**Пояснювальна записка  
до курсової роботи**

на тему: Багатокористувацька веб-книга рецептів

КПІ.ІП-1324.045440.02.81

Київ – 2024

Зміст

[Перелік умовних позначень 4](#_Toc115992741)

[Вступ 5](#_Toc115992742)

[1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 6](#_Toc115992743)

[1.1 Загальні положення 6](#_Toc115992744)

[1.2 Змістовний опис і аналіз предметної області 6](#_Toc115992745)

[1.3 Аналіз існуючих технологій та успішних IT-проєктів 6](#_Toc115992746)

[1.3.1 Аналіз відомих алгоритмічних та технічних рішень 6](#_Toc115992747)

[1.3.2 Аналіз допоміжних програмних засобів та засобів розробки 7](#_Toc115992748)

[1.3.3 Аналіз відомих програмних продуктів 7](#_Toc115992749)

[1.4 Аналіз вимог до програмного забезпечення 7](#_Toc115992750)

[1.4.1 Розроблення функціональних вимог 8](#_Toc115992751)

[1.4.2 Розроблення нефункціональних вимог 9](#_Toc115992752)

[1.5 Постановка задачі 9](#_Toc115992753)

[Висновки до розділу 10](#_Toc115992754)

[2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 11](#_Toc115992755)

[2.1 Моделювання та аналіз програмного забезпечення 11](#_Toc115992756)

[2.2 Архітектура програмного забезпечення 11](#_Toc115992757)

[2.3 Конструювання програмного забезпечення 11](#_Toc115992758)

[2.4 Аналіз безпеки даних 13](#_Toc115992759)

[Висновки до розділу 13](#_Toc115992760)

[3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 14](#_Toc115992761)

[3.1 Аналіз якості ПЗ 14](#_Toc115992762)

[3.2 Опис процесів тестування 14](#_Toc115992763)

[3.3 Опис контрольного прикладу 15](#_Toc115992764)

[Висновки до розділу 15](#_Toc115992765)

[4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 16](#_Toc115992766)

[4.1 Розгортання програмного забезпечення 16](#_Toc115992767)

[4.2 Підтримка програмного забезпечення 16](#_Toc115992768)

[Висновки до розділу 17](#_Toc115992769)

[ВИСНОВКИ 18](#_Toc115992770)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 19](#_Toc115992771)

Перелік умовних позначень

|  |  |  |
| --- | --- | --- |
| IDE | – | Integrated Development Environment – інтегроване середовище розробки. |
| API | – | Application programming interface, прикладний програмний Інтерфейс |
| SDK | – | Software development kit |
| IT | – | Інформаційні технології |
| ER | – | Entity-Relation diagram |
| OC | – | Операційна система. |
| БД | – | База даних. |

Вступ

Розробка веб-додатку «Багатокористувацька веб-книга рецептів» народжена з метою створення центрального хабу для кулінарних ентузіастів, спрямованого на обмін рецептами та кулінарними відкриттями. Цей проект відповідає на зростаючий попит у сучасному світі на інтерактивні та відкриті платформи, де кожен може співпрацювати, ділитися та вдосконалювати кулінарними знаннями та рецептами.

У світі, де технології стають неодмінною частиною нашого щоденного життя, «Багатокористувацька веб-книга рецептів» відповідає потребам спільноти, яка шукає не просто платформу для зберігання рецептів, але й простір для активного обміну ідей та взаємодії.

Цей проект забезпечує зручність та доступність, сприяючи розвитку кулінарної культури та надаючи змогу кожному учаснику відчути себе частиною активної та творчої спільноти кухарів-аматорів та гурманів. «Taste Bud Tales» прагне стати не лише платформою, але й віртуальним майданчиком для кулінарних творців, де кожен може долучитися, співпрацювати та навчатися один від одного.

# АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Загальні положення

Кулінарія у веб-середовищі є ключовою галуззю, яка активно розвивається та прогресує у світі сучасних технологій. Ця область охоплює процеси створення, редагування, зберігання та обміну кулінарною інформацією через Інтернет.

Кулінарна сфера на веб-платформах охоплює широкий спектр аспектів, включаючи рецепти, методи приготування страв, історію кулінарії, харчові тенденції та інновації в кулінарній сфері. Розвиток технологій та доступність Інтернету сприяють розширенню доступу до кулінарної інформації, що робить цей ресурс ширше доступним та більш інтерактивним для аудиторії.

Загальне визначення кулінарної галузі у веб-середовищі включає в себе веб-портали з рецептами, кулінарні блоги, спеціалізовані форуми, мобільні додатки для кулінарії, а також онлайн-курси та спільноти для обміну досвідом та ідеями у галузі кулінарії.

Згідно з дослідженнями[1], платформи кулінарної культури у веб-середовищі стають все більш популярними серед широкого кола користувачів, від початківців до професіоналів, які шукають різноманітні ресурси та інформацію з готування.

Зокрема, розвиток інтерактивних платформ, де користувачі можуть спілкуватися, обмінюватися рецептами та досвідом, є важливим напрямком розвитку веб-середовища кулінарії, що робить кулінарну інформацію більш доступною та цікавою для широкого кола аудиторії.

## Змістовний опис і аналіз предметної області

На сьогоднішній день, інформаційні технології значно удосконалили процеси у сфері кулінарії, спрощуючи доступ до рецептів, кулінарних порад та різноманітних кухонь з усього світу. Програмне забезпечення, пов'язане з кулінарією, зазвичай забезпечує різні можливості для користувачів, такі як пошук рецептів, створення списків покупок, відстеження харчових властивостей, а також навчання і надання порад стосовно технік готування.

Однак, деякі недоліки існують. Наприклад, фрагментація існуючих платформ – наявність великої кількості веб-сайтів, додатків та платформ, що різняться за своїм функціоналом та інтерфейсами, ускладнює вибір користувача та робить процес пошуку рецептів менш зручним.

