**Test Plan-Test Specification**

**Version 2.3**

**Project Management App**

**Team A**

**CSC-355**

**Spring 2016**



03/21/2016

Authors: Jennifer Li – Project Leader & System Tester, Tyler Mariano – Database & Interface Designer

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 ………………………...…………………………………………...3
   1. Unit Testing…………………………………………………………………………3-4
   2. System Testing………………………………………………………………………...5
   3. Scenario Testing…………………………………………………………………….5-6
   4. Regression Testing…………………………………………………………………….6
   5. Reliability Testing……………………………………………………………………..6
   6. Usability Testing………………………………………………………………………7
   7. Scalability Testing…………………………………………………………………….7
4. Software Defect States………………..……………………………………………………….7
5. Software Defect Severity…………….………………………………………………………..7
6. System Test Configuration……………………………………………………………………8
7. Test Responsibilities…………………………………………………………………………..8
8. Test Entrance Criteria…………………………………………………………………………8
   1. Developers…………………………………………………………………………….8
   2. Testers………………………………………………………………………………....9
9. Test Schedule………………………………………………………………………………….9

10.0 Risks and Contingencies……………………………………………………………………10

11.0 Features Test Cases…………………………………..…………………………………11-29

12.0 Scenario Test Cases………………………………….……………………………...…..30-34

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Description | Date |
| 1.0 | Jennifer Li | I created the first draft. | 01/31/2016 |
| 1.1 | Jennifer Li | Added Section 1 and 2 | 02/20/2016 |
|  | Jennifer Li | Added Section 3 and 6 | 02/21/2016 |
| 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. | 02/22/2016 |
|  | Jennifer Li | Added Section 10 and completed all of the Testing cases and scenario cases. | 02/23/2016 |
| 1.3 | Tyler Mariano | Updated every test case to match the app screen flow. | 02/26/2016 - 02/28/2016 |
| 2.0 | Jennifer Li | Made grammatical changes on all of the sections. Also updated and removed F.R.12 from the first release to the second release date. | 02/28/2016 |
| 2.1 | Tyler Mariano | Add changes to all test cases base on the notes from the team meeting held yesterday. | 03/01/2016 - 03/02/2016 |
|  | Tyler Mariano | Fixed ST6.0 and ST23.0 to use the new progress slide bar. | 03/03/2016 |
| 2.2 | Jennifer Li | Fixed the list and tables in Section 2.1, 2.2, 2.3, 3.1, 3.3, 11.0. Also edit test case 13-16 | 03/16/2016-3/27/2016 |
|  | Jennifer Li | Removed F.R 17 from section 2.2, 3.1 and 11.0 | 03/30/2016 |
| 2.3 | Tyler Mariano | Add the edit suggested by Dr. Tan in section 2.2, 7.0, 8.0, 9.0, 10.0 and 11.0. | 4/12/2016 |

**1.0 INTRODUCTION**

This document presents the Test Plan and Test Specification on how and what the Project Management App will be tested. The purpose of testing is to ensure that the delivered app is at an appropriate quality. Thus Quality Assurance must be taken into consideration. Quality assurance was used to prevent mistakes or defects in our app. This helped avoid problems further down the road by setting certain coding standards and commenting practices at the beginning. The document will also be outlining test procedures that are formulated to validate the Project Management App. In addition, 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 risks and contingencies of testing the app.

**2.0 FEATURES TO BE TESTED**

The following features that were agreed upon with the client will be tested to ensure that the Project Management App satisfies all functionality. The list of functionality will be divided into two releases and tested before the determined release dates.

**2.1 Intermediate Release**

The following features will be tested by the intermediate release. The intermediate release date will on March 4th of 2016.

* F.R.1.0 – Verify that a user can create an account
* F.R.2.0 – Verify that a user can login using a valid username and password
* F.R.3.0 – Verify that a user can create a project
* F.R.4.0 – Verify that leader can add members by sending an invitation
* F.R.6.0 – Verify that leader can assign task
* F.R.7.0 – Verify that member can submit completed task
* F.R.8.0 – Verify that leader can review submitted task
* F.R.9.0 – Verify that leader can view overall progress
* F.R.13.0 – Verify that leader can view calendar
* F.R.16.0 – Verify that leader can Logout
* F.R.19.0 – Verify that leader can send invites to members
* F.R.20.0 – Verify that leader can view all members

**2.2 Final Release**

The following features will be tested by the final release, which is currently targeted to be on April 27th of 2016.

