Software Engineering Group Project
Design Specification

Author: Samuel Hyde [sah54]
Date: 23-11-2015
Version: 1.0
Status: Review

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB

Content

1. Introduction

1.1 Purpose of the Document

The purpose of this document is to give an idea into the final product that will be designed. This document will provide information in design specifications that have been produced by the group for the project.

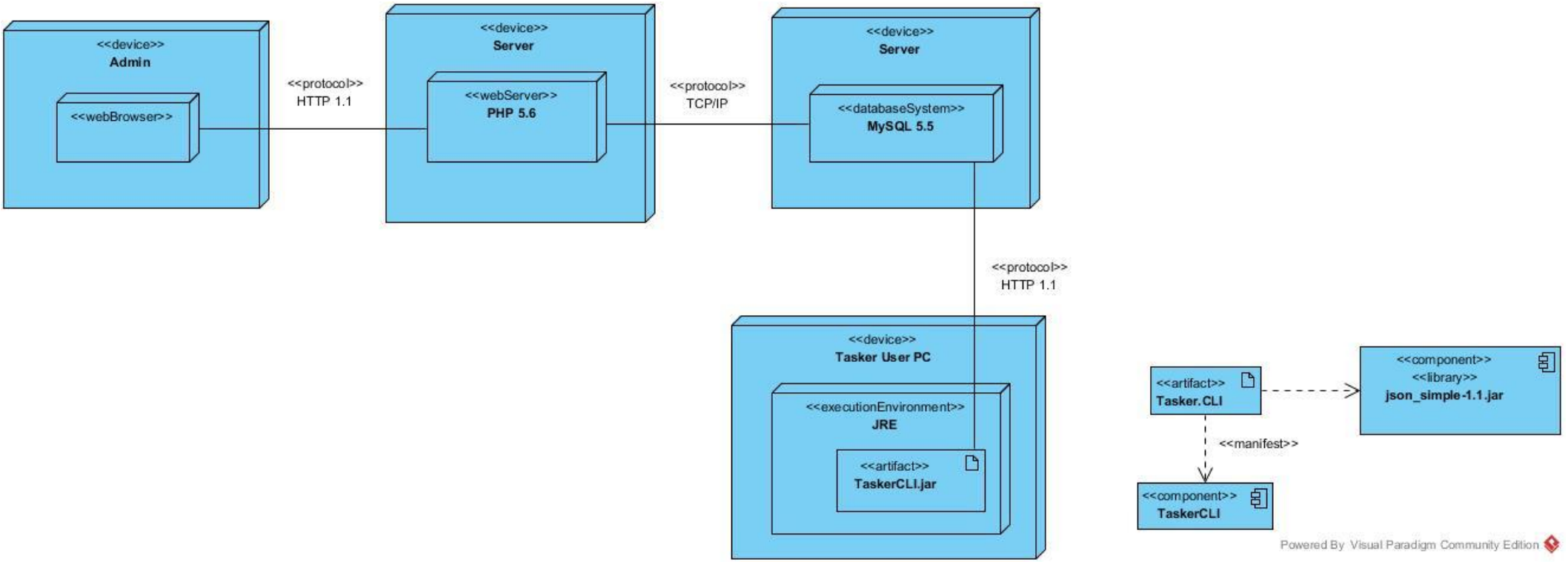
1.2 Scope

This document specifies the layout and content that are specified in Design Specification Standards that is required for the group project Tasker system [2].

1.3 Objectives

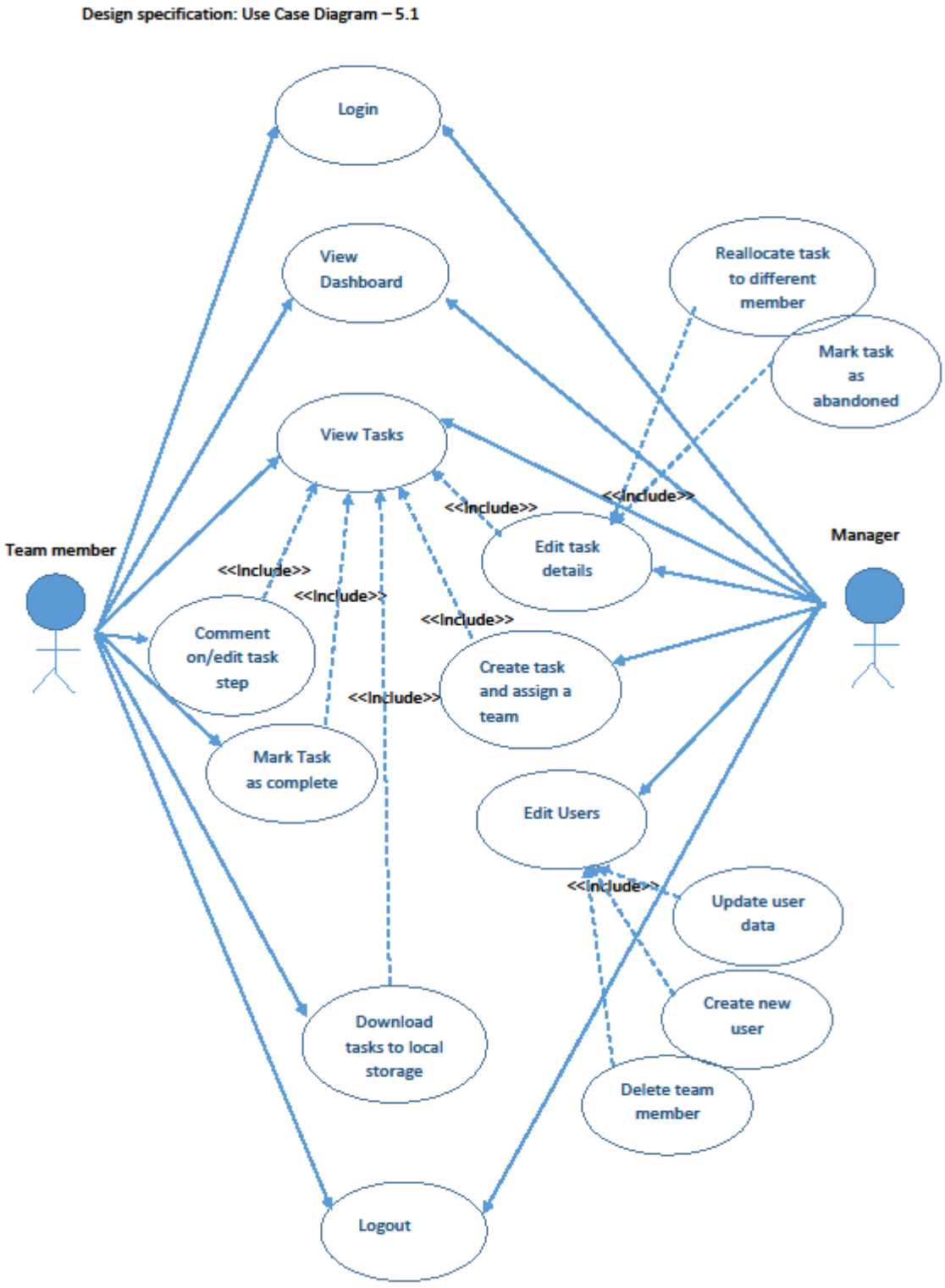
The main objective is to produce a design specification which is complete and accurate that includes the requirements listed in the requirement specification ready for the implementation phase [3].

