**Test Plan-Test Specification**

**Version 1.3**

**Project Management App**

**Team A**

**CSC-355**

**Spring 2016**



02/24/2016

Author: Jennifer Li – System Tester

Submitted To: Dr. Joo Tan

**TABLE OF CONTENTS**

Revision History…………………………………………………………………………………..ii

1. Introduction……..….....…………………………….…...…………………………….……....1
2. Features to Be Tested………………………………………………………………………….1
   1. Intermediate Release…………………………………………………………………..1
   2. Final Release…………………………………………………………………………..2
   3. Scenarios………………………………………………………………………………2
3. Test Approach & Design ………………………...…………………………………………...2
   1. Unit Testing…………………………………………………………………………...3
   2. System Testing………………………………………………………………………...4
   3. Scenario Testing……………………………………………………………………….4
   4. Regression Testing…………………………………………………………………….5
   5. Reliability Testing……………………………………………………………………..6
   6. Usability Testing………………………………………………………………………6
   7. Scalability Testing…………………………………………………………………….6
4. Software Defect States………………..……………………………………………………….6
5. Software Defect Severity…………….………………………………………………………..6
6. System Test Configuration……………………………………………………………………7
7. Test Responsibilities…………………………………………………………………………..7
8. Test Entrance Criteria…………………………………………………………………………7
   1. Developers…………………………………………………………………………….7
   2. Testers………………………………………………………………………………....7
9. Test Schedule………………………………………………………………………………….8

10.0 Risks and Contingencies……………………………………………………………………..8

11.0 Features Test Cases…………………………………..…………………………………..9-24

12.0 Scenario Test Cases………………………………….……………………………...…..25-27

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Description | Date |
| 1.0 | Jennifer Li | I created the first draft. | 1/31/2016 |
| 1.1 | Jennifer Li | Added Section 1 and 2 | 2/20/2016 |
|  | Jennifer Li | Added Section 3 and 6 | 2/21/2016 |
| 1.2 | Jennifer Li | Added Section 4, 5, 7, 8, and 9.  Also made edits on section 2.1, 2.2. and 2.  Added the Testing Case in section 11 and 12. | 2/22/2016 |
|  | Jennifer Li | Added Section 10 and completed all of the Testing cases and scenario cases. | 2/23/2016 |
| 1.3 | Tyler Mariano | Updated test scenarios to match the app screen flow. | 2/26/2016 |

**1.0 INTRODUCTION**

This document presents the test and specification plan on how and what, the Project Management App will be tested. This document will outline, test procedures formulated to validate the Project Management App. This document will describe the test strategy, the test system, and provide an estimation of testing effort. It will also be providing the system test configurations and the risk and contingencies of testing the app.

**2.0 FEATURES TO BE TESTED**

The following features will be tested to ensure that the Project Management App satisfies all functionality. The following functionality will be included and tested before the determined release dates.

**2.1 Intermediate Release**

The following features that will be tested by the intermediate release; which will be within the first week of March.

* RS 1.0 (F.R.1.0) – Verify that a user can create an account
* RS 2.0 (F.R.2.0) – Verify that a user can login using a valid username and password
* RS 3.0 (F.R.3.0) – Verify that a user can create a project
* RS 4.0 (F.R.4.0) – Verify that leader can add members
* RS 5.0 (F.R.6.0) – Verify that leader can assign task
* RS 6.0 (F.R.7.0) – Verify that member can submit completed task
* RS 7.0 (F.R.8.0) – Verify that leader can review submitted task
* RS 8.0 (F.R.9.0) – Verify that leader can view overall progress
* RS 8.1 (F.R.9.1) – Verify that member can view overall progress
* RS 9.0 (F.R.12.0) – Verify that member can accept project invitations
* RS 10.0 (F.R.13.0) – Verify that leader can view due dates
* RS 10.1 (F.R.13.1) – Verify that member can view due dates
* RS 11.0 (F.R.16.0) – Verify that leader can Logout
* RS 11.1 (F.R.16.1) – Verify that member can Logout
* RS.12.0 (F.R.19.0) – Verify that leader can send invites to members
* RS 13.0 (F.R.20.0) – Verify that leader can view all members

**2.2 Final Release**

The following features that will be tested by the final release; which will be within the last week of April.

