**Факультет інформатики та обчислювальної техніки**

**Кафедра автоматизованих систем обробки інформації і управління**

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

\_\_\_\_\_\_\_\_\_\_\_\_ О.А. Павлов

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_ 2019 р.

**WEB-застосування проведення тематичних дискусій та on-line опитувань**

**Програма та методика тестування**

КПІ.ІП-ХХХХ. 045440.02.51

|  |  |
| --- | --- |
| “ПОГОДЖЕНО”  Керівник проекту:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.М. Крамар |  |
| Нормоконтроль:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.І. Ліщук | Виконавець:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ю.О. Ващенко |

Київ – 2019 року

**ЗМІСТ**

[1 Аналіз якості програмного забезпечення 3](#_Toc10319037)

[2 Підходи до тестування 5](#_Toc10319038)

[2.1 Інтеграційне тестування 5](#_Toc10319039)

[2.2 Компонентне тестування 5](#_Toc10319040)

[2.3 Тестування продуктивності 5](#_Toc10319041)

[3 Критерії проходження тестування 6](#_Toc10319042)

[3.1 Інтеграційне тестування, Компонентне тестування 6](#_Toc10319043)

[3.2 Тестування швидкодії 6](#_Toc10319044)

[4 Процес тестування 6](#_Toc10319045)

[4.1 Дані до тестів 6](#_Toc10319046)

[4.2 Задачі тесту 6](#_Toc10319047)

[5 Вимоги до середовища 6](#_Toc10319048)

[5.1 Апаратна частина 6](#_Toc10319049)

[5.2 Вимоги до безпеки 6](#_Toc10319050)

[5.3 Інструменти 6](#_Toc10319051)

[6 Опис контрольного прикладу 6](#_Toc10319052)

# Аналіз якості програмного забезпечення

Аналіз якості розроблених програмних засобів є одним з ключових елементів процесу розробки та впровадження. Якість – суб’єктивне поняття, що позначає відповідність між очікуваною користувачем поведінкою та реальністю. Оскільки якість не є вимірюваною тестування як один з елементів контролю якості займається перевіркою методом порівняння чіткої специфікації з результатами роботи коду у кінцевому наборі тестів.

План тестування наводить набір функцій програмних засобів які будуть протестовані, а також типи тестів, необхідні ресурси, тощо.

План тестування включає виконання тестування всіх частей продукту, від інтерпретатора мови Plotter до авторизаційної частини.

У даному плані будуть протестовані наступні функції:

* Авторизація користувачів;
* Реєстрація користувачів;
* Створення фільмів;
* Додавання страйпів;
* Завантаження відеоматеріалу;
* Збереження Plotter-коду фільму;
* Перегляд списку фільмів;
* Перегляд відео;
* Прийняття рішень;

Тестові модулі:

* Авторизація користувача;
* Авторизація користувача з невалідним паролем;
* Реєстрація користувача;
* Реєстрація користувача з невалідним емейлом;
* Реєстрація користувача з невалідним паролем;
* Створення фільму;
* Видалення фільму;
* Публікування та депублікування фільму;
* Створення та видалення страйпів;
* Завантаження відео для страйпу;
* Завантаження Plotter-коду фільму;
* Завантаження коду, який використовує неіснючі страйпи;
* Завантаження коду з порушеннями синтаксису;
* Завантаження коду, який використовує неіснуючі змінні;
* Завантаження коду з іншими порушеннями семантики;
* Перегляд списку фільмів;
* Отримання стану перегляду та перегляд відео;
* Прийняття рішень та отримання нового відео;
* Рестарт фільму;

# Підходи до тестування

Даний план включає використання наступних методів:

* Інтеграційного;
* Компонентного;
* Продуктивності;

## Інтеграційне тестування

Метод інтеграційного тестування буде застосований до взаємодії між модулями системи:

* Робота API-сервера та Google Cloud Storage;
* Робота API-сервера та фронтенду;
* Робота API-сервера та мови Plotter;

## Компонентне тестування

Метод компонентного тестування буде застосований до частин API-сервера та мови Plotter:

* Сервіси доступу до бази даних;
* Інтерпретація Plotter-коду та перемикання станів перегляду;
* Надавачі API;

## Тестування продуктивності

Методом тестування продуктивності буде перевірена швидкодія надавачів API.