2. Deployment Diagram
2.1 Applications in the System
UML Deployment Diagram
Admins can access the TaskerMAN website via a web browser. The TaskerMAN website will communicate with the web server using the HTTP 1.1 protocol, although HTTPS could be used for better security. This communication will provide information to the admin such as all users and their tasks and allow the admin to allocate new tasks. The database system uses MySQL 5.5 and communicates with the web server using the TCP/IP protocol and also the TaskerCLI program via HTTP 1.1. A regular tasker user will run the TaskerCLI program in the JVM (Java virtual machine) using JRE (Java runtime environment). The TaskerCLI.jar consists of the TaskerCLI component and the JSON external library.



3. Interaction Design

3.1 Use Cases



Description key: Numbers that appear bold refer to the requirement specification.
This is the use case diagram for the task management system. It shows all the actions that can be performed by the different users that will be interacting with it. The diagram also details all the expected functional requirements for the task management system. TaskerSRV doesn't have any user interaction, so it hasn't been included in the diagram.

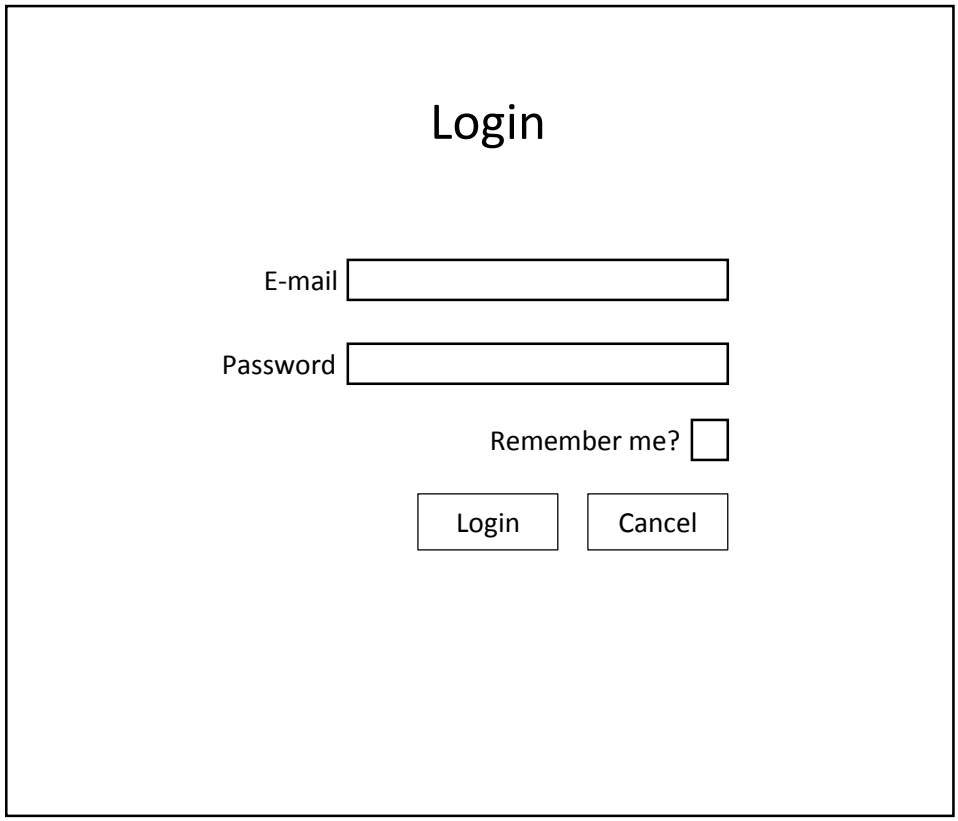
TaskerCLI is the front-end application that team members will have interaction with. From TaskerCLI the team members can identify themselves by logging in (**4.1.3: FR8**), using email authentication. Team members can download tasks for viewing offline (**4.1.3: FR8a**). Other use cases team members perform on TaskerCLI are editing of tasks (**4.1.3: FR10),** this includes commenting on/ editing steps of a task and changing the status of a task to completed.

TaskerMAN is the web-side application of the task management system, which the manager will interact with. From TaskerMAN the manager can create and modify user data. Illustrated in the use case diagram under 'edit users', the manager can create, update and delete users (**4.1.2: FR3**). Also shown in the use case, the manager can 'create tasks and assign a team member' (**4.1.2: FR4**). The manager, illustrated in the diagram ('edit task details'), can also reallocate a task to a different team member (**4.1.2: FR5**) and mark a task as abandoned (**4.1.2: FR6**). The last use case the manager can perform on TaskerMAN is view tasks (**4.1.2: FR7**) [1].

