**Software Requirements Specification**

**Version 3.1**

**Project Management App**

**Team A**

**CSC-355**

**Spring 2016**



2/04/2016

Author: Jennifer Li – Project Leader, System Analyst, and Tester

Submitted To: Dr. Joo Tan

**TABLE OF CONTENTS**

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

1.0 Introduction…………...…………………………….…...……………………….………….….............1

* 1. Product Perspective……………………………………………………………………………1
  2. Purpose………………………………………………………………………………...............1
  3. Scope of Project………………………………………………………………………………..1
  4. Definitions, Acronyms, and Abbreviations….……………………………………………...…1

2.0 Project Description……………………………………………………………………………………...2

2.1 General User Overview……………………………………………………………………...…3

2.2 System Overview……………………………………………………………………................3

2.3 Overview of Non-Functional Requirements………………………………………………...…4

2.3.1 List of Non-Functional Requirements……………………………………………....4

2.4 Overview of Functional Requirements……………………………………………………...…5

3.0 Functional Requirements……………………………………………………………….……................6

3.1: Create User Account Use Case…………………………………………….….......…………..7 3.2: Login to Account Use Case………………………………………………...…........................8

3.3: Create Project Use Case……………………………………………………………………….9

3.4: Add Member Use Case.…………………………………………...........................................10

3.5: Remove Member Use Case…….……………………………….............................................11

3.6: Assign Tasks Use Case……………………………………....................................................12

3.7: Submit Tasks Use Case………………………………………................................................13

3.8: Review Tasks Use Case………………………………………..…………………...………..14

3.9: View Project Progress Use Case……………………………...…….......................................15

3.10: View Assigned Tasks Progress Use Case…………………….…………………………….16

3.11: View Member’s Tasks Progress Use Case……………………............................................17

3.12: Accept Project Invitation Use Case…………..……..…………...........................................18

3.13: View Due Date Use Case………………………………………….......................................19

3.14: Send Messages Use Case……………………………………….……………………..........20 3.15: Set Task Reminders Use Case………………………………………...…………………....21

3.16: Logout of Account Use Case………………………………….............................................22

3.17: Upload Documents Use Case…………………………………............................................23

3.18: Create Custom Roles Use Case…………………………………………..…...……………24

3.19: Send an Invitation to a Possible Member Use Case……….………….……………...….…25

3.20: View All Members Use Case……………………………………………………………….26

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Description | Date |
| 1.0 | Jennifer Li | I created the first draft. | 09/22/15 |
| 1.1 | Tyler Mariano | I made grammatical changes and cosmetic fixes. | 09/29/15 |
| 2.0 | Jennifer Li | Updated document based on teacher’s revisions.  Added two more use cases : “Upload Documents” and “Create Custom Roles”.  Added one new sub sections for Section 2 and added Section 3. | 10/10/15 – 10/14/15 |
| 2.1 | Tyler Mariano | I fixed minor grammatical errors like forgotten periods and identified run on sentences. | 10/14/15 |
| 2.2 | Hector Richiez | Added a functionality where the leader should be able to self-assigned a task as well a functional requirement that addresses the situation when a task is appropriated or abandoned that the leader has to reassigned. | 10/14/15 |
| 3.0 | Jennifer Li | Updated document based on teacher’s revisions: revised section 3.0, removed redundant words, and removed functional requirement 19. | 01/22/16 |
| 3.1 | Jennifer Li | Added functional requirement 19 and 20 to section 2.4. Created use cases in section 3.0 for functional requirements 19 and 20. Edited the wording with the functional and non-functional requirements in section 2.3 and 2.4. | 01/27/16 |
|  | Jennifer Li | Added three sentences describing intermediate and final delivery deadlines in section 2.3 and 2.4. | 02/03/16 |
|  | Tyler Mariano | Changed delivery column for F.R.13.0 from intermediate to final with Dr. Tan’s approval. | 02/29/16 |
|  | Jennifer Li | Edited F.R 18 in section 2.4 | 03/21/16 |
|  | Jennifer Li | Removed F.R 17 in section 2.4 and removed section 3.17 under section 3.0 | 03/30/16 |

**1.0 INTRODUCTION**

This document provides a detailed description of the Project Management App that will be developed. Within the document is a list of non-functional and functional requirements. In addition to that, this document will also provide use cases that will describe the interactions between the system and the user.

