

# **Team Runtime Error Test Plan**

Amin Parvizi, Behrad Rezazadeh Moussavi, Huyang Yu, Nicholas Lozano

## **Purpose**

The program will be tested to ensure it meets all requirements outlined in the “Take Me Out to the Ballgame” project. Testing will be performed to identify and remedy any encountered bugs throughout the development process after each user story is suspected to be complete.

## **Scope**

The program will be tested against each requirement listed in the requirement specifications. Some examples of this follow:

- Validate that teams and stadiums are viewable in a user-specified order of sorting.
- Validate all user input and check that it is valid before proceeding.
- Validate that all UI elements perform their intended behaviors.
- Validate that all stadium tours visit in expected orders.

## **Test Strategy**

1. Members will submit pull requests when they believe their tasks to be complete.
2. Developers will verify that all written code matches the definition of done and adheres to team coding standards.
3. Developers will perform unit testing on finished changes before merging.
4. Black box testing will be performed by developers before project submission to find potential missing elements.

## **Testing from a user’s perspective**

- UI elements, such as buttons, function in the way they are expected to from labels.
- Database can be modified and saved after making changes, even between executions.
- All MLB team and souvenir information can be modified.
- Distances shown from trips and prices shown from purchases are accurate.

## **Testing from a developer’s perspective**

- All slots are connected correctly and provide appropriate functionality.
- All external files are accessible and used properly by the program.
- Software is developed to check for correct input values and alert user if incorrect values are submitted.

## **Entry Criteria**

- Program adheres to all testing standards and has been tested and believed to match the definition of done.

## **Exit Criteria**

- No bugs are found during testing and all requirements are met.

## **Suspension Criteria**

- Bugs have been found during testing or an unexpected result occurs.
- Desired changes cannot be safely merged with past code.

## **Approval Process**

- Member will receive approval from the team to merge any changes made to the project.
- Changes will be committed and merged to existing code on GitHub.

## **Schedule**

- Unit testing is done at any time a new portion of code is believed to meet a requirement.
- Extensive black box testing must be done before presenting anything for sprint reviews, especially during the final sprint.

## **Necessary Training**

- Knowledge of C++ programming and Qt development.
- Knowledge of SQLite operation.
- Knowledge and use of GitHub as a management tool.

## **Environment**

- Hardware
  - Any computer
- Software
  - Microsoft Windows or Mac OS
  - SQLite 3
  - Qt Creator
  - Git

## **Configuration Management**

- The main branch should not be overwritten or written to directly.
- User stories should be handled and tested individually rather than in groups.
- Code that does not meet exit criteria should be modified and re-tested as soon as possible.

## **Support Documents**

- Qt Documentation