями.
Extension	-
Post-Condition	Створюється запит на надання пристрою. Перехід на сторінку зі своїми пристроями.

Таблиця 2.25 - Варіант використання UC-24

Use case name	Підтвердити доставку пристрою
Use case ID	UC-24
Goals	Підтвердити доставку пристрою
Actors	Працівник
Trigger	Працівник бажає підтвердити доставку пристрою
Pre-conditions	Кнопка “Confirm delivery” є активною
Flow of Events	Працівник знаходиться на сторінці з усіма пристроями, обирає необхідний пристрій та натискає кнопку “Confirm delivery”. Сторінка оновлюється, а статус зв'язку стає “In progress”.
Extension	-
Post-Condition	Статус зв'язку між пристроєм та працівником стає “In progress”.

Таблиця 2.26 - Варіант використання UC-25

Use case name	Повернення пристрою
Use case ID	UC-25
Goals	Повернути пристрій
Actors	Працівник
Trigger	Працівник бажає повернути пристрій
Pre-conditions	Кнопка “Return device” є активною
Flow of Events	Працівник знаходиться на сторінці з усіма пристроями, обирає необхідний пристрій та натискає кнопку “Return device”. Сторінка оновлюється, а статус зв'язку стає “Finished”.
Extension	-
Post-Condition	Статус зв'язку між пристроєм та працівником стає “Finished”.

Таблиця 2.27 - Варіант використання UC-26

Use case name	Створення нового запиту
Use case ID	UC-26
Goals	Створити новий запит
Actors	Працівник
Trigger	Працівник бажає створити новий запит
Pre-conditions	-

Продовження таблиці 2.27

Flow of Events	Працівник обирає вкладку “Create a request” на навігаційній панелі та натискає на неї. Працівник перенаправляється на сторінку запиту, де він вводить інформацію про запит у поле та натискає “Submit”. Після цього він перенаправляється на сторінку з усіма запитами користувача.
Extension	У вікні з усіма запитами відображається повідомлення про успішне створення запиту
Post-Condition	Перехід на сторінку з усіма запитами. Створення нового запиту користувачем.

Таблиця 2.28 - Варіант використання UC-27

Use case name	Перегляд списку своїх запитів
Use case ID	UC-27
Goals	Перегляд списку своїх запитів
Actors	Працівник
Trigger	Працівник бажає переглянути список своїх запитів
Pre-conditions	-
Flow of Events	Працівник обирає вкладку “My requests” на навігаційній панелі та натискає на неї. Працівник перенаправляється на сторінку з усіма запитами.
Extension	-
Post-Condition	Перехід на сторінку з усіма запитами

Таблиця 2.29 - Варіант використання UC-28

Use case name	Видалення запиту
Use case ID	UC-28
Goals	Видалення обраного запиту
Actors	Працівник
Trigger	Працівник бажає видалити запит
Pre-conditions	Статус запиту є “Not started”
Flow of Events	Працівник знаходиться на сторінці з усіма заявками, обирає необхідну заявку та натискає активну кнопку “Delete”. Після цього заявка видаляється з системи, а сторінка оновлюється.
Extension	-
Post-Condition	Заявку видалено з системи

Таблиця 2.30 - Варіант використання UC-29

Use case name	Перегляд сторінки виконавця
Use case ID	UC-29
Goals	Перегляд сторінки виконавця
Actors	Працівник, Менеджер
Trigger	Користувач бажає переглянути інформацію про виконавця
Pre-conditions	-
Flow of Events	Користувач обирає посилання на сторінку користувача у таблиці та натискає на нього. Користувач перенаправляється на сторінку виконавця, що містить його контактну інформацію.

Продовження таблиці 2.30

Extension	-
Post-Condition	Перехід на сторінку виконавця

Таблиця 2.31 - Варіант використання UC-30

Use case name	Перегляд персональної інформації користувача
Use case ID	UC-30
Goals	Перегляд персональної інформації користувача
Actors	Працівник, Менеджер
Trigger	Користувач бажає переглянути персональну інформації
Pre-conditions	-
Flow of Events	Користувач обирає вкладку “My profile ” на навігаційній панелі та натискає на неї. Працівник перенаправляється на особисту сторінку, що містить його персональну та контактну інформацію.
Extension	-
Post-Condition	Перехід на персональну сторінку користувача

Таблиця 2.32 - Варіант використання UC-31

Use case name	Зміна персональної інформації
Use case ID	UC-31
Goals	Зміна персональної інформації
Actors	Працівник, Менеджер

Продовження таблиці 2.33

Trigger	Користувач бажає змінити персональну контактну інформацію
Pre-conditions	Користувач знаходиться на персональній сторінці
Flow of Events	Користувач вводить нову контактну інформацію у відповідні поля та натискає кнопку “Submit”. Нова контактна інформація зберігається, сторінка оновлюється.
Extension	-
Post-Condition	Персональна інформація була змінена.