**1.1 Project Perspective**

After soliciting requirements from our client, Dr. Hussain, the Project Management App was requested to be mobile to help improve the overall management of projects. This mobile based Project Management App’s intended purpose is to create a seamless management level system that will save time, reduce the workload of the project leader, and organizes the overall project by the usage of tasks, charts, and dates.

**1.2 Purpose**

This SRS document will describe in detail about the requirements of the Project Management App. In this document, the reader will understand the purpose and the functionality of the Project Management App; by the description of components, the interactions between the user, app interfaces, and features.

**1.3 Scope of Project**

The Project Management App, will be compatible with mobile devices such as android phones and tablets. This mobile app interfaces with an SQL database, to provide the functional and non-functional requirements. In this app the user can create an account; with that account the user can create a project. As the leader of the project, the user can assign tasks, add members and/or remove members, review tasks, etc. As the member of the project, the user can accept project invitation, submit a task, view project progress, view their own progress, etc. With these functionalities, the leader can organize and prioritize the tasks in a centralized application.

**1.4 Definitions, Acronyms, and Abbreviations**

Terms and abbreviations that are used in this document, are described in the table below.

|  |  |
| --- | --- |
| Term | Definition |
| SRS | Software Requirements Specification |
| App | Application and/or System |
| User | Refers to either a Project Leader or Project Member |
| SQL | Structured Query Language |

**2.0 PROJECT DESCRIPTION**

This section, will provide an overview of the overall functionality of the Project Management App, shown in the table below.

|  |  |
| --- | --- |
| User Level | Available Functionality |
| Project Leader can: | * Create an account * Login to the created account * Create a project * Add new members * Remove members * Assign tasks to the members * Self-assigned a task * Submit tasks * Review each task * View the project’s overall progress * View each assigned task’s progress * View each member’s individual task progress * View the calendar dates associated with the project * Send messages to each and every member of the project * Set reminders * Logout of the account * Send invitations to the possible members * View a list of all members |
| Project Member can: | * Create an account * Login to the created account * Submit an assigned task * View the project’s overall progress * View their own progress of the assigned task * Join projects upon receiving an invite from the project leader * View their task’s due date via calendar view * Send messages to each member of the project * Logout of the account * View a list of all members * Choose a role when adding filling out their profile |

**2.1 General User Overview**

User’s computer skills will range from technological illiterate to technological savvy; therefore the app shall accommodate the weakest computer skill level. This will be accomplished by:

* Creating an app with a simple user interface.
* Clear and simple directions for all of the functionalities for both types of users.
* Using a uniform design for all the interfaces used by the user.

**2.2 System Overview Diagram**

This figure displays the system environment with the two actors and the interactive application. All actors can interact with the Project Management App.

Project Management App : Android Based Managing Application with DB

Project Management Application

Database

**2.3 Overview of Non-Functional Requirements**

The Project Management App has three persistent qualities and constraints that will be revised over and over again with each release of a new versions of the app. The non-functional requirements, specifies the criteria on the operational side of the system rather than the functional side. These qualities will measure how the system will operate.

**2.3.1 List of Non-Functional Requirements**

The Project Management App has three non-functional requirements that define the specifications of the quality of the system. The purpose of this is to identify the how the system operates under constraints. This app has reliability, usability, and scalability as its non-functional requirements. Reliability incorporates the precision of the app. Usability deals with the effort required to learn, provide and interpret the results of the app. Lastly, scalability is associated with the sheer number of users and the quality of the data processed in a timely manner.

The non-functional requirements of the Project Management App are listed below. Final delivery deadline is within the last week of April.

|  |  |  |
| --- | --- | --- |
| Requirement | Type | Delivery |
| The app must not be down for more than 1 min per day. | Reliability | Final |
| The user must be able to learn how to use the app under 5 minutes. With the use of a survey of 10 people, the team will be able to determine the apps usability. | Usability | Final |
| The app will be developed for mobile phones during the intermediate delivery; however, during final delivery the app will be able to support tablets that runs on the android system. | Scalability | Final |

**2.4 Overview of Functional Requirements**

The Project Management App has lot of functional requirements that define the specifications of the behavior of the functions. The purpose of this is to identify the possible interactions between the types of users, the Project Management App and its database.