* RS 14.0 (F.R.5.0) – Verify that leader can remove members
* RS 15.0 (F.R.10.0) – Verify that leader can views members’ task progress
* RS 16.0 (F.R.11.0) – Verify that member can view their own task progress
* RS 15.0 (F.R.14.0) – Verify that leader can message members
* RS 15.1 (F.R.14.0) – Verify that member can message leader and other members
* RS 16.0 (F.R.15.0) – Verify that leader can set reminders
* RS 17.0 (F.R.17.0) – Verify that leader can upload documents
* RS 17.1 (F.R.17.1) – Verify that member can upload documents
* RS 18.0 (F.R.18.0) – Verify that leader can create custom roles
* RS 19.0 (NF.R.1.0) – Verify Reliability
* RS 20.0 (NF.R.2.0) – Verify Usability
* RS 21.0 (NF.R.3.0) – Verify Scalability

**2.3 Scenarios**

The system will also be tested using various real-to-life scenarios. This will verify workflow and coordination of functions:

The following scenario test will be conducted during the intermediate release date:

* RS 22.0 : F.R.3.0, F.R.4.0, F.R.19.0, F.R.12.0 – Verify leader and member interaction of, Create a project, Add members, Send invites to members, and Accept project invitations.
* RS 23.0 : F.R.6.0, F.R.7.0, F.R.8.0 – Verify leader and member interaction of, Assign task, Submit task, and Review task.
* RS 24.0 : F.R.2.0, F.R.9.0, F.R.13.0, F.R.16.0 – Verify leader and member can Login, View overall progress, View due dates, Logout

The following scenario test will be conducted during the final release date:

* RS 25.0 : F.R.20.0, F.R.5.0, F.R.10.0 – Verify leader can, View all members, Remove members, and Leader views members’ task progress.
* RS 26.0 : F.R.12.0, F.R.7.0, F.R.11.0 – Verify member can, Accept project invitations, Submit task, and View members’ own task progress.

**3.0 TEST APPROACH & DESIGN**

The overall testing approach includes unit testing, system testing, scenario testing, regression testing, reliability testing, usability testing, and scalability testing. Each type of testing is described in more detail in the following sub-sections.

**3.1 Unit Testing**

All featured describe in section 2 must be tested by the developers before the release dates. The functional requirements will be individually and independently scrutinized for proper operation.

**Table 3.1: Unit Testing Design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement  Definition | System  Test Case ID | Test Input | Test Configuration | Test Objective |
| RS 1.0 | ST 1.0 | Test Plan 1.2 | TC 1.0 | TO 1.0 - Verify that a user can create an account |
| RS 2.0 | ST 2.0 | Test Plan 1.2 | TC 1.0 | TO 2.0 – Verify that a user can login using a valid username and password |
| RS 3.0 | ST 3.0 | Test Plan 1.2 | TC 1.0 | TO 3.0 – Verify that a user can create a project |
| RS 4.0 | ST 4.0 | Test Plan 1.2 | TC 1.0 | TO 4.0 – Verify that leader can add members |
| RS 5.0 | ST 5.0 | Test Plan 1.2 | TC 1.0 | TO 5.0 – Verify that leader can assign task |
| RS 6.0 | ST 6.0 | Test Plan 1.2 | TC 1.0 | TO 6.0 – Verify that member can submit completed task |
| RS 7.0 | ST 7.0 | Test Plan 1.2 | TC 1.0 | TO 7.0 – Verify that leader can review submitted task |
| RS 8.0 | ST 8.0 | Test Plan 1.2 | TC 1.0 | TO 8.0 – Verify that leader can view overall progress |
| RS 8.1 | ST 8.1 | Test Plan 1.2 | TC 1.0 | TO 8.1 – Verify that member can view overall progress |
| RS 9.0 | ST 9.0 | Test Plan 1.2 | TC 1.0 | TO 9.0 - Verify that member can accept project invitations |
| RS 10.0 | ST 10.0 | Test Plan 1.2 | TC 1.0 | TO 10.0 – Verify that leader can view due dates |
| RS 10.1 | ST 10.1 | Test Plan 1.2 | TC 1.0 | TO 10.1 – Verify that member can view due dates |
| RS 11.0 | ST 11.0 | Test Plan 1.2 | TC 1.0 | TO 11.0 – Verify that leader can Logout |
| RS 11.1 | ST 11.1 | Test Plan 1.2 | TC 1.0 | TO 11.1 – Verify that member can Logout |
| RS 12.0 | ST 12.0 | Test Plan 1.2 | TC 1.0 | TO 12.0 – Verify that leader can send invites to members |
| RS 13.0 | ST 13.0 | Test Plan 1.2 | TC 1.0 | TO 13.0 – Verify that leader can view all members |