Також, обмежена персоналізація може стати проблемою. Багато з існуючих платформ намагаються забезпечити персоналізовані функції для користувачів, але існують обмеження у виборі персоналізації належним чином, які не завжди задовольняють потреби кожного користувача.

Для покращення ситуації у сфері ІТ в галузі кулінарії можна розглянути кілька можливих шляхів. Зокрема, створення єдиних стандартів для цифрових кулінарних платформ може сприяти уніфікації користувацького досвіду та полегшити вибір платформи для користувачів. Розширення функціоналу персоналізації та використання штучного інтелекту для забезпечення персоналізованих рекомендацій також можуть вирішити проблеми обмеженої персоналізації.

У рамках курсової роботи обрано шлях вдосконалення функціоналу персоналізації та уніфікації платформи для полегшення процесу вибору та надання користувачам більш зручного інтерфейсу.

## Аналіз існуючих технологій та успішних IT-проєктів

Проаналізуємо відоме на сьогодні алгоритмічне забезпечення у даній області та технічні рішення, що допоможуть у реалізації “Багатокористувацької веб-книги рецептів”. Далі будуть розглянуті допоміжні програмні засоби, засоби розробки та готові програмні рішення.

### Аналіз відомих алгоритмічних та технічних рішень

Під час створення веб-платформи "Багатокористувацька веб-книга рецептів" були використані різноманітні алгоритмічні та технічні рішення для ефективного вирішення завдань, пов'язаних з організацією, пошуком та оптимізованим зберіганням даних за допомогою Firebase.

Алгоритм пошуку і платформа зберігання текстових даних:

Firebase надає різні алгоритми пошуку для роботи з даними. Наприклад, використання Realtime Database забезпечує швидкий доступ до даних у форматі JSON, дозволяючи взаємодіяти в реальному часі. Cloud Firestore використовує індексацію для швидкого пошуку документів у базі даних за різними полями. За допомогою параметрів запиту можна фільтрувати дані за конкретними значеннями та сортувати їх. Також, Firebase може інтегруватися з Algolia, що розширює можливості пошуку та фільтрації даних, використовуючи потужні алгоритми Algolia для поліпшення пошуку в додатках. Для курсової роботи було вибрано зберігання текстових даних у Cloud Firestore і подальший пошук за допомогою його вмонтованих алгоритмів через поєднання швидкості і можливості пошуку за комбінацією параметрів. Дані також були продубльовані у Algolia Index для забезбечення “Full-text search” рецептів у додатку.

Платформа зберігання файлових даних:

Існує кілька платформ для зберігання файлових даних, які можуть використовуватися для мого веб-додатку:

Firebase Cloud Storage:

* Переваги: Легка інтеграція з іншими сервісами Firebase, масштабована інфраструктура, зручний інтерфейс для роботи з файлами.
* Недоліки: Вартість може зростати зі збільшенням обсягу даних, обмеження за кількістю операцій на запис та витрати на передачу даних.

Amazon S3 (Simple Storage Service):

* Переваги: Висока масштабованість, велика доступність даних, різноманітні параметри доступу та управління даними.
* Недоліки: Складніше налаштування порівняно з іншими платформами, може бути дорожче використання великих обсягів даних.

Google Cloud Storage:

* Переваги: Гнучка система доступу, масштабована інфраструктура, хороша швидкість передачі даних.
* Недоліки: Вартість може зростати зі збільшенням використання ресурсів, менша наочність у порівнянні з іншими платформами.

Для курсової роботи був обраний Firebase Cloud Storage, через легку інтеграцію з іншими наявними у ній сервісами Firebase і легкість у використанні і масштабуванні.

Технологія для забезпечення реєстрації:

У якості технології для забезпечення авторизації було використано Firebase Authentication. Це сервіс аутентифікації та авторизації, надається платформою Firebase. Він дозволяє легко додавати систему входу та управління користувачами. Firebase Authentication підтримує різні методи аутентифікації, такі як електронна пошта, номер телефону, соціальні мережі (Google, Facebook, Twitter), а також інші. Інші технології не були розглянуті, тому що у курсовій вже були використані інші сервіси Firebase і інтеграція з Auth0 SDK, Passport.js була б недоцільно складною.

### Аналіз допоміжних програмних засобів та засобів розробки

У якості мови програмування було розглянуто 2 основні мови для веб-розробки: JavaScript і TypeScript:

* JavaScript (JS) - динамічна мова з і проста у використанні, але може призводити до помилок через свою динамічність.
* TypeScript (TS) - статична мова, що розширює JS, додаючи типи, що дозволяє виявляти помилки під час розробки, але вимагає більше часу для налаштуванн. TS дозволяє писати більш безпечний код і поліпшує супровідність проєкту завдяки розвинутій системі типів.

У курсовій роботі було використано JavaScript через простоту у використанні і інтеграції з іншими сервісами.

Для frontend розробки було розглянуто 3 основних фреймворка: Angular, React і Vue.

React:

Переваги:

* Велика спільнота та підтримка: React має велику та активну спільноту, що означає багато ресурсів для навчання та підтримки а також сторонніх бібліотек на його основі.
* Модульність: Його компонентний підхід дозволяє легко розподіляти і керувати кодом.

Недоліки:

* Складність архітектури: Модульний підхід може спричинити складну архітектуру у великому застосунку, за збереженням якої потрібно слідкувати.
* Відсутність вбудованих інструментів для стейт-менеджменту: Для складних застосунків може знадобитися використання зовнішніх бібліотек.

Vue:

Переваги:

* Легкість вивчення: Vue є легшим для вивчення, через детальну документацію і найбільшу збіжність з розробкою веб-застосунків без використання бібліотек або фреймворків.
* Простота використання: Його синтаксис і підхід до роботи забезпечують простоту розробки.

Недоліки:

* Менша спільнота: За порівнянням з React і Angular, у Vue менша спільнота, що може обмежувати доступ до ресурсів для навчання та підтримки.
* Менша кількість додатків великого масштабу: Vue використовується для невеликих проектів, інакше потрібно виконати додаткову роботу.