Таблиця 2.33 - Варіант використання UC-32

Use case name	Перегляд поширених запитань
Use case ID	UC-32
Goals	Переглянути відповіді на поширені запитання
Actors	Працівник, Менеджер
Trigger	Користувач бажає переглянути відповіді на поширені запитання
Pre-conditions	-
Flow of Events	Користувач обирає вкладку “FAQ” на навігаційній панелі та натискає на неї. Працівник перенаправляється на сторінку з поширеними запитаннями. Після натискання на текст питання з’являється відповідь на питання.
Extension	-
Post-Condition	Перехід на сторінку з популярними питаннями

2.2 Аналіз системних вимог

Для успішної роботи даного застосунку необхідне виконання наступних вимог:

Мінімальна конфігурація технічних засобів:

- тип процесору: Intel Core i5;
- об'єм ОЗП: 4 Гб;
- підключення до мережі Інтернет зі швидкістю від 20 мегабіт;

Рекомендована конфігурація технічних засобів:

- тип процесору: Intel Core i7;
- об'єм ОЗП: 8 Гб;
- підключення до мережі Інтернет зі швидкістю від 100 мегабіт

2.3 Розроблення функціональних вимог

В таблиці 2.34 наведено загальну модель вимог, а в таблиці 2.35 наведений опис функціональних вимог до програмного забезпечення. Матрицю трасування вимог можна побачити на рисунку 2.2.

Таблиця 2.34 - Перелік загальних вимог

№	Назва	ID вимоги	Пріоритети	Ризики
1.1	Авторизація користувача	FR-1	Високий	Високий
1.2	Вихід з акаунту	FR-2	Високий	Високий
2.1	Перегляд списку всіх користувачів	FR-3	Середній	Середній
2.2	Перегляд інформації про користувача	FR-4	Низький	Низький
2.3	Створення нового користувача	FR-5	Високий	Високий
2.4	Редагування інформації про користувача	FR-6	Середній	Середній
2.5	Видалення користувача	FR-7	Низький	Середній

Продовження таблиці 2.34

3.1	Перегляд всіх заявок	FR-8	Середній	Низький
3.2	Створення нової заявки	FR-9	Високий	Високий
3.3	Автоматичне визначення типу заявки	FR-10	Середній	Середній
3.4	Автоматичне призначення виконавців заявки	FR-11	Середній	Середній
3.5	Видалення заявки	FR-12	Низький	Середній
4.1	Перегляд всіх завдань користувача	FR-13	Середній	Низький
4.2	Перегляд інформації про завдання	FR-14	Середній	Низький
4.3	Зміна статусу завдання	FR-15	Високий	Високий
4.4	Відмова від виконання завдання	FR-16	Низький	Середній
5.1	Перегляд списку всіх пристроїв	FR-17	Середній	Низький
5.2	Перегляд списку всіх пристроїв, що були надані користувачам	FR-18	Середній	Низький
5.3	Додавання нового пристрою	FR-19	Високий	Високий
5.4	Створення запиту для отримання пристрою	FR-20	Високий	Високий
5.5	Призначення пристрою	FR-21	Середній	Високий
5.6	Повернення пристрою	FR-22	Середній	Середній
5.7	Зміна стану пристрою	FR-23	Середній	Високий
6	Перегляд відповідей на поширені запитання	FR-24	Низький	Низький
7	Використання пошуку і фільтрів	FR-25	Низький	Середній

Таблиця 2.35 – Перелік функціональних вимог

ID Вимоги	Назва та опис
FR-1	<p>Авторизація користувача</p> <p>Система повинна надавати можливість авторизації користувачу шляхом введення логіну та паролю для входу в систему. Система повинна перенаправляти користувача на головну сторінку відповідно до його ролі</p>
FR-2	<p>Вихід з акаунту</p> <p>Система повинна надавати можливість користувачу входити з системи, шляхом натискання відповідної кнопки</p>
FR-3	<p>Перегляд списку всіх користувачів</p> <p>Система повинна надавати можливість перегляду списку всіх користувачів, що зареєстровані в системі</p>
FR-4	<p>Перегляд інформації про користувача</p> <p>Система повинна надавати можливість переглядами інформацію про користувача без можливості редагування</p>
FR-5	<p>Створення нового користувача</p> <p>Система повинна надавати можливість створення нового користувача, шляхом додавання інформації про нього до системи. До інформації про користувача відноситься ім'я та прізвище, контактна інформація, посада, відділ та роль користувача.</p>
FR-6	<p>Редагування інформації про користувача</p> <p>Система повинна надавати можливість редагувати інформацію про користувача, його контактну інформацію та обов'язки</p>
FR-7	<p>Видалення користувача</p> <p>Система повинна надавати можливість видаляти інформацію про користувача та його завдання</p>