**3.2 System Testing**

System testing will be done by the tester(s) before the release dates. This test will be testing the software and hardware. It must be done on a completed, integrated system to evaluate the compliance with all the described functional and non-functional requirements in the Software Requirements Specification document.

**3.3 Scenario Testing**

The system will be tested using a verity of real life scenarios. This testing will verify the work, screen and coordination flow of all mutually dependent functions.

**Table 3.3: Scenario Test Design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement  Definition | System  Test Case ID | Test Input | Test Configuration | Test Objective |
| RS 22.0 | ST 22.0 | Test Plan 1.2 | TC 1.0 | TO 22.0 – Verify leader and member interaction of, Create a project, Add members, Send invites to members, and Accept project invitations. |
| RS 23.0 | ST 23.0 | Test Plan 1.2 | TC 1.0 | TO 23.0 - Verify leader and member interaction of, Assign, task, Submit task, and Review task. |
| RS 24.0 | ST 24.0 | Test Plan 1.2 | TC 1.0 | TO 24.0 – Verify leader and member can Login, View overall progress, View due dates, Logout |

**3.4 Regression Testing**

Regression testing will ensure that bug fixes in the test cycle will not break the previously working functionality during the prior release dates.

* Bugs are to be fixed as they are found. For each software build that is released to the test team, tests will be run to verify that the bugs fixed in that build do not reoccur.
* Once testing of functionally is complete and the app is up and running, there will be one final regression pass before system testing. For the system to pass regression testing, it will require that 100% of major bugs, at least 90% of moderate bugs, and at least 25% of minor bugs be fixed.

**3.5 Reliability Testing**

Reliability testing will be done before the final release of this app. This test is in place to ensure that the application is not down for more than 1 minute per day when the app is fully deployed.

**3.6 Usability Testing**

Usability testing will be done before the final release of this app. This test will ensure that users must be able to grasp the general use of the app under 5 minutes. To determine this app’s usability a survey of 10 people will be conducted.

**3.7 Scalability Testing**

Scalability testing will be done before the final release of this app. This test will encompass the increasing trend of smartphone and tablet use in the modern workforce. When the app is fully functional it must be able to run on a tablet.

1. **SOFTWARE DEFECT STATES**

Software defect states are used for the purpose of tracking the status of all defects, defect states are defined. The five defect states are defined as follows:

* **New:** When a defect has not been acknowledged or addressed by developers.
* **Active:** A defect that is assigned to a developer or team of developers; and is currently being analyzed and worked on.
* **Fixed:** When developer or team of developers makes necessary code changes and verifies the changes. Then the defect will be ready for retesting.
* **Resolved:** Once the defect is fixed, it is tested by the tester. If the tester feels that the defect no longer exists in the software, the status of the defect is changed to “resolved”.
* **Reactive:** If the defect still exists even after the defect is fixed by the developer, the tester changes the status to “reactive”. The defect goes through the cycle once again.