Angular:

Переваги:

* Широкі можливості: Angular має все необхідне для побудови великих, масштабованих додатків.
* Строга структура коду: Його строга структура дозволяє уникнути деяких помилок, які можуть виникнути у великих проектах.

Недоліки:

* Складність і вагомий розмір: Angular має велику кількість вбудованих функцій, а це може призвести до збільшення розміру додатків та ускладнити їх розробку.
* Неактуальність: За статистикою[2] Angular втрачає свою актуальність, що робить його вивчення недоцільним.

У курсовій роботі було використано React, через відносну простоту у використанні, можливість додавання необхідних розширень і складність самого проекту, який вимагає чіткої архітектури.

Для збірки проекту було використано Vite. Інтеграція Vite та React може бути досить простою та ефективною для розробки веб-додатків. Зазвичай, для початку роботи з React у Vite використовують шаблони, які містять попередньо налаштовані конфігурації. Така інтеграція дозволяє поєднати переваги React як потужного фреймворку для розробки користувацьких інтерфейсів з високою продуктивністю та швидкістю побудови проектів у Vite.

Для backend розробки було використано Firebase. Це платформа від Google, яка пропонує цілий набір інструментів для розробки веб-додатків, включаючи робочі та швидкі сервіси для зберігання даних, автентифікації користувачів, хостингу, аналітики та багато іншого. Переваги Firebase над іншими варіантами розробки бекенду включають:

* Розширені можливості: Платформа має різні модулі, такі як база даних (Firestore), аутентифікація, зберігання файлів (Storage), хостинг та інші, що забезпечує розширені можливості розробки додатків.
* Масштабованість: Firebase легко масштабується разом із зростанням вашого додатку, надаючи можливість швидкого розширення та підтримки великої кількості користувачів.
* Інтегрованість і готові рішення: Платформа пропонує готові рішення, що дозволяють швидко налаштувати робочий функціонал без великих зусиль.
* Реальний час: Firebase забезпечує можливість роботи в реальному часі, що корисно для розробки реактивних додатків або коли необхідно бачити зміни відразу після внесення користувачем.

У якості середовища для розробки було використано Visual Studio Code (VS Code). Він є неабияк потужним та легким у використанні текстовим редактором. Відмінність полягає в його розширеній функціональності завдяки величезній кількості доступних розширень, що спрощують та розширюють його можливості для розробки. Він користується великою популярністю завдяки активній спільноті, яка постійно внесенням нововведень та вдосконаленням. Також слід зазначити його високий рівень налаштувань, які дозволяють користувачам адаптувати редактор до своїх власних потреб, що робить його надзвичайно гнучким у використанні.

### Аналіз відомих програмних продуктів

1. Taste Bud Tales
   * Опис: це платформа, спрямована на створення спільноти кулінарних ентузіастів, яка має на меті обмін рецептами, кулінарними ідеями та знаннями між користувачами.
   * Особливості:
     1. Можливість додавання власних рецептів для кожного користувача.
     2. Можливість збереження рецептів інших користувачів.
     3. Стандартизація створення рецептів: усі рецепти створюються за допомогою зручного конструктора, що забезпечує зручний перегляд рецептів для користувачів.
2. KLOPOTENKO
   * Опис: це платформа з рецептами від відомого українського кухаря Клопотенка. Вона має на меті популяризацію і розширення української кухні.
   * Особливості:
     1. Розміщення рецептів виключно української кухні.
     2. Вбудований інтернет-магазин: сайт надає можливість скористатися онлайн-магазином з книгами Клопотенка та інгрєдієнтами для страв.

Для порівняння курсової роботи з аналогом можна скористатись таблицею 1.3.1.

Таблиця 1.3.1 – Порівняння з аналогом

|  |  |  |  |
| --- | --- | --- | --- |
| Функціонал | Taste Bud Tales | KLOPOTENKO | Пояснення |
| Можливість додавання власних рецептів | Так | Ні | Мій сайт надає можливість створювати рецепти кожному користувачу, який пройшов авторизацію |
| Можливість збереження рецептів | Так | Ні | Сайт KLOPOTENKO має проблеми зі збереженням рецептів через проблеми архітектури у коді. |
| Можливість коментарів | Так | Так | Обидва сайти мають можливість коментарів, однак KLOPOTENKO використовує інший, менш зручний підхід |
| Можливість перегляду власних рецептів | Так | Так | Сайт KLOPOTENKO використовує колекції рецептів, Taste Bud Tales показує усі рецепти користувача з пагінацією |
| Можливість перегляду рецептів за категоріями |  |  | Сайт KLOPOTENKO  використовує інші категорії, які більше підходять для кулінарної книги, яка включає лише українську кухню, натомість TasteBudTales сортує рецепти за типом страви |
| Можливість налаштовування профіля і перегляду профілів інших користувачів | Так | Ні | Профілі на сайті KLOPOTENKO існують лише для збереження інформації і перегляду збережених рецептів, натомість Taste Bud Tales дозволяє переглядати профілі інших користувачів і їх власні рецепти |

## Аналіз вимог до програмного забезпечення

Головною функцією програмного забезпечення є додавання рецептів.

Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

Користувач повинен мати можливість зареєструватися використовуючи внутрішню систему реєстрації. В результаті натиску кнопки “Create an account” буде створено нового користувача, якщо всі дані валідні. Після процедури реєстрації користувача переадресовує на “Profile” сторінку.

Користувач повинен мати можливість авторизуватись використовуючи внутрішню систему логінізації. Кнопка Log in відповідає за підтвердження введених даних і у випадку їх правильного вводу, користувач успішно авторизується та переадресовується на сторінку “ Profile”.

Користувач повинен мати можливість переглядати нові і популярні рецепти на сторінці “Home”.

Користувач повинен мати можливість переглядати рецепти за категоріями (супи, випічка, і. т. д.) на відповідних вкладках “Recipes by category”.