The functional requirements of the Project Management App are listed below. Intermediate delivery deadline is within the first week of March. Final delivery deadline is within the last week of April.

|  |  |  |
| --- | --- | --- |
| Functional  Requirements | Requirement | Delivery |
| F.R.1.0 | User shall be able to create an account | Intermediate |
| F.R.2.0 | User shall be able to log-in. | Intermediate |
| F.R.3.0 | User shall be able to create a project. | Intermediate |
| F.R.4.0 | Project leader shall be able to add members to the created project. | Intermediate |
| F.R.5.0 | Project leader shall be able to remove members if the project member is terminated from the project. | Final |
| F.R.6.0 | Project leader shall be able to assign task to the project members. | Intermediate |
| F.R.7.0 | Project members shall be able to submit completed task for review by the project leader. | Intermediate |
| F.R.8.0 | Project leader shall be able to review the submitted task. | Intermediate |
| F.R.9.0 | User shall be able to view the project’s overall progression via graph. | Intermediate |
| F.R.10.0 | Project leader shall be able to view each project members’ task progress via graph. | Final |
| F.R.11.0 | Project members shall be able to view their assigned task progress via graph. | Final |
| F.R.12.0 | Members shall be able to accept project invitations from project leaders. | Final |
| F.R.13.0 | User shall be able to view calendar. | Intermediate |
| F.R.14.0 | User shall be able to message each user through the app. | Final |
| F.R.15.0 | Project leader shall be able to set reminders for all project members. | Final |
| F.R.16.0 | User shall be able to logout. | Intermediate |
| F.R.18.0 | Member shall be able to choose a role when he or she is creating or updating their profile. | Final |
| F.R.19.0 | Project leader shall be able to send invitations to the possible members upon adding members to the project. | Intermediate |
| F.R.20.0 | User shall be able to view a list of all members that are associated with the created project. | Intermediate |

**3.0 FUNCTIONAL REQUIREMENTS**

The functional requirements of the Project Management App specifies the functionality of the system. These functions are defined in use cases; which is a list of action or event steps, typically defining the interactions between an actor and the system. These use cases will contain a use case name, functional requirement number, scenario, trigger event, brief description, actors, stakeholders, pre and post conditions, flow of activities, and exception conditions.

The use case name and functional requirement number section, will be used to identify the specific use case. The scenario section, will describe a plausible case where the functionality may occur. The trigger event section, will describe an action that may cause the functionality to occur. The brief description section, will describe the general purpose of the use case. The actors section, will contain the external force that interacts with the system. The stakeholders section, will contain anyone who can affect or be affected by the system’s actions. The pre and post-conditions sections, will describe what must be done before the use case can occur and what happens after the use case occurs. In the flow of activities section, it will contain internal and external steps an actor or the system will take when the event is occurring. Lastly, the exception conditions section will contain forces that may cause the event to fail.

**3.1:** **Create User Account Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Create User Account |
| Functional Requirement Number: | F.R.1.0 |
| Scenario: | User creates an account on the app. |
| Triggered Event: | New user wants to set up an account in the app. |
| Brief Description: | The user creates an account by entering basic information such as, email, password, confirm password, phone number, biography and etc. |
| Actors: | User. |
| Stakeholders: | Project Leader or Project Member |
| Pre-Conditions: | Create an account subsystem must be available.  Email validation service must be available.  Phone validation service must be available. |
| Post-Conditions: | User account must be created and saved.  Email information must be validated.  Phone information must be validated.  Password must be saved. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. User indicates a desire to create a user account and enters the basic prompted information. | 1.1 System creates a new user.  1.2 System prompts for user email address. | | 2. User enters email address. | 2.1 System creates email address  2.2 System verifies email address.  2.3 System prompts for user phone number. | | 3. User enters phone number. | 3.1 System creates user account.  3.2 System verifies phone number.  3.3 System associates user, email address, phone, and account.  3.4 System return valid user account details. | |
| Exception Conditions: | 1. Basic user information is incomplete.  2. Invalid email address.  3. Invalid phone number.  4. Mixed matched password. |

**3.2 :** **Login to Account Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Login to Account |
| Functional Requirement Number: | F.R.2.0 |
| Scenario: | User logs into the app. |
| Triggered Event: | The user opens the app and navigates to the login interface. |
| Brief Description: | The user must enter their full email address as the username, and must enter the password to their account. Then user must submit the information. |
| Actors: | User. |
| Stakeholders: | Project Leader or Project Member |
| Pre-Conditions: | User must be created in the database.  User must have created a password.  Enter password must match the password in the database. |
| Post-Conditions: | User will gain access to the app and functionality. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. User enters email address, password, and submits the information. | 1.1 Database is search for the email address.  1.2 If found, database record for the user is retrieved.  1.3 Password from database record is compared to the entered password from the user.  1.4 If password matched, session is created. | |
| Exception Conditions: | 1. Invalid username match with the database.  2. Invalid password match with the database.  3. Use account information is not in the database. |

