**CS221 Group 09**

**Design Specification**

**Authors:** Einar Dogger, Jake Doran, James Downing, Nedialko Petrov, Pawel Kabulski, Henry Finlay, Jay Gould

**Config Ref:** SE\_09\_DesignSpecification\_01

**Date:** 28/10/15

**Version:** 1.2

**Status:** Released

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2015

**CS221 Group 09**

**Design Specification Contents**

**1 INTRODUCTION**……………………………………………………………………………..3

1.1 Purpose of this Document………………………………………………..3

1.2 Scope………………………………………………………………………………..3

1.3 Objectives…………………………………………………………………………3

**2 DEPLOYMENT DESCRIPTION………………………………………………………….**4

2.1 Applications in the system……………………………………………….4

2.2 Application interactions…………………………………………………..5

**3 INTERACTION DESIGN…………………………………………………………………...**5

3.1 Use-cases…………………………………………………………………………..7

3.2 User interface design………………………………………………………..9

**4 COMPONENT DESCRIPTION**…………………………………………………………..14

**5** **REFERENCES**…………………………………………………………………………………..16

**6 DOCUMENT HISTORY**……………………………………………………………………..16

**1 INTRODUCTION**

**1.1 Purpose of this document**

The purpose of this document is to highlight how, as a group, we have understood the requirements outlined in the requirements specification[1], a task management system involving a website for use by team managers and a java application for use by team members, and turned it into an outline design of the system.

**1.2 Scope**

This document provides a deployment description which includes an overview of the applications in the system and the interactions between these applications, as well as an interaction design which includes use case diagrams and user interface design for the system, with screenshots and detailed descriptions of the interactions between the user and the system.

This document should be read by the client and compare to the requirements specification[1] they supplied, to ensure that all of their proposed requirements have been met.

**1.3 Objectives**

The objective of this particular document is to present our outline of the design specification[2]. This includes a deployment description and and an interaction design. The goals of this document are:

1.3.1 Provide an overview of the applications in the system.

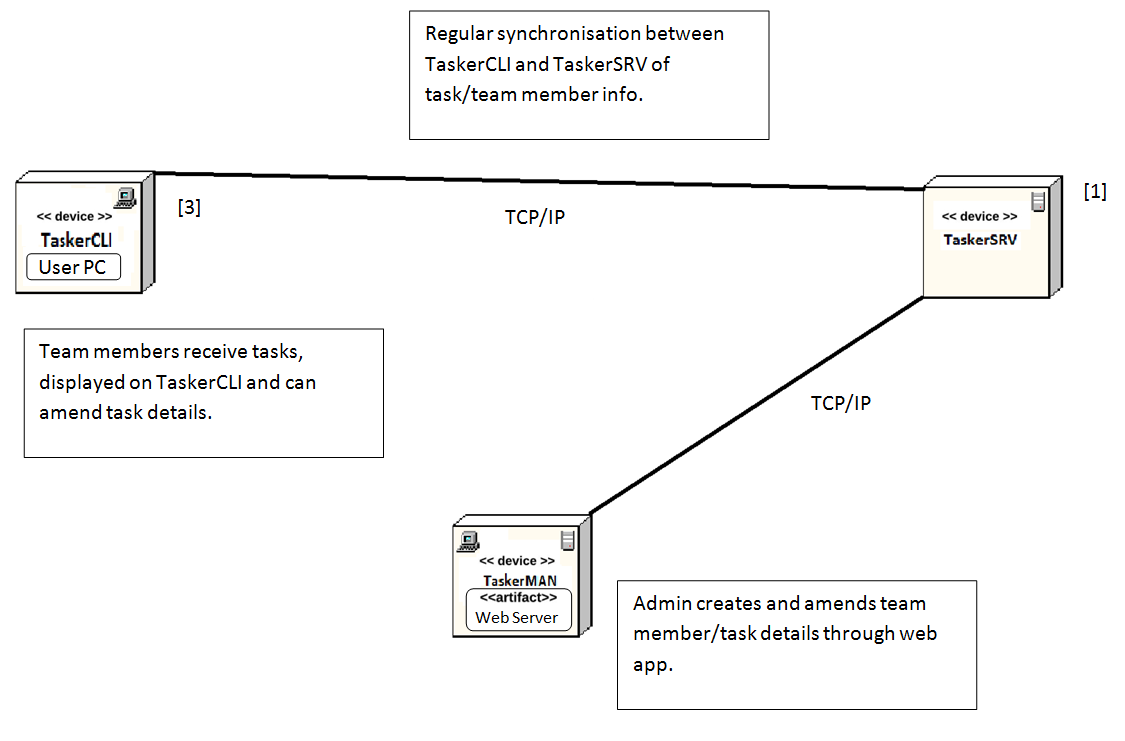
1.3.2 Provide an overview of the interactions between the applications in the system.