Користувач повинен мати можливість відкрити рецепт зі сторінок “My recipes”, “Saved”, “Recipes by category” і “Home” натисканням на картку рецепту і перейти на детальну інформацію, сторінку “Recipe Page”.

Користувач повинен мати можливість поділитися рецептом у соц. мережах за допомогою відповідних кнопок на сторінці “ Recipe Page ”.

Користувач повинен мати можливість пошуку рецептів на сторінці “Search”, на якій можна здійснити “Full text search” за введеним у поле запитом. У результаті користувачу видасть до 6 рецептів, які підходять під запит.

Неавторизований користувач повинен мати можливість перейти на сторінку реєстрації або на сторінку авторизації, якщо має існуючий обліковий запис.

Авторизований користувач повинен мати можливість перейти на сторінку “Profile”, де знаходяться посилання на “My recipes”, “Settings” і “Saved”, а також кнопка Log Out.

Авторизований користувач повинен мати можливість вийти з облікового запису за допомогою кнопки “ Log Out ” на сторінці “Profile”.

Авторизований користувач повинен мати можливість редагувати інформацію профіля на сторінці “Settings”.

Авторизований користувач повинен мати можливість переглядати власні рецепти на сторінці “My recipes”.

Авторизований користувач повинен мати можливість додавати рецепти на сторінці “Recipe Constructor”, використовуючи стандартизований конструктор рецептів.

Авторизований користувач повинен мати можливість видалення рецептів, які були створені ним за допомогою кнопки “DELETE”, яка разташована на сторінці “Recipe Page”.

Авторизований користувач повинен мати можливість зберігати і видаляти зі збережених рецепти за допомогою відповідної кнопки на закритій картці рецепту (на сторінках, де вони присутні) або на сторінках “Recipe Page”.

Авторизований користувач повинен мати можливість переглядати збережені рецепти на сторінці “Saved”, де відображається повний список збережених рецептів у вигляді таблиці закритих карток рецептів.

Авторизований користувач повинен мати можливість залишати коментарі і ставити оцінки під рецептами на сторінці “Recipe Page”.

Більше функцій можна побачити на рисунку 1.4.1.

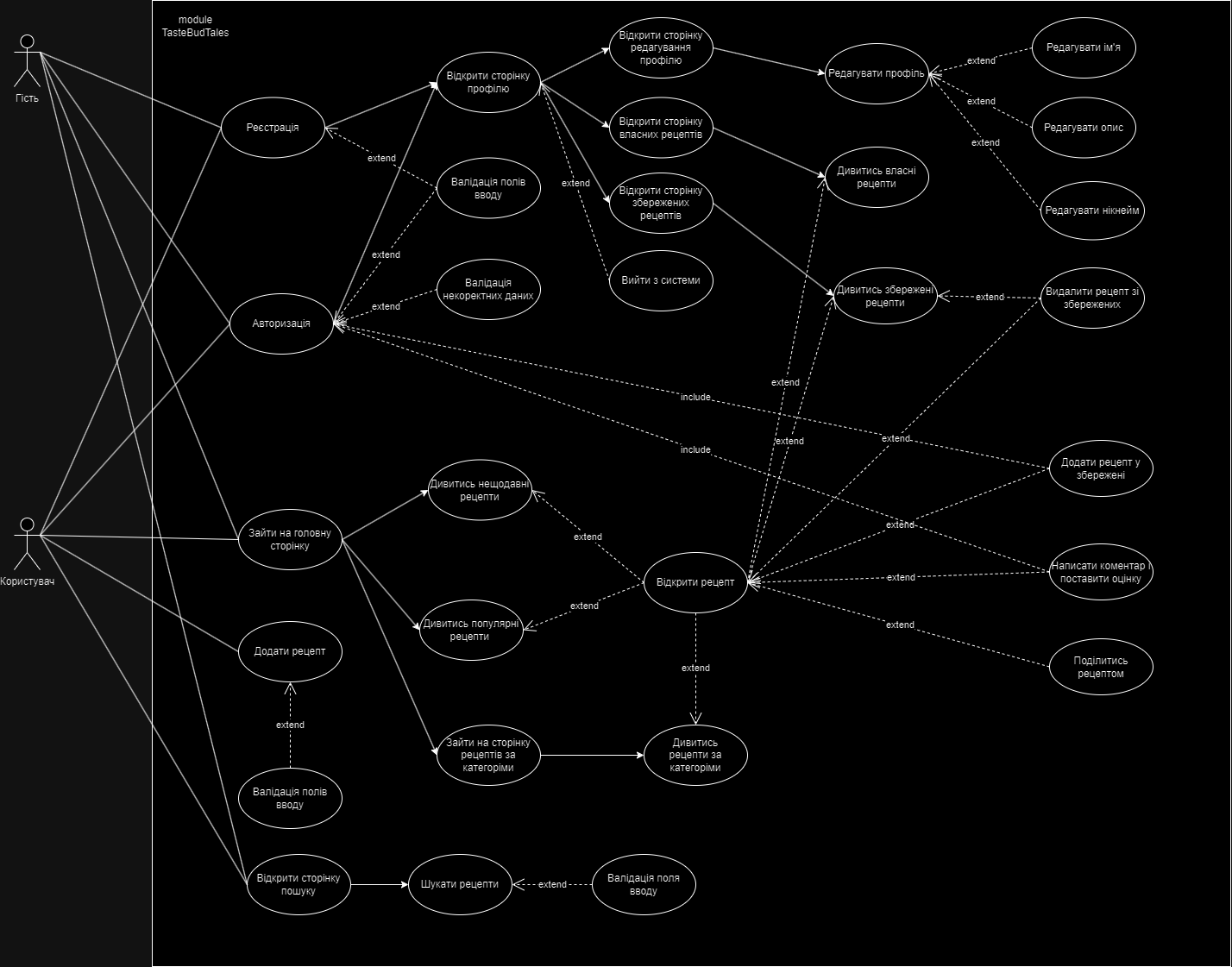


Рисунок 1.4.1 – Діаграма варіантів використання

В таблицях 1.4.1 - 1.4.12 наведені варіанти використання програмного забезпечення.

