Software Engineering Group Project
Design Documentation

Authors:	Ben Dudley, David Fairbrother, Jonathan Englund, Josh Doyle, Liam Fitzgerald, Maurice Corriette, Oliver Earl, Tim Anderson

Config Ref:	SE_05_DEL_03
Date:	13/02/2016
Version:	2.0
Status:	Release

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB

CONTENTS

INTRODUCTION

1.1 Purpose of this Document

The purpose of this document is to describe and specify a full design for all software within this project. This will be used by software engineers to implement several components. It also lists the integration of these components to facilitate a solution matching the client's requirements [1].

1.2 Scope

This document shows a complete and full design for TaskerCLI, TaskerMAN and TaskerSRV. It lists the requirements to run each component and how they integrate and communicate to each other. It goes on to list how these components will be implemented specifying in detail algorithms where required.

1.3 Objectives

This document contains a complete view of the software solution designed. Initially the design considers high level aspects of design such as typical use cases and GUI mock ups it. It then lists classes used within these various components which are further broken down in complex classes into activity diagrams, sequence diagrams and spike work conducted. The reader should be able to visualise the overall look of the software whilst understanding the implementation at lower levels of design.

2. DEPLOYMENT DESCRIPTION

2.1 Applications in the system

2.1.1 TaskerCLI

TaskerCLI is the desktop based application in the system. The software will be written in Java and will be tested with Java 1.7.0_85 running on a Linux 64-bit Operating System. [Appendix A] - using versions of the Java Runtime Environment lower than this may cause unexpected behaviour and therefore is not recommended.

JDBC will be used to facilitate data communication. The version this software will be developed with is 4.2, utilising driver version 5.1.37.

The JUnit testing framework that is used during development will be version 4.12. This requires Java Development Kit 1.5 or above. [1]

2.1.2 TaskerMAN

TaskerMAN is the web-based software component of the system. The website will be built with HTML5, CSS (Cascading Style Sheets), JavaScript and PHP. The PHP tested during development is PHP Version 5.6.13 [Appendix B] running on an Apache server [Appendix C], running on Gentoo Linux 3.18.7 64-bit [Appendix D].

This information is also available by running *phpinfo()* on the targeted web server. [2]

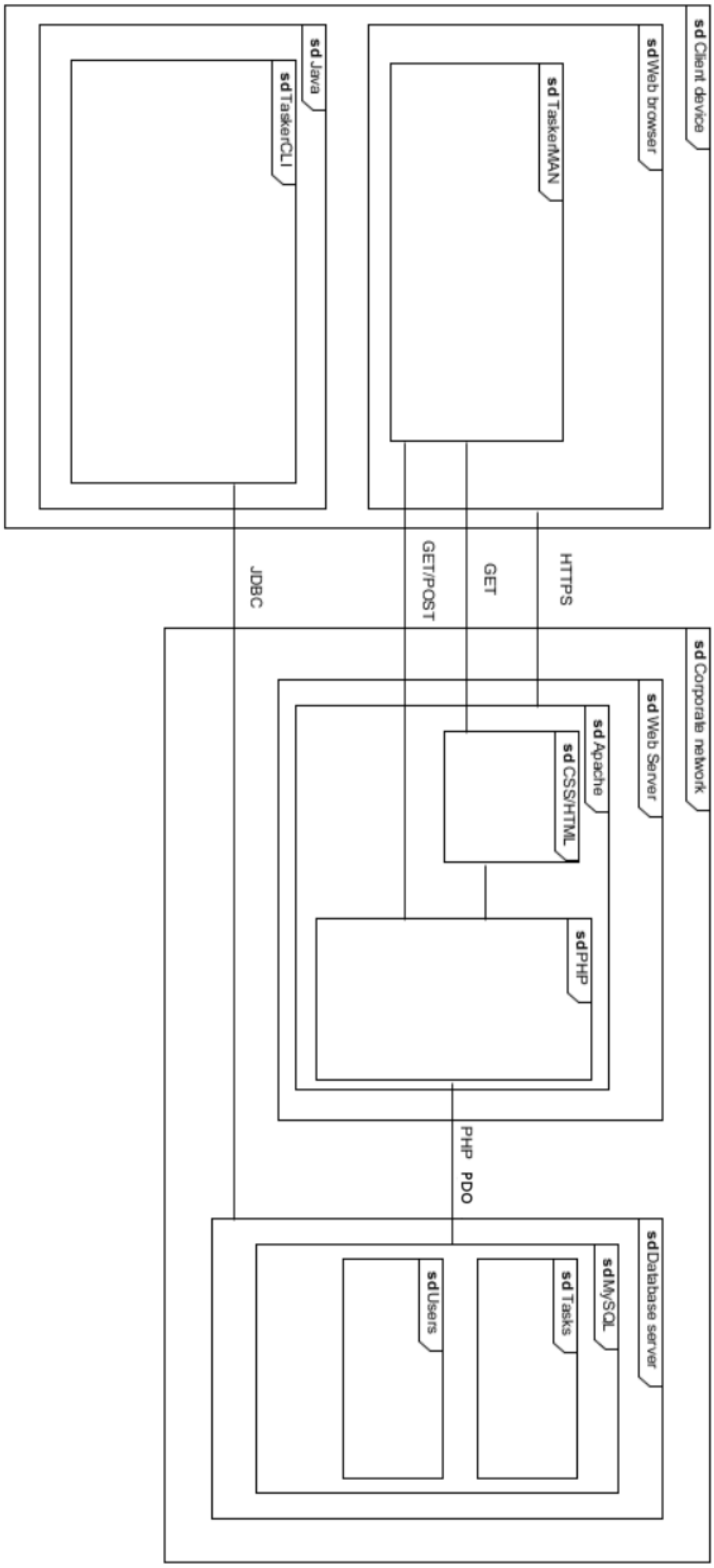
In order to enable the use of the PHPUnit testing framework, a minimal installation of PHP 5.6 is required, but the latest install is highly recommended. [3]

2.1.3 TaskerSRV

TaskerSRV is the database component.

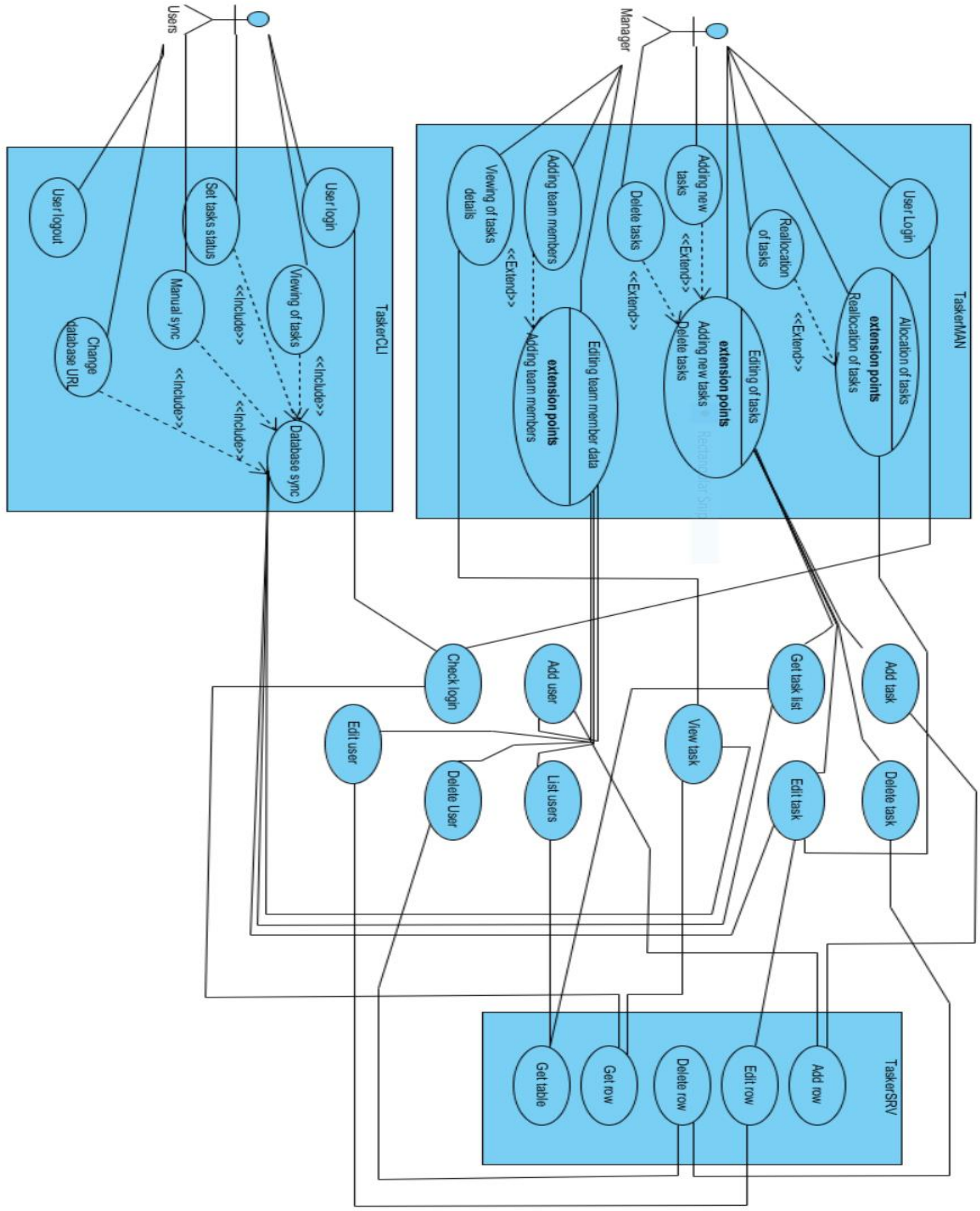
A MySQL relational database will be used. The version will be tested against is MySQL 5.6.26 on a Linux 64-bit Operating System [Appendix E]. The main system requirement for a current MySQL installation is 2.5GB of free hard disk space [4], and any disk space pertinent to the size of the database.

2.2 Application interactions



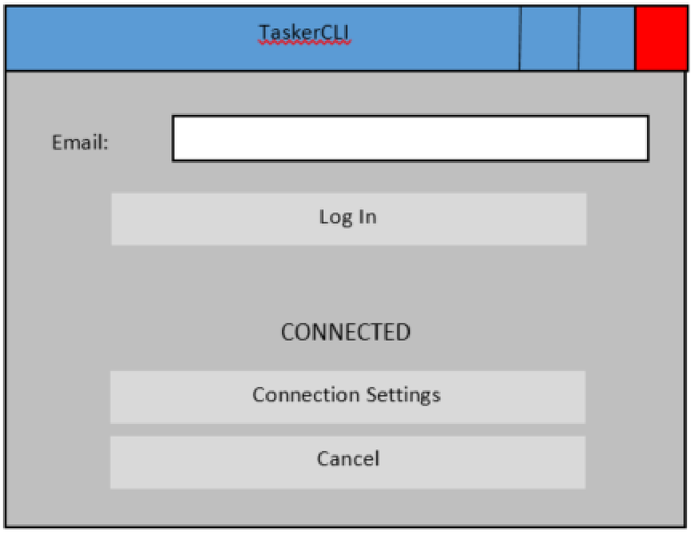
3. INTERACTION DESIGN

3.1 Use-Case Diagrams



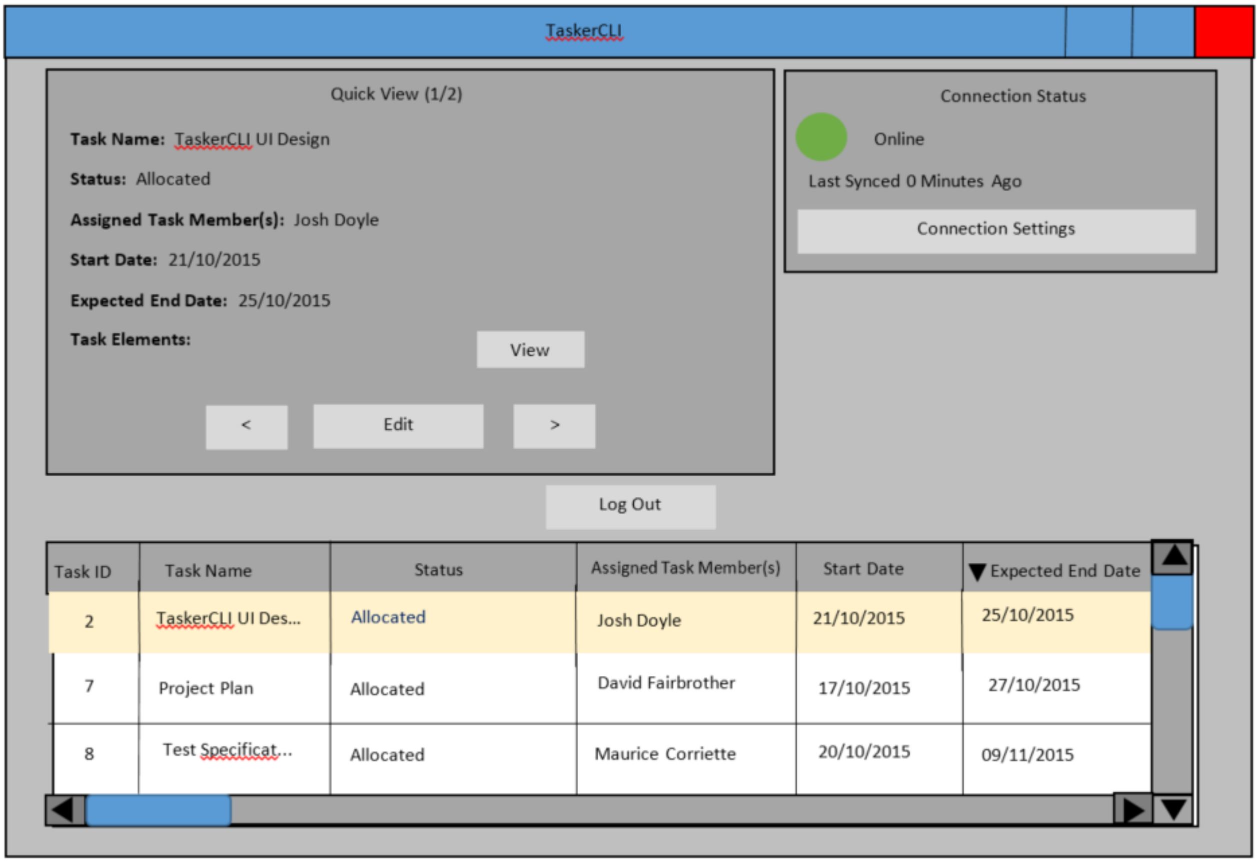
3.2 User Interface Design – Tasker CLI

3.2.1 Log In Window



- Textbox for user to enter their email address for validation, as per requirements specification. [6]
- Log In button opens 'Main Window' when clicked, provided a valid email address has been entered.
- Clicking the 'Connection Settings' button will open the 'Connection Settings' window – this allows the user to configure their connection to the *TaskerSRV* database.
- Closing 'Log in Window' will bring up the 'Exit Confirmation' window.

3.2.2 Main Window



Task Name: TaskerCLI UI Design

Status: Allocated

Assigned Task Member(s): Josh Doyle

Start Date: 21/10/2015

Expected End Date: 25/10/2015

Task Elements:

Online

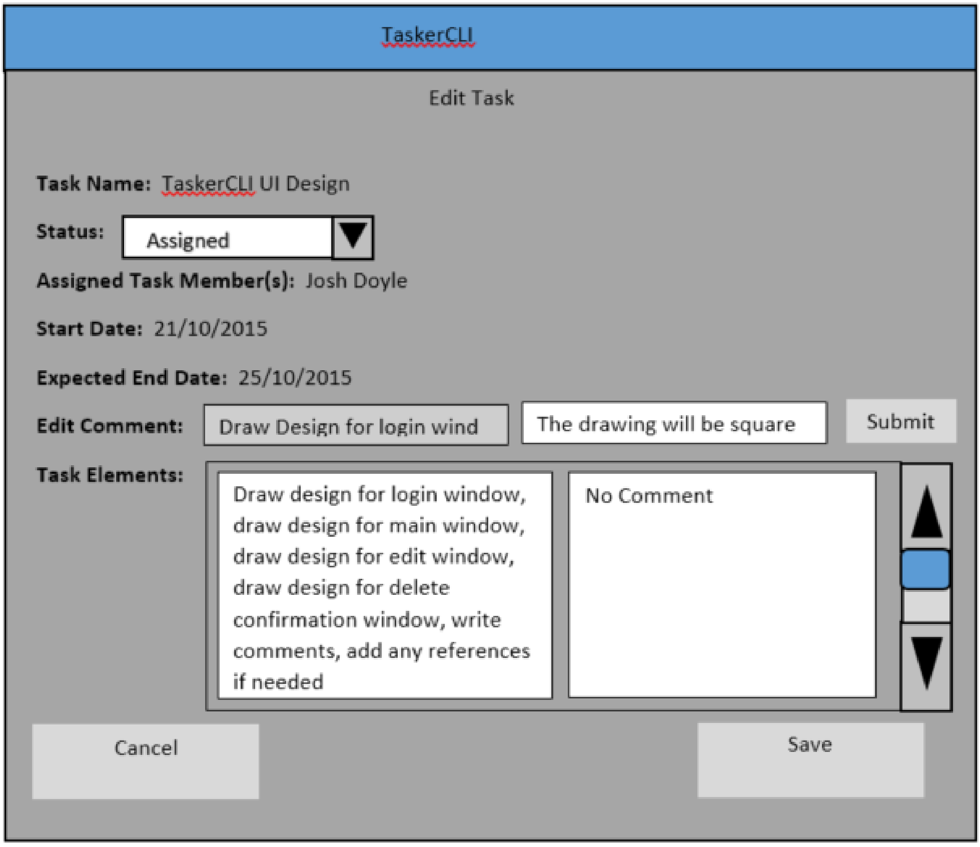
Last Synced 0 Minutes Ago

Task ID	Task Name	Status	Assigned Task Member(s)	Start Date	▼ Expected End Date
2	TaskerCLI UI Des...	Allocated	Josh Doyle	21/10/2015	25/10/2015
7	Project Plan	Allocated	David Fairbrother	17/10/2015	27/10/2015
8	Test Specificat...	Allocated	Maurice Corriette	20/10/2015	09/11/2015

- The table at the bottom of the ‘Main’ window shows all of the tasks currently saved in *TaskerSRV*, the last time the program was synchronised with the database.
    - Clicking on the headings at the top of the columns in the table, will order the table based upon the values in that column. The design shown is in descending order based upon the *Expected End Date* column.
    - Checking the checkboxes next to each task, enables the user to select multiple tasks.
    - The scrollbars are used to navigate the table.
    - Selected tasks are shown in more detail in the ‘Quick View’ panel.
- The ‘Quick View’ panel at the top left of the ‘Main’ window presents the data from the tasks selected from the table.
    - Clicking the arrow keys at the bottom of the ‘Quick View’ panel navigates between all tasks selected from the table at the bottom of the ‘Main’ window.
    - Clicking the ‘Edit’ button opens the ‘Edit’ window, to change completion status and task elements of the current task in the ‘Quick View’ panel.
- The Connection Status panel at the top right of the ‘Main’ window status changes colour depending on the connection status.
    - Green indicates that *TaskerCLI* is currently connected to *TaskerSRV* and that everything is synchronised.
    - Red indicates that the connection between *TaskerCLI* and *TaskerSRV* has been lost and that synchronisation is no longer guaranteed.

  - In the demonstrated design, *TaskerCLI* is connected to *TaskerSRV* and has been synchronised less than a minute ago. The number of minutes increments every minute and returns to 0 after successful synchronisation.
- Closing the window brings up the 'Exit Confirmation' window.

3.2.3 Edit Window



Task Name: TaskerCLI UI Design

Status: Assigned

Assigned Task Member(s): Josh Doyle

Start Date: 21/10/2015

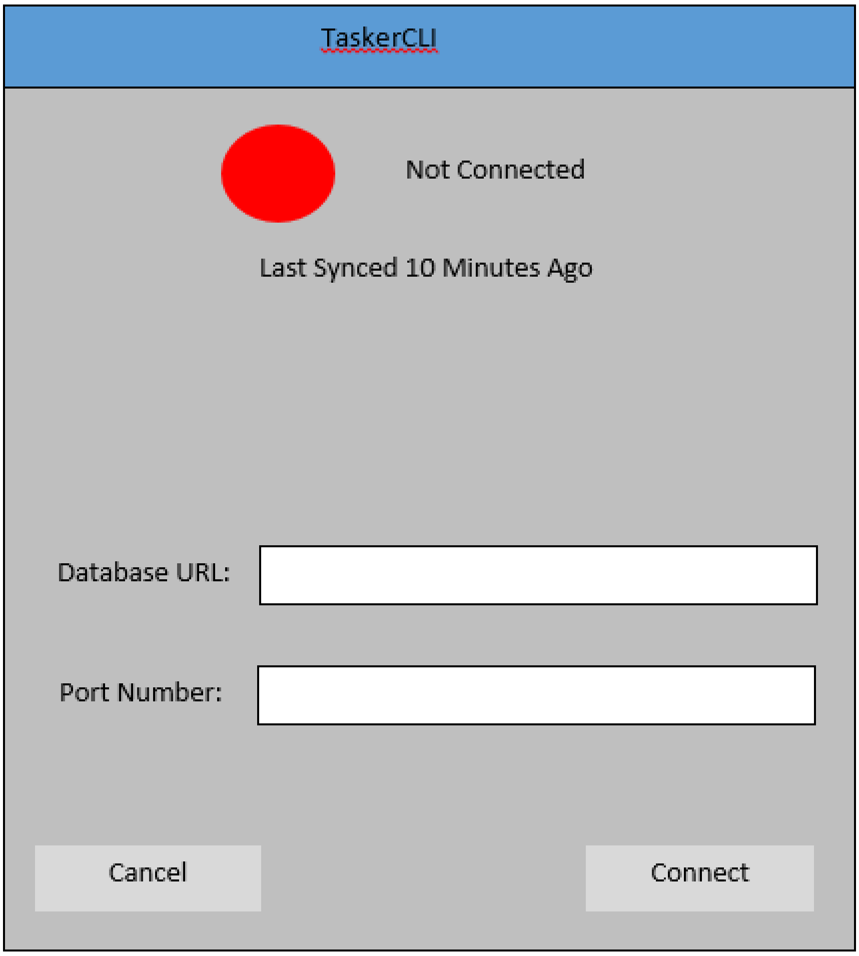
Expected End Date: 25/10/2015

Edit Comment: Draw Design for login wind | The drawing will be square

Task Elements: Draw design for login window, draw design for main window, draw design for edit window, draw design for delete confirmation window, write comments, add any references if needed | No Comment

- The 'Edit Task' window is populated with the data of the task that was in the 'Quick View' panel on the 'Main' window when it was opened.
- The attributes of completion status and task elements are editable from this window.
- The completion status can be selected from a dropdown list. The default value is 'Assigned.'
- The task elements can be changed by typing into the Task Elements textbox.
  - A scrollbar will only appear if the text entry exceeds the size of the textbox.
- When the Save button is clicked, the 'Edit Task' window is closed and the task attributes are updated with their new values.
- Choosing Cancel simply closes the window with no changes.