1.3.3 Detail how “actors” are expected to interact with the system through use case diagrams.

1.3.4 Detail the intended appearance of the user interface with a description of the effects of interactions with the user interface elements by the user.

**2 DEPLOYMENT DESCRIPTION**

**2.1 Applications in the system**



[1]: TaskerSRV (FR1, FR2)

[2]: TaskerMAN (FR3, FR4, FR5, FR6, FR7)

[3]: TaskerCLI (FR8, FR9, FR10, FR11)

**2.2 Application interactions**

**2.2.1 Java Application**

Task list interactions

- Each task name is taken from the database and displayed on each button, with the title of the tasks in the database being the title on the buttons.

- There is a scrollable area if the total number of tasks exceeds the amount being displayed in the above area.

Main area

- There are 2 main areas for the user; the task itself and the progress bar. Both of which will be pulled from the database, showing any previously stored information.

- As the progress text area is optional, the “add comment” button will submit the data inside of the area to the local store area, to then be updated to the online database if a connection exists.

Upper right area

- Update button: Manually submits the locally stored data to the online database, on top of the manually refreshing update done in the background every 5 seconds.

- Logout: Logs the user out and clears all pulled data from the application, sending the user back to the login screen.

- Drop down box: Shows if the user has this task assigned to their account and is actively being edited.

Log in window

- Login: Email and password stored in the database and then retrieved to authenticate the access. Allows the base application to know where to pull data from for the tasks etc.

- Remember me: Locally stores the email and password for the user to have quick access if frequently using the application, stored locally in a separate file.

**2.2.2 Web Application**

Sign up

Input your name , email and password. Search in the database for the email.

If the email already exists output that there is already a user that uses that email.

If the email doesn't exist salt and hash the password and put the name, email

and salted and hashed password in the database and let the newly registered user inside.

Log in

Input your email and password (salted hashing algorithm for security). Search in the database for the email.

If the email has been found then salt and hash the password that has been input to compare to the salted and hashed password we have in the database.

If the comparison returns that they are equal let the user in and assign them a session.

Remember me

If the check box for the remember me gets checked, put a salted and hashed email in it

and save it in database to compare later if the user needs to login automatically.

In TaskerMAN

If successfully authenticated login or sign up th e users will find themselves at the Home page where they are going to see a menu with main pages being the member page and the task page, and a list of the recent tasks.

On the member page there is going to be a list of members and options to delete, edit and add more members.

Respectively on the task page there is going to be a list of tasks and options to delete, edit and add more tasks.

All of the displayed lists will be provided with a SELECT query to the MySQL database with the appropriate parameters for the page.

Editing both members and tasks will be available by clicking on a button/text that will redirect with the help of a HTML GET REQUEST and MySQL SELECT query to a page that will show the current information of the member or task and change it in the database after submitting the edit which is going to be issued with a MySQL UPDATE query.

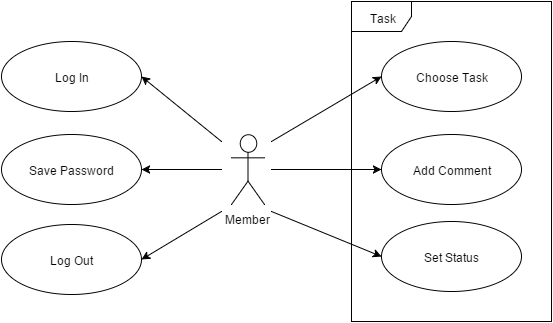
Adding both members and tasks will be available by clicking on a button/text that will redirect to a page with blank fields to fill with the new information of the new member or task and submitting it will be issued with a MySQL INSERT query.

