**College Tour**

**Test Plan**

Project Number: 1

Effective Date: 9/18/2017

Scrum Master: Jesus Sanchez

Product Owner: Brandon Kem

Other Members: Christian Murray,

Brian Landaverde

**I. Test Plan ID**

A. Test Plan #1

**II. Purpose**

A. The purpose of this test plan is to test all features using expected, unexpected and boundary values for both white-box and black-box testing to eliminate all undesired outcomes in the application.

- **White-box testing**: Passing expected, unexpected, and boundary values to functions and displaying the results in the console.

- **Black-box testing:** Using expected, unexpected and boundary values as user input and viewing the results as the application proceeds.

**III. Scope of the Test Plan**

A. Purchasing Souvenirs

B. Editing Purchases

C. Integrity of Database

D. Shortest Trip Algorithm

E. Adding Colleges

F. Modifying Souvenirs

G. Login For Admin

H. Saddleback College Trip

I. University of California, Irvine Trip

J. Michigan Trip

K. Custom Trip

**IV. Documents that Support the Test Plan**

A. UML class diagram

B. Agile Management Tool (waffle.io)

C. State Diagrams

D. Use Case Diagram

E. Activity Diagram

**V. Features Tested from a User’s Perspective**

A. Purchasing Souvenirs

B. Editing Purchases

C. Displaying Campuses

D. Login

E. Ability to Modify Database Data

J. Saddleback College Trip

K. University of Michigan Trip

L. University of Irvine Trip

M. Custom Trip

**VI. Features NOT Tested from a User’s Perspective and What the System Does**

1. Shortest Trip Algrithm – The user has no knowledge of how the trips are calculated. Thus, only the developers are aware that this feature is being tested.
2. Integrity of Database – The admin does not have full access to the database. Thus, only the developers are able to ensure the database is functioning properly.

**VII. Overall Test Strategy**

A. The overall test strategy is to use invalid, valid, and unexpected input values when soliciting for user input. Also, boundary testing will be utilized for testing algorithms and features that allow the user to modify data. From the user’s perspective this will be done during program execution. For non-user features, the testing will be done directily in the code with trial-and-error runs.

**VIII. Entry Criteria**

A. Purchasing Souvenirs.

1. Attempt to purchase the same product more than once.

2. Attempt to purchase items with a valid, invalid, and unexpected quantity.

3. Display total amount spent calculations throughout program. execution. Verify that all output is accurate by comparing total amount spent with calculations done on external system.

4. Attempt to purchase non existent souvenir.

5. Attempt to purchase souvenir without quanity selected.

6. Attempt to purchase souvenir without souvenir selected.

B. Editing Purchases

1. Attempt to change quantity to zero

2. Attempt to change quantity to negative number

3. Attempt to change quantity to values greater than zero

4. Attempt to delete purchase when no purchase is selected

5. Attempt to delete purchase when a purchase is selected.

C. Integrity of Database

1. Execute various queries and output the result to a table

2. Attempt to insert invalid data into database.

3. Attempt to insert valid data into database.

D. Shortest Distance Algorithm

1. Display results in console and compare with calculation done on external system.

2. Use Colleges that are in close proximity with other colleges.

3. Use Colleges that are distant from other colleges.

4. Use Colleges that are approximately the same distance away from a particular college.

5. Create a trip starting from Saddleback College to all other colleges. Compare results with calculations done on external system

6. Create a trip starting from University of California, Irvine to all other colleges. Compare results with calculations done on external system.

7. Create a trip starting from University of Michigan to a number of other campuses. Compare results with calculations done on external system.

8. Create various custom trips. Compare results with calculations done on external system.

E. Adding Colleges

1. Enter completely unique college information.

2. Add a college with the same name as an existing college; display error message.

3. Add a college with a distance that is negative or zero.

4. Use regex

5. Attempt to add a college with no name

6. Attempt to add a college with no distance to another college

F. Modifying Souvenirs

1. Attempt to delete the same souvenir more than once.

2. Attempt to add an existing souvenir.

3. Attempt to add a souvenir without specifying a college.

4. Attempt to add a souvenir with a negative price.

5. Attempt to add a souvenir with a price of zero dollars.

6. Attempt to add an existing souvenir but changing upper case letters with lower case letters.

G. Login For Admin

1. Enter valid administrator login credentials and launch the appropriate window.

2. Leave both fields empty; display error message.

3. Enter a username and no password; display an error message.

4. Enter a password and no username; display an error message.

5. Enter invalid credentials; display error message.

J. Saddleback Trip

1. Create a trip to all other campuses starting from Saddleback College.

2. Use shortest trip algorithm.

3. Display ideal trip route and total distance traveled.

4. Compare results with calculations done on external system.

K. University of California, Irvine Trip

1. Create a trip to all other campuses starting from University of California, Irvine.

2. Use shortest trip algorithm.

3. Display ideal trip route and total distance traveled.

4. Compare results with calculations done on external system.

J. Plan a trip from Michigan to all other campuses

1. Enter a value greater than amount of campuses in database

- User should not have this option

2. Enter a value less than amount of campuses in database.

3. Use shortest trip algorithm.

4. Display ideal trip route and total distance traveled.

5. Compare results with calculations done on external system.

K. Custom Trip

1. Select starting campus.

2. Select other campuses to visit besides campuses already selected.

3. User shortest trip algorithm.

4. Display ideal trip route and total distance traveled.

6. Compare results with calculations done on external system.

**I. Exit Criteria**

A. Purchasing Souvenirs

1. Valid souvenirs and quanities selected for purchase from different colleges will launch appropriate window to display purchase summary.