**3.3 :** **Create Project Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Create Project |
| Functional Requirement Number: | F.R.3.0 |
| Scenario: | User has logged in, they will be able to create a project. |
| Triggered Event: | User will need to manage and organize a project. |
| Brief Description: | The user will be able to create a project by providing information such as, name of project, due date, description, and members. |
| Actors: | User. |
| Stakeholders: | Project Leader. |
| Pre-Conditions: | A project that needs to be managed.  User must be logged in. |
| Post-Conditions: | The project that needs to be managed will be initialized.  The user will automatically become the project leader. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. User will enter the Name of Project, Due Date, Description, and members. | 1.1 A connection is made to the database.  1.2 All of the information is stored in to the database. | | 2. User clicks “Create Project” | 2.1 The system will periodically check the information stored into the database. | |
| Exception Conditions: | 1. Project information entered is incomplete. |

**3.4 :** **Add Member Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Add Member |
| Functional Requirement Number: | F.R.4.0 |
| Scenario: | The project leader wishes to add a member to the project. |
| Triggered Event: | The added user meets the project leader’s need for the certain position. |
| Brief Description: | The project leader will be able to add members to the project by searching the user by username. After reading the biography, if credentials are met the leader will select the user as a possible member to the project. |
| Actors: | Project Leader. |
| Stakeholders: | Project Leader and Project Members. |
| Pre-Conditions: | User must be logged in.  User must create a project.  User must be in the database of the app. |
| Post-Conditions: | A user will be prompted if they would like to be a member of this project. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project leader will click the “Add Member” button. | 1.1 A connection is made to the database. | | 2. Project leader will search through the results. | 2.1 The system will display all users of the app. | | 3. Project leader will choose a user to add as a member. | 3.1 The system will send a confirmation to the leader.  3.2 The system will prompt the chosen user an invitation. | |
| Exception Conditions: | 1. User not in the database. |

**3.5 :** **Remove Member Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Remove Member |
| Functional Requirement Number: | F.R.5.0 |
| Scenario: | The project leader wishes to remove a member from the project. |
| Triggered Event: | An event that causes a member of the project to not be able to complete the assigned task. Examples: illness, termination of contract, misconduct and or etc. |
| Brief Description: | The project leader will be able to functionally remove the Project Member form the project by resigning the assigned task to the member and deleting the user from the project. |
| Actors: | Project Leader. |
| Stakeholders: | Project Leader and Project Member. |
| Pre-Conditions: | Project Leader must be logged in.  Project Member must be a member of the project. |
| Post-Conditions: | A project member will be deleted from the project. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project leader will click the “Remove Member” button. | 1.1 The appropriate SQL statement will execute deletion from the specific project.  1.2 Confirmation will be display of the completion of the deletion of the ex-project member. | |
| Exception Conditions: | 1. User not a project member of the project. |

**3.6 :** **Assign Tasks Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Assign Tasks |
| Functional Requirement Number: | F.R.6.0 |
| Scenario: | The project leader wishes to assign tasks to a member of the project. |
| Triggered Event: | Created project have tasks that needs to be completed. |
| Brief Description: | The project leader will provide basic information about the task such as, description, dependency, due date, and etc. |
| Actors: | Project Leader. |
| Stakeholders: | Project Leader and Project Member. |
| Pre-Conditions: | Project Leader must be logged in.  Project has to be created.  Project must have member or members. |
| Post-Conditions: | Tasks will be assigned to a certain project member. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project leader will click the project member’s username and provide the basic information and or document that are need to complete this task. | 1.1 The system will connect to the database.  1.2 The system will store all the entered information to the database. | |
| Exception Conditions: | 1. Project task information entered is incomplete. |

