# 

# 

# **Mall-E**

**Test Plan**

**Version 1.1**

## **Runtime Error**

# **Contents**

[**1. Introduction**](#_7bawz82z52rc) **3**

[**2. Test Items (Functions)**](#_q747chf2co52) **3**

[**3. Software Risk Issues**](#_vec0utyl6uei) **4**

[**3.1 Critical Areas**](#_slgunl8nov9y) **4**

[**3.2 Software Risks**](#_eclvamtgskqi) **4**

[**4. Features to be Tested**](#_d8l6sc49ehbl) **5**

[**5. Feature not to be Tested**](#_1twou7i1y997) **6**

[**6. Approach (Strategy)**](#_80hsnh5k20aw) **6**

[**6.1 Team Organization**](#_306fvgl6nuvl) **6**

[**6.2 Metrics To Collect**](#_5z16icybx144) **7**

[**6.3 Configuration Management**](#_3rdcrjn) **7**

[**6.3.1 Configurations To Test**](#_26in1rg) **8**

[**6.3.2 Regression Testing**](#_2i8n2090o0n) **8**

[**6.3.3 Handling Untestable Requirements**](#_2jxsxqh)8

[**7. Item Pass / Fail Criteria**](#_3bdv6zz2ux0b) **8**

[**8. Suspension Criteria and Resumption Requirements**](#_eyyjo3g48nor) **9**

[**9. Test Deliverables**](#_1y810tw) **9**

[**10. Environmental Needs**](#_2xcytpi) **9**

[**11. Responsibilities**](#_1ci93xb) **10**

[**12. Staffing And Training Needs**](#_3whwml4) **10**

[**13. Schedule**](#_2bn6wsx) **10**

[**14. Risks and Contingencies**](#_2bn6wsx) **10**

**IEEE 829 Test Plan Template**

**Test Plan Identifier**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 09/Apr | 1.0 | Initial Release – Prototype test plan | Goh Hong Xiang, Bryan  Rajuravi Vishal Raj |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 

# 

# 

# 

# 

# 

# 

# **1. Introduction**

The test plan serves as a bridge to the test cases that have been scripted in accordance to the functional requirements and sequence of the application. It serves as a means of communication between the team members and stakeholders. All logs are kept for any testing done in a particular release, along with any comments or conversation notes.

The test plan also contains detailed documentations of the objectives, resources and processes for a specific software or hardware. The plan typically contains a detailed understanding of the eventual workflow.

# **2. Test Items (Functions)**

Below identifies the main function of our application

|  |  |
| --- | --- |
| Crowd Density per Mall | * Users should be able to view a list of malls / markers of malls on a map screen * Users will then be able to view information on the malls * Users will then be able to view the crowd density level of the particular mall |
| Floor-by-Floor Crowd Density per Mall | * Users should be able to view a list of malls / markers of malls on a map screen * Users will then be able to view information on the malls * Users will then be able to view the floor-by-floor crowd density level of the particular mall |
| Waiting Time of a Particular Restaurant in that Mall | * Users should be able to view a list of malls / markers of malls on a map screen * Users will then be able to view information on the malls * Users will also be able to view the list of restaurants in that mall * Users will also be able to click on the specific restaurant for estimated waiting time details |
| Covid-19 News | * User should be able to view news articles related to Covid-19 under news tab |

# **3. Software Risk Issues**

## **3.1 Critical Areas**

To identify what software is to be tested and what the critical areas are:

|  |  |  |
| --- | --- | --- |
|  | Types of system testing | Functions |
| 1. | Functional Testing | * Testing function / system requirements |
| 2. | Performance Testing | * Testing of application’s general flow * Testing whether application is bug-free * Testing application’s capability to accept high request / demands |
| 3. | Pilot Testing | * Testing of the application’s ability to extract real time data from APIs |
| 4. | Acceptance Testing | * Ability to use and understand a new package / tool / application by third party * Ability to understand the complex functions by third party |
| 5. | Installation Testing | * Testing the modifications to components with past history of failure * Testing the new releases to the application |

## **3.2 Software Risks**

There are some inherent software risks such as complexity; these need to be identified.

|  |  |  |
| --- | --- | --- |
|  | Types of risks |  |
| 1. | APIs’ server | * Data from third party are not under our jurisdiction, any crashes in their server will result in the failure of the application |
| 2. | Government regulations and rules | * With reference to our project, CCTV footage is required. This data needs to be kept confidential and refrained from spreading. |

