# 

# 

**Test Plan**

Campus Tour Application

Team C – Crazy Coders

Team members:

Mallikharjuna Rao Dande

Nayan Reddy Prodduturi

Teja Parimi

Rakesh Chitturi

Vamsi Krishna Solasa

Chiranjeevi Sneha Kotu

Contents

[Introduction 3](#_Toc468821235)

[1.1 Objectives 3](#_Toc468821236)

[1.2 Team Members 3](#_Toc468821237)

[2 Scope 3](#_Toc468821238)

[2.1 Strategy 3](#_Toc468821239)

[2.2 Test Scenarios 4](#_Toc468821240)

[2.3 Assumptions 4](#_Toc468821241)

[2.4 Risks 4](#_Toc468821242)

[3 Test Approach 5](#_Toc468821243)

[3.1 Test Automation 5](#_Toc468821244)

[4 Test Environment 6](#_Toc468821245)

[5 Milestones / Deliverables 6](#_Toc468821246)

[5.1 Test Schedule 6](#_Toc468821247)

[5.2 Deliverables 6](#_Toc468821248)

# **Introduction**

The Test Plan has been created to communicate the test approach to team members while developing the campus tour application. It includes the objectives, scope, schedule, risks and approach during the development. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

## Objectives

We will be using Bugzilla for logging bugs in code which are found during the testing process. Every team member will be maintaining their own version of test suite whose details will be uploaded to Bugzilla once a day to track defects and the test results.

Testing will be done on daily basis whenever the code is updated. The core feature testing will carry highest priority for testing. Since every team member will be coding in this project the developer will choose one of the remaining five team members to test their code and log it in the bug tracker.

## Team Members

|  |  |
| --- | --- |
| **Resource Name** | **Role** |
| Mallikharjuna Rao Dande | Team Lead, Developer & Tester |
| Nayan Reddy Prodduturi | Communications Manager, Developer & Tester |
| Teja Parimi | Developer, Tester and Quality Manager |
| Rakesh Chitturi | Data Manager, Developer & Tester |
| Vamsi Krishna Solasa | Issues Manager, Developer & Tester |
| Chiranjeevi Sneha Kotu | Developer & Tester |

# **Scope**

## Strategy

The initial phase will include all ‘must have’ requirements. These and any other requirements that get included must all be tested. At the end of MileStone 1, a tester must be able to:

1. Create a manual test with as many steps as necessary
2. Save it
3. Retrieve it and can view it when running the test
4. Enter results and appropriate comments
5. View results

As the team works with the product they will define the needs for the second milestone.

Load testing will not be considered part of this project since the user base is known and not an issue.

The above-mentioned tester duties will be managed with the following control procedures

**Control Procedures:**

The team members work on different tasks and integrate the resulting code during milestones mostly working individually and collaborating at milestones.

**Reviews:**

We individually review others work and comment them for better enhancement.

**Review meetings:**

Review meetings every week to discuss, review and fix any errors occurred

**Change Request:**

We need to check with Client Requirements regularly whether we are developing the app accordingly or not, if not change them as per the client request.

**Defect Reporting:**

While using the app, if any defect is occurred, Tester reports it in the bug tracking tool and assigns it to the concerned developer to fix it.

## Test Scenarios

There are few test scenarios which needs to be implemented before the app launches, there are listed below:

* Initially to load the campus map view, the user must enter the desired campus and he would also able to get the data downloaded from the server for the campus.
* To ensure while the user in virtual tour, can he fetches the appropriate data of that building.
* When the user in the campus tour, if he is diverting from the actual directed path then he must alert to get back into the actual route and tour must be resumed from the last point.
* Need the check whether the user data enabled during the campus tour, if not or interrupted, then he must alert to check the data connection.
* Check whether the user is scanning the appropriate QR of the building, and displaying the data with audio feedback.
* If the user is between he should direct one QR code of one of those building, next direct him to towards another building.
* In offline mode (in case data is not available), still the user would able to get over view of the campus and points of interest.

## Assumptions

This section lists assumptions that are made specific to this project.

1. Delivery of the product is in format that the test team can check it into CVS.

## Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Risk | Impact | Trigger | Mitigation Plan |
| 1 | Scope Creep – as testers become more familiar with the tool, they will want more functionality | High | Delays in implementation date | Each iteration, functionality will be closely monitored. Priorities will be set and discussed by stakeholders. Since the driver is functionality and not time, it may be necessary to push the date out. |
| 2 | Changes to the functionality may negate the tests already written and we may lose test cases already written | High – to schedule and quality | Loss of all test cases | Export data prior to any upgrade, massage as necessary and re-import after upgrade. |
| 3 | Weekly delivery is not possible because the developer works off site | Medium | Product did not get delivered on schedule |  |
| 4 | Team Overloaded with development | High | Less time to test the code | Project schedule should give adequate time for testing and not to develop at fast pace. |

# **Test Approach**

The project is using an agile approach, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

Exploratory testing will play a large part of the testing as the team has never used this type of tool and will be learning as they go.

## Test Automation

No scenarios for testing automation have been identified

# **Test Environment**

For testing the software developed we would need

* 1. Android Studio Integrated developing environment.
  2. GenyMotion Emulator software
  3. BugZilla software to log defects and test results
  4. Internet connection to work stations for developed application to work.

# **Milestones / Deliverables**

## Test Schedule

The initial test schedule follows……….

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Effort** | **Comments** |
| Home Screen |  |  | 3 D |  |
| Virtual Tour UI |  |  | 4 D |  |
| Physical Tour |  |  | 7 D |  |
| Super Admin User Interface |  |  | 1 D |  |
| University Admin User Interface |  |  | 2 D |  |
| Voice Assist In Campus Tour |  |  | 2 D |  |
| Campus Tour Functionality |  |  | 5 D |  |
| Virtual Tour Functionality |  |  | 2 D |  |
| QR Code Generation and Functionality |  |  | 4 D |  |
| Database Connectivity |  |  | 2 D |  |

## Deliverables

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **For** | **Date / Milestone** |
| Test Plan | Project Manager; QA Director; Test Team |  |
| Traceability Matrix | Project Manager; QA Director |  |
| Test Results | Project Manager |  |
| Test Status report | QA Manager, QA Director |  |
|  |  |  |
| Metrics | All team members |  |
|  |  |  |