3.2.4 Connection Settings Window



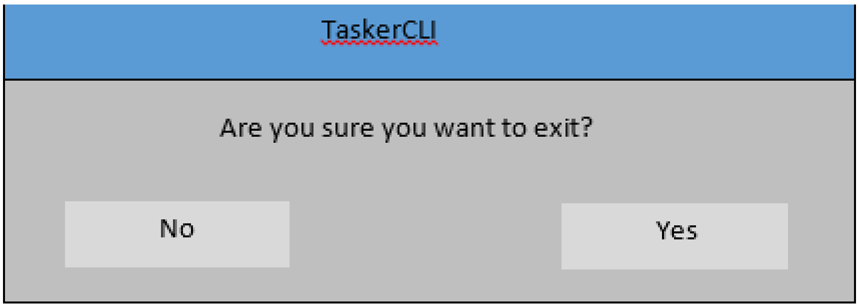
TaskerCLI

Not Connected

Last Synced 10 Minutes Ago

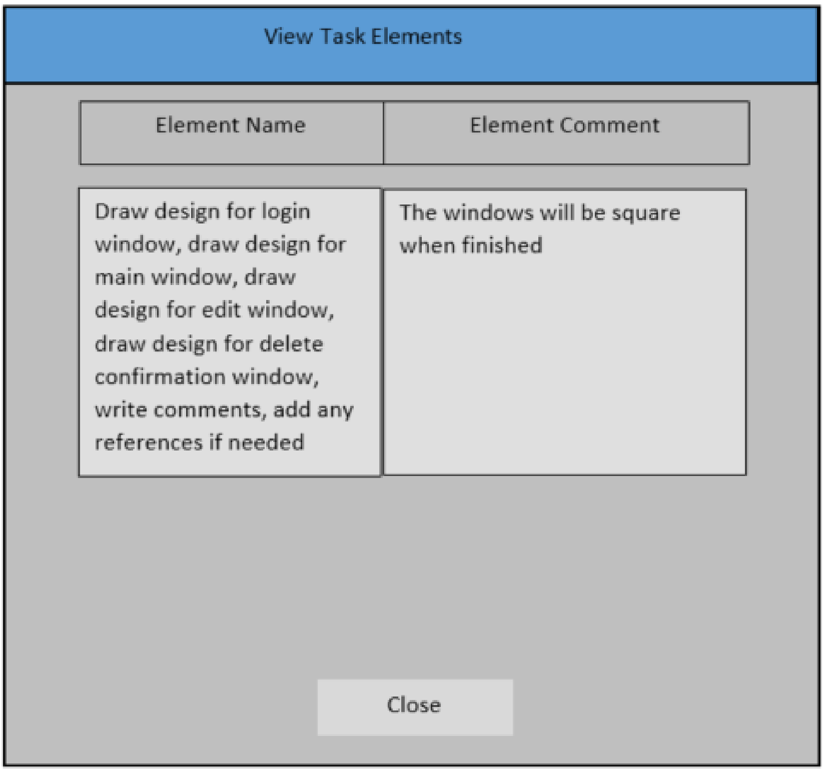
- The connection status text at the top of the window shows the current connection state of *TaskerCLI.*
    - The coloured circle is red when there is no connection established to *TaskerSRV.*
        - Consequently the coloured circle appears green when a connection is successfully established.
    - The time since last sync shows how much time has passed since the last synchronisation.
    - In this design, *TaskerCLI* is not connected to *TaskerSRV* and it has been 10 minutes since the last successful synchronisation.
- The Database URL and Port Number are entered into the respective fields to provide information for connecting to the *TaskerSRV* database.
- Choosing ‘Cancel’ simply closes the window without saving any information.
- Choosing ‘Connect’ will instruct *TaskerCLI* to attempt to connect using the information provided.
- Default window controls and clicking outside of the window are disabled to prevent the user from opening multiple instances of this window and attempting to cause simultaneous connections to be established.

3.2.5 Exit Confirmation Window



- If ‘No’ is selected, the window is closed and the user regains control of the window they were previously using.
- If ‘Yes’ is selected, *TaskerCLI* closes.
- Default window controls are disabled to make it clear to the user that their attention is required and that a decision must be made.
- Clicking away from the window to bring another window into focus is also disabled, to stop the user spawning multiple instances of the ‘Exit Confirmation’ window.

3.2.6 View Tasks Window



Element Name	Element Comment
Draw design for login window, draw design for main window, draw design for edit window, draw design for delete confirmation window, write comments, add any references if needed	The windows will be square when finished

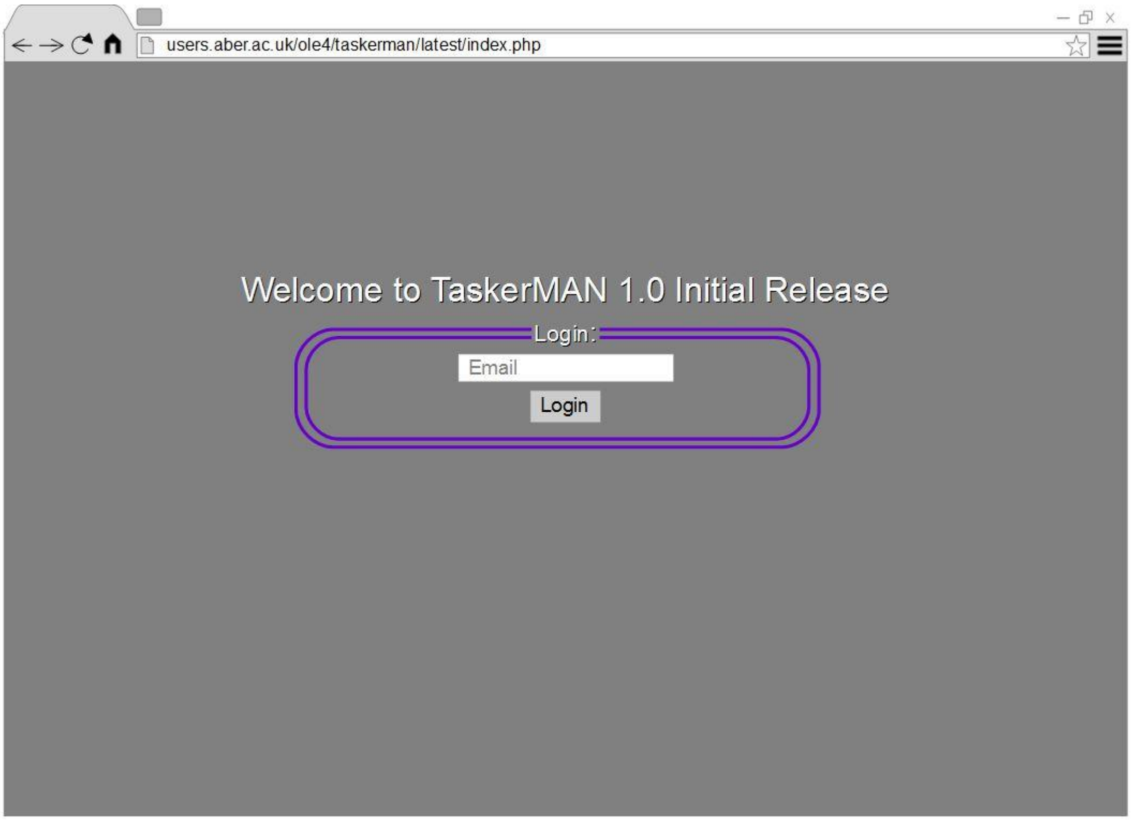
- Grid layout of element and comment pars displayed
- If no elements are associated with task display ‘No Element’ and ‘No Comment’
- Multiple element comment pairs will display in table format.

3.3 User Interface Design - TaskerMAN

3.3.1 General Notes

Mozilla Firefox is used as an example web browser in the following images. [6]

3.3.2 Login Page

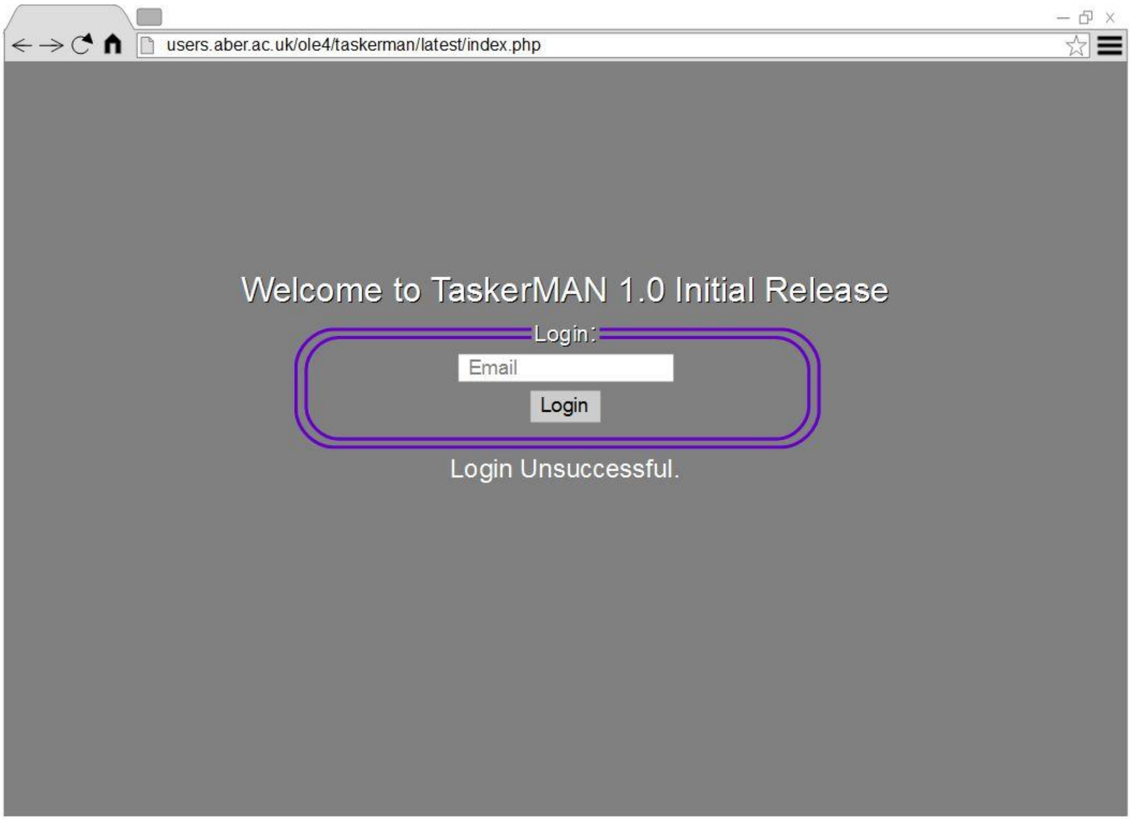


users.aber.ac.uk/ole4/taskerman/latest/index.php

Welcome to TaskerMAN 1.0 Initial Release

Login:

Email



users.aber.ac.uk/ole4/taskerman/latest/index.php

Welcome to TaskerMAN 1.0 Initial Release

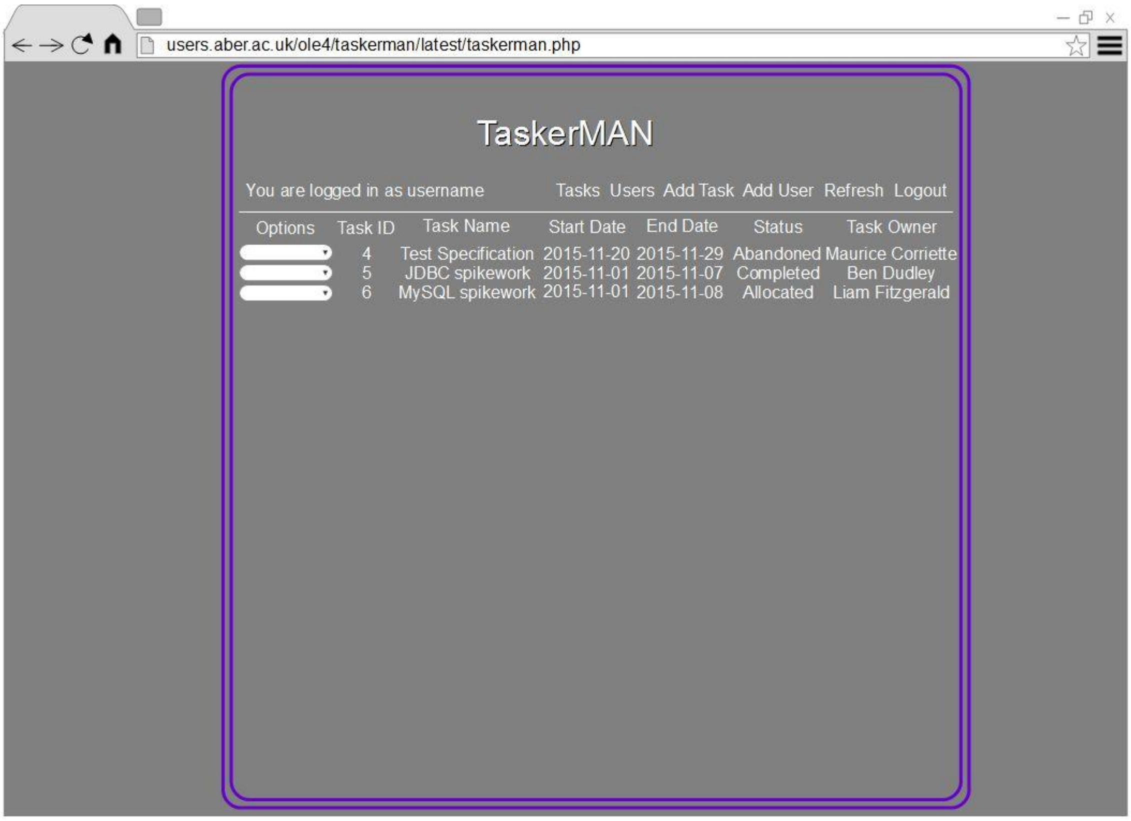
Login:

Email

Login Unsuccessful.

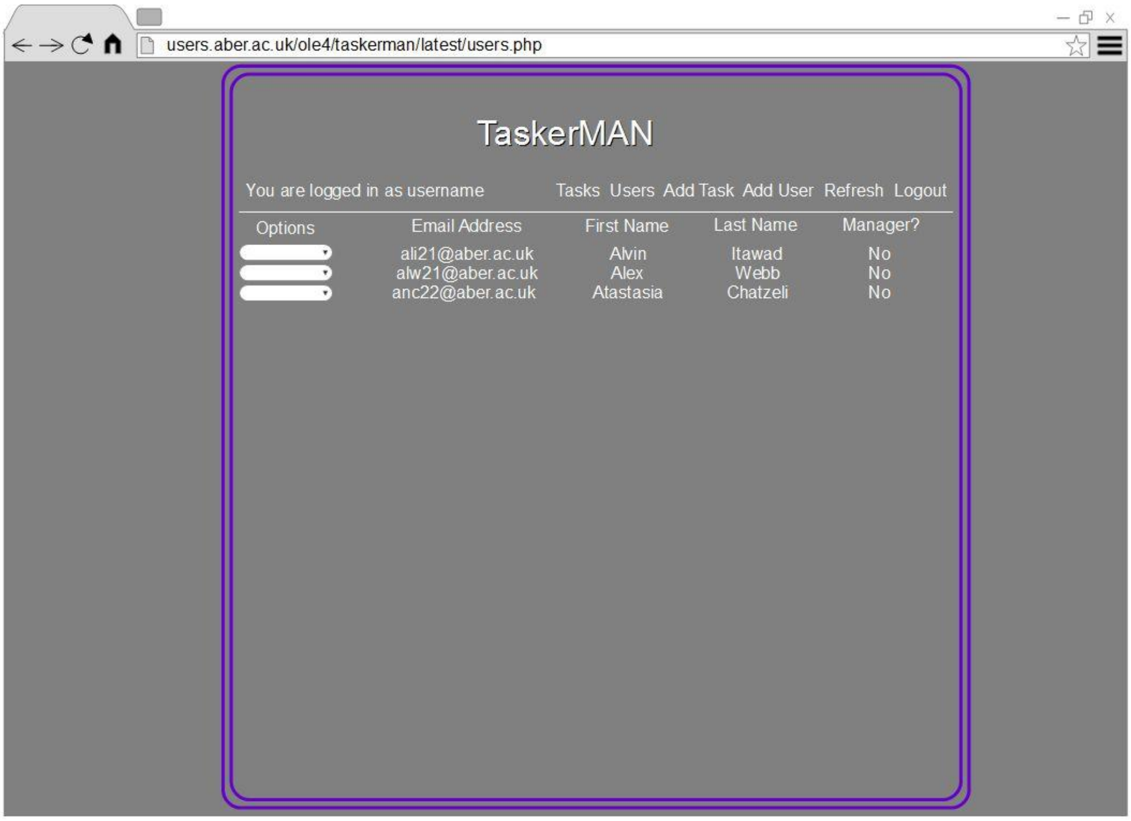
- If a valid email address is entered, the user will be directed to the main page – otherwise access is prohibited, as is required. [6]
- It is also required that the email address belongs to a manager. If this is not the case the user will be denied access.

3.3.3 TaskerMAN Task / User View



You are logged in as username

Options	Task ID	Task Name	Start Date	End Date	Status	Task Owner
	4	Test Specification	2015-11-20	2015-11-29	Abandoned	Maurice Corriette
	5	JDBC spikework	2015-11-01	2015-11-07	Completed	Ben Dudley
	6	MySQL spikework	2015-11-01	2015-11-08	Allocated	Liam Fitzgerald

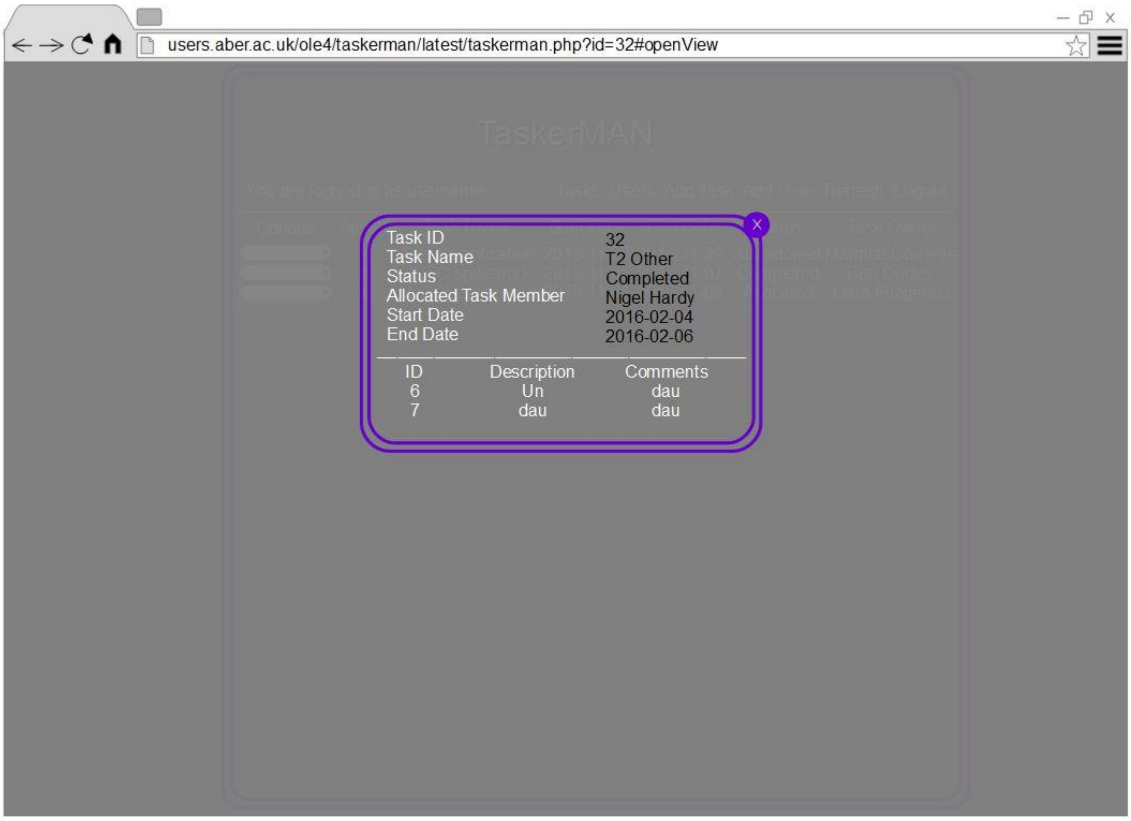


You are logged in as username

Options	Email Address	First Name	Last Name	Manager?
	ali21@aber.ac.uk	Alvin	Itawad	No
	alw21@aber.ac.uk	Alex	Webb	No
	anc22@aber.ac.uk	Atastasia	Chatzeli	No

- Displays the user’s name at the top of the screen, indicating who is currently logged in.
- Tasks and Users change the current TaskerMAN view to the other options.
- Add Task displays the modal window to add a new task to the database.
- Add User displays the modal window to add a new user to the database.
- Drop down next to individual users and tasks to allow users to view a user, task or its elements and edit them.
- Refresh reloads the TaskerMAN interface and retrieves any new data from the database.
- The Logout link simply logs the user out of the system, redirecting them to the Login page.

