

European Vacation

Project Number: 1

Test Plan

Version Number: 1

Product Owner: Nathan Whitney
Scrum Master: Daniel Navarro

Prepared by: Jacob Armstrong, Nathan Whitney,

Mason Emes, Daniel Navarro, Nicholas Hu

1. Purpose of the test plan:

- a. Ensure the project works after completion, with no bugs or unintended behavior due to incomplete code, improperly written code or developer oversight.

2. Scope of the test plan:

- a. Covers testing the project and making sure that the product will be delivered in an acceptable state.

3. Overall Test Strategy

- a. Generally, black box testing, with a focus on improving user experience on top of making sure every feature works properly and is in a deliverable state.

4. What features will be tested from a user's perspective?

- a. The features specified in the requirements for the program that the user should be able to do, such as seeing the restaurant items available for purchase at each city and the ability to calculate distances between cities to see which route is shortest.

5. What features will NOT be tested from a user's perspective?

- a. Things that are required for the program to run properly that the user has no control over, such as the database functionality, making sure all of the algorithms are working properly, and creating an overall GUI.

6. Entry Criteria

- a. Project is finished.
- b. Databases are created.
- c. Able to interact with the program.

7. Exit Criteria

- a. We will confirm there are little to no bugs and can fulfill all the requirements for the project.
- b. Definition of done is met for each story.

8. Suspension Criteria

- a. Found a problem that causes the program to crash.
- b. Realized we had done something wrong and needed to fix it.
- c. If a higher priority issue takes precedence, then that issue shall be taken care of first in order to proceed with testing as usual.

9. Approval Process

- a. Whenever someone finishes a story, have it tested by each member of the team to make sure it fulfills all requirements then push it to their branch.
- b. Upon approval, a merge request shall be created and overlooked by the team to ensure minimal problems when merging to the main branch.

10. Whitebox vs Blackbox Testing

- a. We will do all blackbox testing.

11. Schedule

- a. Sprint 1: test on **09/20**
- b. Sprint 2: test on **10/04**
- c. Finished Project: test on **10/18**

12. Training before Testing

- a. Have whoever did the story go over it with the rest of the team before testing occurs.
- b. Explain what from the stories has been accomplished, and check if our tasks have been completed for each story, explaining the methods used and prerequisites for completion.

13. Environment description

- a. Hardware and software needed for testing:
 - i. Computers are needed for testing, some debugging programs can be used.
 - ii. Qt Open Source development environment needed for testing.

14. Configuration management approach:

- a. GITHUB utilization:
 - i. Agile Board:
 - 1. Contains stories that need to be done or are already completed.
 - ii. Branch Structure:

1. Main Branch, the release branch.

- a. Each has a Team Member Branch that will be branching from main.

- iii. What happens when a test fails:

1. Add it to the backlog on the agile board and re-work until the problem is solved.

15. Test deliverables:

- a. UML Diagrams

- i. Create UML Diagrams of all our databases.

- b. Agile Stories

- i. Create at least 10 agile stories within an agile board on GITHUB.

- c. Coding Standards

- i. Variable declaration/definitions should follow the declaration comments.

- ii. No UPPERCASE or non-camelcase variable names.

- iii. No global variables that can cause memory leaks.

- iv. Good programming standards and variable/function names with readability.

- d. Team Rules

- i. Using QT as our main IDE, C++ as the language.

- ii. Code should work on a Mac and a PC

- iii. Team meeting 3 times a week

- iv. Standup Meetings (approx 10 mins during class times)
- v. Doxygen for automatic documentation.
- vi. Using private Discord server to communicate
- vii. Don't struggle with a problem for more than a day
- viii. 24-hour meeting cancellation notice with reason
- ix. GitHub used for the agile management tool
- x. Keep calendar of important group member dates on scrum log
- xi. Prioritize main requirements over minor details
- xii. When someone completes a story, the scrum master and project owner should REVIEW AND APPROVE IT BEFORE MERGING to the master branch.

e. Doxygen

- i. Document and generate UMLs with class relationships for the program.

f. Scrum Log

- i. Document and record all the meetings and what happened in them.

16. Documents that support the test plan:

- a. Github Project Page - Agile Board
- b. Desk checks to ensure numbers are accurate.

17. Roles and Responsibilities:

a. Team Members

- i. Daniel Navarro - Scrum Master
- ii. Nathan Whitney - Product Owner
- iii. Mason Emes - Team Member
- iv. Jacob Armstrong - Team Member
- v. Nicholas Hu - Team Member

b. We will be planning to follow the definitions of Scrum

Master, Product Owner, and Team Member found in the glossary of terms.

18. Glossary of Terms

a. **UML** - Unified Modeling Language: intended to provide a standard way to visualize the design of a system.

b. Scrum Master

- i. It is the Scrum Master's responsibility to ensure that the team follows the process and practices of Agile methodology.

c. Product Owner

- i. The product owner has the final say if a feature is complete or if a feature is needed in the first place. When a story is proposed to be complete, the product owner is to verify that it is fully functional and complete.

d. Team Member

- i. Every team member is responsible for doing their share of work. This includes the Scrum Master and Product Owner.

e. Agile Board

- i. a visual framework to display and sync upon the tasks moving between the production steps.

f. Merging

- i. Git's way of putting a forked history back together again.