* F.R.12.0 – Verify that member can accept project invitations
* F.R.5.0 – Verify that leader can remove members
* F.R.10.0 – Verify that leader can views members’ task progress
* F.R.11.0 – Verify that member can view their own task progress
* F.R.14.0 – Verify that leader can message members
* F.R.15.0 – Verify that leader can set reminders
* F.R.18.0 – Verify that member can choose a role
* Reliability – Verify Reliability
* Usability – Verify Usability
* Scalability – Verify Scalability

**2.3 Scenarios**

The system will also be tested using various real life scenarios. These hypothetical situations will help the tester test the complex mutually dependent functions and how they correlate to each other. This will verify workflow and coordination of functions:

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

* Scenario 1: F.R.3.0, F.R.4.0, F.R.19.0 – Verify leader and member interaction of Create a project, Add members, and Send invites to members.
* Scenario 2: 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.
* Scenario 3: 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 additional scenario test will be conducted during accepting testing on the final release date:

* Scenario 4: F.R.20.0, F.R.5.0, F.R.10.0 – Verify that the leader can: view all members, remove members, and view all members’ task progress.
* Scenario 5: F.R.12.0, F.R.7.0, F.R.11.0 – Verify that the member can: accept project invitations, submit tasks, and view their 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, user interface testing, and scalability testing. Each type of testing is described in more detail in the following sub-sections.

**3.1 Unit Testing**

All featured that are described in section 2.0 must be tested by the developers during system development. The functional requirements will be individually and independently scrutinized for proper operation. Refer to the tables.

**Table 3.1: Unit Testing Design for Phase 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement  Definition | System  Test Case ID | Test Input | Test Configuration | Test Objective |
| F.R 1.0 | ST 1.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 1.0 - Verify that a user can create an account |
| F.R 2.0 | ST 2.0 | Test and Specification Plan 2.2 | TC 1.0, 20 | TO 2.0 – Verify that a user can login using a valid username and password |
| F.R 3.0 | ST 3.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 3.0 – Verify that a user can create a project |
| F.R 4.0 | ST 4.0 | Test and Specification Plan 2.2 | TC 1.0, 2,0 | TO 4.0 – Verify that leader can add members |
| F.R 6.0 | ST 5.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 5.0 – Verify that leader can assign task |
| F.R 7.0 | ST 6.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 6.0 – Verify that member can submit completed task |
| F.R 8.0 | ST 7.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 7.0 – Verify that leader can review submitted task |
| F.R 9.0 | ST 8.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 8.0 – Verify that leader can view overall progress |
| F.R 13.0 | ST 9.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 9.0 – Verify that leader can view calendar |
| F.R 16.0 | ST 10.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 10.0 – Verify that leader can Logout |
| F.R 19.0 | ST 11.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 11.0 – Verify that leader can send invites to members |
| F.R 20.0 | ST 12.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 12.0 – Verify that leader can view all members |

**Table 3.2: Unit Testing Design for Phase 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement  Definition | System  Test Case ID | Test Input | Test Configuration | Test Objective |
| F.R 12.0 | ST 13.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 13.0 - Verify that Member can accept project invitations via email |
| F.R 5.0 | ST 14.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 14.0 - Verify that leader can remove members |
| F.R 10 | ST 15.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 15.0 - Verify that leader can views members’ task progress |
| F.R 11 | ST 16.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 16.0 - Verify that member can view their own task progress |
| F.R 14 | ST 17.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 17.0 - Verify that leader can message members |
| F.R 15 | ST 18.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 18.0 - Verify that leader can set reminders |
| F.R 18 | ST 20.0 | Test and Specification Plan 2.2 | TC 1.0, 2.0 | TO 20.0 - Verify that member can choose a role |

**3.2 System Testing**

System testing will be done by the white box tester Tyler Mariano and black box tester Jennifer Li, before the each release date. The first release date will be on March 4th of 2016 and the second release date is scheduled for April 27th of 2016. This test will be testing the functionality of the software using white and black box testing. White box testing will be performed by Tyler Mariano to look for any user interface errors. Whereas, black box testing will be performed by Jennifer Li to look for usability problems and bugs that crash the system. Any defects found will be added to the current version of the Modification Request report. Each defect will be tracked by status and classification which will be explained in more detail in section 4.0 and 5.0.

System testing must be performed on a complete, integrated for the system to be 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 variety of real life scenarios. This testing will verify the screen flow and coordination of all mutually dependent functions. Refer to the table 3.3.1 & 3.3.2 for more detail.