Продовження таблиці 2.35

FR-8	<p>Перегляд всіх заявок</p> <p>Система повинна надавати можливість переглядати список всіх заявок, що були створені користувачами.</p>
FR-9	<p>Створення нової заявки</p> <p>Система повинна надавати можливість користувачу створювати нову заявку</p>
FR-10	<p>Автоматичне визначення типу завдання</p> <p>Система повинна надавати можливість автоматичного визначення та призначення типу завдання за ключовими словами</p>
FR-11	<p>Автоматичне призначення виконавців заявки</p> <p>Система повинна надавати можливість автоматичного призначення виконавців заявки, згідно з типом завдання</p>
FR-12	<p>Видалення заявки</p> <p>Система повинна надавати можливість видалення заявки</p>
FR-13	<p>Перегляд всіх завдань користувача</p> <p>Система повинна надавати можливість користувачу бачити список всіх завдань, де він був призначений виконавцем</p>
FR-14	<p>Перегляд інформації про завдання</p> <p>Система повинна надавати можливість виконавцю переглядати детальну інформацію про задачу</p>
FR-15	<p>Зміна статусу завдання</p> <p>Система повинна надавати виконавцю можливість змінювати статус виконання завдання.</p>
FR-16	<p>Відмова від виконання завдання</p> <p>Система повинна надавати виконавцю можливість відмовлятися від виконання завдання</p>

Продовження таблиці 2.35

FR-17	<p>Перегляд списку всіх пристроїв</p> <p>Система повинна надавати можливість перегляд список всіх пристроїв та їх статуси</p>
FR-18	<p>Перегляд списку пристроїв, що були надані користувачам</p> <p>Система повинна надавати можливість перегляд історії надання пристроїв користувачам та інформацію про ці зв'язки</p>
FR-19	<p>Додавання нового пристрою</p> <p>Система повинна надавати можливість додавання нового пристрою до системи</p>
FR-20	<p>Створення запиту для отримання пристрою</p> <p>Система повинна надавати можливість створювати заявку для отримання певного пристрою</p>
FR-21	<p>Призначення пристрою</p> <p>Система повинна надавати можливість виконавцям призначати доступні пристрої користувачам та підтверджувати їх доставку</p>
FR-22	<p>Повернення пристрою</p> <p>Система повинна надавати можливість користувачу повертати пристрій та підтверджувати їх повернення</p>
FR-23	<p>Зміна стану пристрою</p> <p>Система повинна надавати можливість змінювати статус пристрою, в залежності від статусу завдання</p>
FR-24	<p>Перегляд відповідей на поширені запитання</p> <p>Система повинна надавати можливість користувачам шукати та переглядами відповіді на поширені запитання</p>
FR-25	<p>Використання пошуку та фільтрів</p> <p>Система повинна надавати користувачам можливість використовувати фільтри та пошук інформації за ключовими словами у визначених таблицях.</p>

	FR-1	FR-2	FR-3	FR-4	FR-5	FR-6	FR-7	FR-8	FR-9	FR-10	FR-11	FR-12	FR-13	FR-14	FR-15	FR-16	FR-17	FR-18	FR-19	FR-20	FR-21	FR-22	FR-23	FR-24	FR-25
UC-1	"+"																								
UC-2		"+"																							
UC-3					"+"																				
UC-4							"+"																		
UC-5						"+"																			
UC-6		"+"																							"+"
UC-7																									"+"
UC-8												"+"													"+"
UC-9													"+"												"+"
UC-10																		"+"							"+"
UC-11																			"+"						"+"
UC-12																	"+"								"+"
UC-13																							"+"		"+"
UC-14		"+"																							"+"
UC-15						"+"																			"+"
UC-16												"+"													"+"
UC-17																"+"									"+"
UC-18													"+"												"+"
UC-19																				"+"					"+"
UC-20															"+"								"+"		"+"
UC-21																							"+"		"+"
UC-22																		"+"							"+"
UC-23																			"+"						"+"
UC-24																				"+"					"+"
UC-25																					"+"				"+"
UC-26								"+"	"+"	"+"	"+"														"+"
UC-27							"+"																		"+"
UC-28												"+"													"+"
UC-29			"+"																						"+"
UC-30			"+"																						"+"
UC-31					"+"																				"+"
UC-32																									"+"

Рисунок 2.2 – Матриця трасування вимог

2.4 Розроблення нефункціональних вимог

У процесі розробки було визначено наступні нефункціональні вимоги:

- вебзастосунок має мати англійську мову інтерфесу
- вебзастосунок повинен коректно відображатись у найпоширеніших сучасних браузерях
- легкість у підтримці, що буде забезпечена дотриманням єдиного стилю коду, використанням бібліотек для роботи
- легкість у використанні, що буде досягнута за допомогою використання графічного інтерфейсу

Висновки до розділу

В цьому розділі було розроблено програмні вимоги до вебзастосунку. Головною функцією вебзастосунку є обробка заявок, що надходять до служби підтримки ІТ відділу. Для відображення всього функціоналу було створено

діаграму використання застосунку. Також було наведено варіанти використання вебзастосунку за допомогою use cases. Було проаналізовано та наведено мінімальні та рекомендовані системні вимоги для успішної роботи вебзастосунку.