1. **SOFTWARE DEFECT SEVERITY**

Software defect severity are used to classify the level of all defects, defect severities are defined. The three defect severities are defined as follows:

* **Critical:** Defect affects critical functionality or critical data
* Severe**:** Defect affects major functionality or major data
* **Minor:**Defect affects minor functionality or non-critical data
* **Enhancement:** Defect does not affect functionality or data

1. **SYSTEM TEST CONFIGURATION**

For the testing process, we will be using our phones and the nexus tablet, provided by Kutztown University’s Computer Science Department. This testing will take place in 157, one of the Computer Science computer laps in Old Main. The Project Management App is designed to run on android devices. It was created using Google’s Android Java language and shared on GitHub. The test product will be downloaded from android studio via micro USB cable. The test configuration specifies both the hardware and software requirements that the implemented system will be test on. The test configuration to be used is shown in tables 3.1 and 3.3.

* TC1 – Android Version 5.1.1, Nexus 7
* TC2 – Android Version 5.1.1, Samsung Galaxy Note 4
* TC3 – Android Version 5.1.1, Samsung Galaxy 4

1. **TEST RESPONSIBILITIES**

Both the Development and Test Teams have responsibilities for ensuring the quality of the intended product.

Development Team responsibilities include:

* Unit Test Features as they are developed
* Prepare the application for intermediate release and final release
* Fix bugs submitted via MR Report

Test Team responsibilities include:

* Run planned tests and create MR Report for developers
* Classify the defects for the MR Report on defects found

1. **TEST ENTRANCE CRITERIA**

There must be a set standard before the tester will accept the system from the development team. The following are the entry criteria that must be met before the system is accepted.

**8.1 Developers**

Developers must complete unit testing and have fixed all assigned defects by the tester. They must also prepare the system, whether if installing or configuring, the system must be up and running for the tester to start testing.

**8.2 Testers**

For testers to accept the system, all of the critical and major defects found in the project management app must be fixed and unit and integration testing must be completed.

1. **TEST SCHEDULE**

Unit and integration testing is expected to be done by the development team, prior to the start of Testing Cycle #1. Testing Cycle #1 will be done before intermediate release. Testing Cycle #2 will be done before the final release date. All major bugs found in that testing will be fixed prior to the start of system testing. Table 9.1 provides an estimated schedule of the start and the end dates for tasks during each cycle.

**Table 9.1: Estimate Testing Schedule**

|  |  |  |
| --- | --- | --- |
| Task | Start Date | End Date |
| Testing Cycle #1 | 2/27 | 3/01 |
| Testing Cycle #2 | 4/15 | 4/20 |

1. **RISK AND CONTINGENCIES**

The following Table 10.1 outlines possible risks and contingency plans affecting the testing of the Project Management App system:

**Table 10.1: Risks and Contingencies**

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Likelihood | Impact | Mitigation |
| The room that Team A would be using to conduct testing is not available at the testing time. | 25% | Major | The testing will be done using the computer science lab in Old Main. If this room is not available, the backup room would be in one of the professor’s office. |
| Test cases are not ready by the time of testing. | 10% | Moderate | Testing of functionality should be performed once the test cases have been finalized. |
| Test Plan and Test Specification document is not completed on time | 10% | Minor | The first version of the Test Plan and Test Specification document must be completed by 2/24/26. |
| Test cases in Test Plan and Test Specification do not correspond to System Interface delivered | 50% | Moderate | Test Cases may need to be modified to the System Interface that is delivered. |
| People scheduled to test and or take the test survey are not available to complete the desired test | 25% | Moderate | If the survey tester are not available during the scheduled test date; rescheduling the date is a must. Since this is a mobile application, the testing could be done remotely. |
| Due to the releases during the Spring semester, weather may cause an issue which may force a delay in the schedule | 25% | Moderate | If the members of team A are not available during the day of testing due to weather; rescheduling the test day will have to be made. |