Таблиця 1.4.1 - Варіант використання UC-1

|  |  |
| --- | --- |
| Use case name | Реєстрація користувача |
| Use case ID | UC-01 |
| Goals | Реєстрація нового користувача в системі |
| Actors | Гість (незареєстрований користувач) |
| Trigger | Користувач бажає зареєструватися |
| Pre-conditions | Користувач не має акаунту у системі |
| Flow of Events | Користувач переходить на сторінку реєстрації. В поля для реєстрації вводяться відповідні дані: пошта користувача, пароль в системі, та його повтор для підтвердження, нікнейм і повне ім’я. Після заповнення даних користувача натискає кнопку реєстрації. Після цього з’являється повідомлення про успішну реєстрацію, і користувач перенаправляється на сторінку профіля. |
| Extension | В випадку введення не коректних даних, під полями з некоректними даними буде показана помилка. Якщо під час реєстрації відбулася помилка, над формою буде показана помилка з описом, чому не вдалося створити нового користувача. |
| Post-Condition | Створення сторінки користувача, перехід на сторінку користувача |

Таблиця 1.4.2 - Варіант використання UC-2

|  |  |
| --- | --- |
| Use case name | Логін користувача |
| Use case ID | UC-02 |
| Goals | Вхід користувача в систему |
| Actors | Гість (неавторизований користувач) |
| Trigger | Користувач бажає увійти у систему |
| Pre-conditions | Користувач має акаунт у системі |
| Flow of Events | Користувач переходить на сторінку логіну. В поля для логіну вводяться відповідні дані: пошта користувача та пароль в системі. Після заповнення даних користувача натискає кнопку логін. Після цього з’являється повідомлення про успішну логінізацію, і користувач перенаправляється на сторінку профіля. |
| Extension | В випадку введення не коректних даних, під полями з некоректними даними буде показана помилка. Якщо під час логіну відбулася помилка (наприклад неправельні дані входу), над формою буде показана помилка з описом, чому не вдалося створити нового користувача. |
| Post-Condition | Створення сторінки користувача, перехід на сторінку користувача |

Таблиця 1.4.3 - Варіант використання UC-3

|  |  |
| --- | --- |
| Use case name | Збереження рецепту |
| Use case ID | UC-03 |
| Goals | Рецепт буде додано до збережених |
| Actors | Авторизований користувач |
| Trigger | Користувач бажає зберегти рецепт іншого користувача |
| Pre-conditions | Користувач має акаунт у системі |
| Flow of Events | Користувач переходить на сторінку “Recipe Page”. Внизу сторінки буде відображено кнопку “Save”. Користувач нажимає на неї і рецепт додається у список збережних користувача. Також користувач може скористатися такою самою кнопкою на закритій картці рецепту (на сторінках, де вони присутні). |
| Extension | Під час збереження рецепту буде відображено лоудер, після цього кнопка змінить свій зовнішній вигляд, а рецепт додасться до списку збережених рецептів користувача. Збережені рецепти можна буде передивитись на сторінці “Saved”. |
| Post-Condition | Рецепт додано до збережених |

Таблиця 1.4.4 - Варіант використання UC-4

|  |  |
| --- | --- |
| Use case name | Пошук рецепту |
| Use case ID | UC-04 |
| Goals | Буде знайдено рецепти за текстовим запитом |
| Actors | Гість(неавторизований користувач) |
| Trigger | Користувач бажає знайти рецепти за текстом |
| Pre-conditions | - |
| Flow of Events | Користувач переходить на сторінку “Search”. На сторінці буде відображено поле для вводу. Користувач вводить у поле текстовий запит і система видає до 6 співпадінь з бази даних. |
| Extension | Користувач повинен ввести від 3 ненульвих символів для здійснення пошуку, інакше пошук не відбудеться і під текстовим полем буде відображена відповідна помилка. |
| Post-Condition | Рецепт видалено зі збережених |

Таблиця 1.4.5 - Варіант використання UC-5

|  |  |
| --- | --- |
| Use case name | Додавання нового рецепта |
| Use case ID | UC-05 |
| Goals | Рецепт буде створено і відображено на сайті |
| Actors | Авторизований користувач |
| Trigger | Користувач бажає створити новий рецепт |
| Pre-conditions | Користувач має акаунт у системі |
| Flow of Events | Користувач переходить на сторінку “Create Recipe”. У відповідні поля вводяться дані: назва рецепту, фото готової страви. Після заповнення даних користувача натискає кнопку логін, короткий опис(за бажанням), список інгрєдієнтів і кроків приготування, cкладність приготування, час приготування, кількість порцій. Після цього нажимає на кнопку “Submit Recipe” і користувача перенаправляє на щойностворену його картку рецепту. |
| Extension | В випадку введення не коректних даних у лівому нижньому куті буде сповіщення, у чому саме проблема. Усі поля є обов’язковими для заповнення, окрім опису. У рецепта має бути мінімум 2 інгредієнти і 1 шаг приготування. |
| Post-Condition | Створено новий рецепт на сайті |

Таблиця 1.4.6 - Варіант використання UC-6

|  |  |
| --- | --- |
| Use case name | Оцінювання рецептів |
| Use case ID | UC-06 |
| Goals | Буде додано коментар під рецептом |
| Actors | Авторизований користувач |
| Trigger | Користувач бажає залишити коментар під рецептом |
| Pre-conditions | Користувач має акаунт у системі |
| Flow of Events | Користувач переходить на сторінку “Recipe Page”. Внизу сторінки буде відображено вкладку “Leave Comment”. Користувач нажимає на неї і у нього з’являється форма з 2 полями: оцінка від 1 до 5 і поле для вводу тексту коментаря. Після заповнення обох полів користувач нажимає на кнопку “Submit” і його повертає до усіх коментарів, де вже відображено його коментар. Якщо користувач залишає коментар під власним рецептом, він не зможе виставити оцінку, лише написати текстовий коментар. |
| Extension | Поля для вводу валідуються. Якщо поле коментаря пусте, користувач не зможе залишити відгук, поки не заповнить його. |
| Post-Condition | Коментар додано і відображено |