**Table 3.3.1: Scenario Test Design Phase 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement  Definition | System  Test Case ID | Test Input | Test Configuration | Test Objective |
| Scenario 1 | ST 22.0 | Test and Specification Plan 2.0 | TC 1.0, 3.0 | TO 21.0 – Verify leader and member interaction of Create a project, Add members, and Send invites to members. |
| Scenario 2 | ST 23.0 | Test and Specification Plan 2.0 | TC 1.0, 3.0 | TO 22.0 - Verify leader and member interaction of Assign, task, Submit task, and Review task. |
| Scenario 3 | ST 24.0 | Test and Specification Plan 2.0 | TC 1.0, 3.0 | TO 23.0 – Verify leader and member can Login, View overall progress, View due dates, Logout. |

**Table 3.3.2: Scenario Test Design Phase 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement  Definition | System  Test Case ID | Test Input | Test Configuration | Test Objective |
| Scenario 4 | ST 24.0 | Test and Specification Plan 2.0 | TC 1.0, 3.0 | TO 24.0 – Verify leader can View all members, Remove members, and Leader Views members’ task progress. |
| Scenario 5 | ST 25.0 | Test and Specification Plan 2.0 | TC 1.0, 3.0 | TO 25.0 - Verify member can Accept project invitations, Submit task, and View members’ own task progress. |

**3.4 Regression Testing**

Regression testing will help ensure that defect fixes in the test cycle will not break the previously working functionality during the prior release dates. Regression testing will be done by white and black box testers Tyler Mariano and Jennifer Li. This testing will be conducted after each sprint.

Defects and Completion:

* Defects 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 defects 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 severity 1 defects, at least 90% of severity 2 defects, and at least 50% of severity 3 defects 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’s server is not down for more than 1 minute per day when the app is fully deployed. This test is used to satisfy the non-functional requirements necessary for this project; it is used to help improve the overall quality of the app.

**3.6 Usability Testing**

Usability testing will be done conducted after each sprint and 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 15 minutes. To determine this app’s usability a survey of 5-10 people will be conducted. This test is used to satisfy the non-functional requirements necessary for this project; it is used to help improve the usability of the app.

**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. Scalability testing is used to support the increasing use of technology in the workforce. Also, the non-functional requirements necessary for this project will be satisfied by this test.

1. **SOFTWARE DEFECT STATES**

Software defect states are used for the purpose of tracking the status of all defects found during development and testing. The cycle starts at state New, when the defect is found and ends when the defect states have been changed to Resolved. If the defect has not been resolved by the end of the cycle, the defect will then restart the cycle with the state of Reactive. 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:** A defect that is reassigned to a developer or team of developers; and will then restart the cycle as if it was in the Active state again.

1. **SOFTWARE DEFECT SEVERITY**

Defect severity or Impact is a classification of software defect to indicate the degree of negative impact on the quality of software. The classification of severities for this project are defined as follows:

* Severity 1 - A critical error has occurred; the system cannot continue functioning
* Severity 2 - A major functionality does not work; this prevents other functionality from being tested
* Severity 3 - A minor functionality does not work; this is considered a minor defect
* Severity 4 - This is considered an enhancement type of defect that does not affect any functionality

1. **SYSTEM TEST CONFIGURATION**

For testing purposes, team A will be using their own phones as well as the nexus tablet which is provided by Kutztown University’s Computer Science Department. This testing will take place in room 157, one of the Computer Science computer labs 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 hardware and software requirements that will be needed for the implemented system to be tested. The test configuration that will be used is listed here for reference.

* TC 1.0 – Android Version 5.1.1, Nexus 7
* TC 2.0 – Android Version 5.1.1, Samsung Galaxy Note 4
* TC 3.0 – Android Version 5.1.1, Samsung Galaxy s4

1. **TEST RESPONSIBILITIES**

Both the Development and Test Teams have responsibilities for ensuring the quality of the intended product. Development Team will be comprised of developers Steven Grantz and Hector Richiez. Test Team will be comprised of white box tester Tyler Mariano and black box tester Jennifer Li.

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
* Preform integration testing

Test Team responsibilities include:

* Run planned tests and create MR Report for developers
* Classify the defects for the MR Report on defects found
* Preform regression testing
* Preform system testing
* Preform UI testing
* Perform scenario testing

1. **TEST ENTRANCE CRITERIA**

A standard will be established before the test team will accept the system from the development team. The entry criteria that must be met before the system is accepted is specified in the following sub-sections.

**8.1 Developers**

Developers must have completed unit testing and have fixed all defects assigned by the tester as described in the MR report. They must also prepare the system, whether installing or configuring, and the system must be up and running for the tester to start testing.