2. Invalid souvenirs or quanitites selected will display error message and will not be stored.

B. Editing Purchases

1. Editing quanities of purchases is reflected in the table displaying purchase data.

2. Deleting purchase removes souvenir from table.

3. Souvenir must be selected in order to edit the quanitity or delete it.

C. Integrity of Database

1. All added/deleted college data is reflected in database.

2. All queries return appropriate data.

D. Shortest Distance Algorithm

1. All variations of college lists will be sorted and displayed properly.

2. Total distance is calculated and displayed correctly.

E. Adding Colleges

1. Added Colleges are reflected in the database and in each table that displays colleges.

2. Administrator is unable to enter any characters that are not letters (using regex) when prompted for college name.

3. Administrator is unable to enter any characters that are not numbers when prompted for distances from all other colleges.

4. Entering a college name that already exists will display an error message and input will be ignored.

5. An empty college entry will result in an error message.

F. Modifying Souvenirs

1. Adding/Deleting Souvenirs reflects changes in the database and the tables displaying souvenirs.

3. Adding a souvenir without a price displays an error message.

4. Adding a souvenir without a souvenir name will displays an error message.

5. Adding a souvenir without a college selected will display error message.

6. Adding a souvenir that already exists will display an error message.

7. Admin is not allowed to enter letters for souvenir price.

8. Adding more than six souvenirs will result in an error message.

G. Admin Login

1. Valid credentials launch appropriate window and invalid credentials or empty fields display an error message.

H. Saddleback Trip

1. Trip to all other campuses starting from Saddleback College is created.

2. Shortest trip algorithm is used.

3. Displays accurate calculations for total distance.

4. Displays ideal trip route correctly.

I. University of California, Irvine Trip

1. Trip to all other campuses starting from University of California, Irvine is created.

2. Shortest Trip Algorithm is used.

3. Displays accurate calculations for total distance.

4. Displays ideal trip route correctly.

J. University of Michigan Trip

1. User can only select an amount of colleges to visit that is less than

or equal to the total amount of campuses in database.

2. Shortest trip algorithm is used.

3. Displays accurate calculations for total distance.

4. Displays ideal trip route correctly.

K. Custom Trip

1. User can select any starting colleges they choose.

2. User can select any colleges from database to visit except ones already selected.

3. Shortest trip algorithm is used.

4. Displays ideal trip route correctly.

5. Displays total distance correctly.

**II. Suspension Criteria**

A. Any testing should be suspended if certain values entered during program execution cause the program to terminate unexpectedly or statements added to the code prevent the build from occurring. The test team should record the value or statement that caused the issue and revisit the code to determine why the error occurred.

**III. Roles and Responsibilities of the Test Team**

A. The test team will consist of all members of the development team. Each member should test the functions they have written themselves before anything is added to the main executable file. Once the owner of a specific section of code has fully tested and pushed their code to the main branch of the repository, other members should run the program again and utilize black box hacking techniques in an attempt to break the system so that any weaknesses can be addressed and taken care of.

**IV. Schedule**

A. Testing should be performed (by any team member) at least once a day before any new material is added to the project to ensure that the previous day’s work is still functional. On sprint review days, the code should be tested a few hours prior to the beginning of lab in order to ensure that all new functions implemented during the sprint are functional.

**V. Necessary Training Needed for Testing**

A. All members should understand every aspect of the code in order to be able to effectively test it. If there are questions about how certain parts of the code work, the member with the question should ask the owner of that section to explain what is happening if the documentation is not clear enough.

**VI. Test Environment Description**

A. All test team members should have access to the following hardware and software...

1. Windows, Unix or Mac operating system

2. Qt Creator (v. 2.8 or higher)

3. GitHub

4. Waffle.io

5. Travis CI

**VII. Test Deliverables**

A. Documentation

B. Source code

C. **Test environment**

**VIII. Approval Process**

A. Test strategy implemented

B. Successful Jenkins integration

C. Successful **white-box testing**

D. Successful **black-box testing**

E. Product approved upon customer **validation** and successful product owner **review.**

**IX. Glossary of Terms**

A. **Black-box testing -** tests based on the behaviour of the component or system, derived from a specification. Also known as functional testing or behavioural testing.

B. **Review -** a process or meeting during which a work product, or set of work products, is presented to project personnel, managers, users or other interested parties for comment or approval. Types of review include walkthrough, inspection, informal review and technical or peer review.

C. **Test Environment -** a description of the hardware and software environment in which the test will be run, and any other software with which the software under test interacts with under test including stubs and drivers.

D. **Validation -** determination of the correctness of the products of software development with respect to the user needs and requirements.

E. **White-box testing -** test case selection that is based on an analysis of the internal structure of the component. Also known as structural testing or glass box testing.