Таблиця 1.4.7 - Варіант використання UC-7

|  |  |
| --- | --- |
| Use case name | Видалення рецепту |
| Use case ID | UC-07 |
| Goals | Рецепт користувача буде видалено |
| Actors | Авторизований користувач |
| Trigger | Користувач бажає видалити свій рецепт |
| Pre-conditions | Користувач має акаунт у системі |
| Flow of Events | Користувач переходить на сторінку “Recipe Page”. Якщо це дійсно його рецепт, внизу сторінки буде відображено кнопку “Delete”. Користувач нажимає на неї і рецепт видаляється зі сповіщенням про видалення, а користувача перенаправляє на головну сторінку. |
| Extension | Кнопка буде неактивною під час видалення для уникнення зайвих проблем на серверній стороні. |
| Post-Condition | Рецепт видалено |

Таблиця 1.4.8 - Варіант використання UC-8

|  |  |
| --- | --- |
| Use case name | Редагування профілю |
| Use case ID | UC-08 |
| Goals | Профіль користувача буде оновлено |
| Actors | Авторизований користувач |
| Trigger | Користувач бажає оновити інформацію про свій профіль |
| Pre-conditions | Користувач має акаунт у системі |
| Flow of Events | Користувач переходить на сторінку “Settings”. Перед користувачем з’являються 3 поля для вводу і кнопка “Submit”. Користувач вводить нові дані: повне ім’я, нікнейм і опис профіля. Після нажимає кнопку “Submit” і його перенаправляє назад на профіль з повідомлення про успішну зміну даних профіля. |
| Extension | В випадку введення не коректних даних, під полями з некоректними даними буде показана помилка. Якщо користувач залишить поле повністю пустим, то його значення просто не буде оновлено, залишаться старі дані. |
| Post-Condition | Профіль користувача оновлено |

Таблиця 1.4.9 - Варіант використання UC-9

|  |  |
| --- | --- |
| Use case name | Поширення рецепту |
| Use case ID | UC-09 |
| Goals | Користувач відправить посилання на рецепт у соц. мережі |
| Actors | Гість (неавторизований користувач) |
| Trigger | Користувач бажає поділитись рецептом |
| Pre-conditions | - |
| Flow of Events | Користувач переходить на сторінку “Recipe Page”. Внизу сторінки буде відображено декілька кнопок для поширення у Telegram і Twitter, а також кнопка для копіювання посилання у буфер обміну. При натисканні кнопки Telegram буде відкрито вікно Telegram і можна буде вибрати контакт, кому користувач відправить посилання на рецепт. При натисканні на кнопку Twitter буде створено новий пост з акаунта користувача на Twitter. При натисканні на кнопку копіювання, посилання на рецепт буде збережено у буфері обміну з повідомленням про це. |
| Extension | - |
| Post-Condition | Користувач поділився посиланням на рецепт |

Таблиця 1.4.10 - Варіант використання UC-10

|  |  |
| --- | --- |
| Use case name | Вихід із системи |
| Use case ID | UC-10 |
| Goals | Користувач вийшов із системи |
| Actors | Авторизований користувач |
| Trigger | Користувач бажає вийти зі свого акаунту |
| Pre-conditions | Користувач має акаунт у системі |
| Flow of Events | Користувач переходить на сторінку “Profile”. Справа буде розташована кнопка “Log out”. Користувач натискає на кнопку, його перенаправляє на головну сторінку і показує сповіщення про вихід з акаунту. |
| Extension | - |
| Post-Condition | Користувач вийшов із системи |

### Розроблення функціональних вимог

Програмне забезпечення розділене на модулі. Кожен модуль має свій певний набір функцій. У таблиці 1.4.11 наведено загальну модель вимог, а в таблиці 1.4.12 наведений опис функціональних вимог до програмного забезпечення. Матрицю трасування вимог можна побачити у таблиці 1.4.13.

|  |  |  |  |
| --- | --- | --- | --- |
| Опис | Назва | Пріоритет | Ризик |
| 1. Реєстрація нового користувача | FR-1 | Високий | Середній |
| 1. Вхід в обліковий запис | FR-2 | Високий | Середній |
| 1. Перегляд списку рецептів | FR-3 | Високий | Низький |
| * 1. Перегляд рецептів за категорією страви | FR-4 | Високий | Середній |
| * 1. Перегляд нових рецептів | FR-5 | Низький | Низький |
| * 1. Перегляд популярних рецептів | FR-6 | Низький | Високий |
| * 1. Перегляд власних рецептів | FR-7 | Середній | Низький |
| * 1. Перегляд збережених рецептів | FR-8 | Високий | Високий |
| 1. Перегляд відкритої картки рецепту | FR-9 | Високий | Середній |
| * 1. Збереження рецепту | FR-10 | Високий | Середній |
| * 1. Коментування рецепту | FR-11 | Низький | Середній |
| * 1. Менеджмент наявних інгредієнтів | FR-12 | Низький | Низький |
| * 1. Видалення рецепту | FR-13 | Високий | Високий |
| 1. Додавання рецептів | FR-14 | Високий | Високий |
| * 1. Створення рецепту за допомогою конструктора | FR-15 | Середній | Високий |
| 1. Пошук рецептів за текстом | FR-16 | Високий | Високий |
| 1. Перегляд профілю | FR-17 | Високий | Низький |
| * 1. Редагування профілю | FR-18 | Низький | Високий |
| * 1. Вихід із системи | FR-19 | Високий | Середній |

