Software Engineering Group Project

Interaction and High Level Design

|  |  |
| --- | --- |
| Authors: | Ben Dudley, David Fairbrother, Jonathan Englund, Josh Doyle, Liam Fitzgerald, Oliver Earl, Tim Anderson |
| Config Ref: | SE\_05\_DEL\_01 |
| Date: | 2015-10-17 |
| Version: | 1.1 |
| Status: | Review |

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2015

CONTENTS

CONTENTS 2

1. Introduction 3

1.1 Purpose of this Document 3

1.2 Scope 3

1.3 Objectives 3

2. DEPLOYMENT DESCRIPTION 3

2.1 Applications in the system 3

2.2 Application interactions 4

3. INteraction design 5

3.1 Use-Case Diagrams 5

3.2 User Interface Design – Tasker CLI 6

3.3 User Interface Design - TaskerMAN 10

APPENDICES 19

DOCUMENT HISTORY 22

# Introduction

## Purpose of this Document

The purpose of this document is to ensure that the client and design team understand and can visualise the software interface easily. It also lists the minimum requirements and interaction methods the software will use.

## Scope

This document shows the GUI design for TaskerMAN and TaskerCLI to facilitate further design. It goes on to list the communication methods between the various components and the minimum requirements to run those components.

## Objectives

The document aims to inform the reader of high level design choices by including typical use cases. From this the applications and their methods to interact with each other in order to facilitate the functional requirements was selected. It also contains mock up designs of the user interface and the functionality they will provide for users of this software.

# DEPLOYMENT DESCRIPTION

## Applications in the system

### 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 Linux 64-bit. [Appendix A] - using versions of the Java Runtime Environment lower than this may cause unexpected behaviour and is therefore not recommended.

JDBC will be used to allow 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]

### TaskerMAN

TaskerMAN is the web-based software component of the system. The website is built comprising of 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]

### TaskerSRV

TaskerSRV is the database component.

A MySQL relational database will be used. The version we will test against is MySQL 5.6.26 on a Linux 64-bit OS [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.

## Application interactions

# INteraction design

## Use-Case Diagrams



## User Interface Design – Tasker CLI

### Log In Window

TaskerCLI

Email:

Connection Settings

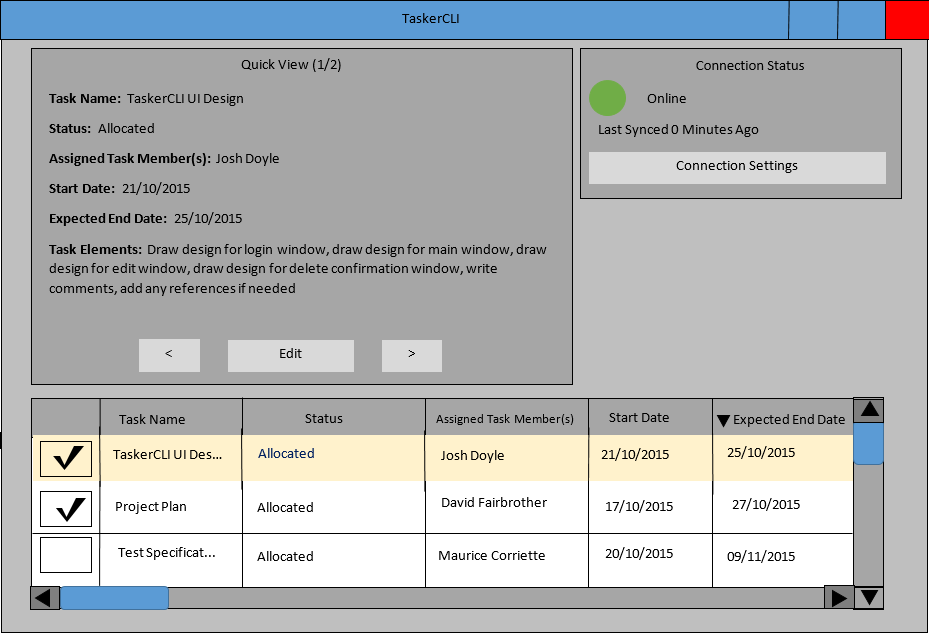
Log In

ddddd

appendices

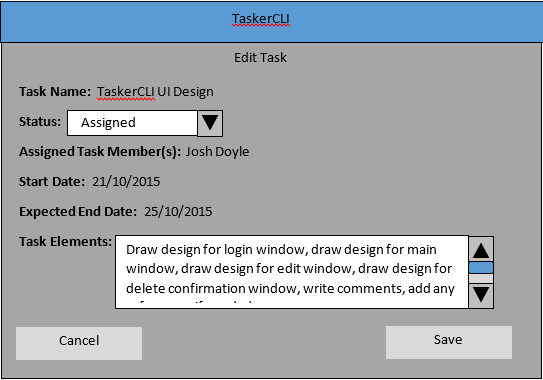
* Textbox for user to enter their email address for validation, as per requirements specification. [5]
* Log In button opens ‘Main Window’ when clicked, provided a valid email address has been entered.
* Clicking the ‘Connection Settings’ button will open the ‘Connection Settings’ window – this allows the user to configure their connection to the *TaskerSRV* database.
* Closing ‘Log In Window’ will bring up the ‘Exit Confirmation’ window.