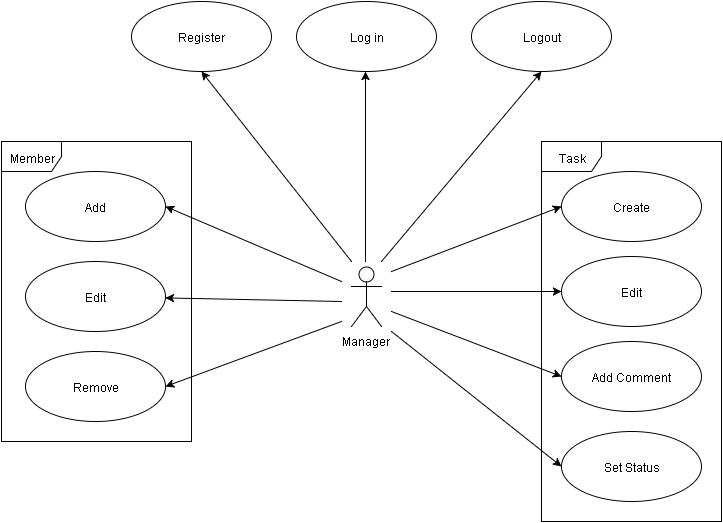
Deleting both members and tasks will be available by clicking on a button/text that will issue a MySQL DELETE query and refresh the page.

**3 INTERACTION DESIGN**

**3.1 Use-cases**

**3.1.1 Java Application**

**3.1.2 Web Application**



**Java Application**

|  |  |
| --- | --- |
| Log in | User will be able to log in to the application if they enter the correct email and password. |
| Logout | User can log out so that another user can log in. |
| Save Password | User can choose to save the login password for next time. |
| Choose Task | User can choose a task that the user wants to see more details about. |
| Add Comment | User can add a comment to the task for later reference. |
| Set Status | User can set the status of the task to “Completed”. |

**Web application**

|  |  |
| --- | --- |
| Register | User can make an account if It’s their first time using the application. |
| Log in | User can log in with the correct email and password. |
| Logout | User can logout of the system. |
| Add | User can add a new team member. |
| Member: Edit | User can edit the information of a team member. |
| Remove | User can remove a team member. |
| Create | User can create a new task using task information and allocate team members to the task. |
| Task: Edit | User can edit the information regarding a task. |
| Add Comment | User can add a comment to the task for later reference. |
| Set Status | User can change the status of the task to “Completed”, “Allocated” or “Abandoned”. |

**3.2 User interface design**

**3.2.1 Website (TaskerMAN)**

Login

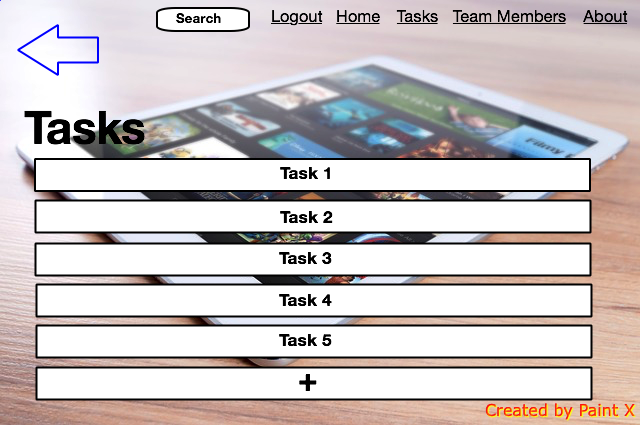
The Login screen will be the default screen users will see when accessing the website. It shall allow users to login via providing valid user information in the form of an email address and optional password. If the user doesn’t have an account, they can create an account by providing their full name, a valid email address and an optional password. It shall be possible for users to save their login information for previous logins by using the “remember me” feature. This shall make for speedier logins in the future.

Home



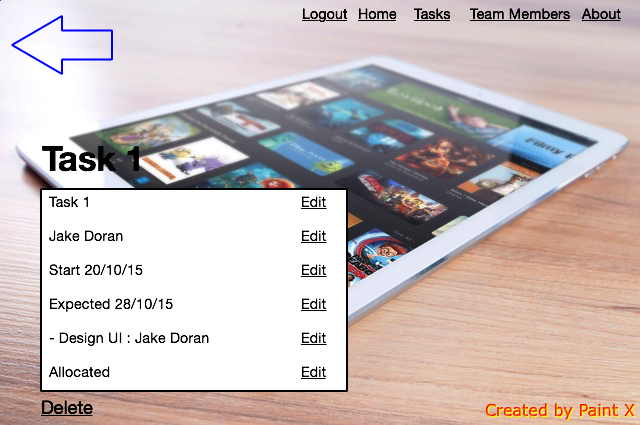
Once the user has successfully logged in, or are still logged in from a previous session, they shall be taken to the home page. From here the user can navigate the website using the menu in the top right of the screen which will be present on all pages through the use of a menu.php file included on all necessary pages. The user will also be shown a feed of recent tasks which are clickable and take the user to the specific task page, as well as a description of the system. The back arrow in the top left of the screen will allow better navigation for mobile users.

