**System Sequence Diagrams**

**Version 2.3**

**Project Management App**

**Team A**

**CSC-354**

**Fall 2015**



11/12/2015

Author: Tyler Mariano and Jennifer Li

Submitted To: Dr. Joo Tan

**TABLE OF CONTENTS**

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

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

1.1 What Is A System Sequence Diagram?………....……………………………….........1

1.2 System Sequence Diagram Notation………………………………………………….1

1.3 System Sequence Diagrams for Project Management App…………………….……..2

2.0 System Sequence Diagrams…………………………………………………………………...2

2.1 User Login…………………………………………………………………………….3

2.2 Leader: Add Member………………………………………………………………….4

2.3 Leader: Review Task………………………………………………………………….5

2.4 Member: View Task Progress…………………………………………………………6

2.5 Member: Upload Document…………………………………………………………..7

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Description | Date |
| 1.0 | Tyler Mariano | I created the first draft. | 10/22/2015 |
| 1.1 | Jennifer Li | I added the paragraph descriptions for the document. | 10/24/2015 |
| 1.2 | Tyler Mariano | I added the five diagrams. | 10/26/2015 |
| 1.3 | Jennifer Li | I edited the sub section titles and paragraphs. | 10/27/2015 |
| 1.4 | Hector Richiez | I changed the stick figure from leader to member to represent the two events performed by the member. | 10/27/2015 |
| 1.5 | Tyler Mariano | I added descriptions and action boxes to the five diagrams. | 10/28/2015 |
| 2.0 | Jennifer Li | I added the section 2 and all of the sub sections incorporated with the diagram. I also made some of the suggested edit from version 1.5. | 11/04/2015 |
| 2.1 | Hector Richiez | I changed the introduction to give a more explicit description of the document’s purpose | 11/06/15 |
| 2.2 | Jennifer Li | Edited sub section 1.2 and added the paragraph for section 2. | 11/08/15 |
| 2.3 | Hector Richiez | Added the paragraphs for sections 2.1 to 2.5. |  |

**1.0 INTRODUCTION**

This document contains sequence system diagrams of particular use case scenarios explained in the use case diagram documentation of the Project Management App. The use case diagrams that were chosen to be used in the sequence system diagrams, shown below, were the events where the leader or members are logging to the system, the leader is adding members to the project, the leader is reviewing tasks, the members are viewing their tasks’ progress, and the members are uploading their completed tasks. To illustrate each particular scenario, we use sequence System diagrams which notations are explained in the paragraph below.

**1.1 What Is A System Sequence Diagram?**

System sequence diagram is a visual representation of a distinct use case. Typically, in a system sequence diagram, it captures the behavior of an individual use case scenario. The diagram would show the order, of how messages interact between the actor and the system, with in the specific use case. In simpler terms, the system sequence diagram will show a scenario of a use case that is triggered by an actor.

**1.2 System Sequence Diagram Notation**

This section explains the notation used in system sequence diagrams. In a system sequence diagram a stick figure is used to represent the actor that has a specific role. A rectangular box that contains “:System” which represents the system as a black box. The diagram also contains two types of arrowed lines that shows the input and output between the actor and the system. The first arrowed line is a solid arrowed line pointing to the system, representing the sent input message from the actor. The second arrowed line is a dashed lined arrow pointing to the actor, representing the output message sent from the system. The system sequence diagram, also has notations for conditions of looping and true/false. For looping, an \* is placed on the input arrow followed by the returned values that would be needed over and over again in [ ] and the specified message sent to the system using the := symbol. All of the looping variables must be incased in a box to represent the case of looping. For the condition of true/false, use the [ ] symbols will represent if true or if false. Last but not least, system sequence diagrams have a life line or a duration which represents the session in which the interactions take place.