3.3.4 View Elements / View Task



users.aber.ac.uk/ole4/taskerman/latest/taskerman.php?id=32#openView

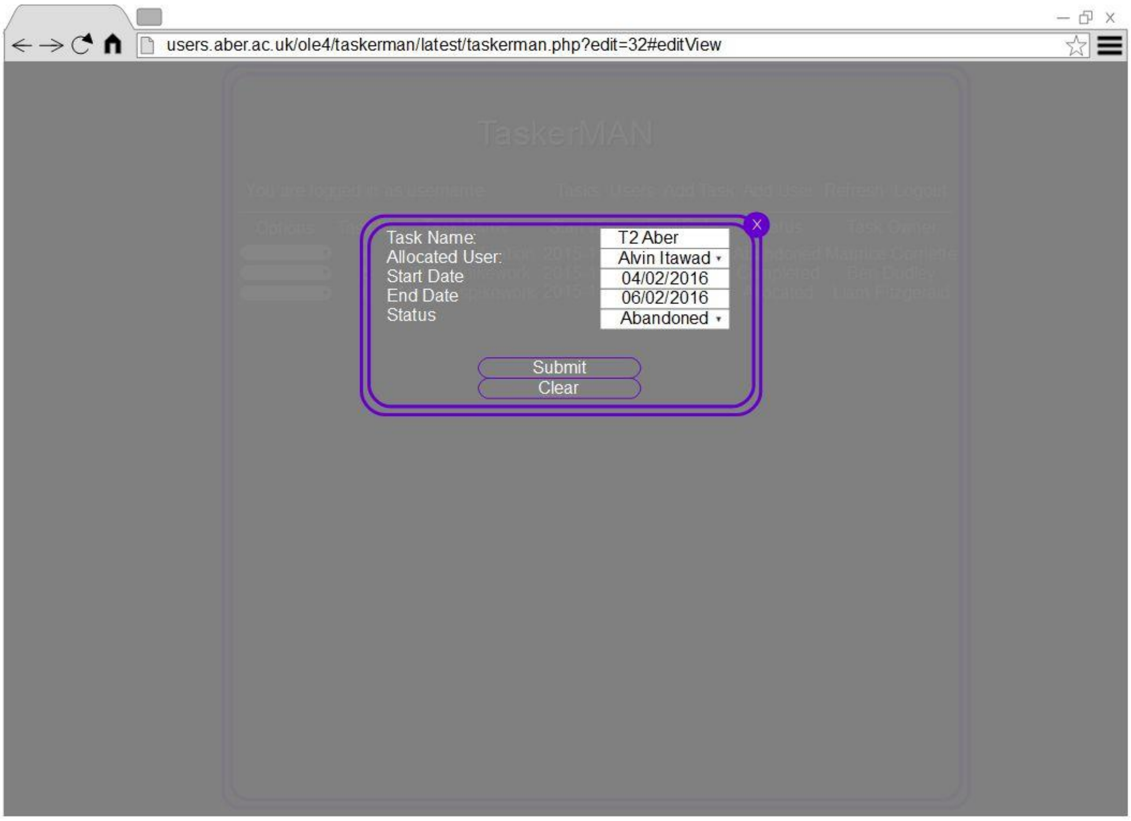
Task ID	32
Task Name	T2 Other
Status	Completed
Allocated Task Member	Nigel Hardy
Start Date	2016-02-04
End Date	2016-02-06

ID	Description	Comments
6	Un	dau
7	dau	dau

TaskerMAN
You are logged in as username
Tasks Users Add Task Add User Refresh Logout
Options
Task Owner
X

- Displays selected tasks elements.
- Read-only – data is not editable here.

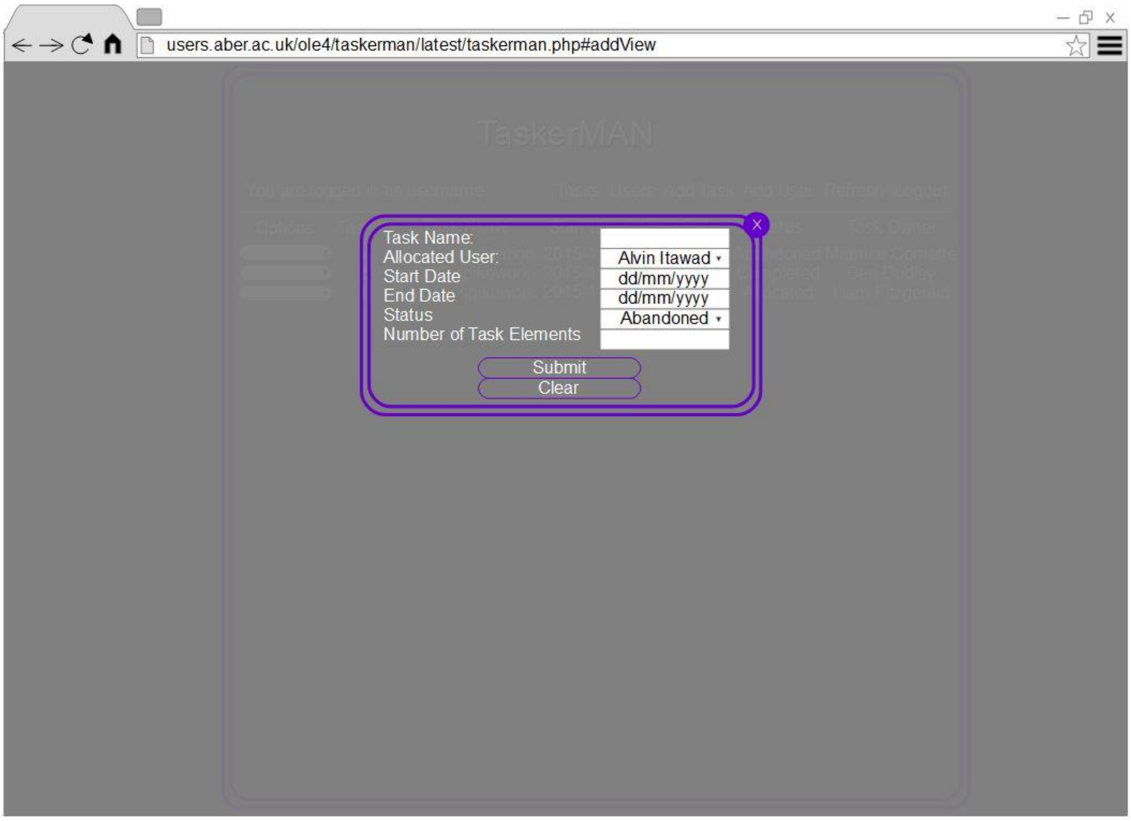
3.3.5 Edit Task Overlay


users.aber.ac.uk/ole4/taskerman/latest/taskerman.php?edit=32#editView

Task Name: T2 Aber
Allocated User: Alvin Itawad
Start Date 04/02/2016
End Date 06/02/2016
Status Abandoned

- All fields here are editable, allowing complete editing of the task.
  - It does not allow the editing of elements as these are contained in a different window.
  - Data is validated and sanitised to prevent invalid input.
- Submit will submit the data. Clear blanks the form.
- Closing the modal window will cancel any changes.
- The database will be updated after each edit, refreshing TaskerMAN in the process.

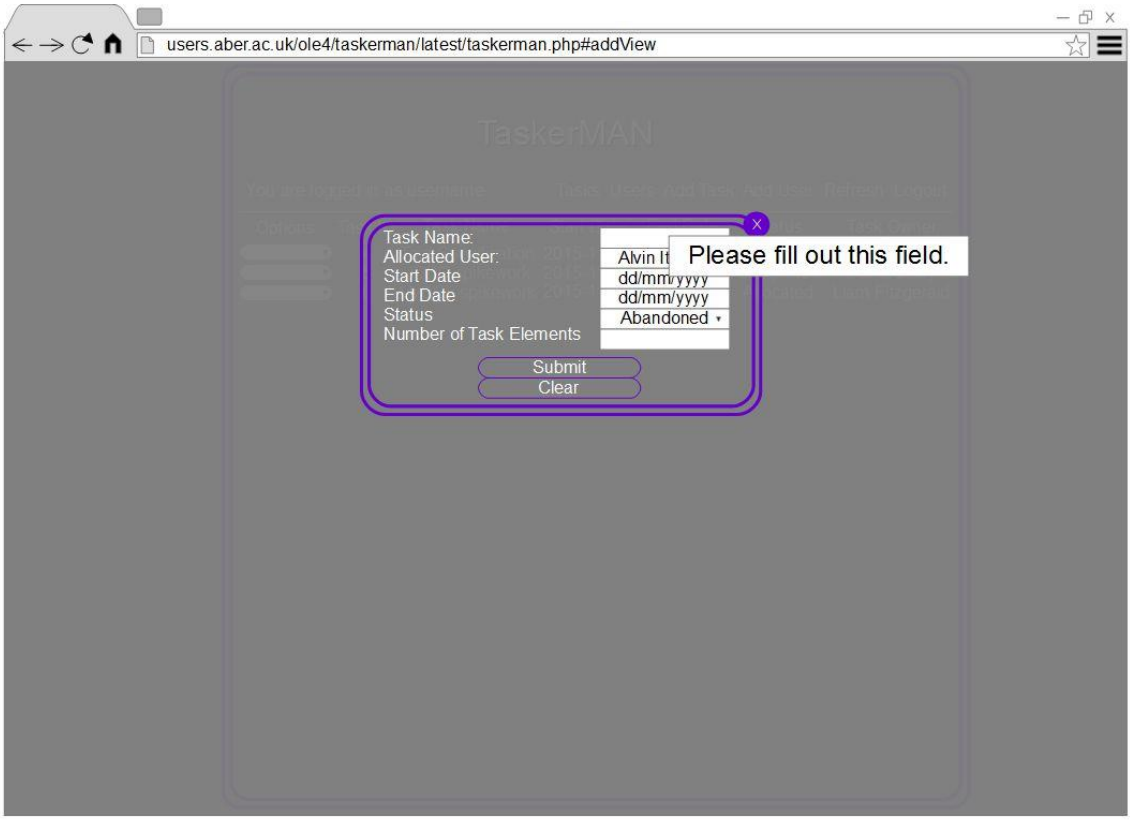
3.3.6 Add Task Overlay



users.aber.ac.uk/ole4/taskerman/latest/taskerman.php#addView

Task Name:
Allocated User: Alvin Itawad
Start Date dd/mm/yyyy
End Date dd/mm/yyyy
Status Abandoned
Number of Task Elements

TaskerMAN
You are logged in as username
Tasks Users Add Task Add User Refresh Logout
X
Submit
Clear



users.aber.ac.uk/ole4/taskerman/latest/taskerman.php#addView

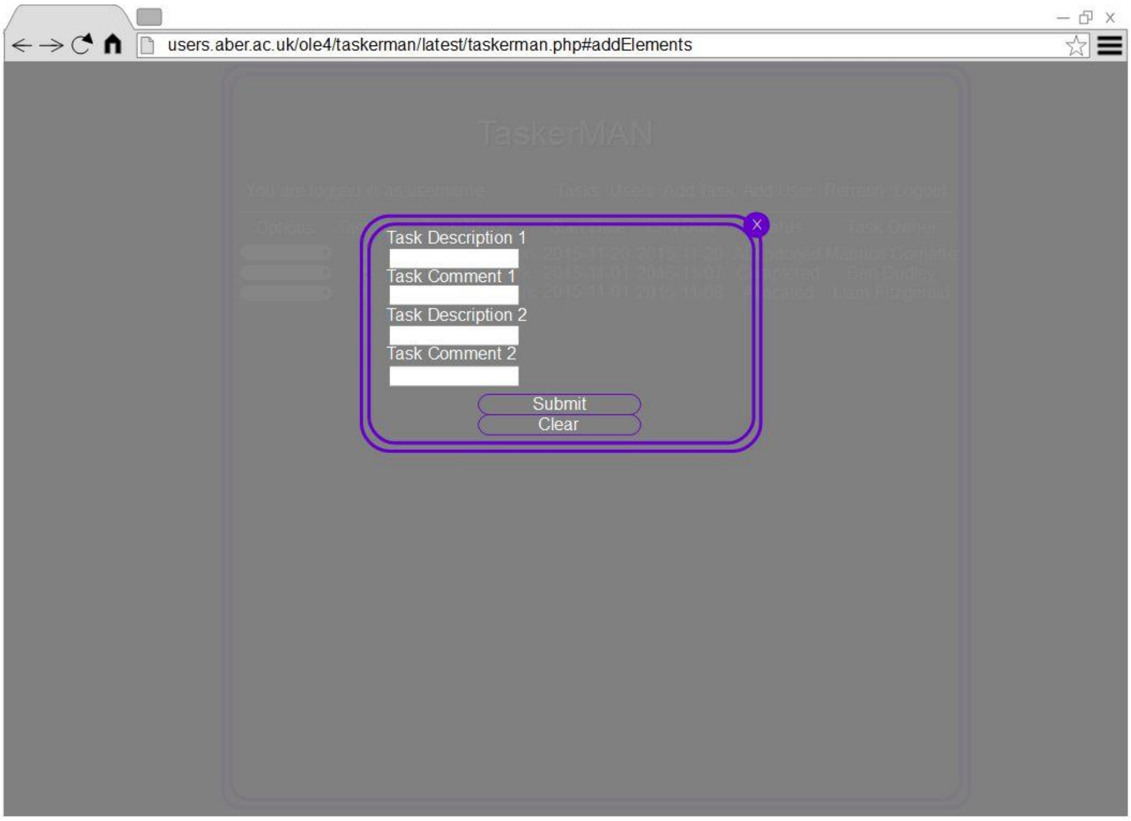
Task Name:
Allocated User: Alvin It
Start Date dd/mm/yyyy
End Date dd/mm/yyyy
Status Abandoned
Number of Task Elements

Please fill out this field.

TaskerMAN
You are logged in as username
Tasks Users Add Task Add User Refresh Logout
X
Submit
Clear

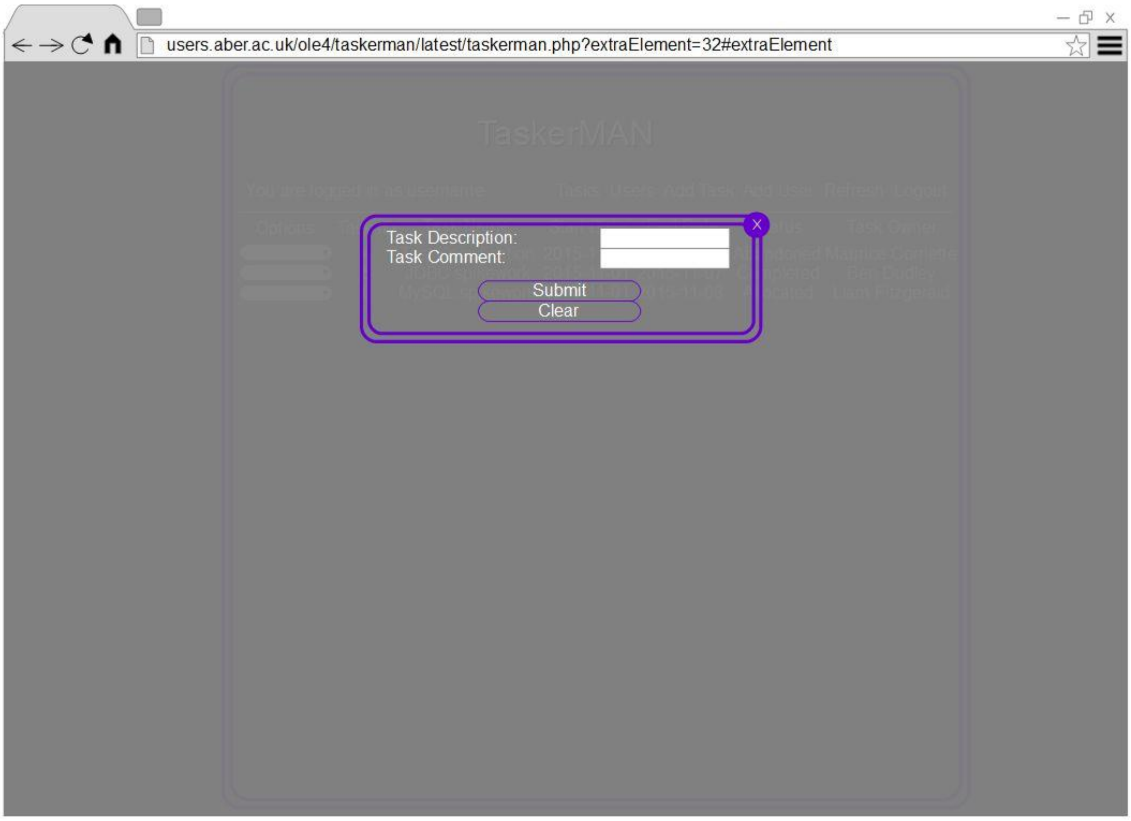
- In visual likeness to the ‘Edit Task’ screen, except blank task details and allows for a completely new entry to be added.
- Option for number of task elements allows the user to add more elements on another modal window that appears afterwards.
- Validation and sanitisation will be used to ensure only correct/meaningful data can be entered.

3.3.7 Add Tasks – Elements Window


users.aber.ac.uk/ole4/taskerman/latest/taskerman.php#addElements
TaskerMAN
You are logged in as username
Tasks Users Add Task Add User Refresh Logout
x
Task Description 1
Task Comment 1
Task Description 2
Task Comment 2
Submit
Clear

- Number of text input boxes for task element’s descriptions and comments is determined by the number of task elements inputted in the previous window.
- This is capped at five elements at a time.
- Inputs are validated and sanitised to ensure valid input.
- The user can add additional elements later via the Add Element option.