**8.2 Testers**

For testers to accept the system, all severity 1 and 2 defects found in the project management app that is described in the MR report, 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 Phase 1 System Testing Cycle. This test cycle will be done before intermediate release on March 4th of 2016. System testing will start on 2/29 to 3/03.

Phase 2 System Testing Cycle will be done before the final release date, April 27th of 2016. Phase 2’s system test cycle will be broken down into three different sprints. Each sprint has its own functional or non-functional requirements to be tested. 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 |
| Phase 1 System Testing Cycle | 2/29 | 3/03 |
|  | F.R. 1,2,3,4,6,7,8,9,13,16,19,20 | Unit testing must be completed by the developers and the start of System testing by the system tester. |
| Phase 2 System Testing Cycle | 4/1 | 4/30 |
|  | Sprint 1  Starts: 3/14  Ends: 3/31 | After the first half of sprint 1, unit testing of F.R. 5, 12, and 18 must be complete and given to the system tester.  After the second half of sprint 1, unit testing of F.R. 10, 11, and 14 must be completed and given to the system tester. |
|  | Sprint 2  Starts: 4/1  Ends: 4/17 | After first half of sprint 2 unit testing of F.R. 15 must be completed and given to the system tester.  The second half of sprint 2 will be used to test the Reliability, Usability, and Scalability of the system |
|  | Sprint 3  Starts: 4/18  Ends: 4/30 | In the last sprint, stress and system testing will be performed until the system results are acceptable |

**10.0 RISK AND CONTINGENCIES**

Table 10.1 outlines possible risks and contingency plans which may affect 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 and updated by \_\_\_\_\_\_\_\_. |
| Test cases in Test Plan and Test Specification do not correspond to User Interface delivered | 50% | Moderate | Test Cases may need to be modified to the User Interface that is delivered. |
| People scheduled to test 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. |
| Not tracking defects found during any of the testing phases. | 25% | Moderate | All defects will be document and posted on GitHub so the system tester to view and update the MR report. |
| Not having the proper hardware with the correct environment for testing. | 50% | Major | Three of the four members of team A have android devices that can be used for testing as described in section 6.0 |
| Not having a plan to test the agreed upon functional requirements. | 10% | Minor | The purpose of this document is to plan for testing of the Project Management App. The testing schedule is given in section 9.0. |