3.2 User Interface Design

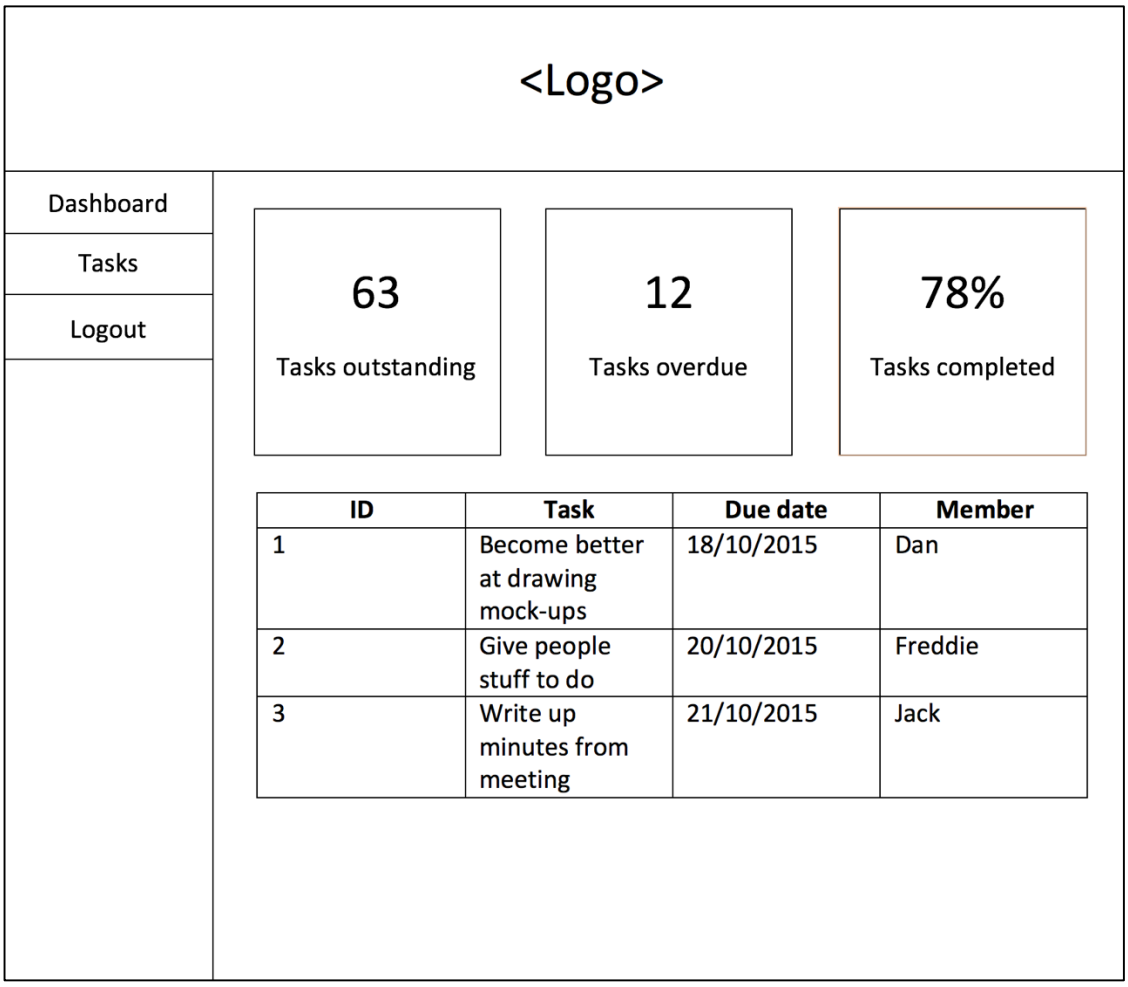
3.2.1 Java Application Design

Login Interface



This is a basic mock-up of the interface design for the user to login to the desktop client to gain access to the system.

Dashboard Interface



ID	Task	Due date	Member
1	Become better at drawing mock-ups	18/10/2015	Dan
2	Give people stuff to do	20/10/2015	Freddie
3	Write up minutes from meeting	21/10/2015	Jack

This interface design shows the main dashboard to the system, this is shown after the user has logged into the desktop client.

Task Details Interface

<Logo>

Dashboard

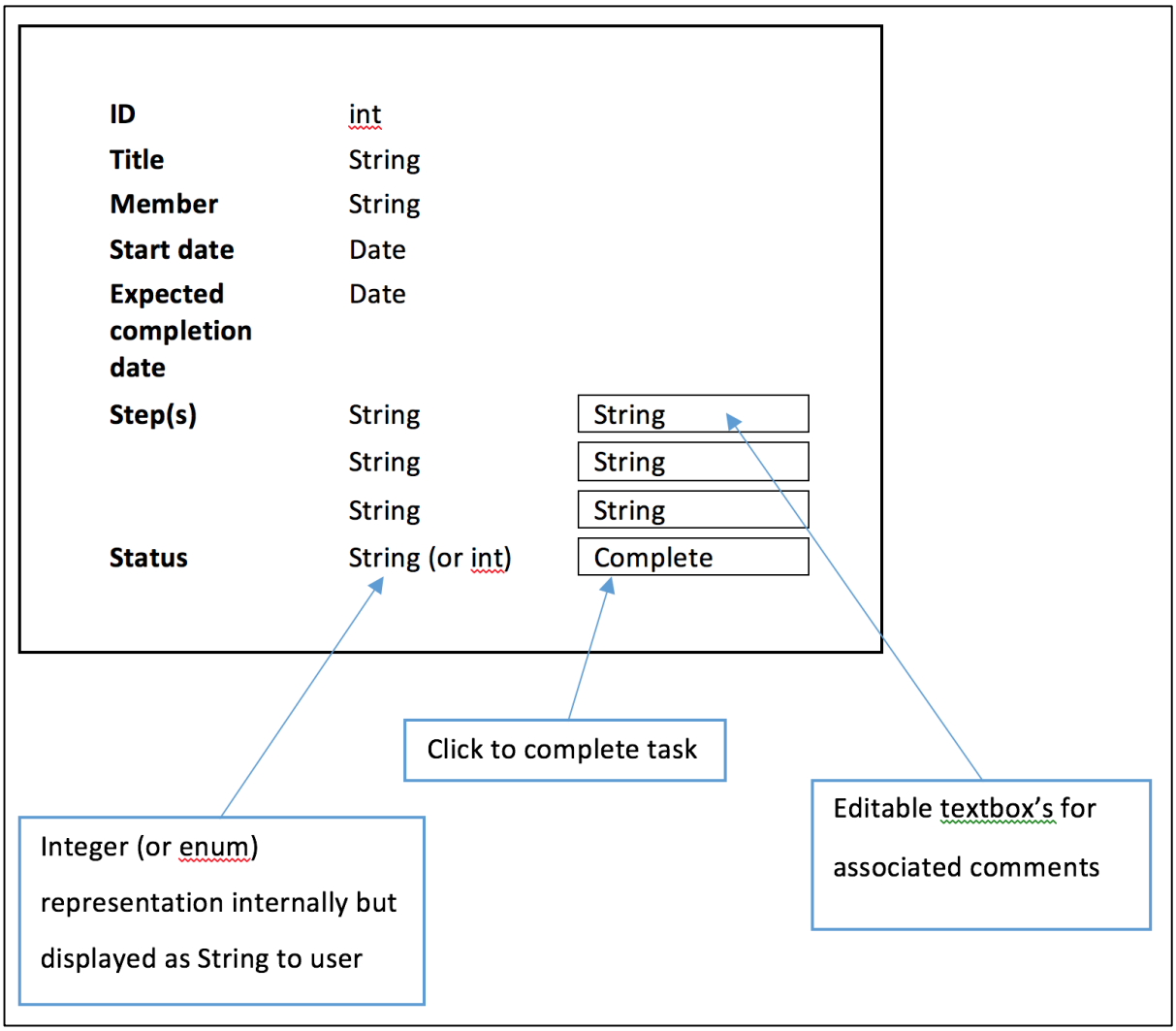
Tasks

Logout

ID	Task	Due date	Member
1	Become better at drawing mock-ups	18/10/2015	Dan
2	Give people stuff to do	20/10/2015	Freddie
3	Write up minutes from meeting	21/10/2015	Jack
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

This design shows the interface that will be shown when the Tasks button is clicked. From here, the user can see all tasks that are within the system and able to see when they are due and who is meant to complete the task.