**11.0 FEATURE TEST CASES**

The following teat cases will be used for functional requirement testing of the Project Management App. (test cases subject to change)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 1.0 | | | | |
| Owner of Test |  | | | | |
| Test Name | Create an Account | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 1.0 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies |  | | | | |
| Test Objective | Verify that a user can create an account | | | | |
| Test Procedure | | | | | |
| Test Setup | Open the application (for the first time) | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Enter “jli725@live.kutztown.edu” in email field | | | **N/A** | | |
| 2. Enter “Jennifer” in the password field | | | **N/A** | | |
| 3. Click the Register Button | | | Call screen appears | | |
|  | | | Direct users to main menu screen | | |
|  | | |  | | |
| Test Data | Email: “jli725@live.kutztown.edu” | | | | |
|  | Password : “Jennifer” | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 2.0 | | | | |
| Owner of Test |  | | | | |
| Test Name | Login | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 2.0 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies |  | | | | |
| Test Objective | Verify that a user can login using a valid username and password | | | | |
| Test Procedure | | | | | |
| Test Setup | Open the application | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Enter “jli725@live.kutztown.edu” in email field | | | 1. N/A | | |
| 2. Enter “Jennifer” in the password field | | | 2. N/A | | |
| 3. Click the “Login” button | | | 3. Call screen | | |
|  | | | 4. User is directed to the Main Menu | | |
|  | | |  | | |
|  | | |  | | |
| Test Data | Email: “jli725@live.kutztown.edu” | | | | |
|  | Password: “Jennifer” | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 3.0 | | | | |
| Owner of Test |  | | | | |
| Test Name | Create a Project | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 3.0 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies | RS 4.0 | | | | |
| Test Objective | Verify that a user can create a project | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged in and on the main menu screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “ + ” button | | | 1. Directed to the create a project screen | | |
| 2. Enter “Flappy Bird” in the project name field | | | 2. N/A | | |
| 3. Enter “Game” in to the project description field | | | 3. N/A | | |
| 4. Click the “Create” button to create a project | | | 5. Directs Leader to the Leader view screen | | |
|  | | |  | | |
| Test Data | Project Name: “Flappy Bird” | | | | |
|  | Project Description: “Game” | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 4.0 | | | | |
| Owner of Test |  | | | | |
| Test Name | Add Members | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 4.0 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies | RS 3.0, RS 9.0, RS 12.0 | | | | |
| Test Objective | Verify that leader can add members | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged in and in the member view screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Members” button | | | 1. Directed to the Member List screen | | |
| 2. Click the “+” button | | | 2. Directed to the Add Member screen | | |
| 3. Click the name of member “Steven” | | | 3. N/A | | |
| 4. Click the “add” button to complete adding | | | 4. An invitation is sent to the checked members and the app directs the leader to the Leader View | | |
|  | | |  | | |
|  | | |  | | |
| Test Data |  | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 5.0 | | | | |
| Owner of Test |  | | | | |
| Test Name | Assign Task | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 5.0 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies |  | | | | |
| Test Objective | Verify that leader can assign task | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged in and in the main menu screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Task” button | | | 1. Directed to the Task List screen | | |
| 2. Click the “+” button | | | 2. Directed to the Add Task screen | | |
| 3. Enter the task name “ ” | | | 3. N/A | | |
| 4. Enter the task description “ “ | | | 4. N/A | | |
| 4. Click the name of member “Steven” from combo box | | | 4. N/A | | |
| 5. Click the “Create” button | | | 5. Directed to the Leader View screen | | |
|  | | |  | | |
| Test Data |  | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 6.0 | | | | |
| Owner of Test |  | | | | |
| Test Name | Submit Task | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 6.0 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies |  | | | | |
| Test Objective | Verify that member can submit completed task | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged in and in the member view screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “ Task ” button | | | 1. Directed to the Task List screen | | |
| 2. Click the task “ “ | | | 2. Directed to the Task View screen | | |
| 3. Click the “Submit” button | | | 3. Directs member back to the member view | | |
|  | | |  | | |
|  | | |  | | |
| Test Data |  | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 7.0 | | | | |
| Owner of Test |  | | | | |
| Test Name | Submit Task | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 7.0 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies |  | | | | |
| Test Objective | Verify that leader can review submitted task | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged in and in the main menu screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “ Task ” button | | | 1. Directed to the Task List screen | | |
| 2. Click the green task “ “ | | | 2. Directed to the Task View screen | | |
| 3. Click the “Accept” button | | | 3. Directs member back to the leader view | | |
| 3. Click the “Deny” button | | | 3. Prompts leader to reassign the task then directs member back to the leader view | | |
|  | | |  | | |
| Test Data |  | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 8.0 | | | | |
