CS3337 Software Test Plan Group 6

1. Introduction

1.1 Purpose:

This document details the software testing strategies employed for EurekaEats, a web-based platform that assists users in discovering local restaurant recommendations. The goal is to ensure the robustness of functionality, seamless integration, and optimal platform performance.

1.2 Scope

The testing encompasses all integral functionalities of EurekaEats, including database reliability through MongoDB, integration with external APIs such as Google Maps and Google Places, and CRUD operations with the database.

1.3 References

EurekaEats Requirements Document
EurekaEats Design Document
MapBox API Documentation
Yelp Fusion API Documentation
Pytest Documentation

2. Test Strategy

2.1 Objective

To detect and address flaws or inconsistencies within EurekaEats, aiming to achieve a user-friendly and seamless experience without errors.

2.2 Approach

We will employ primarily manual testing, with some automated testing. Tests are intended to verify individual components, assess integrations, and evaluate the end-to-end system.

2.3 Tools

pytest 7.4.2 - Automated Python unit testing.

2.4 Environment

Development, testing, and production environments will be set up to ensure isolated and focused testing phases.

2.5 Entry and Exit Criteria

Entry:

- Test environment set up with all necessary configurations
- Test data is available

Exit:

- Test cases are executed
- Critical defects are closed
- Test summary reviewed and approved

3. Scope of Testing

3.1 Functional Testing

3.1.1 Unit Testing

During each sprint, developers will write automated tests for each piece of functionality created. This allows us to test each piece of functionality in isolation before it is integrated with the rest of the system.

3.1.2 Integrated Unit Testing

In the middle to end of each sprint, developers are expected to verify how their components will interact with integrated components.

3.1.3 System Integration Testing

Near the end of a planning increment, the development team must integrate all components into the system and test it to ensure that features are correctly integrated per the requirements document.

3.1.4 User Acceptance Testing

Before a release, the product owner selects a group of test users to perform user acceptance testing. The product owner will use user feedback to determine whether the system has met the requirements.

3.1.5 Regression Testing

Testers should continuously test any parts that are added or modified and any components they may interface with. This will ensure that new features are not changing existing functionality.

3.1.6 Smoke Testing

Testers can check the primary functionality to ensure the main components work as intended whenever testing begins.

3.2 Non-Functional Testing

3.2.1 Usability Testing

The UI design will be compared to the UX standards of our group, and it will be confirmed that the design is accessible and that a user will have a seamless experience.

4. Consideration of Infrastructure

4.1 Server Configuration

Due to many constraints, we don’t have the resources to set up a dedicated test server. For this reason, developers will run the tests on their locally hosted web server.

4.2 Database

MongoDB has regular backup procedures and can perform rollbacks when necessary.

5. Risks or Mitigation Plan

5.1 Risks

- Data loss and security vulnerabilities
- Delays accessing database
- Third-party dependencies
- Scalability

5.2 Mitigation Plan

- Data validation checks
- Regular database backups
- Third-party dependency update + depreciation monitoring
- Proper upkeep of web application documentation

6. Resourcing

6.1 Team Composition

Each team member will receive some experience in each role (test manager, test engineer, database test engineer). However, the developers will specialize in different aspects of the project as development continues.

7. Milestones and Deliverables

7.1 Milestones

Planning increment 23.1

- Testing software decided
- Tests are developed with components

Planning increment 23.2

- Planning Increment 23.1 features tested
- Planning Increment 23.1 features integrated
- Regression Testing
- Smoke Testing

Planning increment 23.3

- Regression Testing
- Coding Standard Testing
- Smoke Testing
- Final Report

7.2 Deliverables

- Bug reports
- Test completion report
- MongoDB database integrity/consistency report