Таблиця 1.4.11 – Модель вимог у загальному вигляді

|  |  |
| --- | --- |
| Назва | Опис |
| FR-1 | Система повинна надавати можливість створити новий акаунт на платформі |
| FR-2 | Система повинна надавати можливість увійти у свій акаунт за допомогою пошти та пароля |
| FR-3 | Система повинна надавати можливість переглядати рецепти у вигляді таблиці карток |
| FR-4 | Система повинна надавати можливість переглядати рецепти за категоріями, список яких представлений системою |
| FR-5 | Система повинна надавати можливість переглядати найновіщі рецепти на головній сторінці |
| FR-6 | Система повинна надавати можливість переглядати популярні рецепти на головній сторінці |
| FR-7 | Система повинна надавати можливість переглядати власні рецепти у вигляді таблиці карток |
| FR-8 | Система повинна надавати змогу переглядати збережені рецепту у вигляді таблиці карток |
| FR-9 | Система повинна надавати можливість відкривати свої рецепти і рецепти інших користувачів для отримання детальної інформації про рецепт |
| FR-10 | Система повинна надавати змогу додати рецепт до списку збережених використовуючи кнопку на у картці рецепту |
| FR-11 | Система повинна надавати змогу коментувати і оцінювати рецепти за допомогою форми системи |
| FR-12 | Система повинна надавати змогу викреслювати наявні у нього інгредієнти зі списку усіх інгредієнтів рецепту |
| FR-13 | Система повинна надавати авторам рецептів видаляти власні рецепти за допомогою кнопки видалення |
| FR-14 | Система повинна надавати можливість додавати власні рецепти |
| FR-15 | Системою повинна бути надана стандартизована форма для створення власних рецептів |
| FR-16 | Система повинна надавати можливість пошуку рецептів за введеним текстом |
| FR-17 | Система повинна надавати можливість перегляду опцій профілю |
| FR-18 | Система повинна надавати можливість редагувати дані профілю |
| FR-19 | Система повинна надавати можливість вийти зі свого акаунту |

Таблиця 1.4.12 – Функціональні вимоги

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UC/FR** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** |
| **1** | ***+*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** |  | ***+*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3** |  |  | ***+*** | ***+*** | ***+*** | ***+*** | ***+*** | ***+*** |  | ***+*** |  |  |  |  |  |  |  |  |  |
| **4** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***+*** |  |  |  |
| **5** |  |  |  |  |  |  |  |  |  |  |  |  |  | ***+*** | ***+*** |  |  |  |  |
| **6** |  |  |  |  |  |  |  |  | ***+*** |  |  | ***+*** |  |  |  |  |  |  |  |
| **7** |  |  |  |  |  |  |  |  |  |  |  |  | ***+*** |  |  |  |  |  |  |
| **8** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***+*** | ***+*** |  |
| **9** |  |  |  |  |  |  |  |  |  |  | ***+*** |  |  |  |  |  |  |  |  |
| **10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***+*** |

Таблиця 1.4.13 – Матриця трасування вимог

### Розроблення нефункціональних вимог

|  |  |  |
| --- | --- | --- |
| Номер | Назва | Опис |
| NFR-1 | Сумісність з різними браузерами | Система повинна бути сумісна з усіма сучасними браузерами, включаючи Safari, Opera, Chrome, та інші. |
| NFR-2 | Масштабованість | Система має ефективно масштабуватися, коли зростає кількість користувачів або обсяги даних без втрати продуктивності. |
| NFR-3 | Приємний користувацький досвід |  |
| NFR-4 |  |  |

## Постановка задачі

<У підрозділі викладають:

що саме ви збираєтесь реалізувати. Це не зовсім вимоги і варіанти використання. Це достатньо узагальнене поняття з метою, цілями та задачами, що підлягають розв’язанню у результаті розробки програмного забезпечення на курсову роботу.>

## Висновки до розділу

< Необхідно стисло описати усе, що було виконано у даному розділі. Обсяг 0,75-1 сторінка>

# МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Моделювання та аналіз програмного забезпечення

<У підрозділі викладають:

опис бізнес процесів вашої розробки та їх наочне представлення за допомогою засобів моделювання (BPMN та інші).>

Для опису бізнес процесу програмного забезпечення використовується BPMN модель (рисунок 2.1).

Опис послідовності створення облікового запису користувача:

* користувач переходить на сторінку реєстрації;
* користувач заповнює поля реєстрації;
* якщо введені поля, не відповідають шаблону заповнення на клієнтській стороні, відповідні поля підсвічуються помилкою;

## Архітектура програмного забезпечення

<У підрозділі викладають:

втілення архітектурного паттерну для вашого ПЗ та деталізація його компонентів за необхідності. Необхідно виконати повний опис архітектури (схеми, таблиці та ін.).>

## Конструювання програмного забезпечення

<У підрозділі викладають:

опис оригінальних алгоритмів чи модифікацій існуючих. Опис структур даних, програмних структур та ін. Опис бази даних з представленням концептуальної, логічної чи фізичної моделі та з описом сутностей чи таблиць. Опис утиліт, бібліотек та іншого стороннього програмного забезпечення, що використовується у розробці. Аналіз системних вимог.>

Бібліотека xpring, яка виконує запити в мережу блокчейн – однопоточна, тобто не може оброблювати паралельні запити. Через таку специфіку бібліотеки, довелось розробляти рішення яке б не блокувало запити з різних потоків, або інших частин коду. За основу ідеї алгоритму була взята багатопоточність самої мови програмування java. Java для вирішення проблем з потоками представляє ділянки коду які можна синхронізувати, що унеможливлює одночасний доступ до нього з різних потоків. Ця синхронізація проходить за допомогою передачі управління об’єкта-монітора. Алгоритм вирішення проблеми з доступом до однопоточної бібліотеки наведено на рисунку 2.3.

В якості системи управління базами даних використовується Postgres. База даних серверу призначена для зберігання користувачів, а також даних про їх …. Опис таблиць бази даних наведено у таблицях 2.11 - 2.14. Модель бази даних наведена на рисунку 2.12.

Таблиця 2.11 – Опис таблиці user

|  |  |  |  |
| --- | --- | --- | --- |
| Таблиця | Назва поля | Тип даних | Опис |
| user | id | serial | ідентифікаційний номер користувача |
| email | varchar | електронна пошта користувача |
| password\_id | int | посилання на запис у таблиці password, де зберігається пароль користувача |

Опис утиліт, бібліотек та іншого стороннього програмного забезпечення, що використовується у розробці наведено в таблиці 2.22.

