# **Test Plan**

This test plan is to list out all the functions, within the project scope, that are required to be tested. It includes description of each test, the testing dates of each function, the testing dates as well as the testers responsible for each testing. It also shows the overall process of testing.

# **Functions Tests**

|  |  |  |  |
| --- | --- | --- | --- |
| **Functions to Test** | **Test Description** | **Test Date** | **Testers** |
| Login | This functionality will validate a user to ensure that they can only access student dashboard and admin dashboard upon valid credentials respectively. | 25 09 17 | Amanda Lim  Joel Tay  Ming Xuan |
| Bootstrap SLOCA System | This functionality will validate an admin user to bootstrap the system and upload additional location and demographics data. |  |  |
| Basic Location Report – Students Breakdown Report | This functionality will allow a user to see a basic report that shows the breakdown of students in the SIS building by their year (2013/2014/2015/2016/2017), by their gender(male/female), and by their school. |  |  |
| Basic Location Report – Top K popular places report | This functionality will allow a user to see a basic report that shows the top-k popular places in the whole SIS building at a specified time. |  |  |
| Basic Location Report – Top k people location report | This functionality will allow a user to see a basic report that shows the top-k other users who were co-located with a specified user (using MAC address) in a specified query window. |  |  |
| Basic Location Report – Top k next locations report | This functionality will allow a user to see a basic report that shows the top-k next popular places that users located at place A are likely to visit in the next 15 minutes. |  |  |
| Automatic Group Identification | This functionality will identify potential groups in the SIS building at a give date/time based on users' location traces. |  |  |
| Heatmap | This functionality will allow a user to view the crowd density of a specified floor in the SIS building, given a particular date and time. The crowd density should be shown in a user-friendly way. |  |  |

Each function test is considered as pass if it passes all the test cases assigned to the respective function test.

If one or more test case has fail, the testers should log the bug and inform the programming pair in-charge of the function to debug if it is critical.

If the bug is critical to the system (e.g. futures tasks cannot start due to this bug), delay the pair programming pair’s future tasks and conduct a pair programming session to debug.

If the bug is not critical to the system (e.g. grammatical error, misalignment of UI), the programming pair will finish their tasks at-hand before proceeding to debugging.

# **Regression Testing**

For every deployment to the Amazon Web Service (AWS) or iteration end, regression testing will be conducted after the team. Regression testing is required to ensure that the previous functions still work with the new changes added. To run old test cases against the new version to make sure that all the old capabilities still work and to uncover new bugs in the system.

# **User Acceptance Testing**

The user acceptance testing (UAT) is done by the clients/users of this system and this is to ensure that this system meets their requirement. If there are any requirements not met by the system, they will be logged in the bug metric and project manager will estimate the amount of time needed to resolve these issues and create the pair programming sessions necessary.

# **Deployment Testing**

The deployment testing is done by the clients/users of this system to ensure that the system is working according to the tested functions after being deployed onto AWS. If there are any requirement that are not met by the system, they will be logged to the bug metrics with the necessary time needed to resolve the issue. Debugging session would be done according when deem fit by the pair programmer or project manager.

# **Testing Process**

1. Update test plan in each iteration

2. Create test cases for each function

3. Perform the test for each test cases after the function is coded

4. Log the bugs in the bug metric (if any) and resolve immediately for bugs which are deemed as critical

5. Fix all the bugs during debugging sessions / iteration overall function testing