Task Details Interface

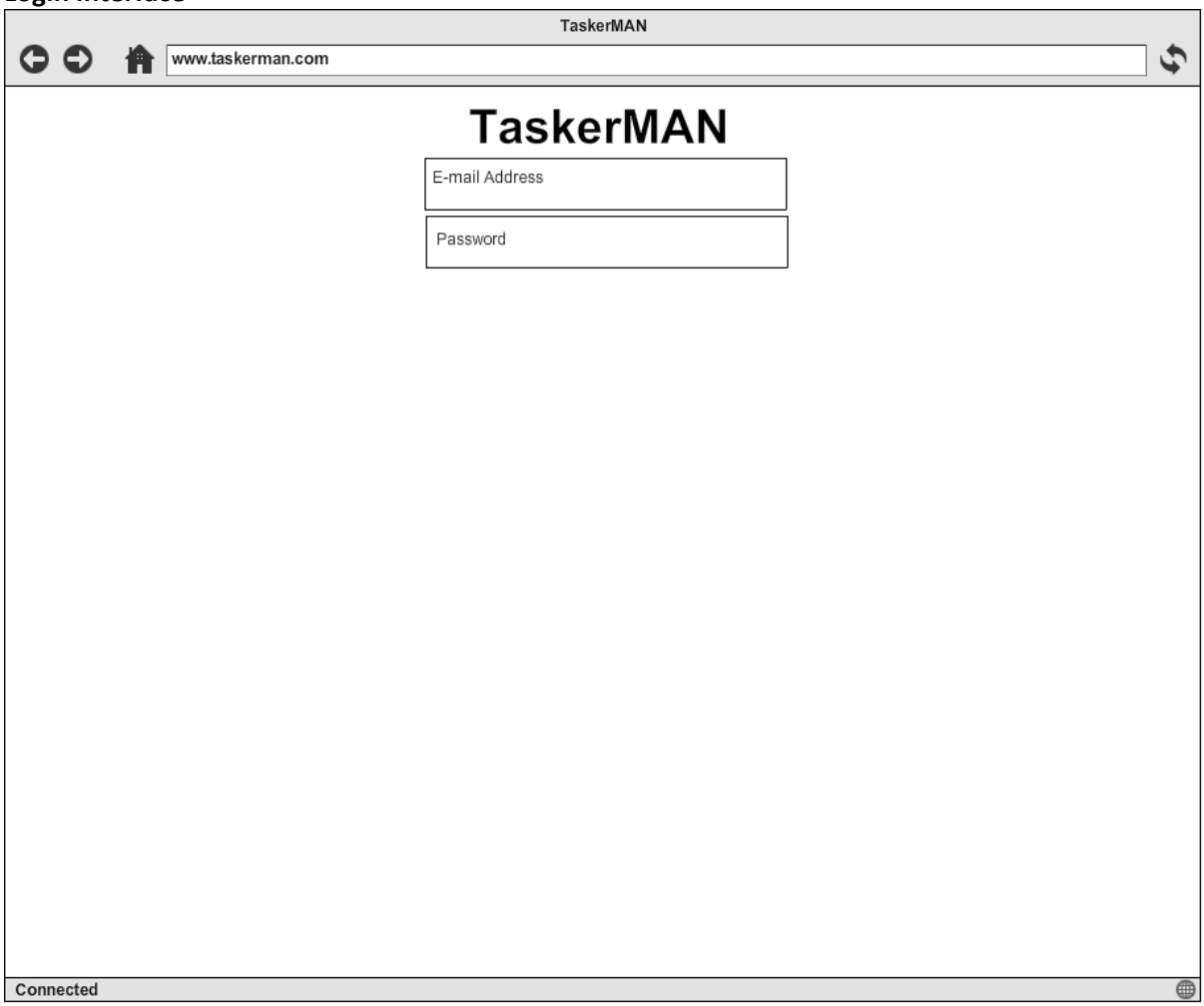


This design shows the interface style when the user wants to see the specific details for a selected task.