Для розроблення функціональних вимог було наведено загальну модель вимог та описано функціональні вимоги до програмного забезпечення. Матриця трасування була наведена для відображення зв'язку між функціональними вимогами та способами використання застосунку. Для розроблення нефункціональних вимог було проаналізовано та наведено засоби, які покращать якість використання застосунку для користувачів.

В цьому розділі було наведено результати аналізу функціональних та нефункціональних вимог до застосунку, що дозволять користувачу використовувати застосунок зі зручністю та ефективністю, забезпечуючи йому потрібні функції та забезпечуючи високу якість взаємодії.

За результатами розділу сформовано технічне завдання на розробку програмного забезпечення.

3 КОНСТРУЮВАННЯ ТА РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Архітектура програмного забезпечення

Архітектура вебзастосунку була створена з використанням паттерну MVC (Model-View-Controller). Цей паттерн розподіляє бізнес-логіку, інтерфейс та операційну частину, що зв'язує бізнес-логіку та інтерфейс[4]. Кожен компонент паттерну виконує чітко розподілені завдання.

- Model: Модель визначає структуру даних та бізнес-логіку застосунку. Модель відповідає за управління даними додатку, відповіді на запити від інших компонентів та обробку бізнес-правил. Наприклад, Модель відповідає за збереження заявок, що надходять, у базі даних.
- View: Цей компонент визначає логіку представлення (виду) вебзастосунку. Цей компонент визначає, як дані з Моделі відображаються користувачу. Наприклад, для отримання списку запитів користувача View отримує дані з Моделі та представляє її, у вигляді графічного інтерфейсу.
- Controller: Контролер діє як посередник між користувацьким інтерфейсом (View) та даними (Model). Він обробляє дані, введені користувачем, ініціює відповідні дії в Моделі і відповідно оновлює Представлення. Наприклад, коли користувач просить переглянути свої минулі запити, контролер отримує ці дані з моделі і дає вказівки представленню, як їх відобразити.

На рисунку 3.1 зображено архітектуру вебзастосунку за допомогою паттерну Model-View-Controller.

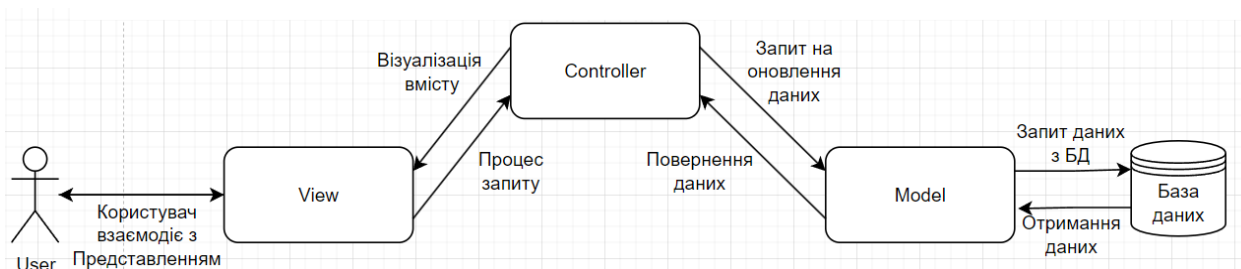


Рисунок 3.1 - Архітектура вебзастосунку

Для управління автентифікацією користувачів використовується бібліотека `werkzeug.security`. Ця бібліотека надає основні функції для хешування та перевірки паролів, забезпечуючи безпеку облікових даних користувача. Під час створення користувача пароль хешується за допомогою хешу SHA-256. Хеш паролю зберігається в базі даних. При спробі авторизації користувач вводить свій пароль, з отриманим паролем виконується хешування і виконується перевірка на співпадіння хешів паролів.

Для управління сесіями та збереження автентичності користувачів використовується бібліотека `flask_login` [5]. Після успішної автентифікації користувача його акаунт зберігається за допомогою функції `login_user` бібліотеки `flask_login`. Для виходу з акаунту використовується функція `logout_user`, що завершує сесію користувача.

Інтерфейс вебзастосунку був розроблений інтуїтивно зрозумілим для користувачів і доступним у найпоширеніших сучасних браузерах, забезпечуючи узгоджене відображення та функціональність. Дизайн сторінок був частково створений за допомогою використання Bootstrap.

Для полегшення підтримки та обслуговування вебзастосунку було використано єдиний стиль код для всіх елементів, що покращує читабельність і спрощує процеси налагодження.

3.2 Обґрунтування вибору засобів розробки

Для розробки вебзастосунку було використано мову програмування Python. Python містить декілька фреймворків для створення вебзастосунків. Найпопулярніші з них - це Django та Flask. Фреймворк Django був створений для простих і базових додатків, є повноцінним веб-фреймворком та URL диспетчер цього фреймворку базується на RESTful запитах. Фреймворк Flask був розроблений для швидкої розробки, є фреймворком Web Server Gateway Interface (WSGI), URL-диспетчер цього фреймворку базується на

контролері-regex. Для розробки вебзастосунку було обрано фреймворк Flask, оскільки він є більш відповідним до завдання проекту.