# Критерії проходження тестування

## Інтеграційне тестування, Компонентне тестування

Критерієм проходження інтеграційного та компонентного методів тестування критерій проходження є повне успішне виконання кожного пункту тесту. Якщо хоч один тест було виконано не успішно тестування вважається проваленим.

## Тестування швидкодії

Критерієм проходження тестування швидкодії є повне успішне виконання кожного пункту тесту з усіма доступними наборами параметрів, як то об’єм запитів чи кількість одночасних запитів за допустимий час. В разі перевищення допустимого на виконання часу або невідповіді на запит взагалі тест вважається проваленим.

# Процес тестування

## Дані до тестів

Вхідними даними інтеграційного тесту є набори повідомлень, який отримує компонент системи ззовні від інших компонентів. Вихідними даними є результати роботи компоненту та його підлеглих.

Вхідними даними компонентного тесту є набори параметрів, контекстів та визначених результатів, які і є вихідними даними компонентного тесту.

Вхідними даними швидкодійного тесту є набори різноманітних даних які імітують натуральне використання програмного забезпечення. Вихідними даними являються швидкості обробки запитів, дані по навантаженню, тощо.

## Задачі тесту

Задачею будь-якого тесту є перевірка правильності роботи програми в умовах виконання тесту, а також виявлення потенційних помилок у роботі.

# Вимоги до середовища

## Апаратна частина

Апаратна частина середовища виконання тестів має відповідати технічним вимогам запуску програмних засобів.

## Вимоги до безпеки

Програмні засоби запущені для виконання тестування повинні працювати в окремій базі даних, а також мати окремий сервісний акаунт для доступу до окремого сховища Google Cloud Storage.

## Інструменти

Інструментами для виконання перевірки програмних засобів є:

* Karma & Jasmine;
* Yandex Tank;
* Postman;

# Опис контрольного прикладу

Таблиця 6.1 – Створення фільму

|  |  |
| --- | --- |
| Мета тесту | Перевірка можливості створення фільму |
| Початковий стан | Відкриті програмні засоби |
| Вхідні дані | Назва та опис фільму |
| Схема проведення тесту | Перейти до сторінки «Editor», натиснути на кнопку «Create Movie», заповнити дані у форму, натиснути «Create». |
| Очікуваний результат | Сторінка оновить список фільмів та в ньому відобразиться новостворений фільм |

Таблиця 6.2 – Редагування Plotter-тексту фільму

|  |  |
| --- | --- |
| Мета тесту | Перевірка аналізатора мови Plotter |
| Початковий стан | Відкриті програмні засоби |
| Вхідні дані | Некоректний текст на мові програмування Plotter |
| Схема проведення тесту | Перейти до сторінки «Editor», обрати фільм, перейти на вкладку «Blocks», ввести дані, натиснути «Save». |
| Очікуваний результат | Сторінка відобразить помилковий стан та пояснення некоректності тексту. |

Таблиця 6.3 – Перегляд фільму

|  |  |
| --- | --- |
| Мета тесту | Перевірка обробника прийнятих користувачем рішень |
| Початковий стан | Відкриті програмні засоби |
| Вхідні дані | Заповнений та опублікований фільм |
| Схема проведення тесту | Обрати фільм серед опублікованих, продивитись страйп до кінця, обрати певне рішення |
| Очікуваний результат | Сторінка завантажить новий страйп у відпвідності до plotter-коду фільму, у кінці нового страйпу запитання та відповіді відповідатимуть plotter-коду фі |