**3.7 :** **Submit Tasks Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Submit Tasks |
| Functional Requirement Number: | F.R.7.0 |
| Scenario: | The project member has complete the task and wishes to submit to the project leader. |
| Triggered Event: | Project member has finished all of the requirements that were need to complete the task. |
| Brief Description: | The project member will provide all document that will be submitted for review and any necessary comments about the task. |
| Actors: | Project Member or Project Leader. |
| Stakeholders: | Project Member and Project Leader. |
| Pre-Conditions: | Project member must be logged in.  Project member must have been assigned a task.  The completion of task must be before the due date. |
| Post-Conditions: | The task, with all of the documents and comments will be reviewed. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project member or project leader will click the “Submit Task” button. | 1.1 The system will connect to the database. | | 2. Project member or project leader will upload all document and comments. | 2.1 The system will store all the entered information to the database.  2.2 The system will prompt the project leader about the task submission. | |
| Exception Conditions: | 1. Missed the due date. |

**3.8 :** **Review Tasks Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Review Tasks |
| Functional Requirement Number: | F.R.8.0 |
| Scenario: | The project leader has seen the prompted message and wishes to review the submitted task. |
| Triggered Event: | The project leader has received the completed documents and comments on or before the set due date. |
| Brief Description: | The project leader will review the submitted documents with comments. If the task is done to the project leader’s liking, progress will be updated for the entire project and the specific member. Also the project leader will unlock any dependent task that depends on the completion of the completed task. If the task is not completed to the Project leaders liking they can re-assign the task. |
| Actors: | Project Leader. |
| Stakeholders: | Project Leader and Project Member |
| Pre-Conditions: | Project leader must be logged in.  Task must be submitted.  Project leader must be prompted. |
| Post-Conditions: | Update in the overall project progress.  Update in the project member’s progress.  Or resigned if not completed. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project leader will click the “Review Task” button. | 1.1 The system will connect to the database. | | 2. Project leader will download all document and comments. | 2.1 The system will retrieve all the entered information for this task from the database. | | 3. If complete both progress will be updated; if not task will be re-assigned. | 3.1 Both progresses will be updated in the database.  3.2 –Refer to F.R.6.0 | |
| Exception Conditions: | 1. Must update the overall progress if completed properly.  2. Must update the project member’s progress if completed properly.  3. If not must re-assign task. |

**3.9 :** **View Project Progress Use Case**

|  |  |
| --- | --- |
| Use Case Name: | View Project Progress |
| Functional Requirement Number: | F.R.9.0 |
| Scenario: | Project leader or project member wishes to view the progress of the overall progress of the project. |
| Triggered Event: | Possible trigger would be the want of viewing the progression of the project after the project leader updates it. |
| Brief Description: | Project leader updates the task which will be represented in a Gantt chart. It will display the overall progress of the project by the percent of completion of project tasks. |
| Actors: | Project Member and Project Leader. |
| Stakeholders: | Project Member and Project Leader. |
| Pre-Conditions: | Both project member and project leader must be logged in.  Project must be created.  Project must have assigned task weather completed or not.  Project leader must update or updated it. |
| Post-Conditions: | The progress display in Gantt chart format. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project leader or project member will click the “View Project Progress” button. | 1.1 A connection made to the database.  1.2 The appropriate SQL statement will executed.  1.3 The results are displayed | |
| Exception Conditions: | 1. Faulty connection with database. |

**3.10 :** **View Assigned Tasks Progress Use Case**

|  |  |
| --- | --- |
| Use Case Name: | View Assigned Tasks Progress |
| Functional Requirement Number: | F.R.10.0 |
| Scenario: | Project leader wishes to view the progress of each task of the project. |
| Triggered Event: | The project leader needs to view the tasks’ progress. |
| Brief Description: | Project leader updates the task which will be represented in a Gantt chart. It will display the progress of each tasks’ percent of completion. |
| Actors: | Project Leader. |
| Stakeholders: | Project Leader. |
| Pre-Conditions: | Project leader must be logged in.  Project must be created.  Project must have assigned task weather completed or not.  Project leader must update or updated it. |
| Post-Conditions: | The progress display in Gantt chart format. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project leader will click the “View Assigned Tasks Progress” button. | 1.1 A connection made to the database.  1.2 The appropriate SQL statement will executed.  1.3 The results are displayed | |
| Exception Conditions: | 1. Faulty connection with database. |