# **4. Features to be Tested**

Purpose of the application: To allow users to be notified of crowd density levels within a mall so as to allow them to plan their schedule in advance and avoid crowded areas to reduce the spread of COVID-19.

|  |  |  |
| --- | --- | --- |
| **Features** | **Risk Level** | |
| Users must be able to view the markers on the map interface in the application. Markers shows all of the malls in Singapore | **Medium** | * This function brings convenience to the users for mall searching. * Users can go to the list views of malls to view the specific mall (Back-up) |
| Users must be able to view the list of malls under malls tab | **Medium** | * This function brings convenience to the users for mall searching. * Users can go to the map interface to view the cluster of malls to view the specific mall (Back-up) |
| Users must be able to view crowd density level within multiple malls | **High** | Our application would not be functional and serve its purpose if users are unable to view the crowd density level within each malls |
| Users must be able to view floor-by-floor crowd density level within multiple malls | **High** | Our application would not be functional and serve its purpose if users are unable to view the floor-by-floor crowd density levels |
| Users must be able to view the list of restaurants within the mall | **High** | If the list of restaurants fails, users will not be able to view the restaurant waiting time simply because there is no option of clicking on any restaurants |
| Users must be able to view the waiting time of restaurants within that specific mall of multiple malls | **High** | Our application would not be functional and serve its purpose if users are unable to view the waiting time of restaurants in a mall |
| Users must be able to view COVID-19 related news | **Medium** | * This function would be great to have function on top of the other functions * It may be a changing factor if there was a previous COVID-19 case that previously visited the mall, people will avoid the mall due to fear * Yet it would not be of much hindrance if it fails to work as there are other platforms for news. |
| User Login Function | **High** | Users will not be able to enjoy the application functions if the user fails to login using their credentials. There is no way to bypass the login page to the functional pages. |
| Bookmark favourite / frequented malls | **Low** | Users should be able to bookmarks favourite malls / frequented malls. Users should be able to view the list of bookmarked malls |
| Search Function | **Medium** | To improve user-friendliness and convenience, users should be able to search for specific malls and restaurants. |

# **5. Feature not to be Tested**

Purpose of the application: To allow users to be notified of crowd density levels within a mall so as to allow them to plan their schedule in advance and avoid crowded areas to reduce the spread of COVID-19.

|  |  |  |
| --- | --- | --- |
| **Features** | **Risk Level** | |
| Setting Button | **Low** | Main functionalities are to be up first before working on miscellaneous items |
| Profile Page | **Low** | Main functionalities are to be up first before working on miscellaneous items |

# **6. Approach (Strategy)**

## **6.1 Team Organization**

|  |  |  |
| --- | --- | --- |
| **Role(s)** | **Name** | **Roles** |
| Project / Test Manager / | Das Atrik | • Oversees project progress  • Approves and executes project plan  • Assigns tasks and reports status of the project to team members |
| Quality Assurance Manager / | Rajuravi Vishal Raj | • Ensures acceptable software qualities  • Crafts the project quality plan  • Ensures acceptable software qualities |
| Release Manager / Test Administrator | Kelvin Chua Chong Hung | • Build up the test environment and maintenance of assets  • Aiding the Tester with the use of test environment for test plan execution |
| Developers | Melvin Chua Chong Wei  Goh Hong Xiang, Bryan | • Support in implementing the test cases and test programs |
| Tester | Padhi Abhinandan | • Identify the appropriate test technique and tools for the application  • Verification and assessment of the test approach |

## **6.2 Metrics To Collect**

* Number of subclasses (collected at Integration testing level)
* Length of identifiers (collected at Unit testing level)
* Depth of inheritance (collected at Acceptance testing level)
* Cyclomatic Complexity (collected at System testing level)
* Fog Index (collected at System testing level)
* Weighted methods per class (collected at System testing level)
* Number of overriding operations (collected at System testing level)
* Length of Code (collected at Integration testing level)
* Depth of conditional nesting (collected at Integration testing level)
* Class coupling (collected at Integration testing level)
* Fan-in/Fan-out (collected at System testing level)

## **6.3 Configuration Management**

During the SDLC of Mall-E, the software configuration management was split into four main chronological steps which are as follows:

1. Configuration Identification: To identify the items that had to be controlled during SDLC.
2. Configuration Control: To handle change management processes during the SDLC.
3. Configuration Status Accounting: To back-track all the configuration items during its lifecycle.
4. Configuration Audit: To review the software product and meet the requirement functional standards.