3.2 User Interface Design

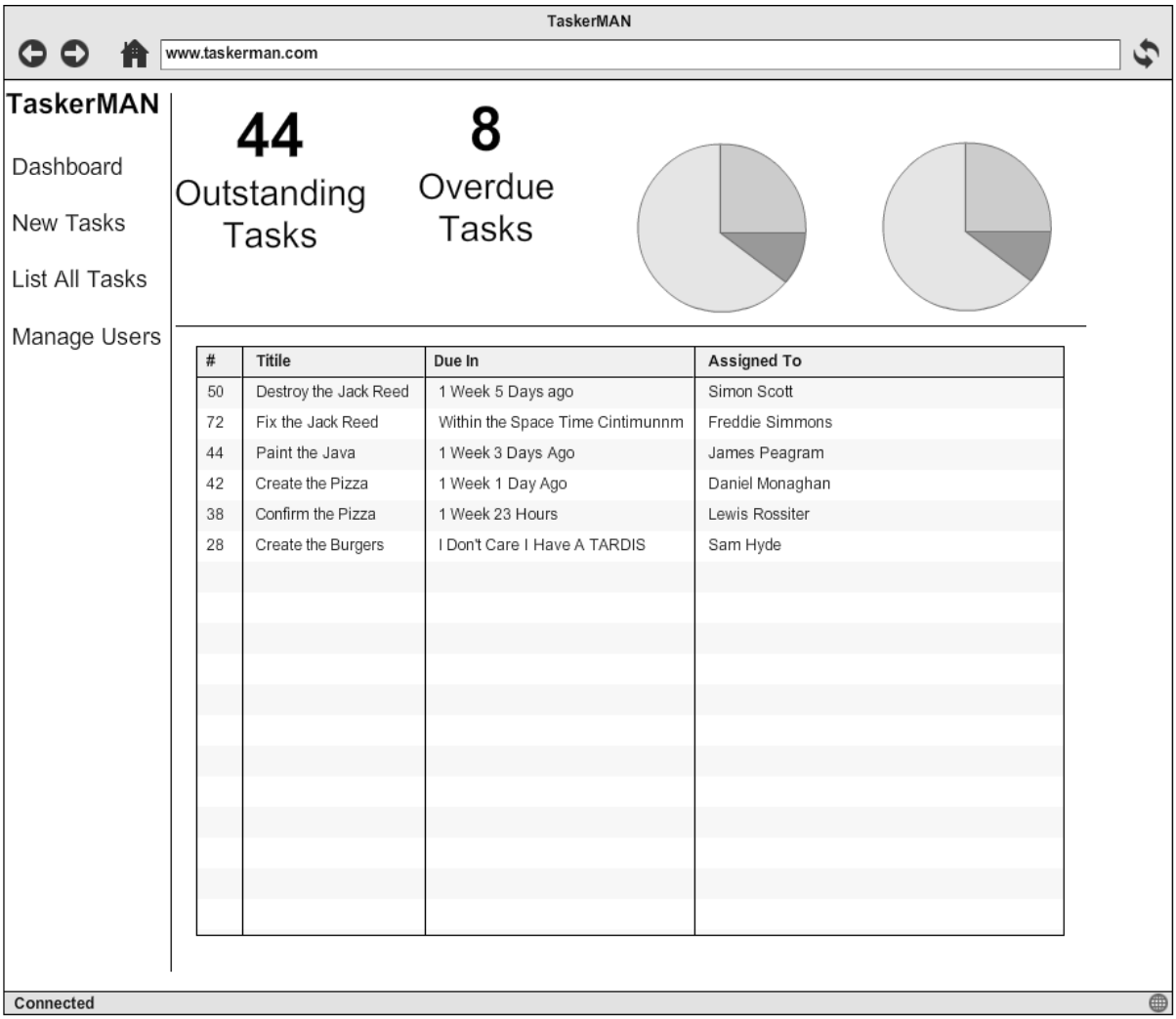
3.2.2 Web Interface

Login Interface



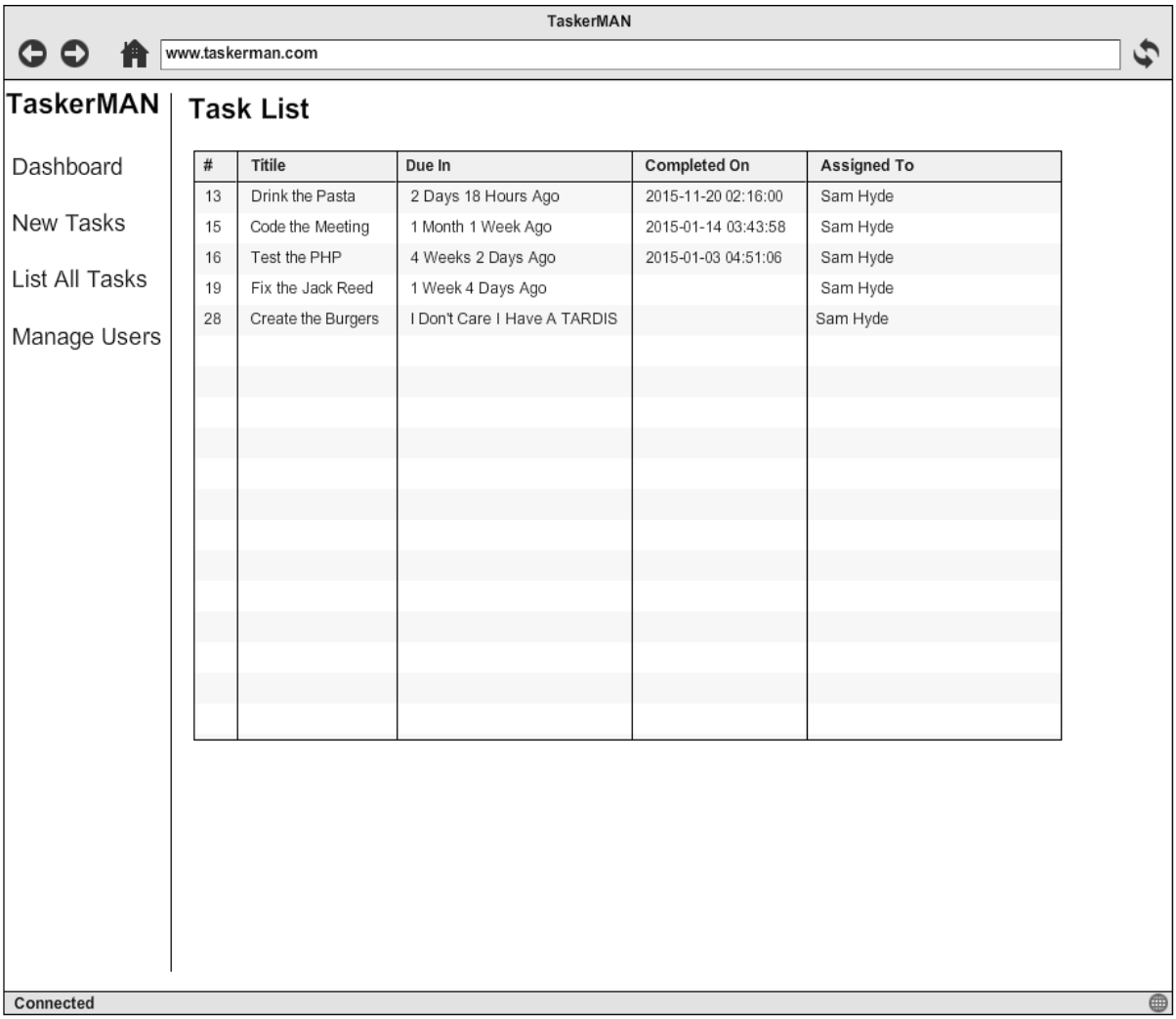
Mock-up design for the interface for the web application. This is the main screen where the user will log into the system and be able to access the Tasker system.

Dashboard Interface



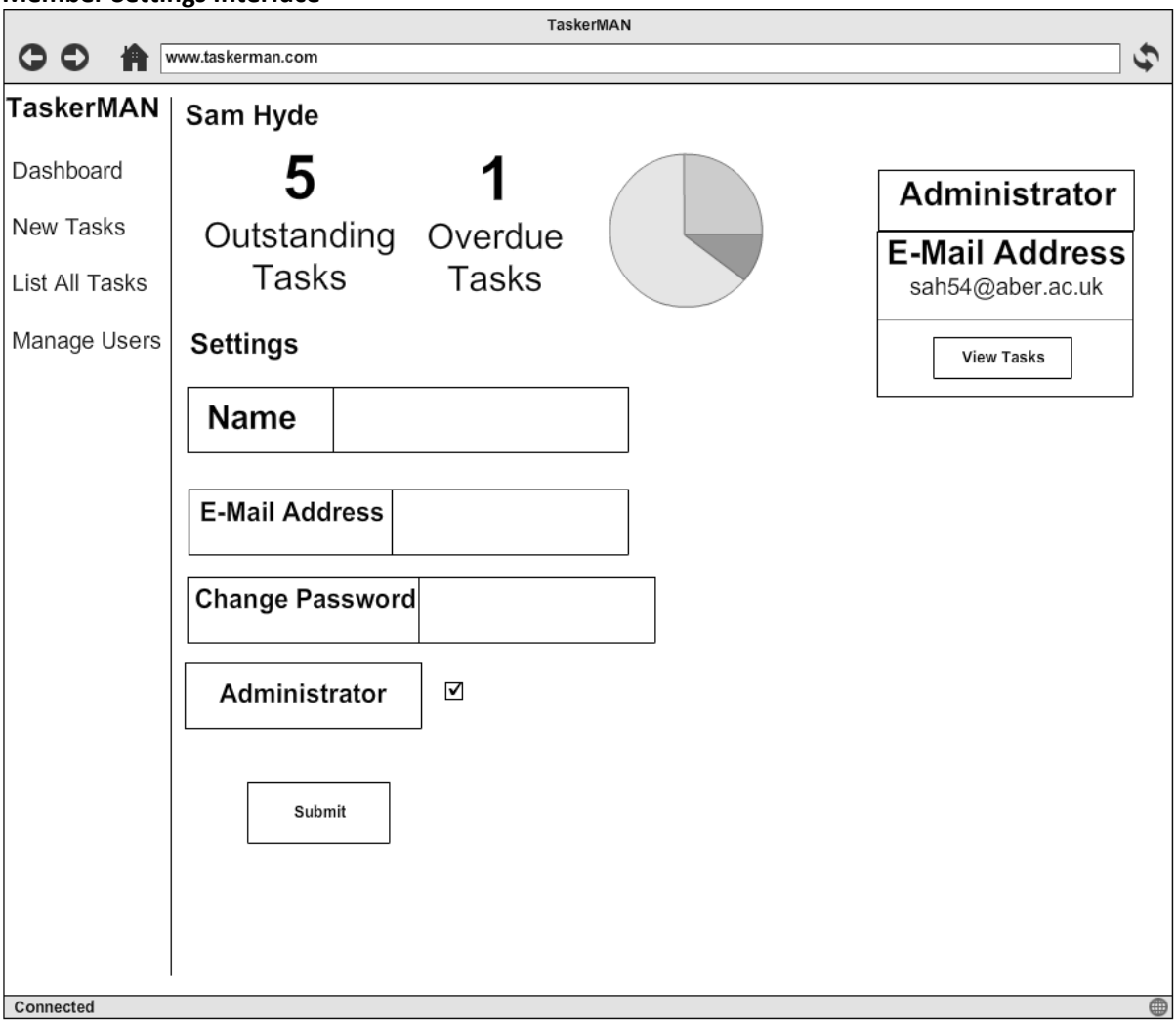
After logging in to the web site, the user will be presented with the dashboard that shows a collective overview of the current task list; this includes the task number, title, due date and the team member assign to it.