Найпопулярнішими IDE для роботи на Python є PyCharm від JetBrains та Visual Studio Code[6]. Visual Studio Code є безкоштовним, не використовує багато ресурсів пристрою при роботі, містить велику кількість плагінів для покращення роботи та містить інтеграцію з git для зручного збереження прогресу роботи. Для створення програмного забезпечення було використано Visual Studio Code.

В якості бази даних вебзастосунку було обрано базу даних MySQL. База даних MySQL була обрана через її надійність, масштабованість та сумісність із сучасними фреймворками для веб-розробки, такими як Flask. Для роботи з базою даних було обрано SQLAlchemy, завдяки його можливостям в якості API, що дозволяє безперешкодно взаємодіяти з базою даних MySQL. Абстрагування операцій SQL полегшило ефективне управління базою даних, підвищивши гнучкість і масштабованість вебзастосунку.

3.3 Конструювання програмного забезпечення

Запити до служби підтримки майже завжди є однотипними та їх можна розподілити в певні категорії. Наприклад, запити для отримання пристроїв, запити для встановлення програмного забезпечення, запити для ремонту пристроїв. Особиста обробка кожного запиту менеджером, призначення типу завдання та виконавців може займати велику кількість часу, тому запропонований застосунок буде використовувати методи роботи з текстом для визначення типу завдання. Менеджер матиме можливість змінити тип завдання у разі виникнення помилки, проте більшість завдань будуть задані автоматично без необхідності менеджера.

Для визначення типу заявки було застосовано функцію аналізу тексту запиту, яка використовувала списки ключових слів для різних типів завдань. Текст запиту було перетворено в нижній регістр, щоб забезпечити

відповідність, незважаючи на реєстр. Перебираючи кожен список ключових слів, функція визначила відповідні ключові слова в тексті запиту, що дозволило класифікувати завдання за окремими категоріями. Такий підхід уможливив ефективну класифікацію завдань на основі наявності конкретних ключових слів, що вказують на проблеми з управлінням програмним забезпеченням, обліковими записами або ремонтом пристроїв. Якщо ключових слів не було знайдено, функція повертала класифікацію за замовчуванням "Інше". Менеджер матиме можливість знайти завдання з цим типом за допомогою фільтрів і вручну задати тип заявки та виконавця.

Для покращення роботи користувачів було створено функції для пошуку та фільтрації даних у таблицях. Дозволяючи користувачам здійснювати пошук, додаток забезпечує швидкий доступ до потрібної інформації, скорочуючи час, витрачений на навігацію у великих наборах даних. Крім того, функції фільтрів дозволяють користувачам уточнювати результати пошуку на основі певних критеріїв, що дає змогу переглядати статуси завдань з урахуванням індивідуальних особливостей.

У таблицях 3.1 - 3.4 наведено усі доступні до використання методи/запити з коротким описом, вхідними та вихідними даними.

Таблиця 3.1 - Методи користувачів з роллю "Admin"

Запит	Опис	Вхідні дані	Вихідні дані
GET /admin_dashboard	Головна сторінка адміністратора	-	список всіх користувачів, відділів та ролей
GET /create_user	Сторінка створення нового користувача	-	список всіх ролей та відділів

Продовження таблиці 3.1

POST /create_user	Сторінка створення нового користувача	firstName, lastName, email, phone, title, department_id, role_id, username, password_hash	новий об'єкт класу Worker
POST /delete_user/<user_id>	Видалення користувача за Id	user_id	-
GET /edit_user/<user_id>	Отримання інформації про користувача для редагування	-	користувач за user_id, список всіх ролей та відділів
POST /edit_user/<user_id>	Редагування інформації про користувача	firstName, lastName, email, phone, title, department_id, role_id, username, password_hash	-

Таблиця 3.2 - Методи користувачів з роллю "Manager"

Запит	Опис	Вхідні дані	Вихідні дані
GET /dashboard	Головна сторінка менеджера	-	current_user, список всіх завдань

Продовження таблиці 3.2

GET /edit_request/<task_id>	Отримання інформації про заявку	task_id	заявка, список потенційних виконавців, список типів завдань
POST /edit_request/<task_id>	Редагування інформації про заявку	task_id, worker_id, taskType_id	-
GET /profile	Особиста сторінка користувача	current_user	-
POST /profile	Редагування особистої інформації	email, phone_number	-
GET /user_profile/<user_id>	Перегляд особистої сторінки користувача	user_id	працівник, список всіх типів завдань, що може виконувати працівник
GET /manage_workers	Перегляд списку всіх користувачів	-	список всіх працівників відділу та їх зобов'язань
GET /questions	Перегляд відповідей	-	-