| Owner of Test |  | | | | |
| Test Name | View Overall Progress Leader | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 8.0 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies |  | | | | |
| Test Objective | Verify that leader can view overall progress | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged in and in the main menu screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Overall Progress ” button | | | 1. Directed to the Overall Progress screen | | |
| 2. Hit the back button to go back to the main menu | | | 2. Directs leader back to the menu screen | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
| Test Data |  | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 8.1 | | | | |
| Owner of Test |  | | | | |
| Test Name | View Overall Progress Member | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 8.1 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies |  | | | | |
| Test Objective | Verify that member can view overall progress | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged in and in the main menu screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Overall Progress ” button | | | 1. Directed to the Overall Progress screen | | |
| 2. Hit the back button to go back to the main menu | | | 2. Directs member back to the menu screen | | |
|  | | |  | | |
|  | | |  | | |
| Test Data |  | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 9.0 | | | | |
| Owner of Test |  | | | | |
| Test Name | Accept Project Invites | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 9.0 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies | RS 3.0, RS 4.0, RS 12.0 | | | | |
| Test Objective | Verify that member can accept project invitations | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged in. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Accept ” button when prompt | | | 1. N/A | | |
| 2. Hit the app to be directed to the project list | | | 2. Directs member to the project list screen | | |
|  | | |  | | |
|  | | |  | | |
| Test Data |  | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 10.0 | | | | |
| Owner of Test |  | | | | |
| Test Name | View Due Dates Leader | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 10.0 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies |  | | | | |
| Test Objective | Verify that leader can view due dates | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged in and at the main menu screen. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Calendar ” button | | | 1. Directs leader to calendar screen | | |
| 2. Hit the back button to go back to the main menu | | | 2. Directs leader back to the menu screen | | |
|  | | |  | | |
|  | | |  | | |
| Test Data |  | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 10.1 | | | | |
| Owner of Test |  | | | | |
| Test Name | View Due Dates Member | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 10.1 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies |  | | | | |
| Test Objective | Verify that member can view due dates | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged in and at the main menu screen. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Calendar ” button | | | 1. Directs member to calendar screen | | |
| 2. Hit the back button to go back to the main menu | | | 2. Directs member back to the menu screen | | |
|  | | |  | | |
|  | | |  | | |
| Test Data |  | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 11.0 | | | | |
| Owner of Test |  | | | | |
| Test Name | Logout Leader | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 11.0 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies |  | | | | |
| Test Objective | Verify that leader can Logout | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged in and at the main menu screen. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Logout ” button | | | 1. Logs the user out and directed to Login screen | | |
|  | | |  | | |
|  | | |  | | |
| Test Data |  | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 11.1 | | | | |
| Owner of Test |  | | | | |
| Test Name | Logout Member | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 11.1 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies |  | | | | |
| Test Objective | Verify that member can Logout | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged in and at the main menu screen. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Logout ” button | | | 1. Logs the user out and directed to Login screen | | |
|  | | |  | | |
|  | | |  | | |
| Test Data |  | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 12.0 | | | | |
| Owner of Test |  | | | | |
| Test Name | Send Invites | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 12.0 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies | RS 3.0, RS 4.0, RS 9.0 | | | | |
| Test Objective | Verify that leader can send invites to members | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged in and is in the process of creating a project. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the name of member “Steven” | | | 1. N/A – request is sent to member | | |
| 2. Click the “Done” button to complete adding | | | 2. Directs leader back to create a project | | |
|  | | |  | | |
|  | | |  | | |
| Test Data |  | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 13.0 | | | | |
| Owner of Test |  | | | | |
| Test Name | View All Members | | | | |
| Test location | Project Management App | | | | |
| Date of Last Revision |  | | | | |
| Requirement Tested | RS 13.0 | | | | |
| Test Configuration | TC 1.0 | | | | |
| Test Interdependencies | RS 4.0 | | | | |
| Test Objective | Verify that leader can view all members | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Members” button | | | 1. Directs leader to the member list screen | | |
| 2. Hit the back button to go back to the main menu | | | 2. Directs leader back to the menu screen | | |
|  | | |  | | |
|  | | |  | | |
| Test Data |  | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