Member Task List



In the event that a team member want to look at task assigned to a specific member, they will be directed to a similar page to the design above. From here, a list detailing the assigned tasks to that team member.

Member Settings Interface

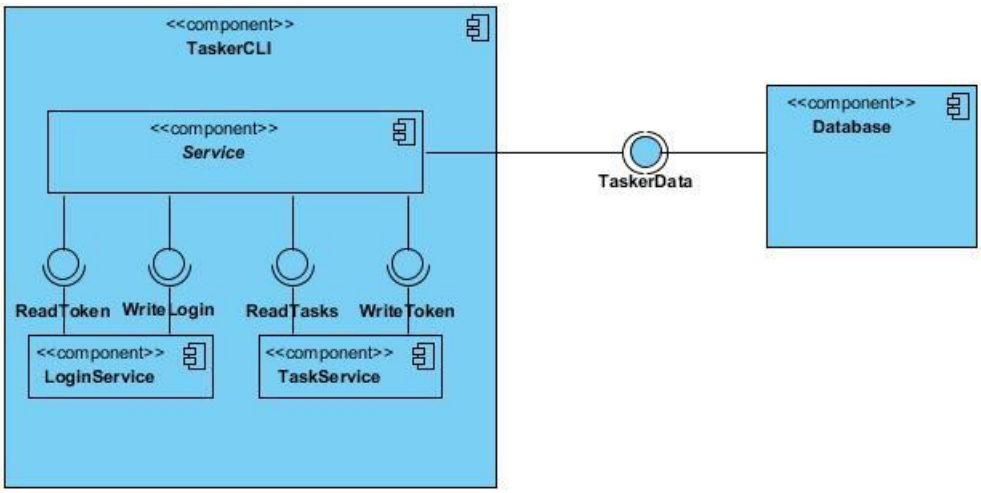


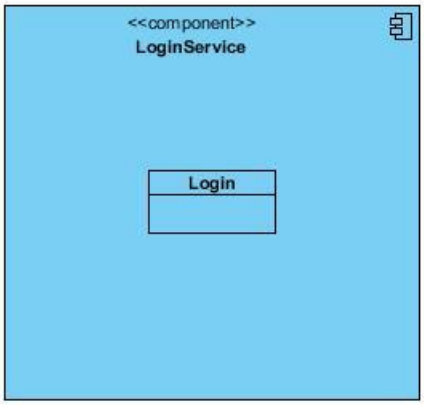
If a team member needs to modify either their own or another team members information in the system, they will be directed to an interface design similar to this. From here, the team members name, e-mail address, password and administrative privileges can be changed.

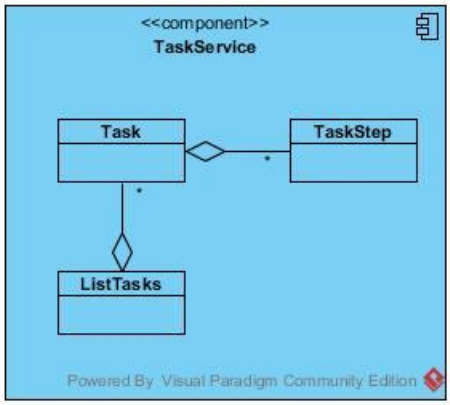
4. Component Description

Component Diagram

The TaskerCLI component contains two Service implementations; LoginService and TaskService. These services communicate with the database system, exchanging TaskerData which can include (but not limited to) information such as login credentials or user tasks. The LoginService consists only of the Login class and the TaskService contains a Task class which has many TaskSteps and ListTasks will produce those Tasks.