Продовження таблиці 3.2

GET /edit_profile/<user_id>	Перегляд сторінки користувача	user_id	працівник та завдання, які він може виконувати
POST /edit_profile/<user_id>	Редагування сторінки робітника	user_id, tasktype_id	user_id
GET /manage_devices	Сторінка для роботи з усіма пристроями	-	список всіх зв'язок між пристроями та працівниками
GET /add_new_device	Форма для додавання нового пристрою	-	список всіх типів та брендів пристроїв
POST /add_new_device	Додавання нового пристрою	device_name, device_type, device_brand, serial_number	новий об'єкт класу Device
GET /view_devices_list	Перегляд списку всіх пристроїв	-	список всіх пристроїв, типів пристроїв та брендів
POST /make_not_available/<device_id>	Зробити пристрій недоступним	device_id	-

Таблиця 3.3 - Методи користувачів з роллю “Worker”

Запит	Опис	Вхідні дані	Вихідні дані
GET /user/dashboard	Головна сторінка користувача	current_user	список всіх завдань користувача
GET /new_request	Форма для створення нового запиту	current_user	-
POST /new_request	Створення нового запиту	current_user, request_text	новий об'єкт класу Task
GET /user_requests	Перегляд всіх запитів працівника	current_user	список всіх запитів користувача
POST /delete_request/ <task_id>	Видалення заявки	current_user, task_id	-
GET /profile	Особиста сторінка користувача	current_user	-
POST /profile	Редагування особистої інформації	email, phone_number	-
GET /user_profile/<user_id>	Перегляд особистої сторінки	user_id	-

Продовження таблиці 3.3

GET /review_request/<task_id>	Перегляд сторінки заявки	task_id	завдання, список статусів завдання
GET /reassign_request/<task_id>	Відмова від завдання та призначення нового виконавця	task_id	task
POST /reassign_request/<task_id>	Відмова від завдання та призначення нового виконавця	task_id	оновлений виконавець завдання
GET /user_devices	Перегляд всіх пристроїв, що належать користувачу	current_user	список пристроїв, що належать користувачу
GET /request_new_device	Форма для запиту нового пристрою	current_user	список типів пристроїв,
POST /request_new_device	Створення запиту на отримання нового пристрою	current_user, device_type, additional_notes	новий об'єкт класу Task

Продовження таблиці 3.3

POST /confirm_device/ <device_id>	Підтвердити отримання пристрою	current_user, device_id	-
GET /return_device/ <device_id>	Повернення пристрою	current_user, device_id	-
POST /return_device/ <device_id>	Повернення пристрою	current_user, device_id	новий об'єкт класу Task
GET /review_device_req uest/<task_id>	Перегляд запиту на отримання пристрою	task_id	список доступних пристроїв
POST /review_device_req uest/<task_id>	Призначення пристрою для передачі	task_id	новий об'єкт класу DeviceWorker
GET /return_device_requ est/<task_id>	Перегляд запиту про повернення пристрою	task_id	-
POST /return_device_requ est/<task_id>	Підтвердження повернення пристрою	task_id	-
GET /questions	Перегляд відповідей	-	-

Таблиця 3.4 - Методи неавторизованих користувачів

Запит	Опис	Вхідні дані	Вихідні дані
POST /	Сторінка авторизації	username, password	current_user

В якості системи управління базами даних використовується MySQL. База даних серверу призначена для зберігання користувачів, а також даних про їх завдання, пристрої та типи завдань, які працівники можуть виконувати. Модель бази даних наведена на рисунку 3.2. Опис таблиць бази даних наведено у таблицях 3.5 - 3.17.