**3.11 :** **View Member’s Tasks Progress Use Case**

|  |  |
| --- | --- |
| Use Case Name: | View Member’s Tasks Progress |
| Functional Requirement Number: | F.R.11.0 |
| Scenario: | Project member wishes to view their own progress of task they completed for the project. |
| Triggered Event: | The project member would like to view their own progress. |
| Brief Description: | After the project leader updates the task which will be represented in a Gantt chart. It will display the progress of a project member’s percent of completion of project tasks. |
| Actors: | Project Member. |
| Stakeholders: | Project Member. |
| Pre-Conditions: | Project member must be logged in.  Project must be created.  Project must have assigned task weather completed or not.  Project leader must update or updated it. |
| Post-Conditions: | The progress display in Gantt chart format. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project member will click the “View Member’s Progress” button. | 1.1 A connection made to the database.  1.2 The appropriate SQL statement will executed.  1.3 The results are displayed | |
| Exception Conditions: | 1. Faulty connection with database. |

**3.12 :** **Accept Project Invitation Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Accept Project Invitation |
| Functional Requirement Number: | F.R.12.0 |
| Scenario: | Users will receive prompted messaging asking if the user would like to join a Project. |
| Triggered Event: | Project leader adds a certain user to the project, which prompts a message of invitation to the project. |
| Brief Description: | One invitation is accepted the user will become a Project Member. Which lets the project leader assign task to the newly added project member. |
| Actors: | Project Member. |
| Stakeholders: | Project Member. |
| Pre-Conditions: | Project member must be logged in.  Project leader must have sent an invitation to the project. |
| Post-Conditions: | Acceptance to be a Project Member.  If denied, the user will stay a user. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project member will click the “Accept” button. Or the “Deny” button. | 1.1 If accepted connection made to the database to add the new member.  1.2 Confirmation message will be sent to the project leader.  1.3 If deny – system will not connect to the database. | |
| Exception Conditions: | 1. Faulty connection whether database, cell phone, and or internet.  2. Phone is out of power or turned off when message is sent. |

**3.13 :** **View Calendar Use Case**

|  |  |
| --- | --- |
| Use Case Name: | View Calendar |
| Functional Requirement Number: | F.R.13.0 |
| Scenario: | Project leader or project member wishes to view the upcoming due dates for the upcoming month or months. |
| Triggered Event: | Possible trigger, newly added members view upcoming due dates or add tasks to the overall project. |
| Brief Description: | This feature will contain a calendar view of all due dates. This can be viewed by all users involved in the project. |
| Actors: | Project Leader or Project Member. |
| Stakeholders: | Project Leader and Project Member. |
| Pre-Conditions: | Both project members and project leader must be logged in.  Project must be created.  Project must have assigned task with set due dates. |
| Post-Conditions: | Display of a calendar with all upcoming due dates that are incorporated with the project. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project leader or project member will click the “View Due Dates” button. | 1.1 A connection made to the database.  1.2 The appropriate SQL statement will executed.  1.3 The results are displayed | |
| Exception Conditions: | 1. Faulty connection with database. |

**3.14 : Send** **Messages Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Send Messages |
| Functional Requirement Number: | F.R.14.0 |
| Scenario: | Project leader and or project member wishes to contact the people involved in the project. |
| Triggered Event: | Possible trigger would be an issue with the assigned task or a simple question. |
| Brief Description: | The messaging functionality is for all users that are involved in the project. Generally used for questions, and or quick response to pressing matters. |
| Actors: | Project Leader or Project Member. |
| Stakeholders: | Project Leader and Project Member. |
| Pre-Conditions: | Both project members and project leader must be logged in.  Project leader and or project member must have a question about the project. |
| Post-Conditions: | Activation of chatting functionality.  Communication connection link set with sender and receiver.  Message sent to the receiver. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project leader and or project member will click the “Message” button. | 1.1 A connection made to the database.  1.2 connect to the appropriate receiver.  1.3 The message reminder will be sent. | |
| Exception Conditions: | 1. Faulty connection whether database, cell phone, and or internet.  2. Phone is out of power or turned off when message is sent. |

**3.15 : Set Task Reminders Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Set Task Reminders |
| Functional Requirement Number: | F.R.15.0 |
| Scenario: | Project leader and project members will receive message reminders of due dates. |
| Triggered Event: | Approaching due dates. |
| Brief Description: | Project leader will activate the reminder feature which sends notifications to both the project leader and the project members of the upcoming due dates. |
| Actors: | Project Leader. |
| Stakeholders: | Project Leader and Project Members |
| Pre-Conditions: | Project leader must be logged in.  Project must be created.  Project must have assigned task with set due dates. |
| Post-Conditions: | Prompted messages about upcoming due dates. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project leader will click the “Set Reminder” button. | 1.1 A connection made to the database.  1.2 The appropriate SQL statement will execute to retrieve all upcoming due dates.  1.3 The message reminder will be sent. | |
| Exception Conditions: | 1. Phone is out of power or turned off when message is sent. |