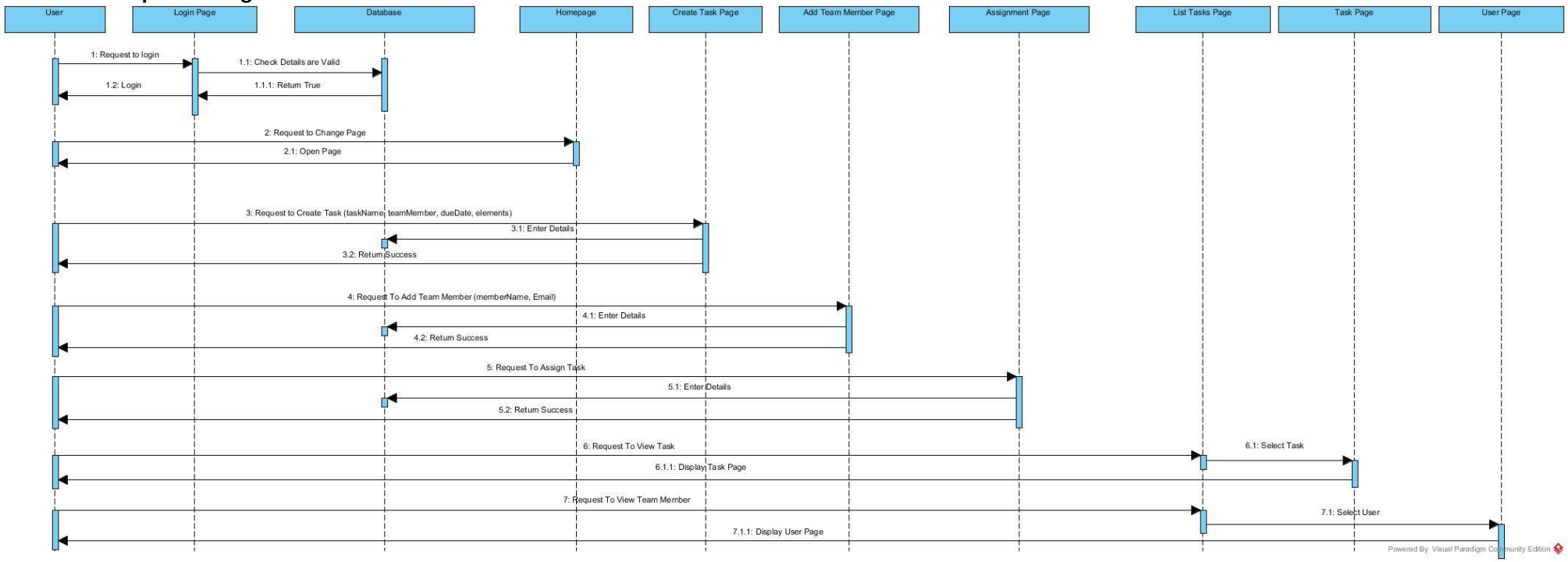






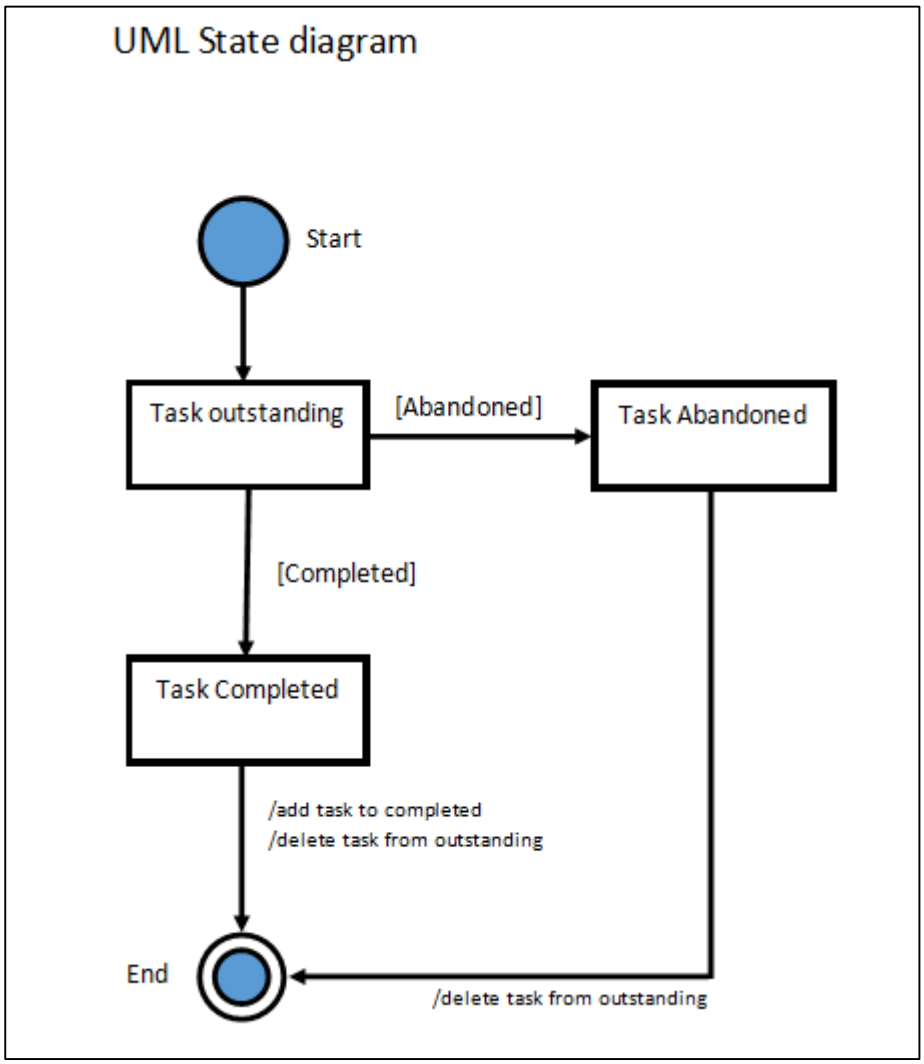
5. Detailed Design

5.1 Sequence Diagram



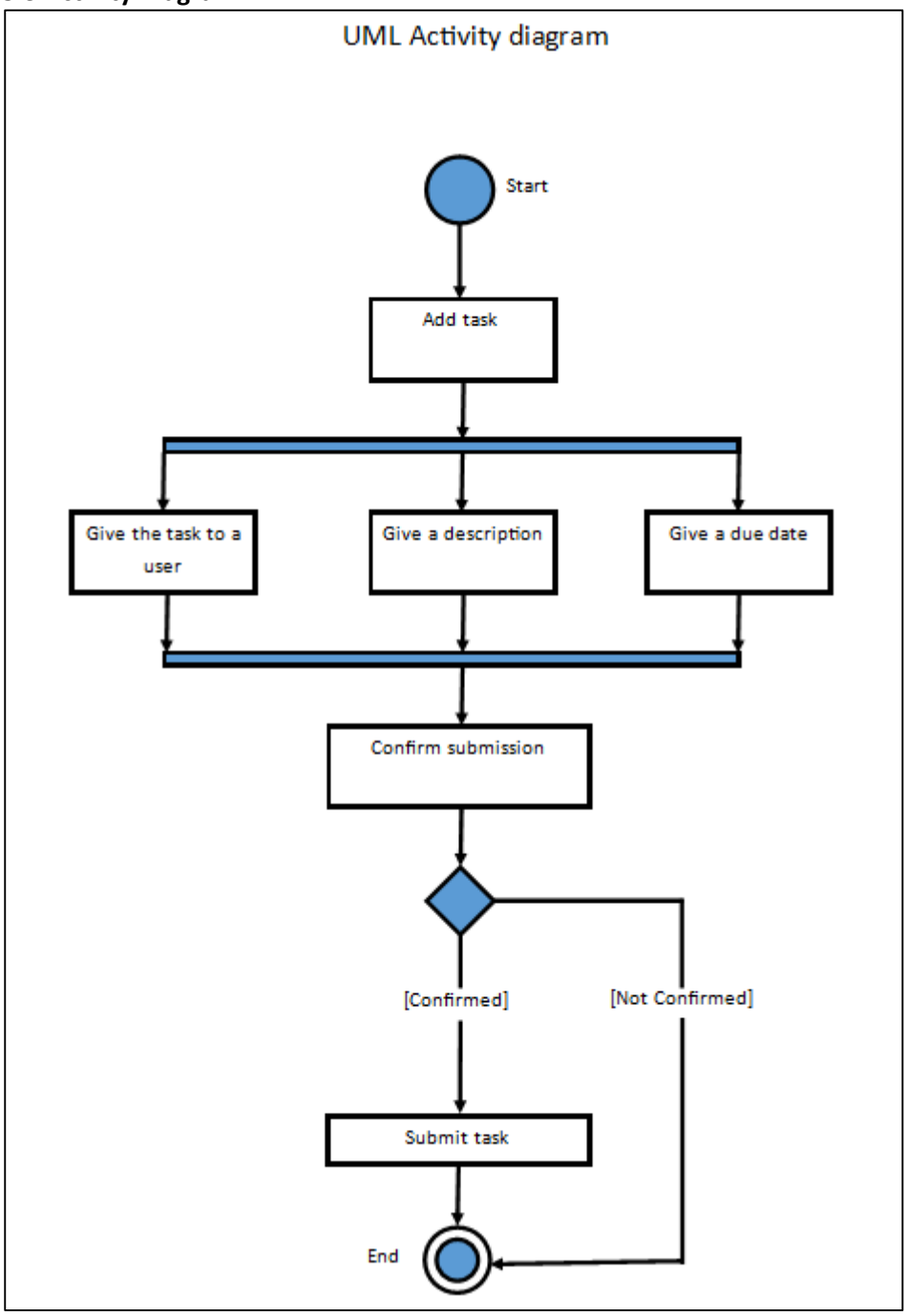
The User first views the Login Page, and makes a request to the page. The website first needs to check the database to see if the details are real, and returns true if it is so, allowing the program to continue by logging the User in to the Homepage. From the Homepage, the user can select any page to open next. In the task page, they can request to create a task, and the page will return as a success. If the user is on the Add team Member Page, they can try to add a team member, at which point the Team Member Page must put the details into the Database, and Return Success to the User. At the Assignment Page, they can request to assign tasks, at which point the Assign tasks Page will enter the details of the request into the Database, and return a success. From the List Tasks Page, The user can request to visit the page of a team member, or of a task. If they request to see a task, the website will display the correct task, and if they request to see the page of a User, the website will show that page.