3.3.8 Add Element



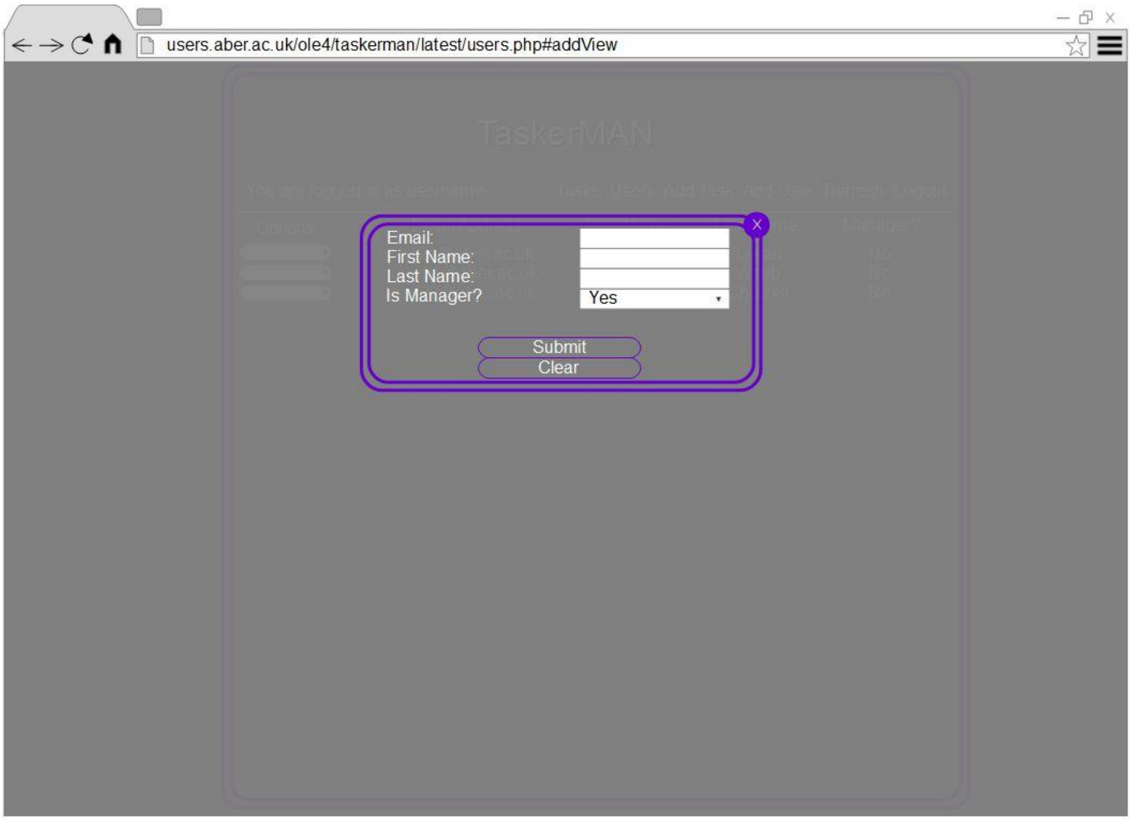
users.aber.ac.uk/ole4/taskerman/latest/taskerman.php?extraElement=32#extraElement

Task Description:
Task Comment:

TaskerMAN
You are logged in as username
Tasks Users Add Task Add User Refresh Logout
Options
Task Owner
X
Submit
Clear

- Allows adding a single extra task – both the description and task comment
- Sanitised and validated to ensure valid input

3.3.9 Add User

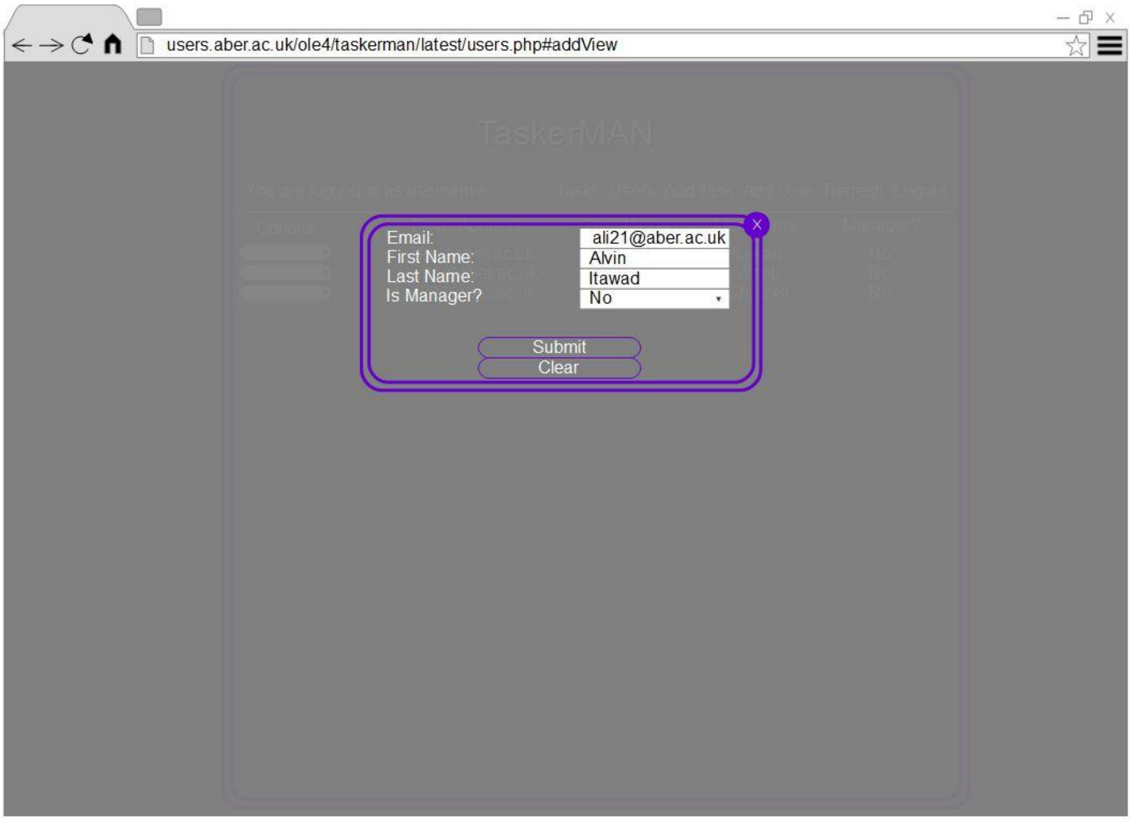


users.aber.ac.uk/ole4/taskerman/latest/users.php#addView

Email:
First Name:
Last Name:
Is Manager? Yes

- Allows adding a new user to the system including setting their managerial status
- Validated and sanitised

3.3.10 Edit User



users.aber.ac.uk/ole4/taskerman/latest/users.php#addView

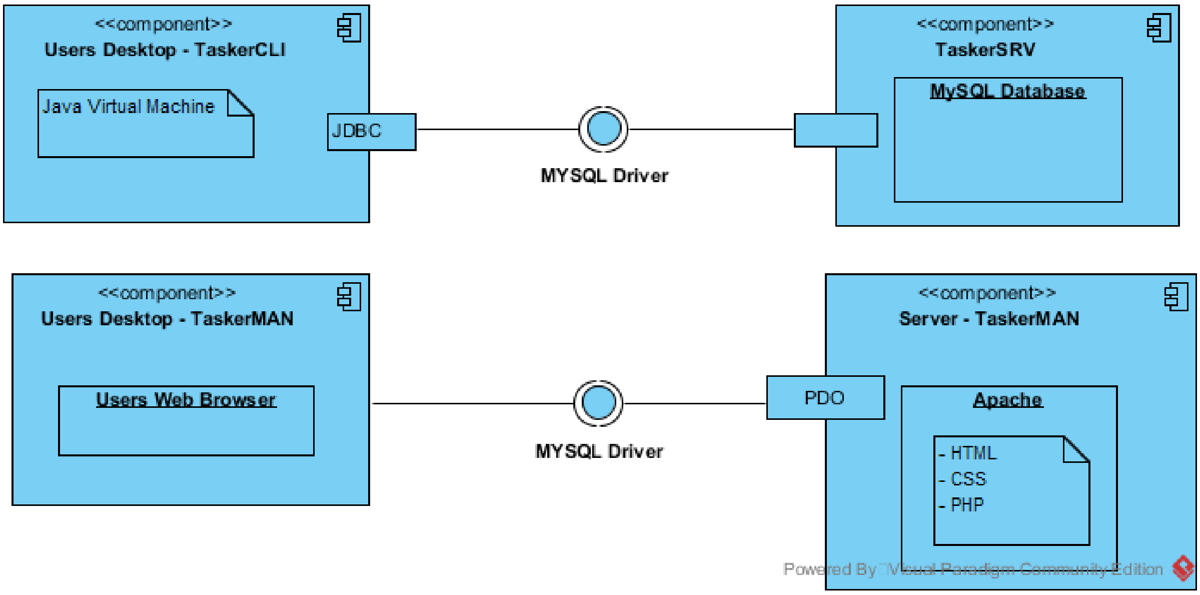
Email: ali21@aber.ac.uk
First Name: Alvin
Last Name: Itawad
Is Manager? No

TaskerMAN
You are logged in as username
Tasks Users Add Task Add User Refresh Logout
Options
Manager?
Submit
Clear

- Allows user to edit an existing user in the system
- This includes modifying their managerial status
- Validation and sanitisation are used to ensure data is valid

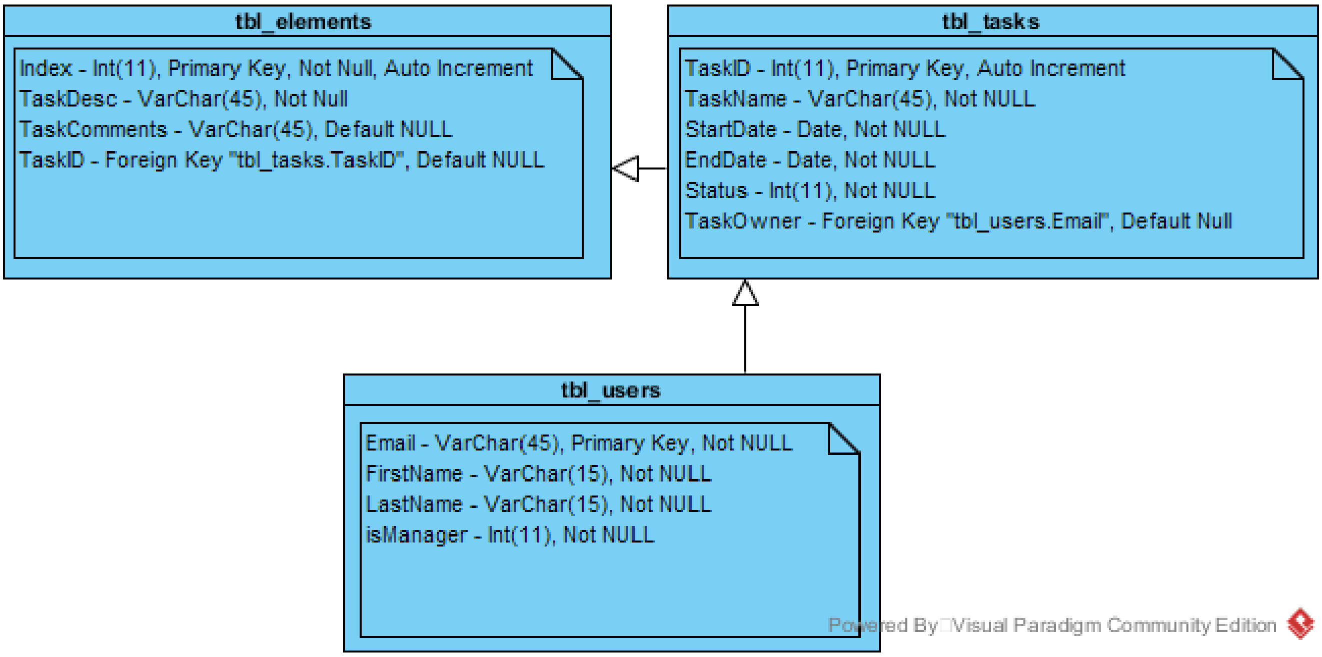
4. COMPONENT DESCRIPTION

By using standard libraries to connect various components to TaskerSRV we can utilise standard protocols provided by the MYSQL driver avoiding the need to specify interfaces for inter component communication. Clients will initiate the connection to TaskerSRV and handle the connection through JDBC and PDO for TaskerCLI and TaskerMAN respectively.



4.1 TaskerSRV Database Design

Using this design the database in TaskerSRV must use a standard naming scheme and have fixed properties. These are listed in the diagram below.



tbl_elements

Index - Int(11), Primary Key, Not Null, Auto Increment
TaskDesc - VarChar(45), Not Null
TaskComments - VarChar(45), Default NULL
TaskID - Foreign Key "tbl_tasks.TaskID", Default NULL

tbl_tasks

TaskID - Int(11), Primary Key, Auto Increment
TaskName - VarChar(45), Not NULL
StartDate - Date, Not NULL
EndDate - Date, Not NULL
Status - Int(11), Not NULL
TaskOwner - Foreign Key "tbl_users.Email", Default Null

tbl_users

Email - VarChar(45), Primary Key, Not NULL
FirstName - VarChar(15), Not NULL
LastName - VarChar(15), Not NULL
isManager - Int(11), Not NULL

5. SIGNIFICANT CLASSES

5.1 TaskerCLI

TaskerCLI classes can be broken down into functional groups. Classes which handle data including editing, database synchronisation and ordering are grouped as “Logic Classes”. The remaining Classes are used to power the GUI such as getting user inputs and displaying or closing windows, these are grouped as “GUI Classes”

5.1.1 Logic Class Diagram

See Appendix F – Logic Class Diagram.

The classes and descriptions are as follows:

Database: - Holds a JDBC connection and performs execution of SQL statements in order to both send and receive data to TaskerSRV.

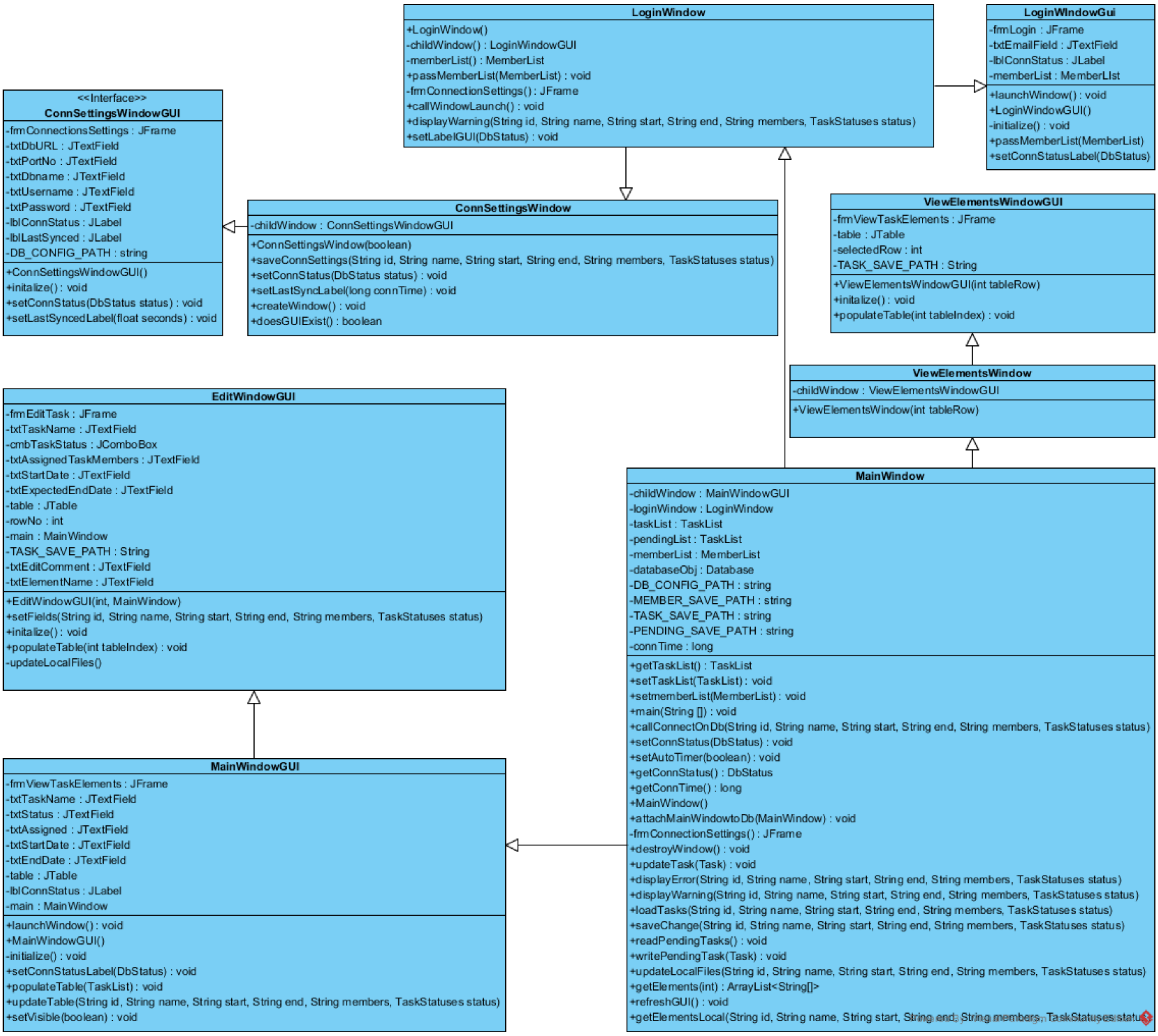
Task: - Represents an individual task and holds the name, elements, dates and assigned users of a task.

Member: - Represents an individual user of the system. Contains the users email address and name.

MemberList: - A class which holds all members found in TaskerSRV as Member objects.

TaskList: - A class which holds all tasks found in TaskerSRV as Task objects.

5.1.2 GUI Class Diagram


LoginWindow
+LoginWindow()
-childWindow() : LoginWindowGUI
-memberList() : MemberList
+passMemberList(MemberList) : void
-frmConnectionSettings() : JFrame
+callWindowLaunch() : void
+displayWarning(String id, String name, String start, String end, String members, TaskStatuses status)
+setLabelGUI(DbStatus) : void
LoginWIndowGui
-frmLogin : JFrame
-txtEmailField : JTextField
-lblConnStatus : JLabel
-memberList : MemberLIst
+launchWindow() : void
+LoginWindowGUI()
-initialize() : void
+passMemberList(MemberList)
+setConnStatusLabel(DbStatus)
<<Interface>>
ConnSettingsWindowGUI
-frmConnectionsSettings : JFrame
-txtDbURL : JTextField
-txtPortNo : JTextField
-txtDbname : JTextField
-txtUsername : JTextField
-txtPassword : JTextField
-lblConnStatus : JLabel
-lblLastSynced : JLabel
-DB_CONFIG_PATH : string
+ConnSettingsWindowGUI()
+initalize() : void
+setConnStatus(DbStatus status) : void
+setLastSyncedLabel(float seconds) : void
ConnSettingsWindow
-childWindow : ConnSettingsWindowGUI
+ConnSettingsWindow(boolean)
+saveConnSettings(String id, String name, String start, String end, String members, TaskStatuses status)
+setConnStatus(DbStatus status) : void
+setLastSyncLabel(long connTime) : void
+createWindow() : void
+doesGUIExist() : boolean
ViewElementsWindowGUI
-frmViewTaskElements : JFrame
-table : JTable
-selectedRow : int
-TASK_SAVE_PATH : String
+ViewElementsWindowGUI(int tableRow)
+initalize() : void
+populateTable(int tableIndex) : void
ViewElementsWindow
-childWindow : ViewElementsWindowGUI
+ViewElementsWindow(int tableRow)
EditWindowGUI
-frmEditTask : JFrame
-txtTaskName : JTextField
-cmbTaskStatus : JComboBox
-txtAssignedTaskMembers : JTextField
-txtStartDate : JTextField
-txtExpectedEndDate : JTextField
-table : JTable
-rowNo : int
-main : MainWindow
-TASK_SAVE_PATH : String
-txtEditComment : JTextField
-txtElementName : JTextField
+EditWindowGUI(int, MainWindow)
+setFields(String id, String name, String start, String end, String members, TaskStatuses status)
+initalize() : void
+populateTable(int tableIndex) : void
-updateLocalFiles()
MainWindow
-childWindow : MainWindowGUI
-loginWindow : LoginWindow
-taskList : TaskList
-pendingList : TaskList
-memberList : MemberList
-databaseObj : Database
-DB_CONFIG_PATH : string
-MEMBER_SAVE_PATH : string
-TASK_SAVE_PATH : string
-PENDING_SAVE_PATH : string
-connTime : long
+getTaskList() : TaskList
+setTaskList(TaskList) : void
+setmemberList(MemberList) : void
+main(String []) : void
+callConnectOnDb(String id, String name, String start, String end, String members, TaskStatuses status)
+setConnStatus(DbStatus) : void
+setAutoTimer(boolean) : void
+getConnStatus() : DbStatus
+getConnTime() : long
+MainWindow()
+attachMainWindowtoDb(MainWindow) : void
-frmConnectionSettings() : JFrame
+destroyWindow() : void
+updateTask(Task) : void
+displayError(String id, String name, String start, String end, String members, TaskStatuses status)
+displayWarning(String id, String name, String start, String end, String members, TaskStatuses status)
+loadTasks(String id, String name, String start, String end, String members, TaskStatuses status)
+saveChange(String id, String name, String start, String end, String members, TaskStatuses status)
+readPendingTasks() : void
+writePendingTask(Task) : void
+updateLocalFiles(String id, String name, String start, String end, String members, TaskStatuses status)
+getElements(int) : ArrayList<String[]>
+refreshGUI() : void
+getElementsLocal(String id, String name, String start, String end, String members, TaskStatuses status
MainWindowGUI
-frmViewTaskElements : JFrame
-txtTaskName : JTextField
-txtStatus : JTextField
-txtAssigned : JTextField
-txtStartDate : JTextField
-txtEndDate : JTextField
-table : JTable
-lblConnStatus : JLabel
-main : MainWindow
+launchWindow() : void
+MainWindowGUI()
-initialize() : void
+setConnStatusLabel(DbStatus) : void
+populateTable(TaskList) : void
+updateTable(String id, String name, String start, String end, String members, TaskStatuses status)
+setVisible(boolean) : void

5.1.3 Logic Classes Interface List

Task:

```
String getID();
String getName();
String getStart();
String getEnd();
String getMembers();
String getStatus();
Void setStatus(TaskStatuses newStatus);
Void addElement(String elementName, String elementComment, String index);
Void clearAllElements();
Element getElement(int index);
ArrayList<Element> getAllElements();
```

Member:

```
String getName();
String getEmail();
void setName();
void setEmail();
```

MemberList:

```
Members getMember(int index);
Void addMember(Members member);
Void loadMembers(String filename);
Boolean memberExists(String email);
Int getLength();
```

TaskList:

```
ArrayList<Task> getTaskList();
void addTask(Task task);
void setAssignedTasks();
Task getTask(int index);
int getListSize();
void changeTask(int taskPos, Task newTask);
```

