Matt Jang

Lamson Bui

Angaar Hamid

Suqi Hu

Team Go Big O Go Home Test Plan

Test plan ID: A800

Purpose of test plan: Design a well suited GUI application that supports the intentions and purposes of the College Tour. We will be testing that there are no errors present throughout the software process in order to ensure that the program runs successfully and meets all of its requirements.

Scope of the test plan: Various different stories will be tested so that the program maintains its effectiveness. To name a few, we will mostly be concerning:

- Validating information regarding colleges including their distances and souvenirs so that they can be added to the College Tour database.
- Validating user inputs from the administrator in order to access individual pages.
- Sending and receiving information to the database whenever a certain story requires such detail from the database.
- The proper and correct values pertaining to a certain category in the database (i.e. souvenirs, new campuses, college distances).
- Proper functionality of certain widgets so that they meet the specifications of the stories.

Overall test strategy:

● Each developer will utilize the concept of unit testing such that the code written is working properly.

● Each developer will check to make sure that their code meets all requirements by error checking inputs, outputs, boundary values, etc during the development process.

● Once a developer finishes, he or she will let the other team members know and will consult with the Product Owner to gain approval.

● During the verification process, the Product Owner will also be testing the developer's code so that there are no missing requirements.

● If the Product Owner finds that bugs are still present, then the developer must fix his or her code once more until no errors are present.

● Towards the conclusion of each sprint, the team will gather to perform Black Box testing in order to find any missing requirements that need to be accounted for.

What features will be tested from a user's perspective:

● The correct storage of information within the tables in the database.

● The correct display of information from the tables within the database.

● Each widget has their own functionality and works properly.

● Sending and receiving of information from the database.

● The addition or deletion of customers in the database.

● Valid and invalid inputs and the outputs each of them produce.

What features will not be tested from a user's perspective and what the system does:

● Code that links the QT application and SQL database together.

● The SQL queries that allow for the execution of certain information from the database.

● Much of the code that implements the specific functions needed to meet the requirements for all college students and administrators.

Entry Criteria:

● For unit testing strategies, each developer must have their code written so that they can check for proper functionality. Each developer will test their code during the development process.

● For Black Box Testing, every developer must have their software written so that the program can be combined. Once merging issues are dealt with, the team must make sure that the software runs properly.

Exit Criteria:

● When the program has no bugs present and much of the testing strategies have been completed with successful outcomes.

● The program accounts for invalid or undesirable inputs and prevents any harm to the code/executions.

Suspension Criteria: When bugs continue to persist despite various testing being conducted.

Approval process: This case will be approved once the product owner agrees with the conditions.

Schedule: Whenever a developer is working on the story, he or she will use testing strategies such as unit testing while coding so that if bugs are present, then they are fixed right away. Towards the end of each sprint, every developer must have their code finished so that Black Box testing can be performed with the combined code needed to meet each particular sprint's requirements.

Necessary training needed for testing: Each developer has familiarity with using version control systems such as Github. Advanced knowledge of QT or C++ isn't required, but each developer must be proficient with both. Same ideas apply to SQL and using the SQL database.

Environment description:

Hardware: laptops/desktops that have good internet connections and work well.

Software: QT creator, SQLITE database browser such as DB Browser.

Configuration management (GITHUB):

` A developer who creates the story will upload the code to his or her branch. The Product Owner will test the code along with the developer, if errors are found, then they will update the code until no bugs are present. Towards the end of each sprint, each developer must have their own separate branches filled with the finished code. The merging process will proceed afterwards. Black box testing will commence and if testing fails, then the debugging process will begin. The team will then work together to fix the error, then the corrected code will replace the previous code and will be merged into the master branch.

Documents that support the test plan: UML diagrams, QT reference pages, SQL reference pages.

Glossary of terms:

- Unit testing - A testing process which may involve either a certain method or an entire class. Usually occurs either during or after the development process. For this project, each developer must perform unit testing during the software development process.

- Integrated Testing - Tests that verify the unit collaborations. Occurs during unit testing, where each developer will check to see that the collaboration of various units work accordingly and do not produce errors.
- Black box Testing - A method of software testing that examines the functionality of an application without peering into its internal structures or workings. This testing strategy will be used towards the end of each sprint when every developer is finished with his or her code such that consolidation can occur. The team as a whole will perform black box testing so that the program meets all its requirements.