5.2 State Diagram



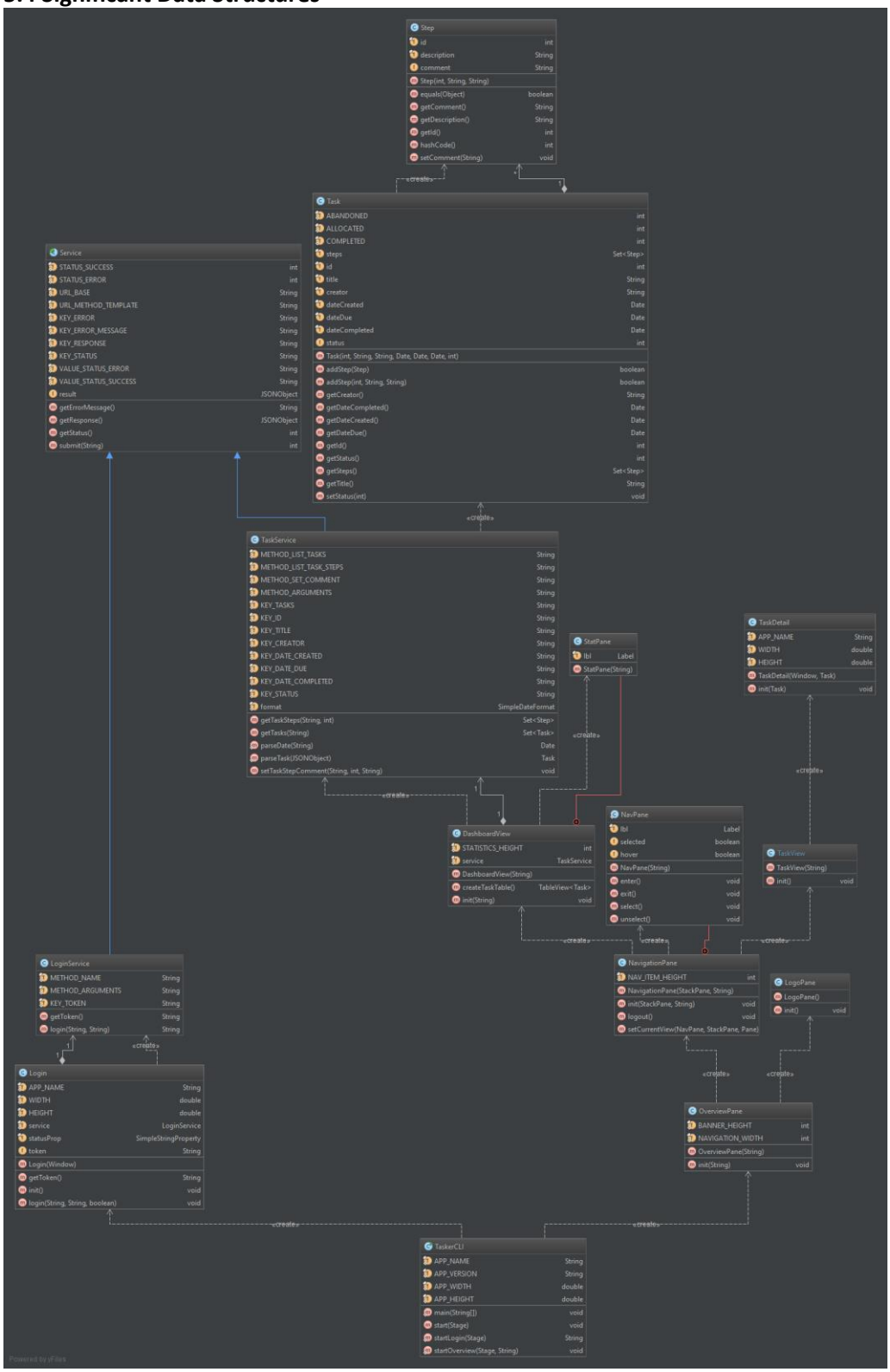
There are three different states that the task can be in, Outstanding, Abandoned or Completed. When the task is first added it will automatically be Outstanding, once it has been completed the user marks it as such, and it will change states from outstanding to completed so you can still track how many task you have completed compared to ongoing tasks. If the task is changed to Abandoned then it is just deleted and no record is kept.

5.3 Activity Diagram



This is the activity diagram for adding a task, once the user has initiated creating a class a user has to be allocated to it, a description has to be given and a due date has to be allocated, otherwise the task cannot be created. Once this has all been done, there is a confirmation box, if this is accepted then the task is created and saved to the database, if the confirmation box is declined then the task is not created.

5.4 Significant Data Structures



References

[1] Software Engineering Group Project. Requirement Specification. N.W.Hardy SE.QA.RS. 1.2. (Release)

[2] Software Engineering Group Projects – Design Specification Standards for SE&Comp Sci projects/1.8 (Release)

Document History

Version	CCF No.	Date	Changed Made	Changed By
1.0	N/A	2015-11-14	First draft of complete design specification	sah54