**12.0 SCENARIO TEST CASES**

The following test cases will be used for scenario testing for the Project Management App.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Information | | | | |
| Test Case ID | ST 22.0 | | | |
| Owner of Test |  | | | |
| Test Name | Create an Project, Add Members, Send invites and Accept infatuations | | | |
| Test location | Project Management App | | | |
| Date of Last Revision |  | | | |
| Requirement Tested | RS 22.0 | | | |
| Test Configuration | TC 1.0 | | | |
| Test Interdependencies |  | | | |
| Test Objective | Verify leader and member interaction of, Create a project, Add members, Send invites to members, and Accept project invitations. | | | |
| Test Procedure | | | | |
| Test Setup | Assumed that leader and member are both logged in | | | |
|  | | | | |
| Step Action | | **Expected Result** | | **Pass( )** |
| 1. Leader clicks “Create a Project” and enter information | | |  | |
| 2. Leader then clicks “Add Members” button which will send invites to the corresponding member. | | |  | |
| 3. Member clicks accept project invitation. | | |  | |
| Test Data |  | | | |
| Test Result | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** |
| Notes: |  | | | |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Information | | | | |
| Test Case ID | ST 23.0 | | | |
| Owner of Test |  | | | |
| Test Name | Assign, Submit, and Review Task | | | |
| Test location | Project Management App | | | |
| Date of Last Revision |  | | | |
| Requirement Tested | RS 23.0 | | | |
| Test Configuration | TC 1.0 | | | |
| Test Interdependencies |  | | | |
| Test Objective | Verify leader and member interaction of, Assign, task, Submit task, and Review task. | | | |
| Test Procedure | | | | |
| Test Setup | Assumed that leader and member is logged in | | | |
|  | | | | |
| Step Action | | **Expected Result** | | **Pass( )** |
| 1. Leader clicks “Assign Task” button | | | 7. Member clicks the task to turn in | |
| 2. Leader clicks a task and a member of the project | | | 8. Leader clicks “Review Task” button | |
| 3. Leader clicks enter | | | 9. Leader will determine Pass or Reassign | |
| 4. Member clicks “Task” button | | |  | |
| 5. Member perform the task | | |  | |
| 6. Member clicks “Submit Task” button | | |  | |
| Test Data |  | | | |
| Test Result | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** |
| Notes: |  | | | |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Information | | | | |
| Test Case ID | ST 24.0 | | | |
| Owner of Test |  | | | |
| Test Name | Login, Views, and Logout | | | |
| Test location | Project Management App | | | |
| Date of Last Revision |  | | | |
| Requirement Tested | RS 24.0 | | | |
| Test Configuration | TC 1.0 | | | |
| Test Interdependencies |  | | | |
| Test Objective | Verify leader and member can Login, View overall progress, View due dates, Logout | | | |
| Test Procedure | | | | |
| Test Setup |  | | | |
|  | | | | |
| Step Action | | **Expected Result** | | **Pass( )** |
| 1. Login in as a leader or a member | | |  | |
| 2. Click the “View Overall Progress” button | | |  | |
| 3. View it, Then click the back key | | |  | |
| 4. Click the “View Due Dates” button | | |  | |
| 5. View it, Then click the back key | | |  | |
| 6. Then click the “Logout” button | | |  | |
| Test Data |  | | | |
| Test Result | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** |
| Notes: |  | | | |
|  |