### ***6.3.1 Configurations To Test***

1. UML Class Diagram design: Testing the full component design.
2. UML Sequence diagrams: Testing the full component design.
3. React Native Source Code: Testing of the functional requirements.
4. Unit Test cases: Revising and updating the unit test cases during the SDLC as per requirements.
5. Integration test cases: Revising and updating the integration test cases during the SDLC as per requirements.

### ***6.3.2 Regression Testing***

The team would be required to perform the following types of regression testing as mentioned:

1. *Selective Regression Testing :* Any new piece of code added to the initial software is analysed. In order to avoid redundancy and cost for retesting, subsets from the initial test cases are used during this regression testing process.
2. *Progressive Regression Testing :* When changes were made to the program specifications along with designing of new test cases, progressive regression testing was conducted. This regression testing aids in maintaining the quality of the features in the newer versions.
3. *Complete Regression Testing :* When several changes were made to the existing code, complete regression testing is used. This testing helps in identification of unexpected errors. This was used mainly when changes were made to the root code of the application.

### ***6.3.3 Handling Untestable Requirements***

In case of any untestable requirements, the development team will come in contact with the project manager and the stakeholders to make final decisions of the untestable requirements.

# **7. Item Pass / Fail Criteria**

This is broken down into steps.

1. Define Test Cases.
2. Execute Test Cases.
3. Check if Test Cases **fulfil** functional requirements / system requirements.
4. If criteria **met**, finish the testing.
5. If criteria are **not met**, suspend testing.
6. Developers **handle** the bugs.
7. Repeat step 2.

# **8. Suspension Criteria and Resumption Requirements**

Test suspension will only occur if the following 1 or more scenarios are met:

* Unavailability of related external dependent systems (i.e. APIs are revoked by service providers).
* Further testing will result in the critical path submission deadline to be missed.
* Public holidays.
* Fatal Errors that result in program abortion.

Testing will resumed if the following requirements are met:

* External Dependent Systems are functional and made available again.
* Extension of the critical path submission deadlines.
* End of Holidays.
* Fatal Errors were debugged and fixed.

The following defects are considered acceptable-level defects and will be assigned a “P3-low” priority while the development team works to fix more urgent defects:

* UI /Visual Defects that do not affect core functionalities of the app.
* Defects that are non-reproducible by normal user behavior.

# **9. Test Deliverables**

The deliverables after the test plan are:

1. Test Plan Document
2. Test Design Specifications
3. Test Cases
4. Tools and their outputs
5. Test Simulation Logs
6. Error and Execution Logs
7. Problem Reports and Corrective Actions

# **10. Environmental Needs**

For smooth execution of the test plan, the following actions are taken:

* Hardware: Having additional backup devices such as test computers and test mobile phones in case of any unexpected breakdown.
* Maintenance Process: Having the list of all expected team members along with their contact details for the test environment running. Also, ensure the leaders are the single point of contact.

# **11. Responsibilities**

|  |  |  |
| --- | --- | --- |
| **Human Resources** | | |
| **Role** | **Minimum Manpower Recommended** | **Specific Responsibilities/Comments** |
| Test Manager | Das Atrik | * Selection features to be tested and not tested * Finalize the overall strategy and approach for the plan |
| Test Designer | Melvin Chua Chong Wei  Bryan Goh Hong Xiang | * Selection of the tasks for testing * Provide the required training needed to the tester |
| System Tester | Padhi Abhinandan | * Execution of the tests |

# **12. Staffing And Training Needs**

Every team member will be actively involved in the testing and each of them will be designated with a specific testing role assessing their strength and weakness. Training will be given every team member personally by the testing manager to ensure smooth sailing of the testing.

# **13. Schedule**

The SDLC used for the development of Mall-E is an incremental development model and carried out in the form of iterations which lasts for 2-3 weeks depending on the workload. Regression testing is conducted to verify that additional features do not dampen the existing functionalities. Moreover, unit testing and integration testing is carried out at every iteration.

# **14. Risks and Contingencies**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. NO. | Risk | Mitigation Strategy | Probability | Severity |
| 1 | Testing Flow is affected by tight schedule | Design a project tracking timeline to maintain steady progress | High | High |
| 2 | New features are not technically feasible | Allocate time for reviewing these features during prototype stage | Medium | High |
| 3 | Unexpected delays in bug-fixes | Establish procedure for tackling and fixing bugs | Medium | High |
| 4 | Unexpected issues with the hardware and software testing tools | Plan backup of the test cases and plan ahead for future developments | Low | High |
| 5 | Features that are not implemented are not testable | Plan which all features need to be tested | High | Low |