5.1.4 GUI Classes Interface List

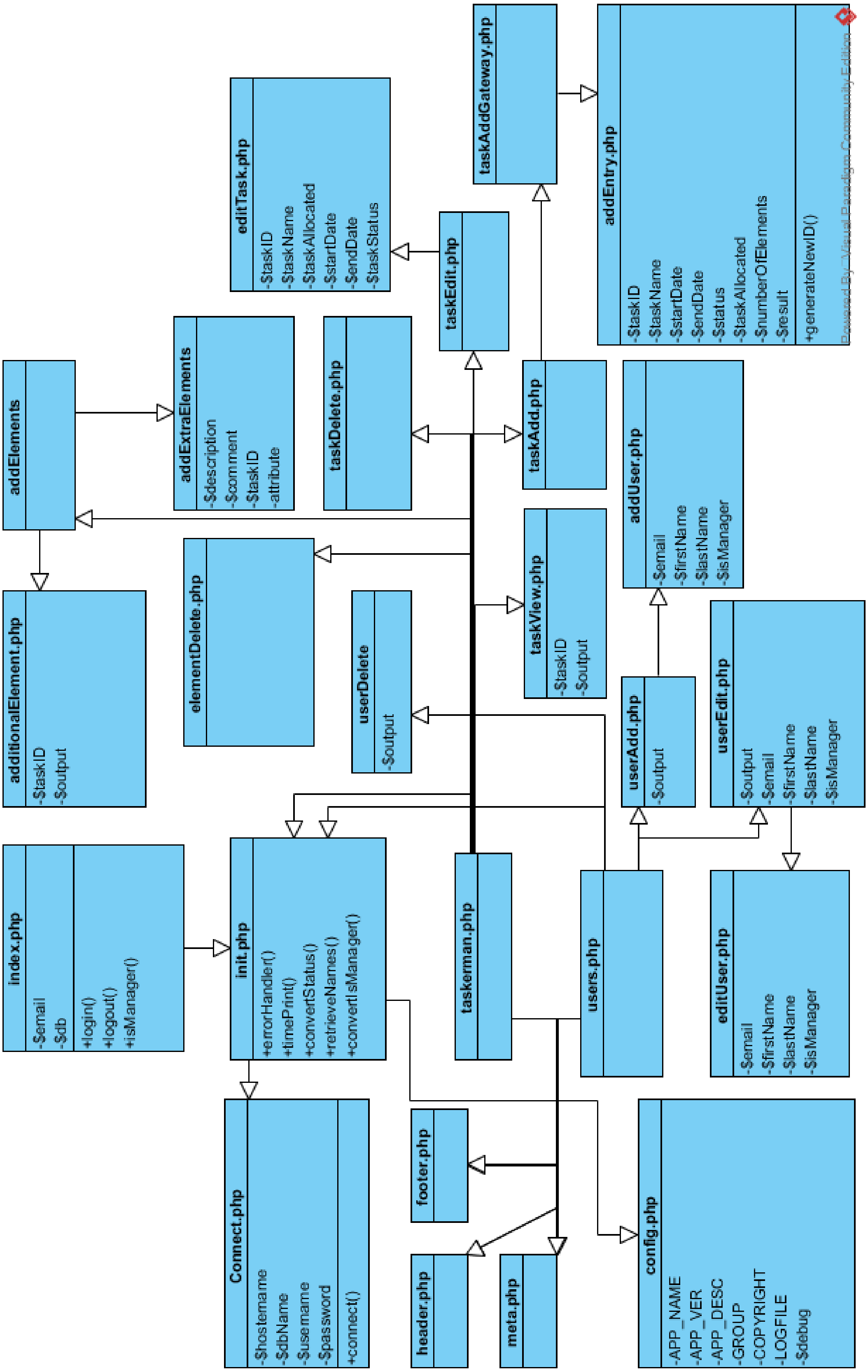
Window:

```
void populateWindowArray();
void initialize (windowIndex, windowName);
void exit(windowIndex, windowName);
void setFocus(windowIndex, windowName);
```

5.2 TaskerMAN

5.2.1 Class Diagram

TaskerMAN is written in procedural PHP, and therefore does not have classes. To compensate, the diagram simply represents the relationships between each PHP file in the program and any contained variables, functions and constants.



5.2.2 PHP File List

index.php

Login page, index page as it is the first page a user lands on when they access TaskerMAN. Not possible to proceed to any other file if they are unable to authenticate.

meta.php, footer.php, header.php

These files make up the static UI components, including the navigation and the HTML includes to any external stylesheets or JavaScript files.

init.php, config.php, connect.php

The three main supporting files of the software. Connect establishes and maintains a database connection to TaskerSRV. Config contains global configuration and settings. Init provides access to these resources as well as useful global functions throughout the entirety of the software.

taskerman.php, users.php

Task and User view of the TaskerMAN software respectively. The main files in the software that the user navigates to after authentication.

taskAdd.php, taskEdit.php, userAdd.php, userEdit.php, addElements.php, additionalElement.php

These are modal windows that are displayed on the Task or User view respectively. They provide functionality to add a task, edit an existing task, add a user, edit a user, add elements to a new task, and to add an additional element to an existing task.

addEntry.php, addUser.php, editTask.php, editUser.php

Carries out the adding or editing operations of a task or user, generating SQL and running them on TaskerSRV.

taskAddGateway.php

This file sanitises taskAdd input and prepares the data to be combined with the new task element data before posting it to addEntry. Since this is the most complicated operation, this file was needed.

taskDelete.php, userDelete.php

Deletes selected task or user. Will refuse to delete a user if they have tasks currently assigned.

addExtraElement.php

Adds an additional element to an existing task.

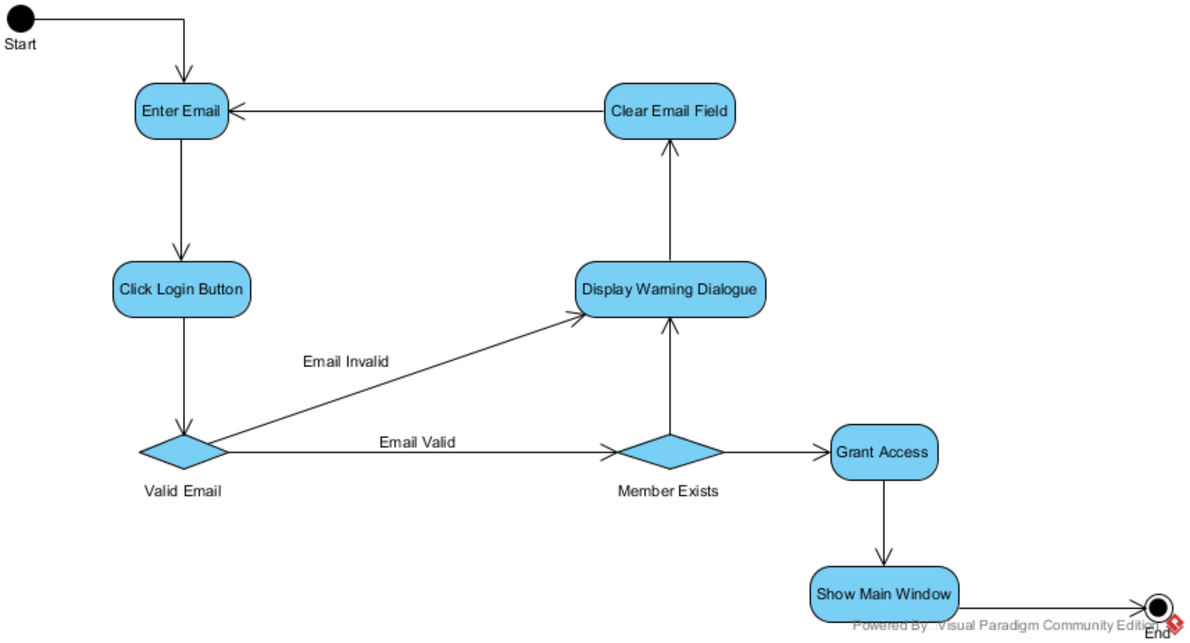
elementDelete.php

Deletes elements corresponding to a selected task.

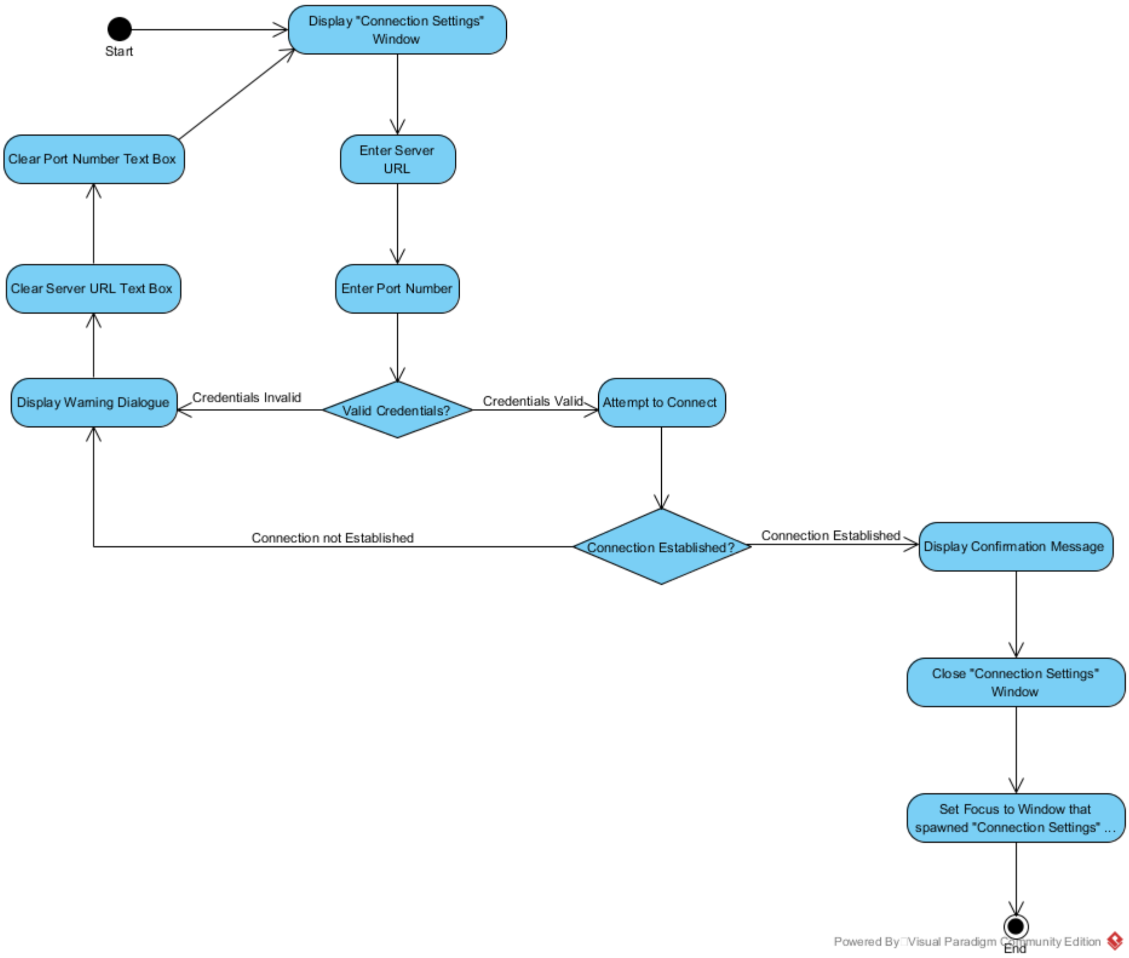
6. DETAILED DESIGN

6.1 Activity Diagrams

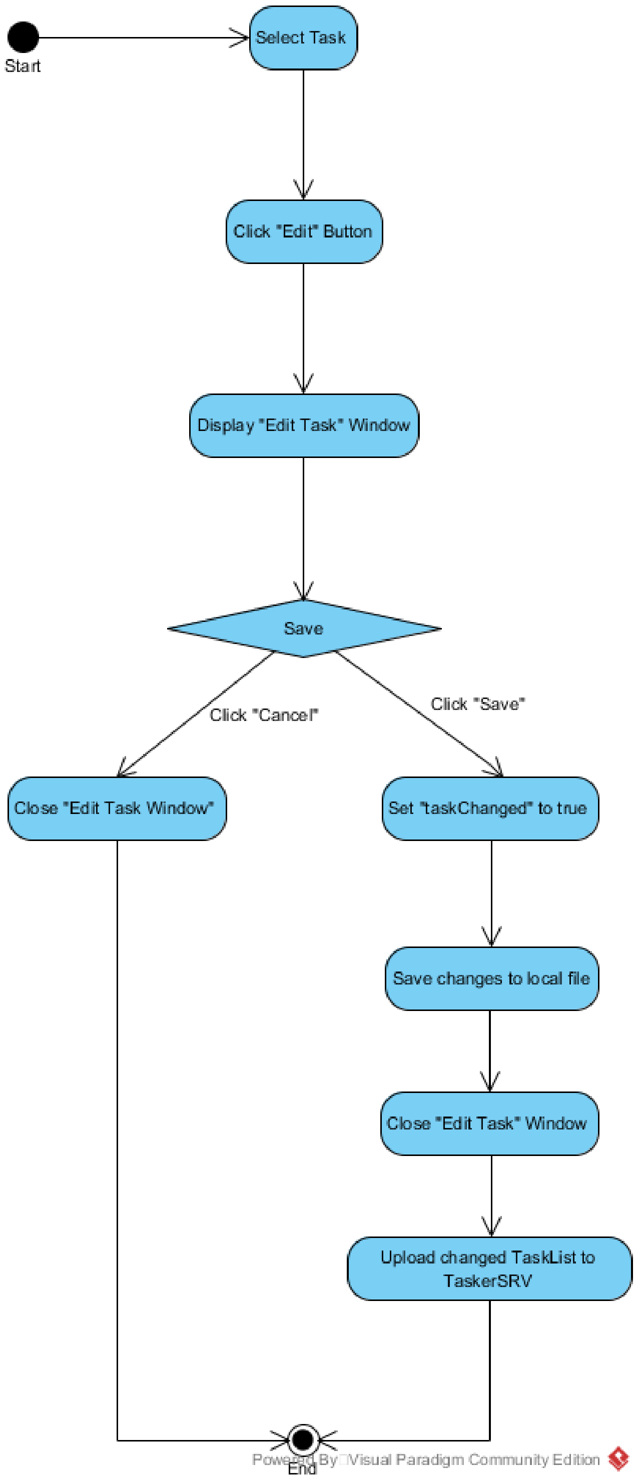
6.1.1 TaskerCLI User Login



6.1.2 TaskerCLI connecting to TaskerSRV

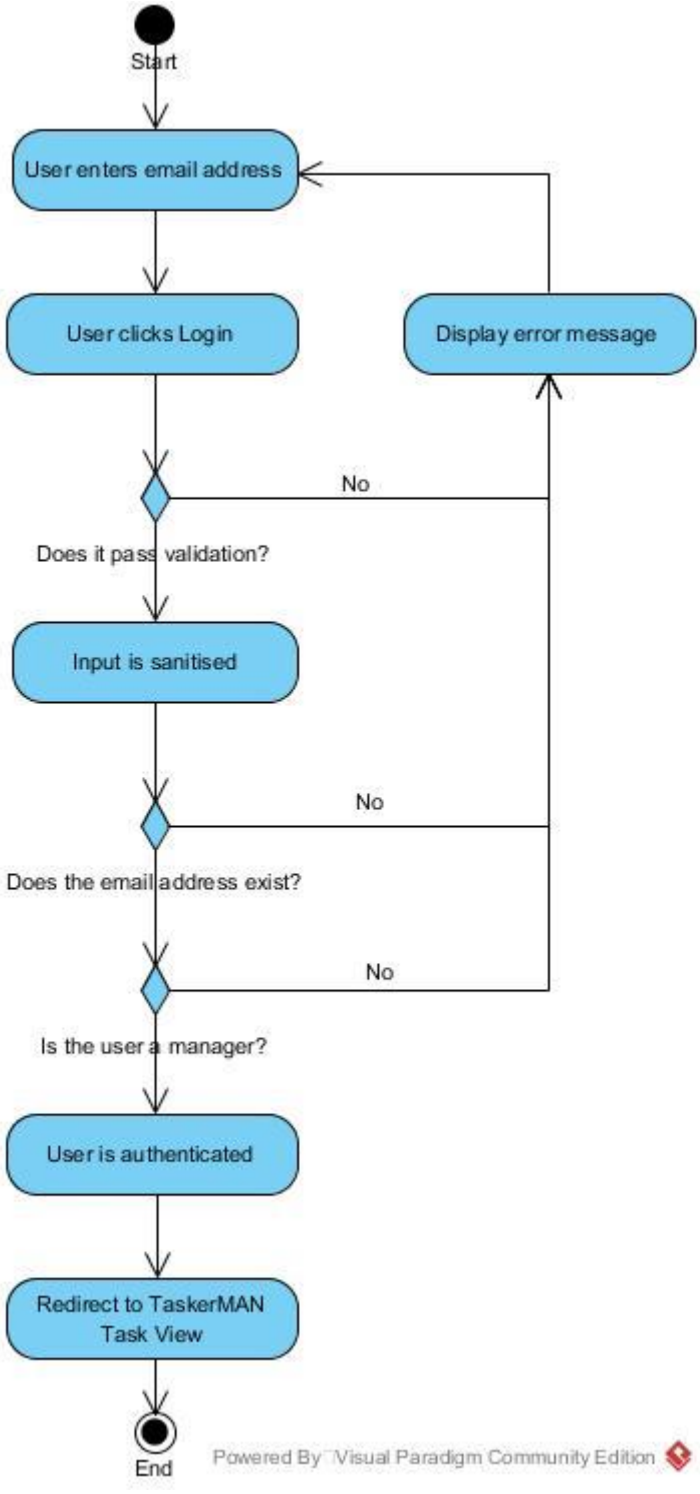

Start
Display "Connection Settings" Window
Clear Port Number Text Box
Enter Server URL
Clear Server URL Text Box
Enter Port Number
Display Warning Dialogue
Credentials Invalid
Valid Credentials?
Credentials Valid
Attempt to Connect
Connection not Established
Connection Established?
Connection Established
Display Confirmation Message
Close "Connection Settings" Window
Set Focus to Window that spawned "Connection Settings" ...
End
Powered By Visual Paradigm Community Edition

6.1.3 TaskerCLI Task Editing


Start
Select Task
Click "Edit" Button
Display "Edit Task" Window
Save
Click "Cancel"
Click "Save"
Close "Edit Task Window"
Set "taskChanged" to true
Save changes to local file
Close "Edit Task" Window
Upload changed TaskList to TaskerSRV
End
Powered By Visual Paradigm Community Edition

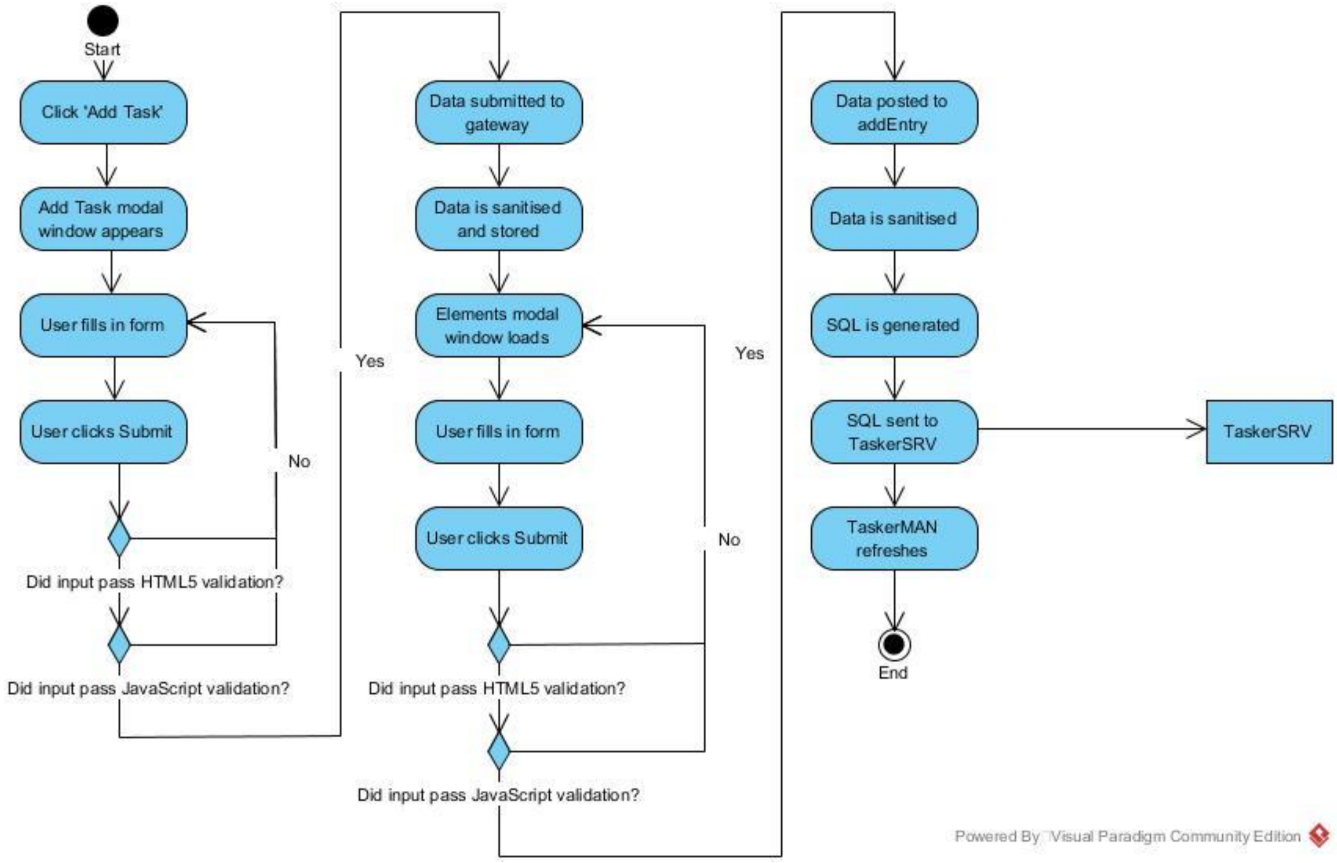
6.1.4 TaskerMAN Login

The login process for TaskerMAN was decomposed into discreet steps which are shown in the diagram below. This then displays the main window specified in FR7 to the user.