**11.0 FEATURE TEST CASES**

The test cases that are listed in this section will be used for functional requirement testing of the Project Management App. Details of each test case will be shown in separate tables. (all test cases subject to change)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 1.0 | | | | |
| Owner of Test | Jennifer and Tyler | | | | |
| Test Name | Create an Account | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | F.R 1.0 | | | | |
| Test Configuration | TC 1.0 , 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| 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. Click the “Register” Button | | | 1. Directs the User to the Create Account page | | |
| 2. Enter “jli724@live.kutztown.edu” in email field | | | 2. N/A | | |
| 3. Enter “Jennifer” in the password field | | | 3. N/A | | |
| 4. Click the “Create Account” button | | | 4. Direct users to Main Menu screen | | |
|  | | |  | | |
| Test Data | Email: “jli724@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 | Jennifer and Tyler | | | | |
| Test Name | Login | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | F.R 2.0 | | | | |
| Test Configuration | TC 1.0, 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| 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 “jli724@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: “jli724@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 | Jennifer and Tyler | | | | |
| Test Name | Create a Project | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | F.R 3.0 | | | | |
| Test Configuration | TC 1.0 , 2.0 | | | | |
| Test Interdependencies | F.R 4.0 | | | | |
| Test Objective | Verify that a user can create a project | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged into Jennifer’s account and is the Main Menu screen. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “ + ” button | | | 1. Directs to the Create 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 | | | 4. 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 | Jennifer and Tyler | | | | |
| Test Name | Add Members | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | F.R 4.0 | | | | |
| Test Configuration | TC 1.0, 2.0 | | | | |
| Test Interdependencies | F.R 3.0, F.R 13.0, F.R 20.0 | | | | |
| Test Objective | Verify that leader can add members | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged into Jennifer’s account and is in the Leader View screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Members” button | | | 1. Directs to the Member List screen | | |
| 2. Click the “+” button | | | 2. Directs to the Add Member screen which is a list of all members | | |
| 3. Click the email of member “sgant869@live.kutztown.edu” | | | 3. Opens Android mail client selector to choose an email client that allows the Leader to email the user an invitation | | |
| 4. Click “Send” | | | 4. Member will be added to the project pending their acceptance | | |
|  | | |  | | |
| Test Data | N/A | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 5.0 | | | | |
| Owner of Test | Jennifer and Tyler | | | | |
| Test Name | Assign Task | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | F.R 6.0 | | | | |
| Test Configuration | TC 1.0, 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify that leader can assign task | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged into Steven’s account and is in the Leader View screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Task” button | | | 1. Directs to the Task List screen | | |
| 2. Click the “+” button | | | 2. Directs to the Add Task screen | | |
| 3. Enter the task name “Say your Name” | | | 3. N/A | | |
| 4. Enter the task description “ Name” | | | 4. N/A | | |
| 5. Click the email “sgant869@live.kutztown.edu” from combo box | | | 5. Combo box label says sgant869@live.kutztown.edu | | |
| 6. Click the “Create Task” button | | | 6. Directs to the Leader View screen | | |
|  | | |  | | |
| Test Data | Task Name: “Say your Name” | | | | |
|  | Task Description: “Name” | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 6.0 | | | | |
| Owner of Test | Jennifer and Tyler | | | | |
| Test Name | Submit Task | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | F.R 7.0 | | | | |
| Test Configuration | TC 1.0, 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify that member can submit completed task | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged into Steven’s account and is in the Leader View screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the task ID under the task section of the screen. | | | 1. Directs to the Task View screen | | |
| 2. Slide the progress slider bar to 100 | | | 2. The Update button changes to a Submit button | | |
| 3. Click the “Submit” button | | | 3. Directs to the Task View screen | | |
|  | | |  | | |
|  | | |  | | |
| Test Data | N/A | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 7.0 | | | | |
| Owner of Test | Jennifer and Tyler | | | | |
| Test Name | Review Task | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | F.R 8.0 | | | | |