Таблиця 2.22 – Опис утиліт

|  |  |  |
| --- | --- | --- |
| №  п/п | Назва утиліти | Опис застосування |
| 1 | Visual Studio Code | Головне середовище розробки програмного забезпечення серверної і браузерної частини курсової роботи. |
| 2 | Postman | Програмне забезпечення необхідне для тестування rest запитів. Використовувалось для тестування API інтерфейсів, та клієнтських запитів. |
| 3 | MySQL Workbench | Програмне забезпечення яке надає легкий графічний інтерфейс для доступу до бази даних. |

## Аналіз безпеки даних

<У підрозділі викладають:

аналіз вразливостей ПЗ та будь-які питання пов’язані з безпекою даних.>

## Висновки до розділу

< Необхідно стисло описати усе, що було виконано у даному розділі. Обсяг 0,75-1 сторінка>

# АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Аналіз якості ПЗ

<У підрозділі викладають:

аналіз якості ПЗ за певними метриками.>

## Опис процесів тестування

<У підрозділі викладають:

опис процесів тестування та приклади тестів.>

Було виконане мануальне тестування програмного забезпечення, опис відповідних тестів наведено у таблицях 3.3 – 3.30.

Таблиця 3.3 – Тест 1.1

|  |  |
| --- | --- |
| Тест | Реєстрація користувача |
| Модуль | Реєстрація користувача |
| Номер тесту | 1.1 |
| Початковий стан системи | Користувач знаходиться на сторінці реєстрації |
| Вхідні данні | Електронна пошта, пароль, підтвердження паролю |
| Опис проведення тесту | У відповідні поля вводяться: коректна електронна пошта, яка до цього не була зареєстрована в системі, пароль від 10 до 64 символів, який містить хоча б з одну англійську літеру, одне число і один спеціальний символ, і який не входить у топ 10000 найпопулярніших паролей, підтвердження паролю, яке співпадає з раніше введеним паролем. Після цього … |
| Очікуваний результат | Реєстрація проходить успішно, користувач додається у систему і перенаправляється на сторінку авторизації. |
| Фактичний результат | Реєстрація проходить успішно, користувач додається у систему і перенаправляється на сторінку авторизації. |

## Опис контрольного прикладу

<У підрозділі викладають:

повний опис контрольного прикладу з усіма можливими розгалуженнями та особливостями. Кроки доповнюють ілюстраціями.>

## Висновки до розділу

< Необхідно стисло описати усе, що було виконано у даному розділі. Обсяг 0,75-1 сторінка>

# ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Розгортання програмного забезпечення

<У підрозділі викладають:

повний опис покрокового розгортання ПЗ. Кроки доповнюють ілюстраціями.>

Клієнтську і серверну частини програмного забезпечення було вирішено розгорнути на платформі Heroku. Для розгортання було використано сервіс GitHub Actions, який надає можливості для постійної інтеграції і розгортання [13].

Розгортання починається коли новий код застосунку доставляється у репозиторій у гілку main. Тоді у середовищі GitHub Actions створюється Docker image за допомогою Dockerfile, що знаходиться у проекті. Цей image розгортається у Heroku за допомогою пакету heroku-deploy. Інформацію про розгортання клієнтської і серверної частини проекту можна побачити на рисунках 4.1 і 4.2.

Рисунок 4.1 - Інформація про розгортання клієнту

## Підтримка програмного забезпечення

<У підрозділі викладають:

опис того, як буде виконуватись підтримка програмного забезпечення. Ілюстрації.>

Користувачі повинні мати можливість отримати нову версію консольного застосунку з кожною версією. До того ж кожна нова версія консольного застосунку повинна бути опублікована в npm. Для автоматизації цього процесу був використаний сервіс GitHub Actions.

Створення нового випуску починається, коли нова версія консольного застосунку доставляється у репозиторій у гілку main, тобто коли commit має tag формату “v\*.\*.\*.”, де замість “\*” знаходиться число. Тоді у середовищі GitHub Actions встановлюється NodeJS. Після цього для проекту встановлюються залежності і проект збирається. Bash скрипт за допомогою бібліотеки pkg генерує виконувані файли (executables) для Linux і для Windows, та пакує файл для Linux у .deb пакет. Після цього .deb пакет і файл для Windows архівуються,

## Висновки до розділу

< Необхідно стисло описати усе, що було виконано у даному розділі. Обсяг 0,75-1 сторінка>

# ВИСНОВКИ

У висновках викладають найважливіші наукові й практичні результати роботи та наводять:

* оцінку одержаних результатів і їх відповідність сучасному рівню наукових і технічних знань;
* ступінь впровадження та можливі галузі або сфери використання результатів роботи;
* наукову, науково-технічну, соціально-економічну значущість роботи;
* доцільність продовження досліджень за відповідною тематикою тощо.

Також у висновках необхідно відобразити стан вирішення усіх поставлених в курсовій роботі задач.

В результаті виконання курсової роботи було спроєктовано …

В якості середовища розробки обрано Visual Studio Code

У якості БД використано FireStore і Firabase Cloud Storage

Після реалізації застосунку він був протестований на пристроях з різними версіями Android, IOS і Windows у браузерах Chrome і Safari з різними розмірами екранів щоб переконатися, що додаток акуратно відображається на різних пристроях.

Наукова новизна роботи полягає в наступному:

Модифіковано:

* Стандартизація створення рецептів для кулінарних веб-сайтів.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. https://www.similarweb.com/top-websites/food-and-drink/cooking-and-recipes/
2. https://bluebirdinternational.com/frontend-frameworks/#:~:text=Popularity%20of%20Frontend%20Framework,-Frontend%20frameworks%20are&text=They%20make%20the%20development%20process,%2C%20Remix%2C%20Preact%2C%20Ember.