Tasks



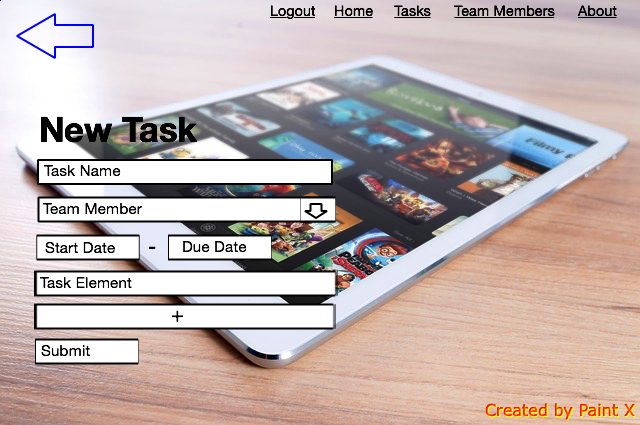
If the user has clicked on the tasks menu option they shall be taken to the task page. This page lists all the tasks that the user manages. Again the menu.php file will be included on this page but this time search.php shall also be included to allow the user to search for a specific task. Every task in the list shall be clickable and will take the user to the specific task page to view all information about that task. By clicking the “+” button at the bottom of the page, the user will be taken to the new task page where they can create a new task.

Specific Task



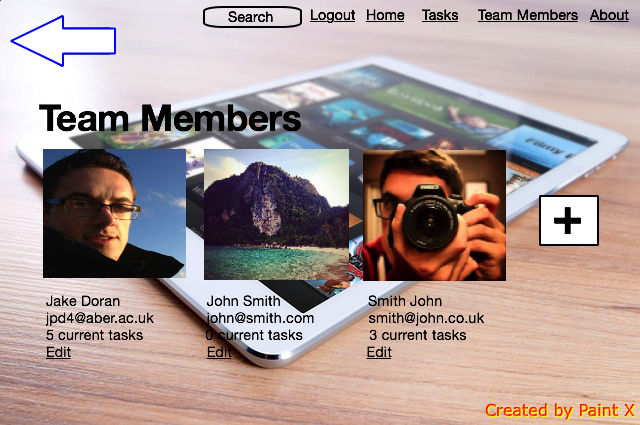
If the user clicked on task from either the home page feed or the tasks page, they shall be taken to a page that lists the information to do with that task. All the information listed about that task will be followed by an edit button to allow live editing of the task information. The delete button at the bottom of the page will delete the task. Again this page will have the menu.php file and shall include the back button for mobile users.

New Task



If the user clicks on the “+” (add task) on the tasks page, they will be taken to the new task page where they can create a new task. They can fill in the information for: Task Name, Start Date, Due Date and Task Elements. They can add further task elements by clicking on the “+” and they can allocate a team member by selecting a member from the drop down list. Once they have filled in the information, the user can click the “submit” button to finalise the task. Again this page will have the menu.php file and the back button.

Team Members



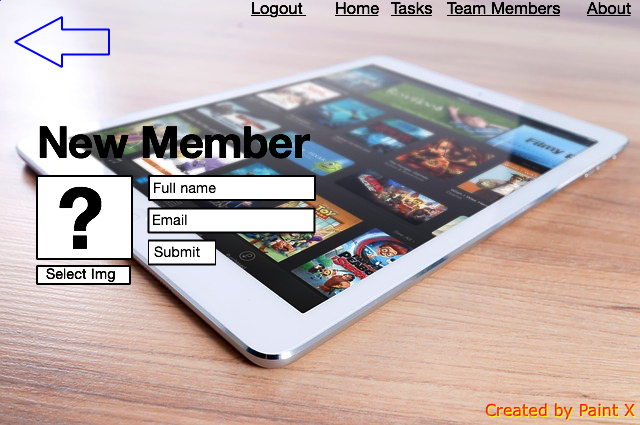
If the user clicks on the team members option from the menu, they will be taken to the team member page. From here the user can see a list of all the members in their team as well as some basic information about them and how many tasks they are currently allocated to. They can edit the information about a specific team member by clicking the edit button below each member profile. They can also add a new team member by clicking on the “+”. The search.php file will be included on this page to allow the user to search for a team member.