| Test Configuration | TC 1.0, 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify that leader can review submitted task | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged into Steven’s account and is in the Leader View screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the ID of the task under the completed task section of the screen. | | | 1. Directs to the Task View screen | | |
| 2. Slide the progress bar back to half way. | | | 2. N/A | | |
| 3. Click the “Update” button to deny the task | | | 3. Directs back to the Task View screen | | |
|  | | |  | | |
|  | | |  | | |
| Test Data | N/A | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 8.0 | | | | |
| Owner of Test | Jennifer and Tyler | | | | |
| Test Name | View Overall Progress Leader | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | F.R 9.0 | | | | |
| Test Configuration | TC 1.0, 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify that leader can view overall project progress | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged into Steven’s account and is in the Leader View screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Progress” button | | | 1. Directs to the project Progress screen | | |
| 2. Hit the back button to go back to the Leader View screen | | | 2. Directs leader back to the Leader View screen | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
| Test Data | N/A | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 9.0 | | | | |
| Owner of Test | Jennifer and Tyler | | | | |
| Test Name | View Calendar Leader | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | F.R 13.0 | | | | |
| Test Configuration | TC 1.0, 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify that leader can view calendar | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged into Steven’s account and is in the Leader View screen. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Calendar ” button | | | 1. Directs leader to calendar screen, view it | | |
| 2. Hit the back button to go back to the Leader View screen | | | 2. Directs leader back to the Leader View screen | | |
|  | | |  | | |
|  | | |  | | |
| Test Data | N/A | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 10.0 | | | | |
| Owner of Test | Jennifer and Tyler | | | | |
| Test Name | Logout Leader | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | F.R 16.0 | | | | |
| Test Configuration | TC 1.0, 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify that leader can Logout | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged into Jennifer’s account and is in the Leader View screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the menu button | | | 1. Opens the drop down menu | | |
| 2. Click the “Logout ” menu item | | | 2. Logs the user out and directed to Login screen | | |
|  | | |  | | |
| Test Data | N/A | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 11.0 | | | | |
| Owner of Test | Jennifer and Tyler | | | | |
| Test Name | Send Invites | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | F.R 19.0 | | | | |
| Test Configuration | TC 1.0, 2.0 | | | | |
| Test Interdependencies | F.R 3.0, F.R 4.0, F.R 13.0 | | | | |
| Test Objective | Verify that Leader can send invites to members | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged into Jennifer’s account and is in the Leader View screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “+” button | | | 1. Directs to the Add Member screen | | |
| 2. Click the email of member “tmari795@live.kutztown.edu” | | | 2. Opens Android mail client selector to choose an email client that allows the Leader to email the user an invitation | | |
| 3. Click “Send” | | | 3. Invitation will be sent to the specified user to join the project | | |
|  | | |  | | |
|  | | |  | | |
| Test Data | N/A | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 12.0 | | | | |
| Owner of Test | Jennifer and Tyler | | | | |
| Test Name | View All Members | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | F.R 20.0 | | | | |
| Test Configuration | TC 1.0, 2.0 | | | | |
| Test Interdependencies | F.R 4.0 | | | | |
| Test Objective | Verify that Leader can view all members | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged into Steven’s account and is in the Leader View screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Enter “sgant869@live.kutztown.edu” in email field | | | 1. N/A | | |
| 2. Enter “Steven” in the password field | | | 2. N/A | | |
| 3. Click the “Login” button | | | 3. Call screen | | |
| 4. N/A | | | 4. User is directed to the Main Menu | | |
| 5. Click the ID of the Pre-created project | | | 5. Directs leader to the Leader View screen | | |
| 6. Click the “Members” button | | | 6. Directs leader to the Member List screen | | |
| 7. Hit the back button to go back to the Leader View screen | | | 7. Directs leader back to the Leader View screen | | |
|  | | |  | | |
| Test Data | Email: “sgant869@live.kutztown.edu”, “tmbeast38@gmail.com” | | | | |
|  | Password: “Steven”, “Tyler” | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