**3.16 :** **Logout of Account Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Logout of Account |
| Functional Requirement Number: | F.R.16.0 |
| Scenario: | User logs out of the app. |
| Triggered Event: | The user closes out of the app or session times out. |
| Brief Description: | The user will click the logout button or be inactive for 30 minutes. |
| Actors: | User. |
| Stakeholders: | Project Leader or Project Member. |
| Pre-Conditions: | User must be logged into the app. |
| Post-Conditions: | App will destroy the session. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. User click logout button. | 1.1 Session is destroyed. | | 2. User is inactive for 30 minutes. | 2.1 Session is destroyed. | |
| Exception Conditions: |  |

**3.18 :** **Choose a Role Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Choose a Role |
| Functional Requirement Number: | F.R.18.0 |
| Scenario: | The member wishes to choose a role when updating their profile. |
| Triggered Event: | The members wants a specific role for the project. |
| Brief Description: | The member will be able to choose a role when creating or updating their profile. |
| Actors: | Project Member. |
| Stakeholders: | Project Leader and Project Member. |
| Pre-Conditions: | User must be logged in.  Account must have been created.  User must be in the database of the app. |
| Post-Conditions: | A user will be prompted if they would like to be a member with the specific role of this project. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project leader will click the “Add Member” button. | 1.1 A connection is made to the database. | | 2. Project leader will search through the results. | 2.1 The system will display all users of the app. | | 3. Project leader will choose a user to add as a member with the created role. | 3.1 The system will send a confirmation to the leader.  3.2 The system will prompt the chosen user an invitation with the appropriate role that the project leader has created. | |
| Exception Conditions: | 1. User not in the database.  2. Project Member does not use the specific feature. |

**3.19 :** **Send an Invitation to a Possible Member Use Case**

|  |  |
| --- | --- |
| Use Case Name: | Send an Invitation to a Possible Member |
| Functional Requirement Number: | F.R.19.0 |
| Scenario: | The project leader wishes to send an invitation to a possible member, when adding a new member to the project. |
| Triggered Event: | The added user meets the project leader’s need for the certain position. |
| Brief Description: | After the project leader has clicked the “Add Member” button and received the sent invitation confirmation message, the system will send an invitation to the possible member. On the invitation the possible member can either accept or decline the invitation. |
| Actors: | Project Leader. |
| Stakeholders: | Project Leader and Project Member. |
| Pre-Conditions: | User must be logged in.  Project must have been created.  User must be in the database of the app. |
| Post-Conditions: | A user will be prompted if they would like to be a member of this project. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. After the project leader clicks the “Add Member” button and received the sent invitation confirmation message. | 1.1 An persistent connection is open to the database.  1.2 The system will prompt the chosen user an invitation to join the project. | | 2. Possible member will hit the “Accept” button to join the project. | 2.1 The system will correlate with the database to add the new member.  2.2 The system will send a confirmation to the leader. | | 3. Project leader will receive a confirmation of the added new member of the project. | 3.1 Database connections is closed. | |
| Exception Conditions: | 1. User not in the database.  2. Project leader does not receive a confirmation message . |

**3.20 :** **View All Members Use Case**

|  |  |
| --- | --- |
| Use Case Name: | View All Members |
| Functional Requirement Number: | F.R.20.0 |
| Scenario: | The project leader wishes to view all members that are associated with the project. |
| Triggered Event: | The project leader needs to assign a task. |
| Brief Description: | The project leader will be able to view all added members of the project. |
| Actors: | Project Leader or Project Member |
| Stakeholders: | Project Leader and Project Member. |
| Pre-Conditions: | Both project members and project leader must be logged in.  Project must be created.  Both project members and project leader must be in the database of the app. |
| Post-Conditions: | All members associated with the project will be displayed in a list view. |
| Flow of Activities: | |  |  | | --- | --- | | Actor | System | | 1. Project leader or project member will click the “View Members” button. | 1.1 A connection made to the database.  1.2 The appropriate SQL statement will executed.  1.3 The results are displayed | |
| Exception Conditions: | 1. Faulty connection to the database. |