Edit Member



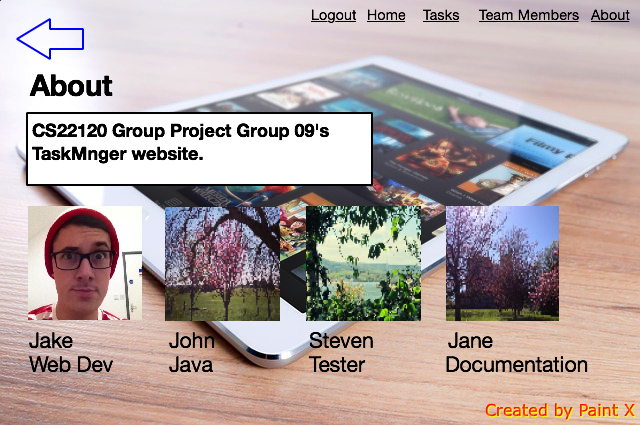
If the user clicks on the edit button for a team member they will be taken to the edit member page. Here the user can edit the basic information that is held for each member i.e the team member’s full name and their email address. The user will also be allowed to delete the team member if necessary. Although an image of the team member is shown this may not make it to the final design of the system since this isn’t a functional requirement. Again the menu.php file will be included on this page as well as the back button.

Add Member



If the user clicked on the “+” on the team members page they shall be taken to the add member page. This page allows the user to create a new team member. To do this the user needs to specify the full name and provide a valid email address for the team member. The user can then finalise this member by clicking the “submit” button which will add the team member to the database. This newly created member will now be able to be allocated to tasks. Again menu.php will be included on this page and so shall the back button.

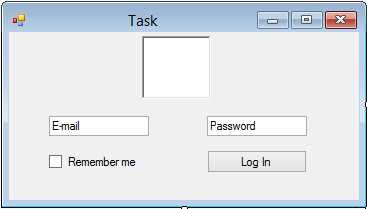
About



If the user clicks on “About” from the menu they shall be taken to the about page. Here the user can see information about the system through the means of a description and a list of contributors and the roles they fulfilled. Again the menu.php file and the back button will be included on this page to allow easier navigation between pages.

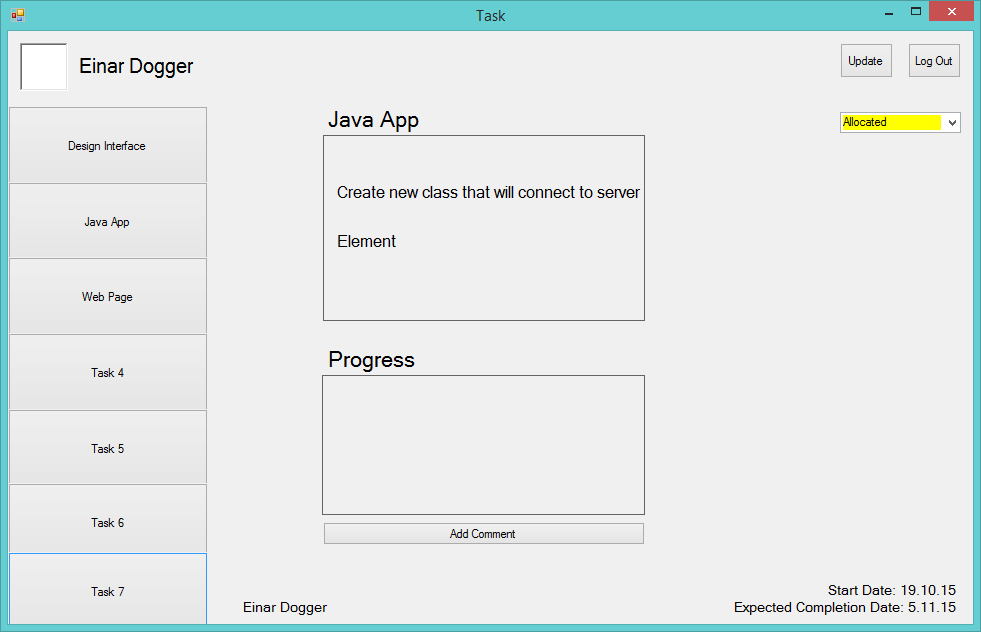
**3.2.2 Java Application (TaskerCLI)**

Login



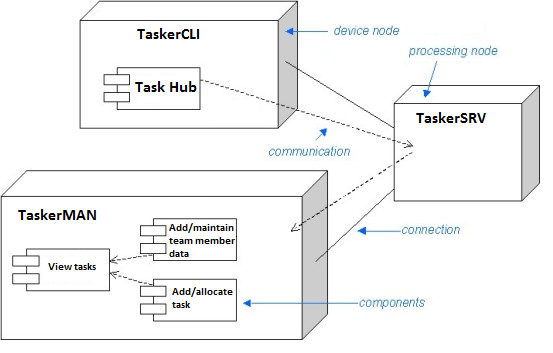
The login screen is the first part of the program the user will see if they have not logged in before or clicked on remember me. The user has to provide a valid email address which will identify them and their password. If the user clicks on the box where it says "Remember me", then this box will not show up next time unless the user logs out.