**------------------Final Release Test Cases--------------------**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 13.0 | | | | |
| Owner of Test | Jennifer | | | | |
| Test Name | Accept Project Invites | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/19/16 | | | | |
| Requirement Tested | F.R 12.0 | | | | |
| Test Configuration | TC 1.0 , 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify that Member can accept project invitations via email | | | | |
| Test Procedure | | | | | |
| Test Setup | Assume that Jennifer has been invited to a project and has open the appropriate email. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the link in the email | | | 1. Directs to the web server and will display a “1” to signify that the request went through. | | |
| 2. Leave the web server, reopen the app and loin. | | | 2. Directs to the Main Menu screen. | | |
| 3. Select the project ID | | | 3. Directs to the Member screen. | | |
| Test Data | N/A | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 14.0 | | | | |
| Owner of Test | Jennifer | | | | |
| Test Name | Remove Members | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/19/16 | | | | |
| Requirement Tested | F.R 5.0 | | | | |
| Test Configuration | TC 1.0 , 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify that leader can remove members | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed that Leader is logged into Steven’s account and is in the Leader View screen of a project. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Leader then clicks the “Members” button via the Leader View screen | | | 1. Directs to the Member List screen | | |
| 2. Click the email “jli724@live.kutztown.edu” and hold that email until the delete user box appears. | | | 2. Pop up box appears and ask if leader wants to delete the member. | | |
| 3. Hit the “YES” button | | | 3. Directs to the Leader View screen | | |
| 4. Hit the back key and choose the project that the member was deleted from. | | | 4. Directs to Main Menu and Leader View screen | | |
|  | | |  | | |
| Test Data | N/A | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 15.0 | | | | |
| Owner of Test | Jennifer | | | | |
| Test Name | View Members’ Task Progress | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/19/16 | | | | |
| Requirement Tested | F.R 10.0 | | | | |
| Test Configuration | TC 1.0 , 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify that leader can views members’ task progress | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed that Leader is logged into Steven’s account and is in the Leader View screen of a project. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Members” button | | | 1 Directs to the Member List screen | | |
| 2.1 Hit “jli724@live.kutztown.edu”  2.2 Hit the back button to go back to the Member List screen | | | 2.1 Directs Members’ progress screen  2.2 Directs leader back to the Member List screen | | |
|  | | |  | | |
| Test Data | N/A | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 16.0 | | | | |
| Owner of Test | Jennifer | | | | |
| Test Name | Member View Own Task Progress | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/19/16 | | | | |
| Requirement Tested | F.R 11.0 | | | | |
| Test Configuration | TC 1.0 , 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify that member can view their own task progress | | | | |
| Test Procedure | | | | | |
| Test Setup | Assume that Jennifer is at the Member View screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Members” button | | | 1 Directs to the Member List screen | | |
| 2.1 Hit “jli724@live.kutztown.edu”  2.2 Hit the back button to go back to the Member List screen | | | 2.1 Directs Members’ progress screen  2.2 Directs member back to the Member List screen | | |
|  | | |  | | |
| Test Data | N/A | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 17.0 | | | | |
| Owner of Test | Jennifer | | | | |
| Test Name | Messaging | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/19/16 | | | | |
| Requirement Tested | F.R 14.0 | | | | |
| Test Configuration | TC 1.0 , 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify that leader can message members | | | | |
| Test Procedure | | | | | |
| Test Setup | Assume that Steven is at the Leader View screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Message” button | | | 1. Directs to the inbox screen | | |
| 2. Click “jli724@live.kutztown.edu” as the recipient | | | 2. Directs to the message screen | | |
| 3. Type “Hello” in the message box and click the “Send” button | | | 3. Adds user message to the message table | | |
|  | | |  | | |
| Test Data | Message Box Text: “Hello” | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 18.0 | | | | |
| Owner of Test | Jennifer | | | | |
| Test Name | Set Reminders | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/19/16 | | | | |
| Requirement Tested | F.R 15.0 | | | | |
| Test Configuration | TC 1.0 , 2.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify that leader can set reminders | | | | |
| Test Procedure | | | | | |
| Test Setup | Assume that Steven is at the Leader View screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Click the “Calendar” button | | | 1. Directs to the Calendar screen | | |
| 2. Click April 27th and hit the “Set Reminder Button”  2.1 Fill in text box “Project is Due Today” then hit the “Create” button | | | 2. Directs to the create reminder screen  2.1 Directs to the Calendar screen | | |
|  | | |  | | |
| Test Data | Text Box: “Project is Due Today”, “New Message” | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 20.0 | | | | |
| Owner of Test | Jennifer | | | | |
| Test Name | Create Custom Roles | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/19/16 | | | | |
| Requirement Tested | F.R 18.0 | | | | |
| Test Configuration | TC 1.0 , 2.0 | | | | |
| Test Interdependencies | F.R 5.0 | | | | |
| Test Objective | Verify that member can choose a role | | | | |
| Test Procedure | | | | | |
| Test Setup | Assume that Steven is in the process of creating an account | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Enter “Steven” in the first name field and Enter “Gantz” for the last name field. | | | 1. N/A – Already at the Profile screen | | |
| 2. select the role of member: “Programmer” | | | 2. N/A | | |
| 3. Click the “Save” Button | | | 3. Directs to the Member View screen | | |
| Test Data | First Name: “Steven”, “Tyler” Last Name: “Gantz”, “Mariano” | | | | |
|  | Role: “Programmer”, “Designer” | | | | |
| 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.