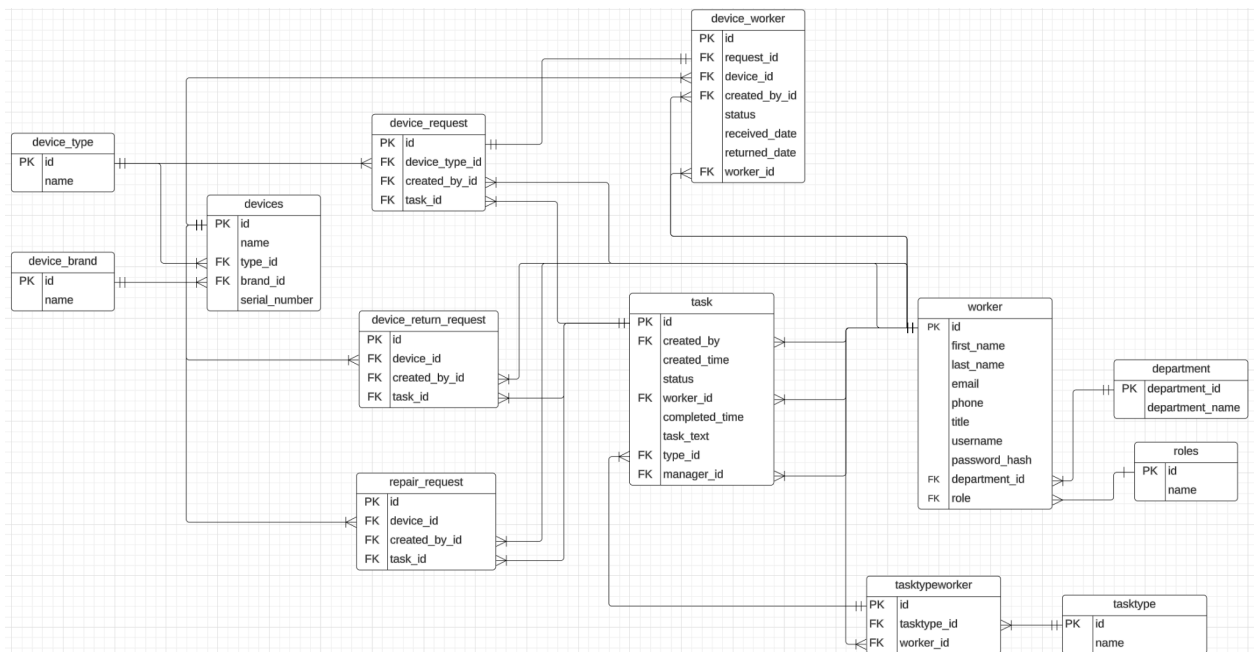


Рисунок 3.2 – ER діаграма сутностей User, Messages, Contacts

Таблиця 3.5 – Опис таблиці worker

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер користувача
first_name	varchar	ім'я користувача

Продовження таблиці 3.5

last_name	varchar	прізвище користувача
email	varchar	адреса електронної пошти
phone	varchar	номер телефону користувача
title	varchar	назва посади користувача
username	varchar	логін користувача
password_hash	varchar	хеш паролю користувача
department_id	int	посилання на запис у таблиці department, де зберігається інформація про відділ
role	int	посилання на запис у таблиці roles, де зберігається інформація про роль

Таблиця 3.6 – Опис таблиці department

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер департаменту
name	varchar	назва департаменту

Таблиця 3.7 – Опис таблиці roles

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер ролі
name	varchar	назва ролі

Таблиця 3.8 – Опис таблиці tasktype

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер типу завдань
name	varchar	назва типу завдань

Таблиця 3.9 – Опис таблиці task

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер завдання
created_by	int	id користувача, який створив завдання (посилання на таблицю worker)
created_time	datetime	час створення завдання
status	enum	статус виконання завдання
worker_id	int	id користувача, який виконує завдання (посилання на таблицю worker)
completed_time	datetime	час завершення виконання завдання
task_text	varchar	текст завдання
type_id	int	id типу завдання (посилання на таблицю tasktype)
manager_id	int	id користувача, який є менеджером завдання (посилання на таблицю worker)

Таблиця 3.10 – Опис таблиці tasktypeworker

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер зв'язку між типом завдання та працівником
tasktype_id	int	id типу завдання, яке може виконувати робітник (посилання на таблицю tasktype)
worker_id	int	id робітника, який може виконувати завдання (посилання на таблицю worker)

Таблиця 3.11 – Опис таблиці devices

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер пристрою
name	varchar	назва пристрою
type_id	int	id типу пристрою (посилання на таблицю device_type)
brand_id	int	id бренду (посилання на таблицю device_brand)
serial_number	varchar	серійний номер пристрою

Таблиця 3.12 – Опис таблиці device_request

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер запиту на отримання пристрою

device_type_id	int	id типу пристрою, який потрібно отримати (посилання на таблицю device_type)
created_by_id	int	id робітника, який створив запит (посилання на таблицю worker)
task_id	int	id завдання, яке відповідає цьому запиту (посилання на таблицю task)

Таблиця 3.13 – Опис таблиці device_type

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер типу пристрою
name	varchar	назва типу пристрою

Таблиця 3.14 – Опис таблиці device_brand

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер бренду пристрою
name	varchar	назва бренду

Таблиця 3.14 – Опис таблиці device_request

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер запиту на отримання пристрою
device_type_id	int	id типу пристрою, який потрібно отримати (посилання на таблицю device_type)

Продовження таблиці 3.14

created_by_id	int	id робітника, який створив запит (посилання на таблицю worker)
task_id	int	id завдання, яке відповідає цьому запиту (посилання на таблицю task)

Таблиця 3.15 – Опис таблиці device_worker

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер зв'язку між пристроєм та працівником
request_id	int	id запиту на отримання пристрою (посилання на таблицю device_request)
device_id	int	id пристрою, який було надано (посилання на таблицю device)
created_by_id	int	id працівника, якому надано пристрій
status	enum	статус зв'язку
received_date	datetime	час отримання пристрою
returned_date	datetime	час повернення пристрою
worker_id	int	id працівника, який надав пристрій

Таблиця 3.16 – Опис таблиці device_return_request

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер заявки на повернення пристрою

Продовження таблиці 3.16

device_id	int	id пристрою, який потрібно повернути
created_by_id	int	id робітника, який створив заявку (посилання на таблицю worker)
task_id	int	id завдання, яке відповідає цій заявці (посилання на таблицю task)

Таблиця 3.17 – Опис таблиці repair_request

Назва поля	Тип даних	Опис
id	int autoincrement	ідентифікаційний номер заявки на ремонт пристрою
device_id	int	id пристрою, що потрібно відремонтувати
created_by_id	int	id робітника, який створив заявку (посилання на таблицю worker)
task_id	int	id завдання, яке відповідає цій заявці (посилання на таблицю task)

Опис утиліт, бібліотек та іншого стороннього програмного забезпечення, що використовується у розробці наведено в таблиці 3.18.