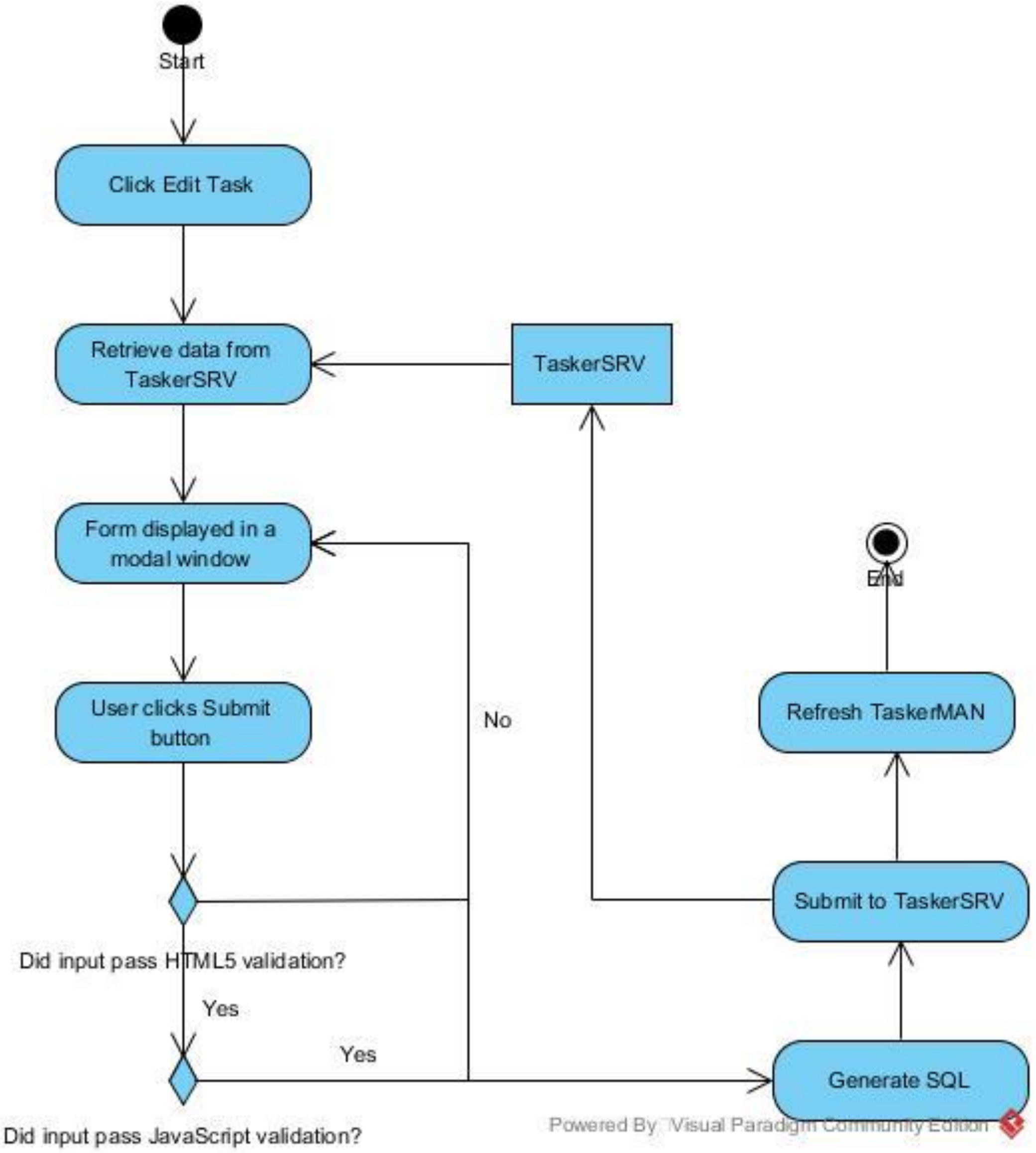
6.1.5 Adding Tasks in TaskerMAN

TaskerMAN must be able to add new tasks and task data to the system as per requirement FR4. This diagram below lists the activities which take place when a task is being added.


Start
Click 'Add Task'
Add Task modal window appears
User fills in form
User clicks Submit
No
Did input pass HTML5 validation?
Did input pass JavaScript validation?
Yes
Data submitted to gateway
Data is sanitised and stored
Elements modal window loads
User fills in form
User clicks Submit
Did input pass HTML5 validation?
Did input pass JavaScript validation?
No
Yes
Data posted to addEntry
Data is sanitised
SQL is generated
SQL sent to TaskerSRV
TaskerSRV
TaskerMAN refreshes
End
Powered By Visual Paradigm Community Edition

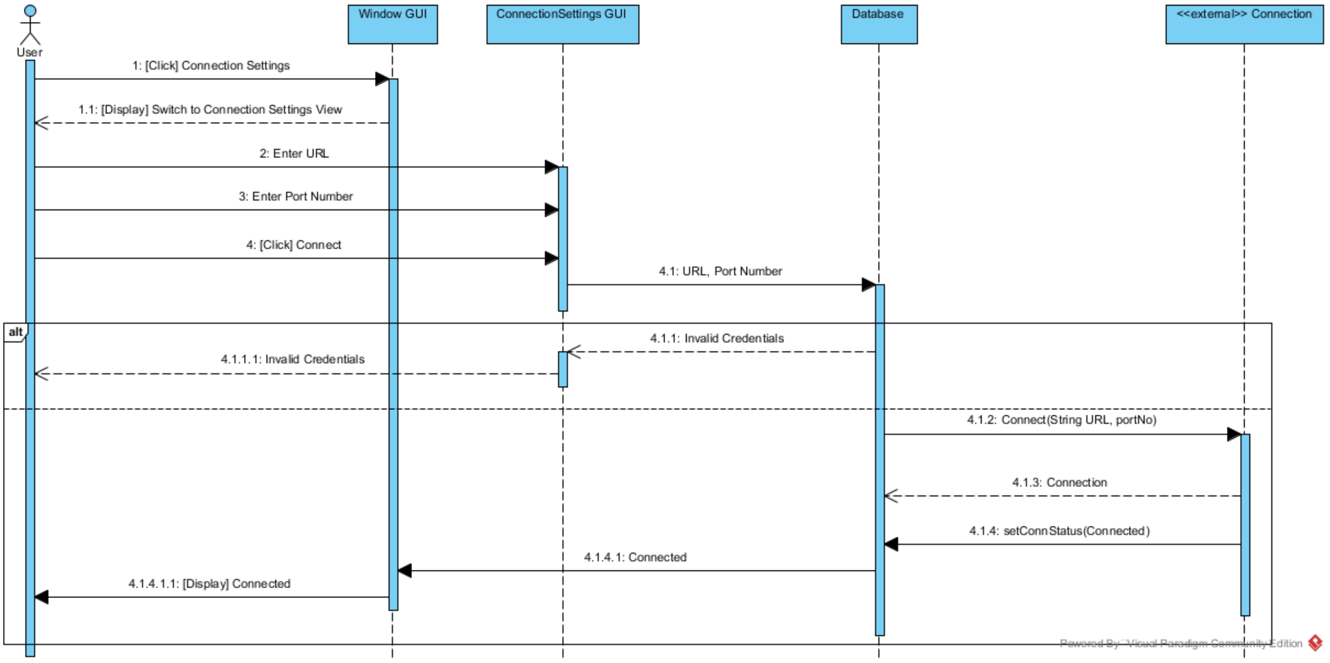
6.1.6 Editing Tasks in TaskerMAN

TaskerMAN must be able to edit tasks to allow reallocation or maintenance of task data as specified by requirement FR5. This diagram below lists the activities which take place when a task is being edited.

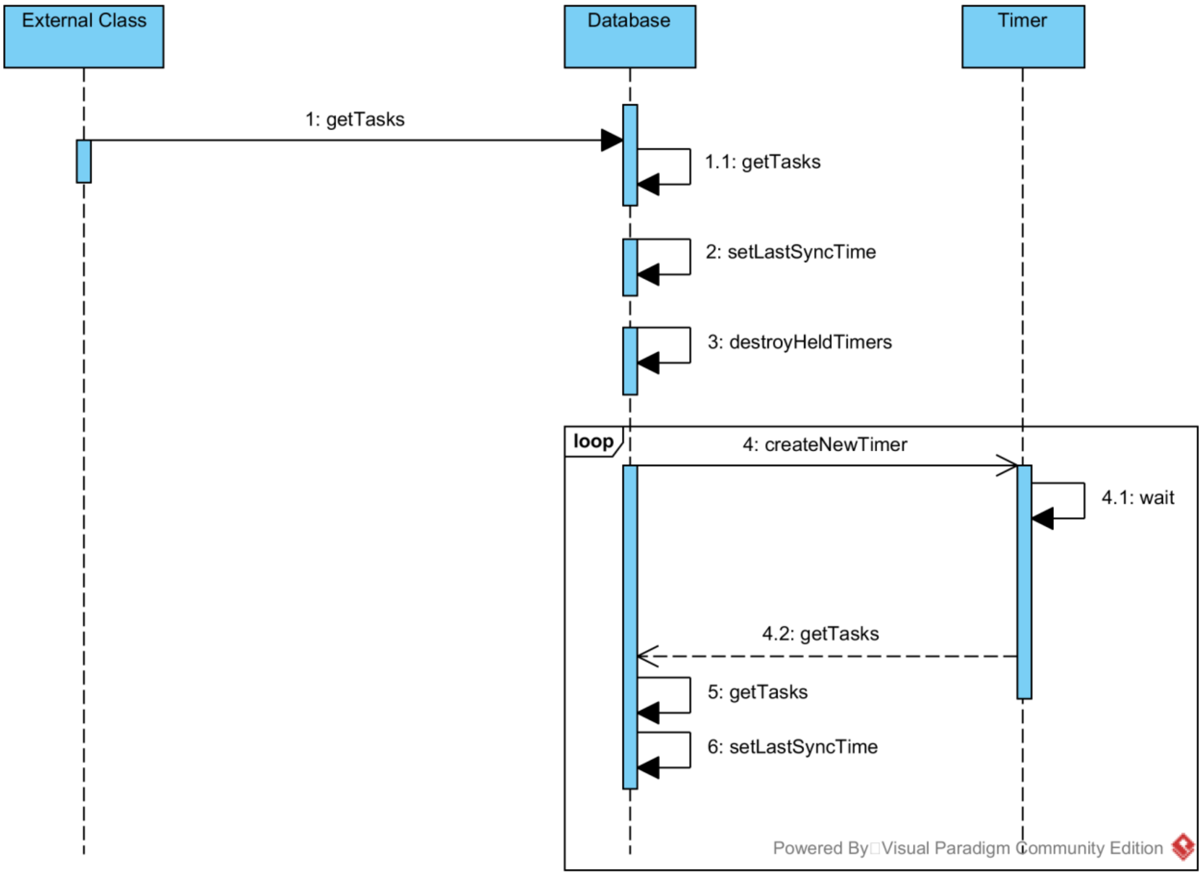

Start
Click Edit Task
Retrieve data from TaskerSRV
TaskerSRV
Form displayed in a modal window
User clicks Submit button
No
Did input pass HTML5 validation?
Yes
Yes
Did input pass JavaScript validation?
End
Refresh TaskerMAN
Submit to TaskerSRV
Generate SQL
Powered By Visual Paradigm Community Edition

6.2 Sequence Diagrams

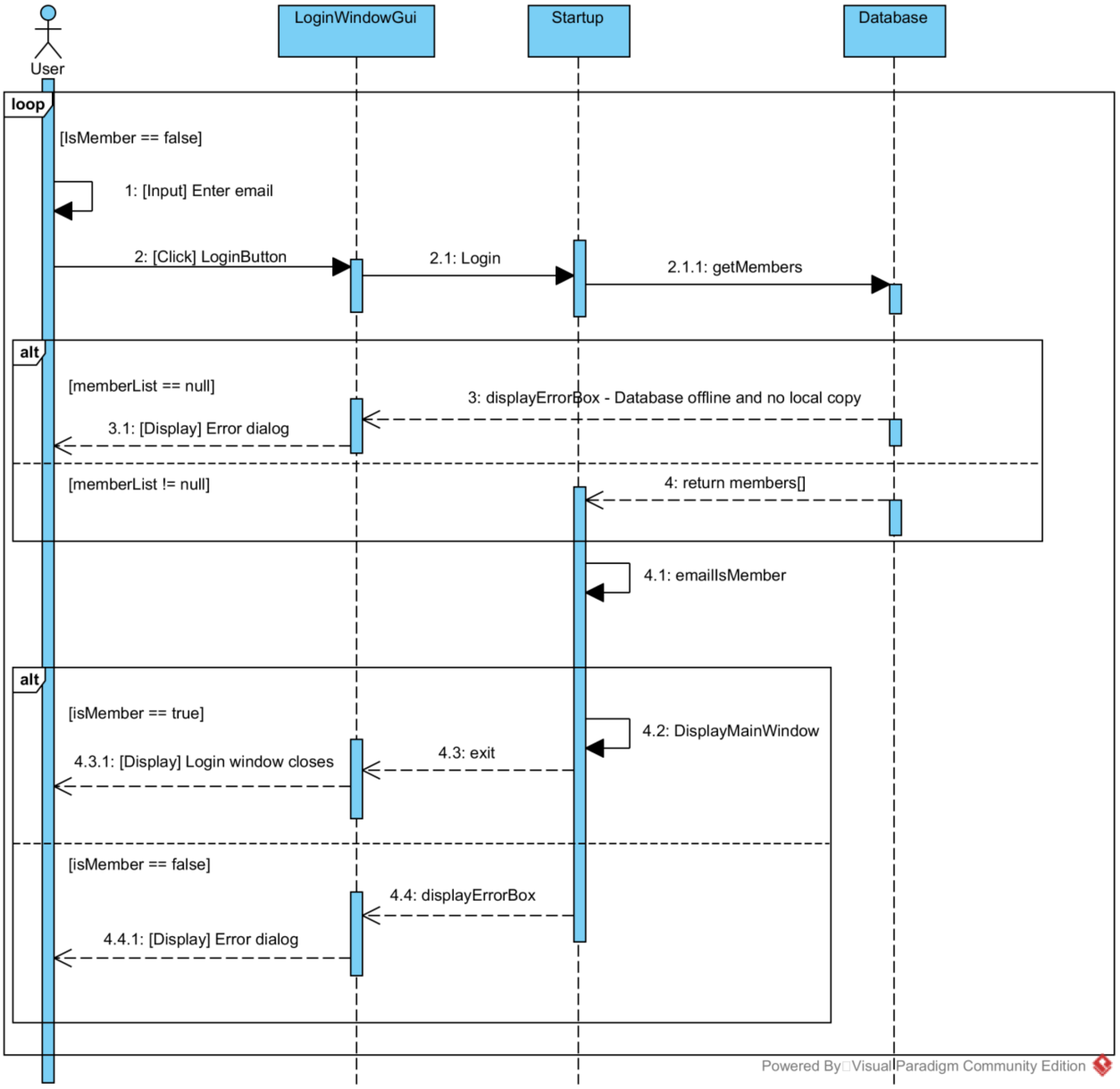
6.2.1 Connecting to TaskerSRV from TaskerCLI


User
Window GUI
ConnectionSettings GUI
Database
<<external>> Connection
1: [Click] Connection Settings
1.1: [Display] Switch to Connection Settings View
2: Enter URL
3: Enter Port Number
4: [Click] Connect
4.1: URL, Port Number
alt
4.1.1: Invalid Credentials
4.1.1.1: Invalid Credentials
4.1.2: Connect(String URL, portNo)
4.1.3: Connection
4.1.4: setConnStatus(Connected)
4.1.4.1: Connected
4.1.4.1.1: [Display] Connected
Powered By Visual Paradigm Community Edition

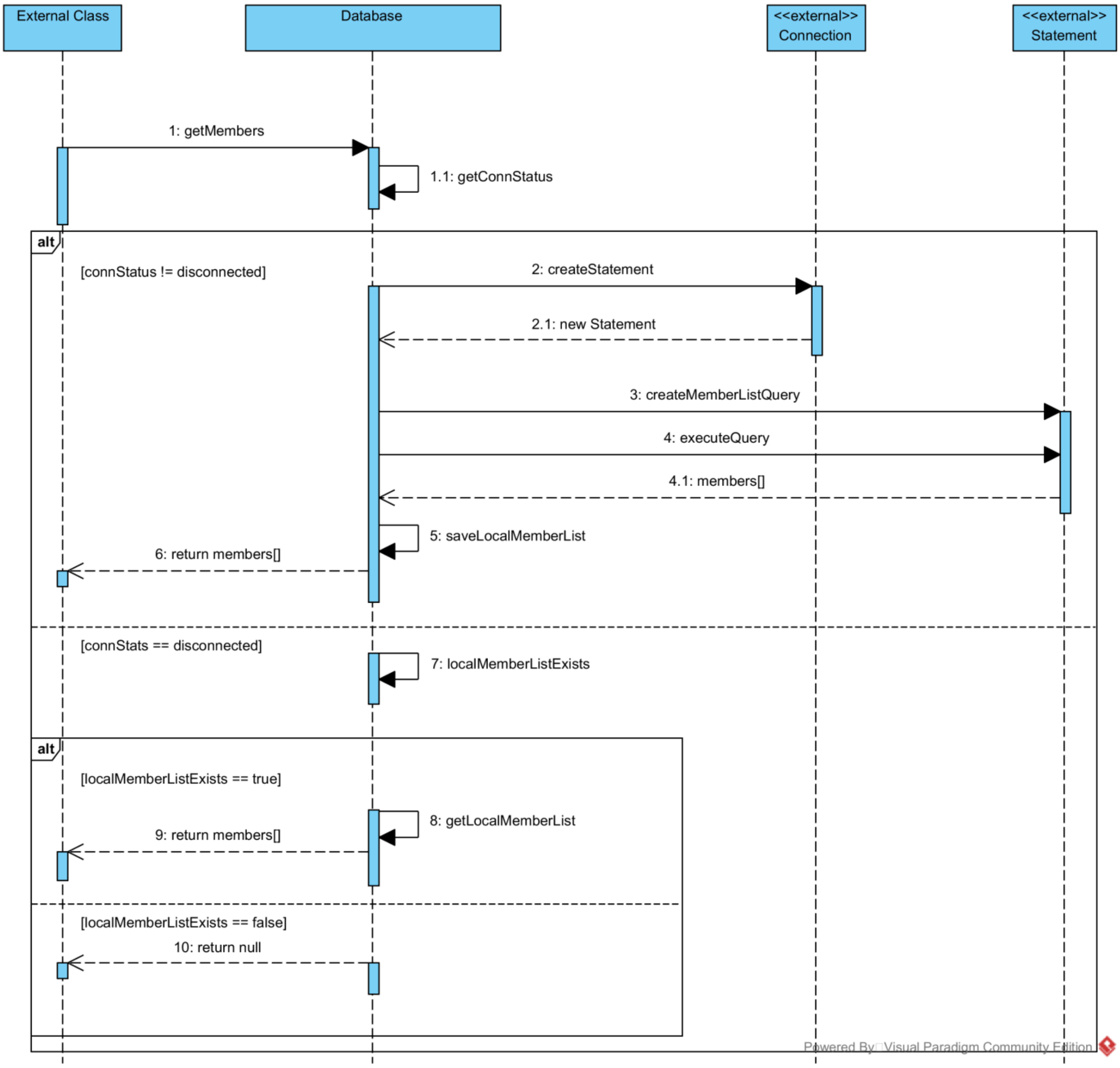
6.2.2 Automatic synchronization to TaskerSRV from TaskerCLI