The system sequence diagram notation and definitions are displayed below.

|  |  |
| --- | --- |
| Notation | Definition |
|  | **Actor** |
|  | **System** |
|  | **Input Arrow** |
|  | **Output Arrow** |
|  | **Life Line or Duration** |
|  | **Looping** |
| [ ] | **True/False Condition** |

**1.3 System Sequence Diagrams for Project Management App**

The project management app will have two major components involved in the system sequence diagram. The first component would be the actor; in these cases the actor can be a project leader or a project member. The other major component would be the system. In this document, five system sequence diagrams will be drawn to illustrate the appropriate notation. Since the project management app has two specific actors, our team decided to pick one use case that affects all actors, two specific use cases that affects the project leader, and two specific use cases that affects the project member.

**2.0 SYSTEM SEQUENCE DIAGRAMS**

This document diagrams, 5 different use case scenarios in system sequence diagram format. The use cases are, User Login which any user of the app can do. Add Member and Review Task, which are actions that only a leader of the created project can do; and lastly, View Task Progress and Upload document are actions that the member will perform in this specific scenario.

**2.1 User Login**

This pictogram shows the interaction between the user, which could be a leader or a member trying to login to the system or app. The input for this event is the username and password. Whether the input data is correct or not, the system will respond with a successful welcome message or with an error indicating the kind of error the system experienced with the data provided.

:System

Leader and Member

Prompts user for their userName and password

Login(userName, password)

successMessage: “Welcome back, userName”

\*sends user to main menu screen\*

Successful(successMessage)

errorMessage: “You have entered the wrong userName and/or password.”

\*clear fields\*

\*prompt user for forgot password\*

Unsuccessful(errorMessage)

**2.2 Leader: Add Member looping case**

This sequence diagram shows the event when the leader is adding members to the project by sending invitations to other application users asking if they want to be part of the project. We show this event in a looping diagram because the leader could perform this action as much as there is members available to choose from.

Leader

Allows the leaders to send invitations to other application users to invite them to be part a given project.

:System

AddMember(userName, projectName, inviteMessage)

sucessMessage: “userName has agreed to work on the projectName project”

\*userName is added to projectName‘s Datebase\*

Successful(successMessage)

errorMessage: “The inviteMessage was not received”

errorMessage: “The inviteMessage was declined”

Unsuccessful(errorMessage)

**2.3 Leader: Review Task True /False case**

This diagram denotes an event whether a condition have been satisfied or not. The leader reviews the submitted tasks by the members and if they are completed, they are marked as completed tasks. If any of the submitted task is not completed, the leader could reassign it to the member responsible for it or to another member that is part the project.

Leader

This allows the leader to review a task submitted to them by one of the members

:System

ReviewTask(submittedTask, userName, comment)

sucessMessage: “submittedTask has been marked as complete”

\*Update progress charts\*

\*Unlock dependent tasks (if needed)\*

Successful(successMessage)

errorMessage: “submittedTask has been marked as incomplete”

\*leader must reassign the task\*

Unsuccessful(errorMessage)

**2.4 Member: View Task Progress**

This sequence diagram shows the interaction between the members and the system, whom want to view their assigned task or tasks progress. If the task’s name provided as input is valid, the system response with a Gant chart view. If there is no connection or the task’s name doesn’t exist, the system response with an error message that addresses one of the two errors.

:System

Member

ViewTaskProgress(taskName)

Successful(successMessage)

Unsuccessful(errorMessage)

Members can view only their own taskName’s progress

sucessMessage: “Gant chart view has been opened for task taskName”

\*app displays chart screen\*

errorMessage: “There was a connection error”

\*returns to the previous screen\*

**2.5 Member: Upload Document**

This sequence system diagram shows the event when a member wants to upload completed task. The data provided for uploading the document is the document name and taskName. The system response will be merely related to whether the document was uploaded successfully and the task’s name is valid. If one of the two is false, the system will response with an error related to the outcome. If both of them true then the system will display a message with the name of the document and task, saying that it was successfully uploaded.

:System

Member

Allows the member to upload documents along with their task submissions

UploadDocument(documentName, taskName)

sucessMessage: “The document doumentName has been upload for the task taskName”

Successful(successMessage)

errorMessage: “There was a connection error”

\*returns to the previous screen\*

Unsuccessful(errorMessage)