Таблиця 3.18 – Опис утиліт

№ п/п	Назва утиліти	Опис застосування
1	Visual Studio Code	Головне середовище розробки програмного забезпечення серверної частини дипломної роботи.

Продовження таблиці 3.18

2	Flask	Фреймворк мови Python, який був використаний для розробки вебзастосунку
3	MySQL Workbench	Програмне забезпечення яке надає легкий графічний інтерфейс для доступу до бази даних.
4	Jinja	Бібліотека мови Python у складі Flask, що використовувалась для створення шаблонів веб-сторінок
5	SQLAlchemy	Бібліотека мови Python, яка використовується для взаємодії з базами даних через їх API
6	werkzeug.security	Бібліотека мови Python, що використовується для безпечного хешування та перевірки паролів у веб-додатках [7]

Тексти програмного коду наведені в окремому документі «Текст програми».

3.4 Аналіз безпеки даних

Неавторизовані користувачі не мають доступу до загальних сторінок. Наявність авторизації в користувача перевіряється за допомогою декоратору `login_required`. Декоратор `login_required` фреймворку Flask гарантує, що певні маршрути або сторінки доступні лише авторизованим користувачам. Неавторизованим користувачам потрібно авторизуватись для доступу до сторінок.

В вебзастосунку користувачі мають 3 ролі: адміністратор, менеджер та користувач. Для обмеження доступу до сторінок користувачів, що не мають дозволу доступу, згідно зі своєю роллю, було створено декоратор `role_required`, який перевіряє чи співпадає роль користувача, що намагається перейти на сторінку, та дозволених ролі методу. Користувачі з роллю, не зазначеній в дозволених, перенаправляються на сторінку з відповідною помилкою.

Паролі користувачів не зберігаються у базі даних, проте зберігаються хеші SHA-256, які створюються за допомогою функції `generate_password_hash()` бібліотеки `werkzeug.security`. При спробі авторизації користувача отримується хеш введеного пароля, та цей хеш порівнюється зі збереженим у базі даних.

Висновки до розділу

У цьому розділі було розглянуто аспекти конструювання та розробки вебзастосунку. В якості архітектури проекту було використано паттерн `Model-View-Controller`, що дозволяє кожному компоненту паттерну виконувати чітко розподілені завдання. Цей паттерн сприяє організації коду, масштабованості та простоті обслуговування.

Було обгрунтовано вибір мови програмування та фреймворків для створення вебзастосунку. Для покращення роботи служби підтримки було використано функцію для аналізу тексту заявок, що роботу якої було описано в цьому розділі. В цьому розділі було наведено структуру бази даних, описано дані, що зберігаються, та методи роботи з базою даних. Також було описано доступні методи та запити, що доступні до використання.

В цьому розділі було розглянуто та порівняно методи та засоби конструювання вебзастосунку, розглянуто структуру бази даних до проекту, описано доступні методи для використання. Було проаналізовано слабкі місця в безпеці даних та розглянуто методи для забезпечення високого рівню безпеки застосунку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) 10 Essential Self-Service Statistics & Trends [2024 Data]. *Tidio*. URL: <https://www.tidio.com/blog/self-service-statistics/>.
- 2) The Value of Customer Self-Service in the Digital Age. *SuperOffice CRM*. URL: <https://tinyurl.com/2u94d8w5>.
- 3) Customer Service Statistics: Trends, Challenges, Business Impact. *Wavetec*. URL: <https://www.wavetec.com/blog/customer-service-statistics/>.
- 4) MVC: Model, View, Controller – Flask-Diamond 0.5.1 documentation. *Flask-Diamond – Flask-Diamond 0.5.1 documentation*. URL: <https://flask-diamond.readthedocs.io/en/latest/model-view-controller/>.
- 5) Flask-Login 0.7.0 documentation. *Flask-Login 0.7.0 documentation*. URL: <https://flask-login.readthedocs.io/en/latest/#api-documentation>.
- 6) Microsoft. Documentation for Visual Studio Code. *Visual Studio Code - Code Editing. Redefined*. URL: <https://code.visualstudio.com/docs>.
- 7) Utilities – Werkzeug Documentation (2.3.x). *Werkzeug – Werkzeug Documentation (3.0.x)*. URL: <https://werkzeug.palletsprojects.com/en/2.3.x/utils/>.

ДОДАТКИ