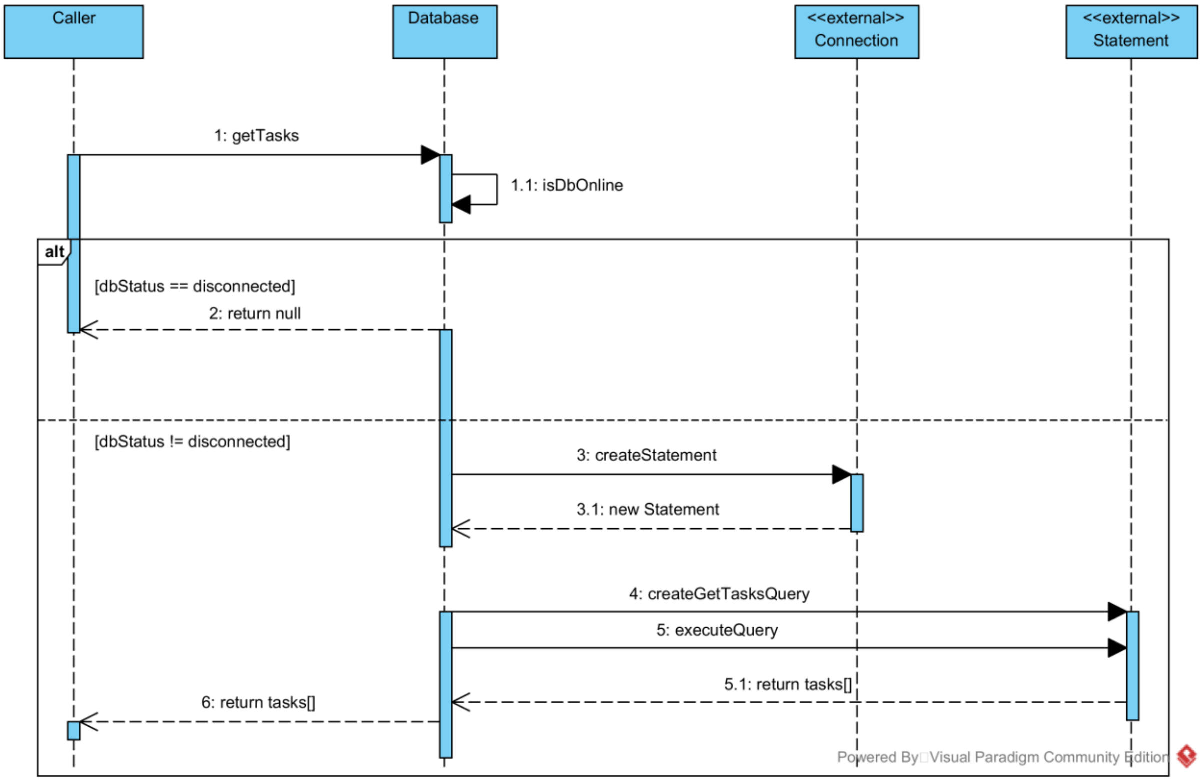
6.2.3 User Login to TaskerCLI



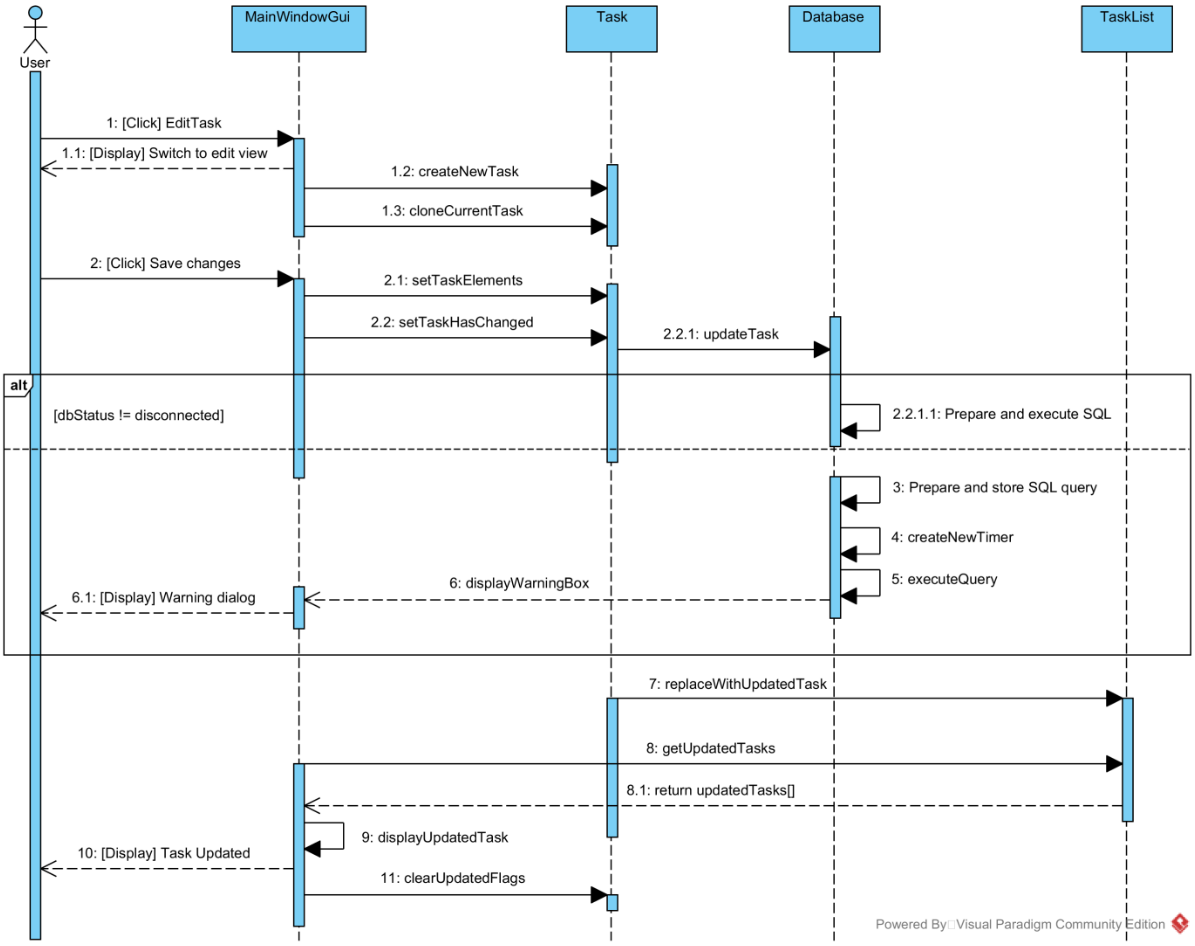
6.2.4 Get list of members in TaskerCLI from TaskerSRV



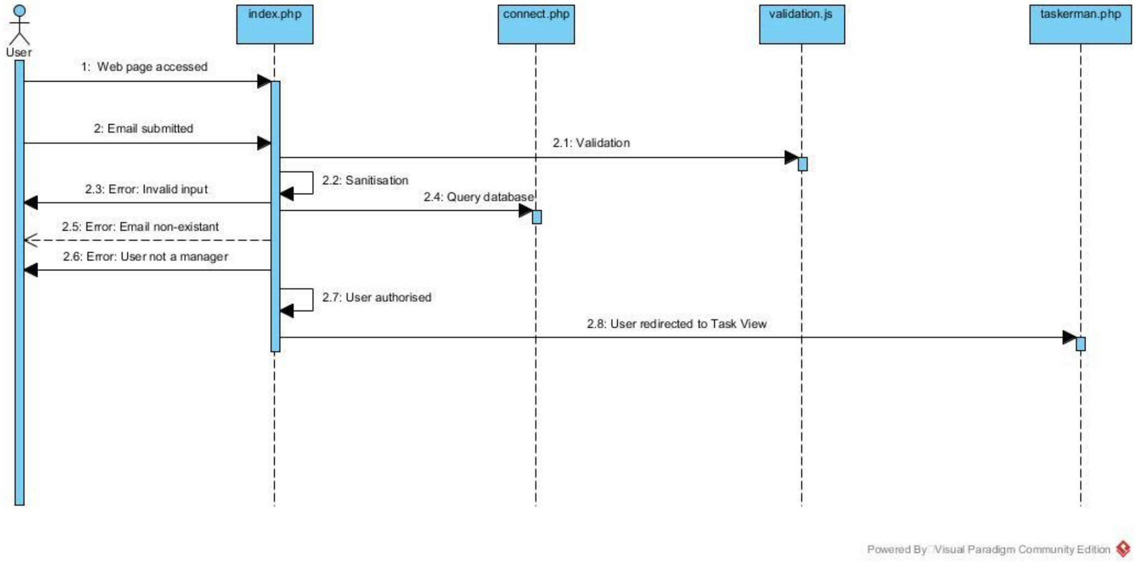
6.2.5 Get list of tasks in TaskerCLI from TaskerSRV



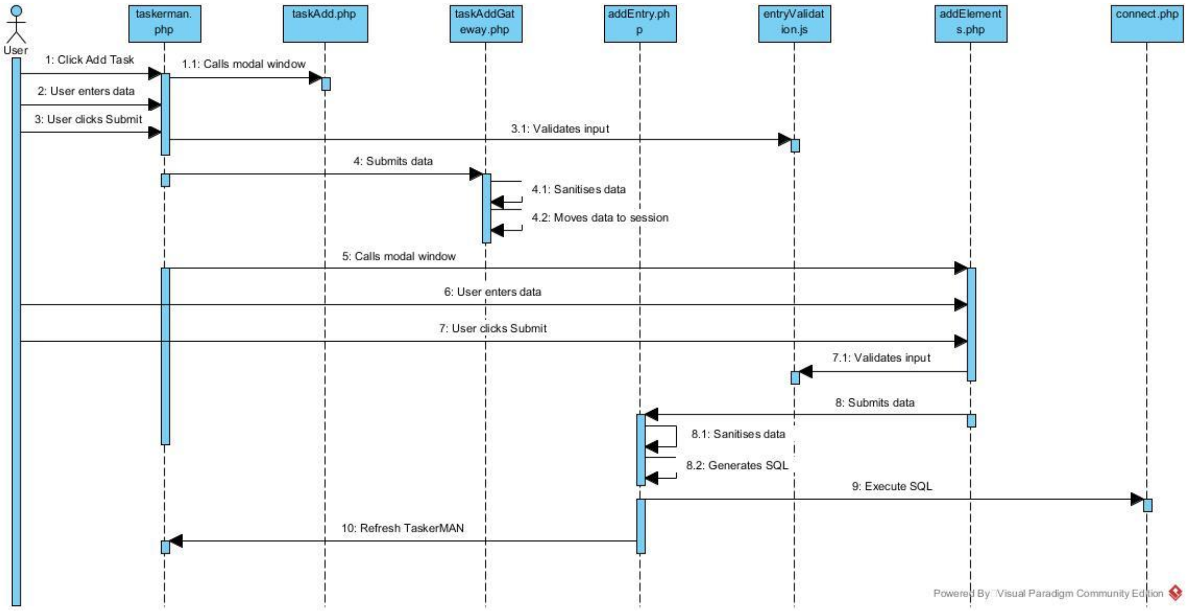
6.2.6 Editing tasks in TaskerCLI



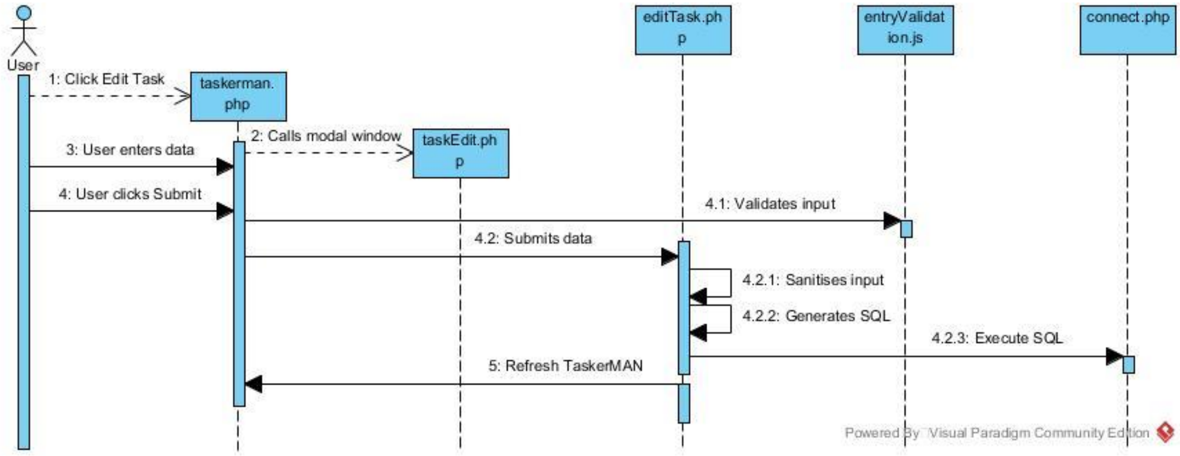
6.2.7 User Login in TaskerMAN


User
index.php
connect.php
validation.js
taskerman.php
1: Web page accessed
2: Email submitted
2.1: Validation
2.2: Sanitisation
2.3: Error: Invalid input
2.4: Query database
2.5: Error: Email non-existant
2.6: Error: User not a manager
2.7: User authorised
2.8: User redirected to Task View
Powered By Visual Paradigm Community Edition

6.2.8 Adding Tasks in TaskerMAN



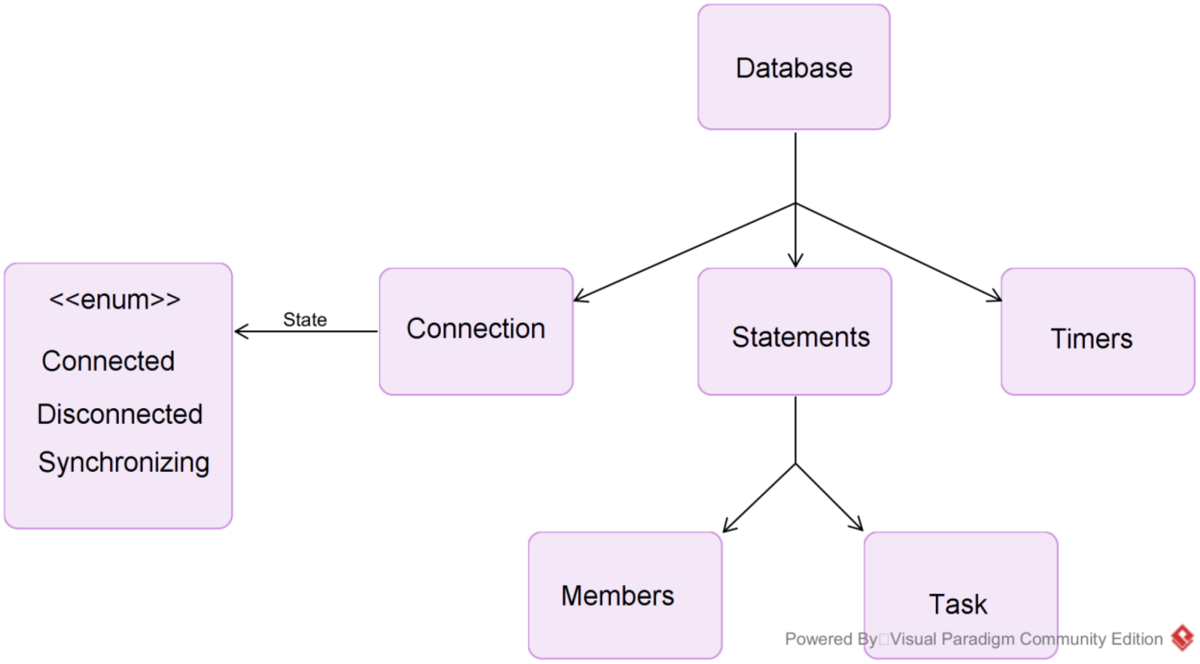
6.2.9 Editing Tasks in TaskerMAN



6.3 TaskerCLI Data Structures

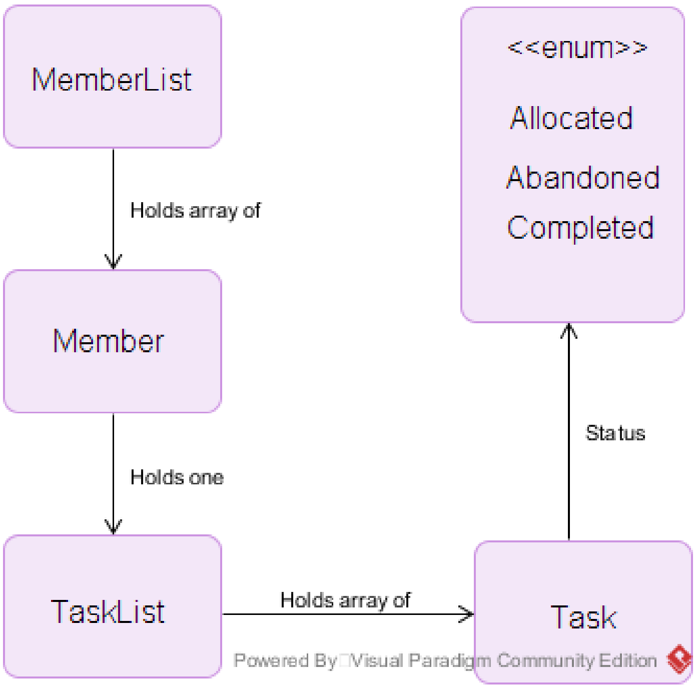
6.3.1.1 Database Structure

The diagram below shows the data classes the Database class uses in order to function



6.3.2 Members and Tasks Structure

The below diagram shows the hierarchy of Member related classes and Task related classes. From this diagram it is obvious the number and type of objects a class will hold or use.



6.4 Spike Work

6.4.1 TaskerCLI

The largest risk to TaskerCLI was implementing GUI's. This was identified as a significant risk early in planning so spike work was created to assess and mitigate any issues. For this spike work several people created the mock ups in Java using various tools.

Window builder allowed us to rapidly and accurately create the required design for the GUI; however the designer can be unintuitive with minor changes requiring major layout method changes. The group met and discussed finding and identified common pitfalls and how to avoid them, such as using several frames where layout could change.

In the process we established that by using an interface on all window classes we could easily construct and destruct these windows whilst abstracting over the minor differences between them.

This concept leads to creating a window manager which handles opening and closing windows on behalf of the software. Refining the code the final design was an array of window interface objects. By using an enumerated list we can access windows and manipulate them by doing a single line such as *"windowManager.createWindow("MAIN_WINDOW_ENUM");"*

6.4.2 TaskerMAN

Once PHP was chosen to drive TaskerMAN the team looked into PHP Unit. Having all used and understood the Java equivalent Junit the group needed to check if this knowledge was applicable to PHP Unit for creating unit tests.

This spike work consisted of setting up and learning how to implement PHP unit tests. One outcome which came to light was that our planned usage for PHP is mostly procedural whilst PHP unit caters for object oriented paradigms. The group also had to train each other in configuring the IDE to correctly implement these tests and writing suitable tests in PHP Unit.

Additional spike work on input validation was performed for TaskerMAN. As emails are used as an input we needed to check the input before sending it to the server. Several approaches were discussed such as using HTML 5 and JavaScript tests to detect invalid input on the client side. The approach decided upon was regular expression tests to ensure input is valid.

REFERENCES

[1] N. W. Hardy, *Tasker Team Tasking System - Requirement Specification 1.2,* Aberystwyth University: Software Engineering Group Project, 2015.

[2] Tutorialspoint, "TutorialsPoint JUnit Environment Setup," TutorialsPoint, 27 10 2015. [Online]. Available: http://www.tutorialspoint.com/junit/junit_environment_setup.htm. [Accessed 27 10 2015].

[3] Earl, Oliver; The PHP Group, "PHPInfo running on Apache," 27 10 2015. [Online]. Available: http://users.aber.ac.uk/ole4/phpinfo.php. [Accessed 28 10 2015].

[4] S. Bergmann, "PHPUnit English Documentation," 28 10 2015. [Online]. Available: https://phpunit.de/manual/current/en/phpunit-book.html#installation.requirements. [Accessed 28 10 2015].

[5] Oracle, "Oracle Software Delivery Cloud - MySQL Standard Edition for Linux x86-64," Oracle, 2015. [Online]. Available: https://edelivery.oracle.com/osdc/faces/SearchSoftware?_adf.ctrl-state=nmw6458k7_28&_afrLoop=2752661125326430. [Accessed 21 10 2015].

[6] N. W. Hardy, C. J. Price and B. P. Tiddeman, *SE.QA.05 A 1.8 - Design Specification Standards,* Aberystwyth University: Software Engineering Group Project, 2015.

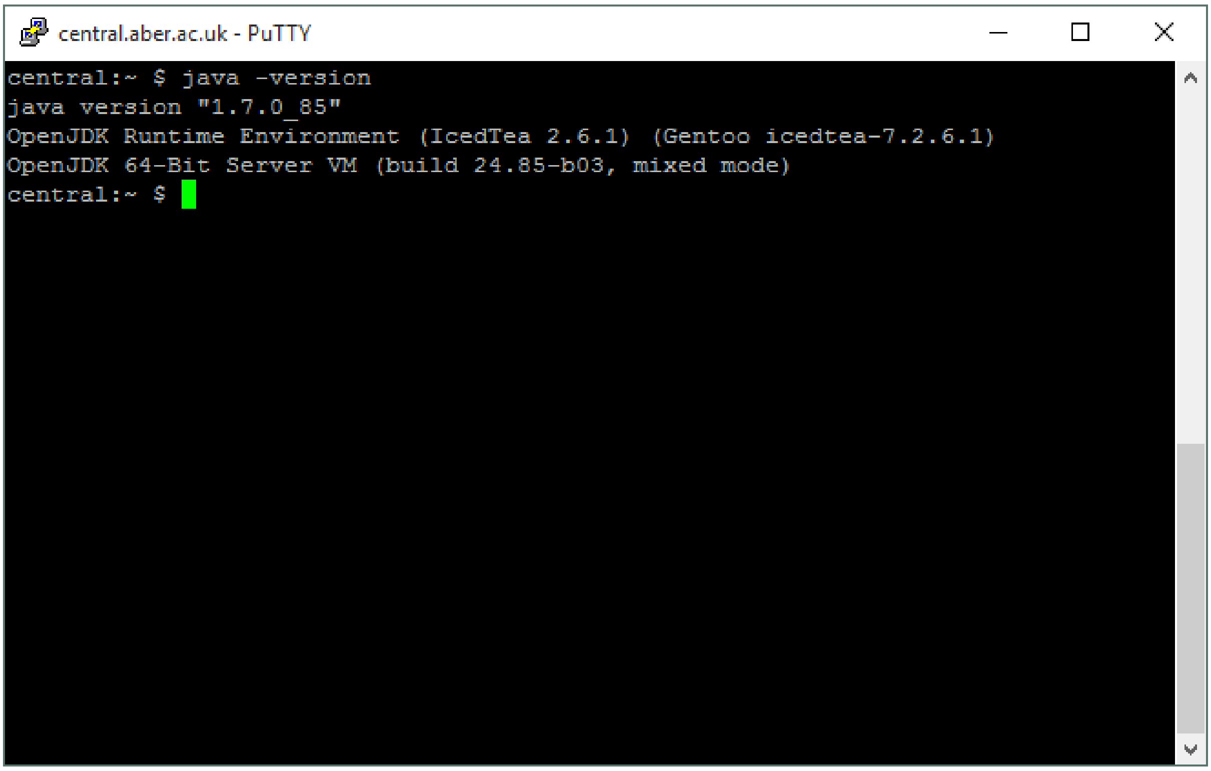
[7] Pixeden, "Firefox Web Browser Mockup Template," CorruptedDevelopment, 26 08 2014. [Online]. Available: http://corrupteddevelopment.com/firefox-web-browser-mockup-template/. [Accessed 26 10 2015].