Tasks

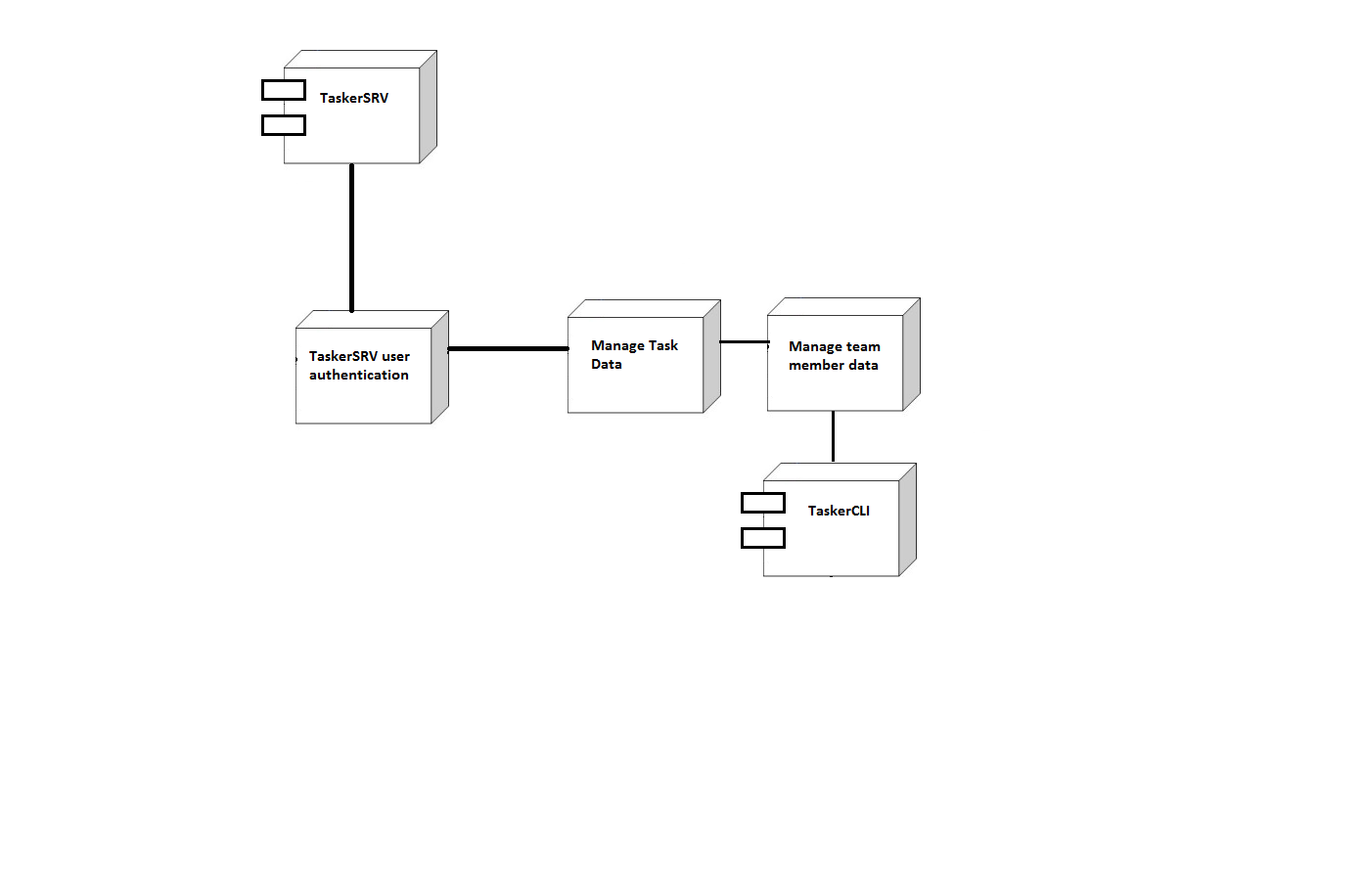


This is what the program will look like once the user has logged in. On the left there will be a column of tasks which the user is allocated to. The user will be able to scroll through the tasks if there are more tasks than what can fit in the window. The user can click on any of these tasks to show more detail. In the middle of the window the user can read the task and in the panel below the user can read and post their progress. They can do so, by typing in the text field and clicking on the "Add Comment" button. In the bottom right corner, the user can see the date when the task was created, and when it should be completed by. In the top right corner, the user can choose to log out so another user can log in instead. Under the logout button is a drop down menu where the user can change the task to completed.

