**CS1D Baseball Project**

**Test Plan**

**ID: 001**

**Scrum Master:** Jorge Rodriguez

**Product Owner:** Nicholas Hu

**Team:** Quarantine Coders

**Purpose of the test plan**

* To test front end aspect of the product using appropriate testing techniques
* Test the functionalities of the program that is meant for two users: Baseball fan and Admin
* Outcome to ensure that program has minimal bugs allowing the Baseball Fan to create their dream vacation and allows the Admin to maintenance the program as smoothly as possible

**Scope of the test plan (What will be tested)**

* Using Agile development to help us break the project into smaller portions to make the process efficient.
* Objective allow user to plan their vacation trip with the shortest trip, display traveled distance, purchase multiple souvenirs
* Objective allow admin to add/delete or modify existing data such as stadiums or souvenirs

**Support Documents**

* Doxygen - class relationships and data in source code
* UML diagrams - show the flow in use cases

**Features tested from a User’s perspective**

* The capability to visit any team starting at Dodger Stadium (shortest distance)
* The capability to visit any team starting at Marlins Park (shortest distance)
* The capability to create a custom trip (shortest distance)

1. Recursively choose team closest to the previous

* Display distance travelled for every trip
* Offer option to purchase multiple souvenirs

1. Display total spent at each stadium
2. Display grand total spent

* Maintenance Login

1. Capability to add new stadium & corresponding souvenirs
2. Capability to modify/add/delete souvenirs
3. Capability to modify stadium information

* Using Dijkstra’s or A\* algorithm to find shortest distance
* Determine MST using Prim’s or Kruskal’s algorithm to display mileage
* Perform DFS starting at Oracle Park
* Perform BFS starting at Target Field

**Features not tested from a user’s perspective (What the system does)**

1. Create, read & write to data base
2. Requiring an encrypted password to gain access (Admin only)

**Overall test strategy (White vs black)**

* Once a team member completes a story, update Trello and notify product owner.
* Product owner will test code for story completeness and bug checks
* Product owner will notify team member if any errors occur
* Black Box
* Test cases will be tested based on a user-functionality.

**Entry criteria**

* Test cases are developed
* Testable code is available
* Requirements are met and approved

**Exit criteria**

* Product owner deadline met
* Sufficient coverage of all functionalities
* All errors are corrected and verified by Scrum Master

**Suspension criteria**

* Error that does not allow for continuous progression must be re-evaluated by team members and fixed

**Roles and Responsibilities**

*Scrum Master*: Jorge

* Logging of tests, and their completion state

*Product Owner*: Nicholas

* Ensuring test conditions are in line with product goals

*Team Member*: Alvaro, Kristine

* Implementing the test code protocols
* Testing your own code as well as others

**Schedule**

* Testing will occur before merging code on git
* Product owner will test master branch at the beginning of every team meeting
* If test unsuccessful by team member address it at team meeting

**Environment description**

* Team members must be able to run and use QT
* Team members must be familiar with push/pulling/merging on GITHUB
  + Each member will have their own branch
  + Master branch for final product
* Team members have access to SQLite
* Team members must use Trello, for agile management

**Test deliverables**

* UML Diagrams
* Agile Stories
* Coding Standards
* Team Rules
* Doxygen
* Test Plan
* Scum Logs

**Approval process**

* Team members will submit stories for testing to product owner
  + Product owner test for story completeness
* Scrum master will assign team stories
* Product owner will go over case testing for approval
* Issues will be discussed during team meetings

**Glossary**

* QT
* GIT
* Black Box