DOCUMENT HISTORY

Version	*CCF No.*	*Date*	*Changes made to document*	*Changed by*
1.0	N/A	27/11/15	Original version	DAF5
1.11	34	14/12/15	Changed interaction diagram to correctly show PDO instead of MYSQL	DAF5
1.2	48	14/12/15	Updated TaskerCLI class diagram with correction	DAF5
2.0	187	13/02/16	Major revision to design documentation following multiple changes throughout implementation of software.	DAF5

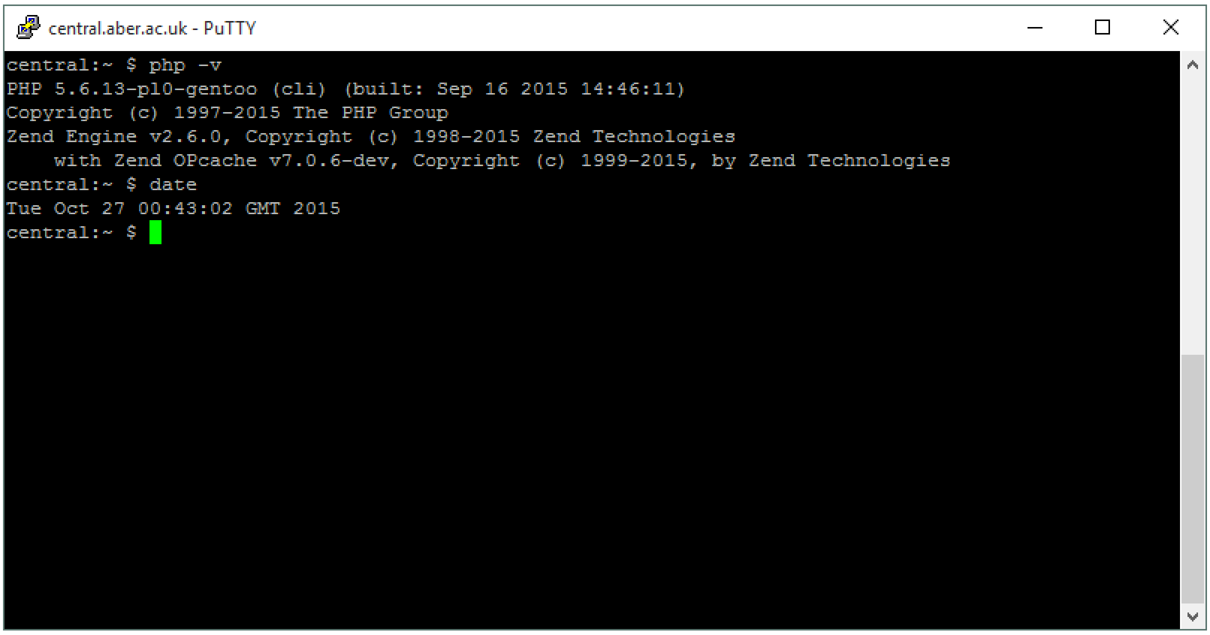
APPENDICES

APPENDIX A – Java Version



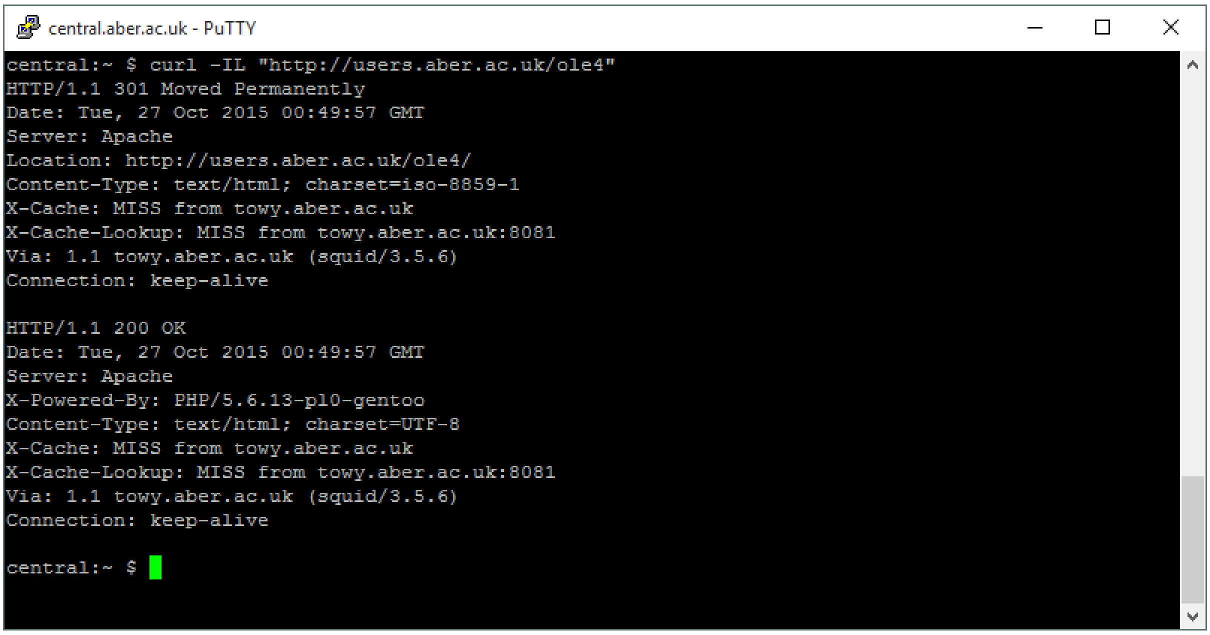
```
central:~ $ java -version
java version "1.7.0_85"
OpenJDK Runtime Environment (IcedTea 2.6.1) (Gentoo icedtea-7.2.6.1)
OpenJDK 64-Bit Server VM (build 24.85-b03, mixed mode)
central:~ $
```

APPENDIX B – PHP Version



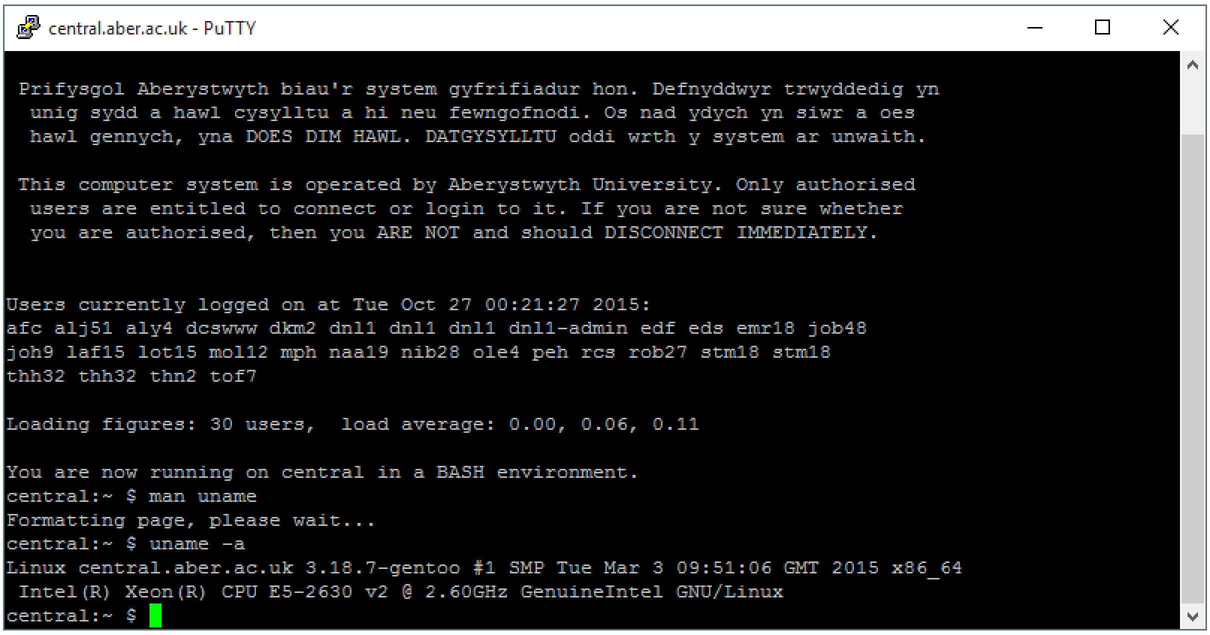
```
central:~ $ php -v
PHP 5.6.13-pl0-gentoo (cli) (built: Sep 16 2015 14:46:11)
Copyright (c) 1997-2015 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2015 Zend Technologies
    with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2015, by Zend Technologies
central:~ $ date
Tue Oct 27 00:43:02 GMT 2015
central:~ $
```

APPENDIX C – Apache Information



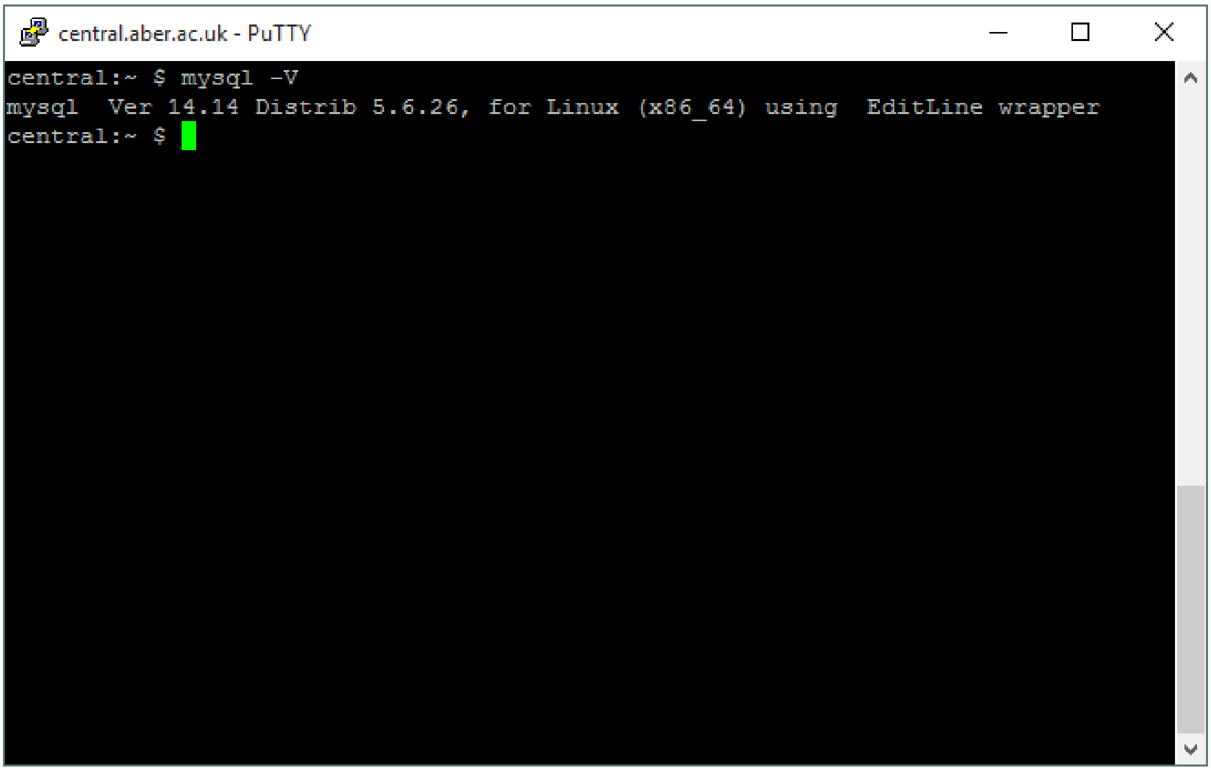
```
central:~ $ curl -IL "http://users.aber.ac.uk/ole4"
HTTP/1.1 301 Moved Permanently
Date: Tue, 27 Oct 2015 00:49:57 GMT
Server: Apache
Location: http://users.aber.ac.uk/ole4/
Content-Type: text/html; charset=iso-8859-1
X-Cache: MISS from towy.aber.ac.uk
X-Cache-Lookup: MISS from towy.aber.ac.uk:8081
Via: 1.1 towy.aber.ac.uk (squid/3.5.6)
Connection: keep-alive

HTTP/1.1 200 OK
Date: Tue, 27 Oct 2015 00:49:57 GMT
Server: Apache
X-Powered-By: PHP/5.6.13-p10-gentoo
Content-Type: text/html; charset=UTF-8
X-Cache: MISS from towy.aber.ac.uk
X-Cache-Lookup: MISS from towy.aber.ac.uk:8081
Via: 1.1 towy.aber.ac.uk (squid/3.5.6)
Connection: keep-alive

central:~ $
```

APPENDIX D – Linux information



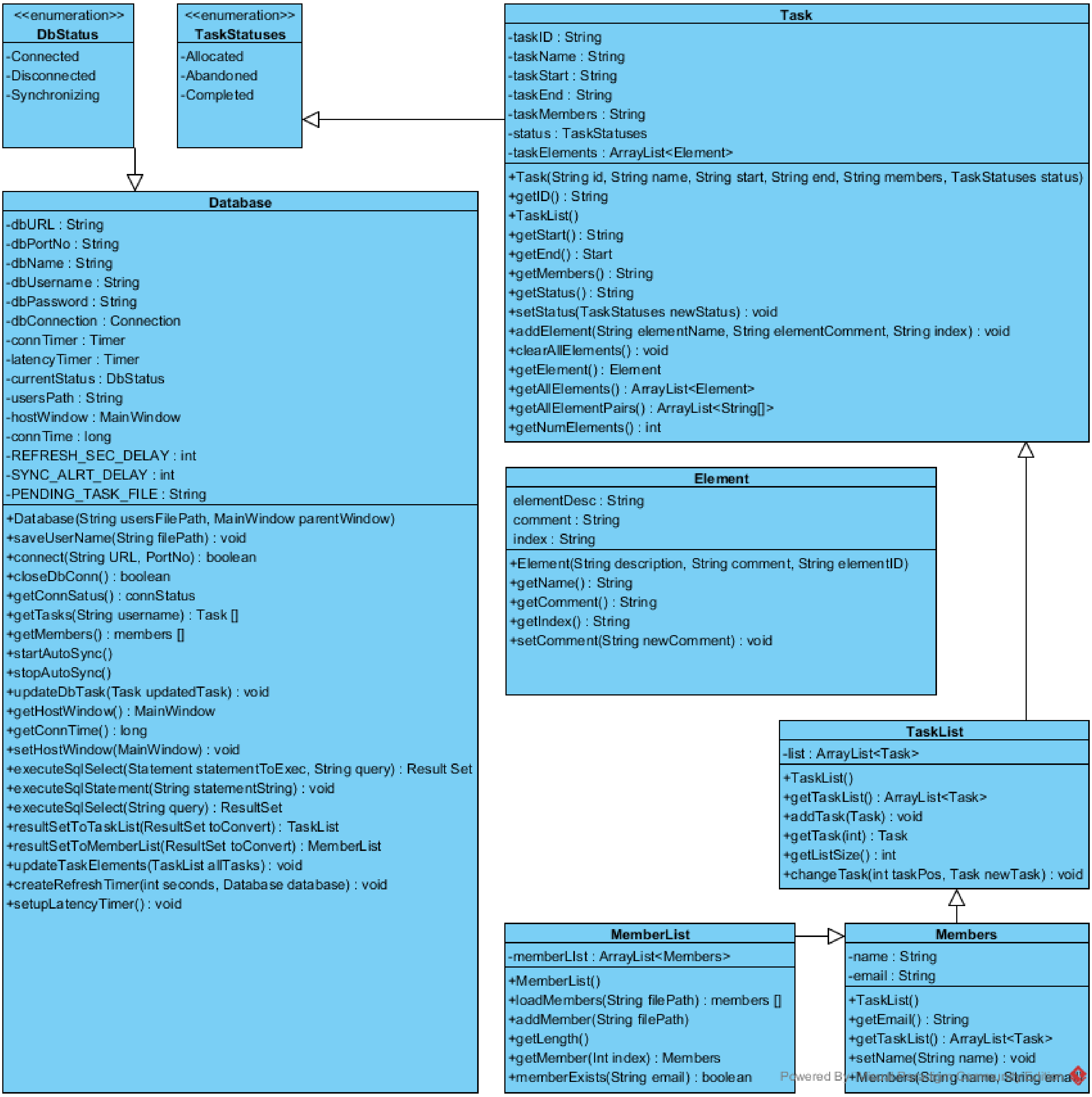
```

 Prifysgol Aberystwyth biau'r system gyfrifiadur hon. Defnyddwyr trwyddedig yn
  unig sydd a hawl cysylltu a hi neu fewngofnodi. Os nad ydych yn siwr a oes
  hawl gennych, yna DOES DIM HAWL. DATGYSYLLTU oddi wrth y system ar unwaith.

 This computer system is operated by Aberystwyth University. Only authorised
  users are entitled to connect or login to it. If you are not sure whether
  you are authorised, then you ARE NOT and should DISCONNECT IMMEDIATELY.


Users currently logged on at Tue Oct 27 00:21:27 2015:
afc alj51 aly4 dcswww dkm2 dnl1 dnl1 dnl1 dnl1-admin edf eds emr18 job48
joh9 laf15 lot15 mol12 mph naa19 nib28 ole4 peh rcs rob27 stm18 stm18
thh32 thh32 thn2 tof7

Loading figures: 30 users,  load average: 0.00, 0.06, 0.11

You are now running on central in a BASH environment.
central:~ $ man uname
Formatting page, please wait...
central:~ $ uname -a
Linux central.aber.ac.uk 3.18.7-gentoo #1 SMP Tue Mar 3 09:51:06 GMT 2015 x86_64
 Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz GenuineIntel GNU/Linux
central:~ $
```

APPENDIX E – MySQL Version



```
central:~ $ mysql -V
mysql  Ver 14.14 Distrib 5.6.26, for Linux (x86_64) using  EditLine wrapper
central:~ $
```

APPENDIX F – Logic Class Diagram


<<enumeration>>
DbStatus
-Connected
-Disconnected
-Synchronizing
<<enumeration>>
TaskStatuses
-Allocated
-Abandoned
-Completed
Task
-taskID : String
-taskName : String
-taskStart : String
-taskEnd : String
-taskMembers : String
-status : TaskStatuses
-taskElements : ArrayList<Element>
+Task(String id, String name, String start, String end, String members, TaskStatuses status)
+getID() : String
+TaskList()
+getStart() : String
+getEnd() : Start
+getMembers() : String
+getStatus() : String
+setStatus(TaskStatuses newStatus) : void
+addElement(String elementName, String elementComment, String index) : void
+clearAllElements() : void
+getElement() : Element
+getAllElements() : ArrayList<Element>
+getAllElementPairs() : ArrayList<String[]>
+getNumElements() : int
Database
-dbURL : String
-dbPortNo : String
-dbName : String
-dbUsername : String
-dbPassword : String
-dbConnection : Connection
-connTimer : Timer
-latencyTimer : Timer
-currentStatus : DbStatus
-usersPath : String
-hostWindow : MainWindow
-connTime : long
-REFRESH_SEC_DELAY : int
-SYNC_ALRT_DELAY : int
-PENDING_TASK_FILE : String
+Database(String usersFilePath, MainWindow parentWindow)
+saveUserName(String filePath) : void
+connect(String URL, PortNo) : boolean
+closeDbConn() : boolean
+getConnSatus() : connStatus
+getTasks(String username) : Task []
+getMembers() : members []
+startAutoSync()
+stopAutoSync()
+updateDbTask(Task updatedTask) : void
+getHostWindow() : MainWindow
+getConnTime() : long
+setHostWindow(MainWindow) : void
+executeSqlSelect(Statement statementToExec, String query) : Result Set
+executeSqlStatement(String statementString) : void
+executeSqlSelect(String query) : ResultSet
+resultSetToTaskList(ResultSet toConvert) : TaskList
+resultSetToMemberList(ResultSet toConvert) : MemberList
+updateTaskElements(TaskList allTasks) : void
+createRefreshTimer(int seconds, Database database) : void
+setupLatencyTimer() : void
Element
elementDesc : String
comment : String
index : String
+Element(String description, String comment, String elementID)
+getName() : String
+getComment() : String
+getIndex() : String
+setComment(String newComment) : void
TaskList
-list : ArrayList<Task>
+TaskList()
+getTaskList() : ArrayList<Task>
+addTask(Task) : void
+getTask(int) : Task
+getListSize() : int
+changeTask(int taskPos, Task newTask) : void
MemberList
-memberLIst : ArrayList<Members>
+MemberList()
+loadMembers(String filePath) : members []
+addMember(String filePath)
+getLength()
+getMember(Int index) : Members
+memberExists(String email) : boolean
Members
-name : String
-email : String
+TaskList()
+getEmail() : String
+getTaskList() : ArrayList<Task>
+setName(String name) : void
+Members(String name, String ema