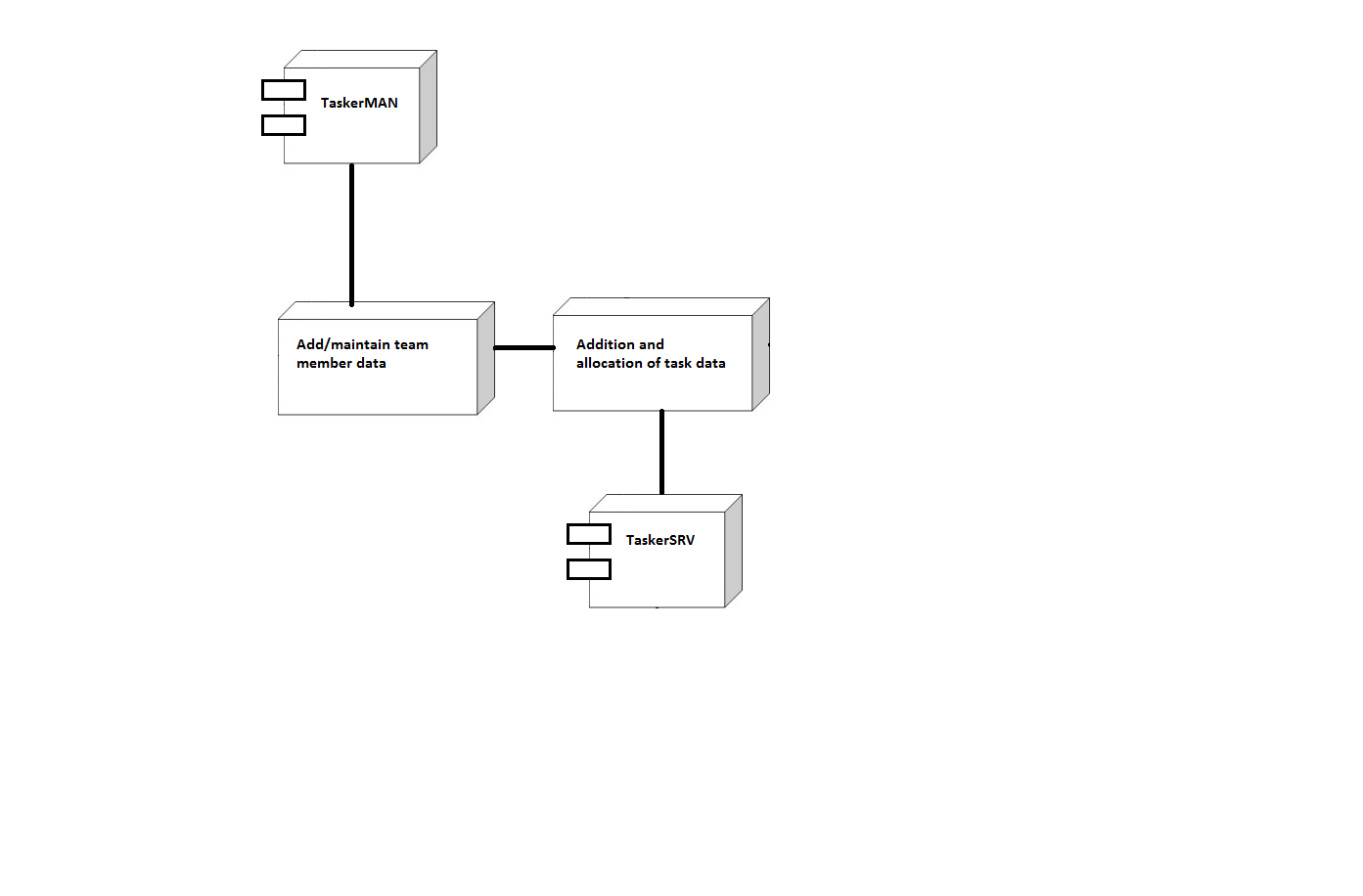
**Component Description**



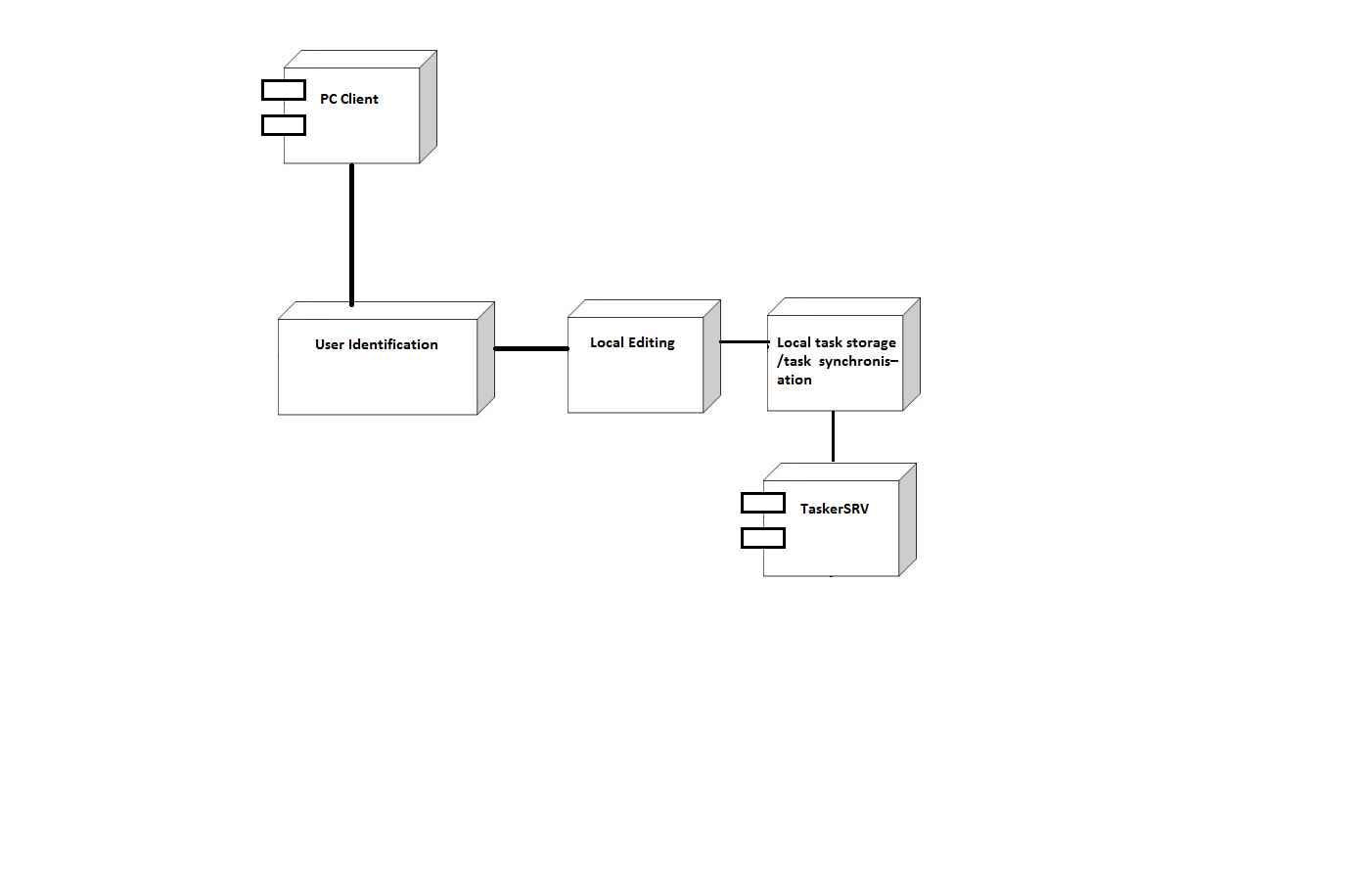
**TaskerSRV**



**TaskerMAN**



**TaskerCLI**



**4 REFERENCES**

[1] QA Document SE.QA.RS - Requirements Specification

[2] QA Document SE.QA.5a - Design Specification

**5 DOCUMENT HISTORY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | CCF# | Date | Change Description | Changed By |
| 1.0 | N/A | 20/10/15 | Document Layout Added | J.D |
| 1.1 | N/A | 26/10/15 | Compiled Draft | J.D |
| 1.2 | N/A | 28/10/15 | Formatted Document | J.D |