(all test cases subject to change)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 21.0 | | | | |
| Owner of Test | Jennifer and Tyler | | | | |
| Test Name | Create an Project, Add Members, and Send invites | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | Scenario 1 | | | | |
| Test Configuration | TC 1.0, 3.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify Leader and Member interaction of, Create Project, Add Members, Send invites to Members, and Accept project invitations. | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed user is logged into Steven’s account and is in the Leader View screen | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Leader clicks “+” on the Main Menu screen and enters “Angry Birds” for Project Name and enter “Game” for the project description and then clicks “Create”. | | | 1. Directs to the create a project screen.  1.2. N/A  1.3. N/A  1.4. 5. Directs Leader to the Leader View screen | | |
| 2. Leader then clicks the “Members” button via the Leader View screen and then clicks the “+” button. | | | 2. Directs to the Member List screen  2.1. Directs to the Add Member screen | | |
| 3. Click the email of member “jli724@live.kutztown.edu” | | | 3. Opens Android mail client selector to choose an email client that allows the Leader to email the user an invitation | | |
| 4. Click “Send” | | | 4. Invitation will be sent to the specified user to join the project | | |
|  | | |  | | |
| Test Data | Project Name: “Angry Birds” | | | | |
|  | Project Description: “Game” | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 22.0 | | | | |
| Owner of Test | Jennifer and Tyler | | | | |
| Test Name | Assign, Submit, and Review Task | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | Scenario 2 | | | | |
| Test Configuration | TC 1.0, 3.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify Leader and Member interaction of Assign Task, Submit Task, and Review Task. | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed that Leader is logged into Steven’s account and is in the Leader View screen of a project. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Leader clicks “+” button via Task View screen | | | 1. Directs to the Add Task screen | | |
| 2. Leader inputs a task name: “Age”, task description: “The age of the Leader” and selects member “sgant869@live.kutztown.edu” form the combo box | | | 2. N/A | | |
| 3. Leader clicks “Create Task” button | | | 3. Directs to the Task List screen | | |
| 4. Select the ID of the task from the Tasks section of the screen | | | 4. Directs to Task View screen | | |
| 5. Performs the task | | | 5. Say your age | | |
| 6. Slide the progress bar to 100 and clicks “Submit” button | | | 6. Directs to the Task List screen | | |
| 7. Select the ID of the task under the completed task section of the screen | | | 7. Directs to the Task View screen | | |
| 8. Accept the completed task by not changing the progress bar. | | | 8. N/A | | |
| 9. Hit the back button on the phone to be directed to the Task List screen | | | 9. Directs to the Task List screen | | |
| Test Data | Task Name: “Age” Task Description: “The age of the member” | | | | |
|  |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 23.0 | | | | |
| Owner of Test | Jennifer and Tyler | | | | |
| Test Name | Login, Views, and Logout | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/03/16 | | | | |
| Requirement Tested | Scenario 3 | | | | |
| Test Configuration | TC 1.0, 3.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify Leader and Member can login, view overall project Progress, view Calendar, logout | | | | |
| Test Procedure | | | | | |
| Test Setup | N/A | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Login in as a leader or a member – Enter “jli724@live.kutztown.edu” and “Jennifer” in the email and password field. Then click “Login” button. And selects a project | | | 1. N/A  1.2. N/A  1.3. Call screen  1.4. User is directed to the Main Menu  1.5. User is directed to the Leader View or Member View | | |
| 2. Click the “Progress” button | | | 2. Directs to the project Progress screen | | |
| 3. View it, Then click the back key | | | 3. Directs Leader back to the Leader View | | |
| 4. Click the “Calendar” button | | | 4. Directs Leader to Calendar screen | | |
| 5. View it, Then click the back key | | | 5. Directs Leader back to the Leader View screen | | |
| 6. Click the menu button | | | 6. Opens the drop down menu | | |
| 7. Then click the “Logout” menu item | | | 7. Logs the user out and directed to Login screen | | |
| Test Data | Email: “jli724@live.kutztown.edu” Password: “Jennifer” | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 24.0 | | | | |
| Owner of Test | Jennifer | | | | |
| Test Name | View, Remove, and View Members’ Task Progress | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/19/16 | | | | |
| Requirement Tested | Scenario 4 | | | | |
| Test Configuration | TC 1.0, 3.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify leader can, View all members, Remove members, and Leader Views members’ task progress. | | | | |
| Test Procedure | | | | | |
| Test Setup | Assumed that Leader is logged into Steven’s account and is in the Leader View screen of a project. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Leader then clicks the “Members” button via the Leader View screen | | | 1. Directs to the Member List screen | | |
| 2. Click the email “jli724@live.kutztown.edu” and then hit the “Remove Button” and hit the “Refresh button” | | | 2. Removes member and then directs to the Member List screen | | |
| 3.1 Hit the back button  3.2 Click the “Progress” button | | | 3.1 Directs to the Leader View screen  3.2 Directs to the project Progress screen | | |
| 4.1 Hit the “Member Progress” button  4.2 Hit the back button to go back to the Leader View screen | | | 4.1 Directs leader to the Member’s progress screen  4.2 Directs leader back to the Leader View | | |
| Test Data | N/A | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Information | | | | | |
| Test Case ID | ST 25.0 | | | | |
| Owner of Test | Jennifer | | | | |
| Test Name | Accept Invitation, Submit Task, View Member’s Own Task Progress | | | | |
| Test location | Old Main 157 | | | | |
| Date of Last Revision | 3/19/16 | | | | |
| Requirement Tested | Scenario 5 | | | | |
| Test Configuration | TC 1.0, 3.0 | | | | |
| Test Interdependencies | N/A | | | | |
| Test Objective | Verify member can, Accept project invitations, Submit task, and View members’ own task progress. | | | | |
| Test Procedure | | | | | |
| Test Setup | Assume Jennifer has been invited to a project and has been assigned a task once accepted. | | | | |
|  | | | | | |
| Step Action | | **Expected Result** | | | **Pass( )** |
| 1. Open email client, and hit the accept project link | | | 1. Directed to the Member View or Member View | | |
| 2. Click the “Task” button | | | 2. Directs to the project Task screen | | |
| 3.1 Select the ID of the task from the Tasks section of the screen  3.2 Performs the task  3.3 Slide the progress bar to 100 and clicks “Submit” button  3.4 Then click the back key | | | 3.1 Directs to Task View screen  3.2 Say your age  3.3 Directs to the Task List screen  3.4 Directs Member back to the Member View | | |
| 4. Click the “Progress” button | | | 4. Directs to the project Progress screen | | |
| 5.1 Hit the “Member Progress” button  5.2 Hit the back button to go back to the Leader View screen | | | 5.1 Directs Member to the Member’s progress screen  5.2 Directs Member back to the Member View | | |
| Test Data |  | | | | |
| Test Result | | | | | |
| Tester: | **Date of Test:** | | | **Test Result (P/F/B):** | |
| Notes: |  | | | | |
|  |