### Main Window



* 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.

### Edit Window



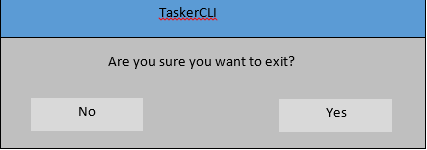
* The ‘Edit Task’ window is populated with the data of the task that was in the ‘Quick View’ panel on the ‘Main’ window when it was opened.
* The attributes of completion status and task elements are editable from this window.
* The completion status can be selected from a dropdown list. The default 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.

### Connection Settings Window



* 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.

### 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.

## User Interface Design - TaskerMAN

### General Notes

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

### Login Page



* If a valid email address is entered, you will be directed to the main page – otherwise access is prohibited, as is required. [5]

### Main Page



* Displays user’s email address at the top of the page, indicating who is currently logged in.
* The ‘Logout’ button displays the ‘Log Out Prompt’, and directs the user to the ‘Logout Successful’ screen upon successful logging out.
* The ‘Synced’ button will show a green dot if currently synchronised with *TaskerSRV*. Red if not.
* The ‘Settings’ button will bring up the ‘Connection Setting’ screen.
* All information shown in the database table and tasks can be selected and edited in bulk if desired.
* Buttons at the top provide self-explanatory functionality, namely the viewing, adding, editing and deleting of tasks, or refreshing the table.

### View Task Overlay



* Displays selected task. ‘Next’ and ‘Previous’ allow the browsing of other entries.
* Read-only

### Edit Task Overlay



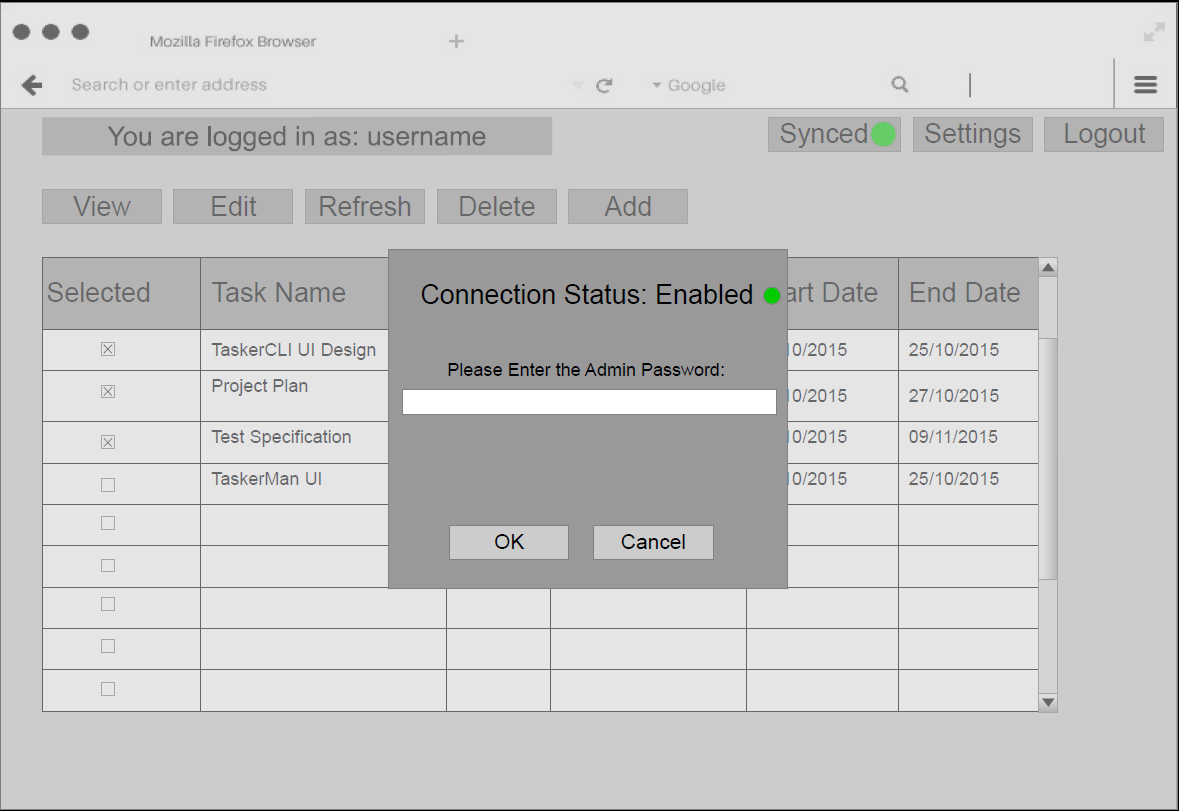
* By selecting one or more tasks, the ‘Edit Task’ overlay appears where details can be changed. Clicking ‘OK’ will save these changes. ‘Next’ and ‘Previous’ allow the navigation through other entries.
  + Next/Previous do not appear if you have only selected one task.
* The database will be updated after each edit, so one task can be modified before cancelling.

### Add Task Overlay



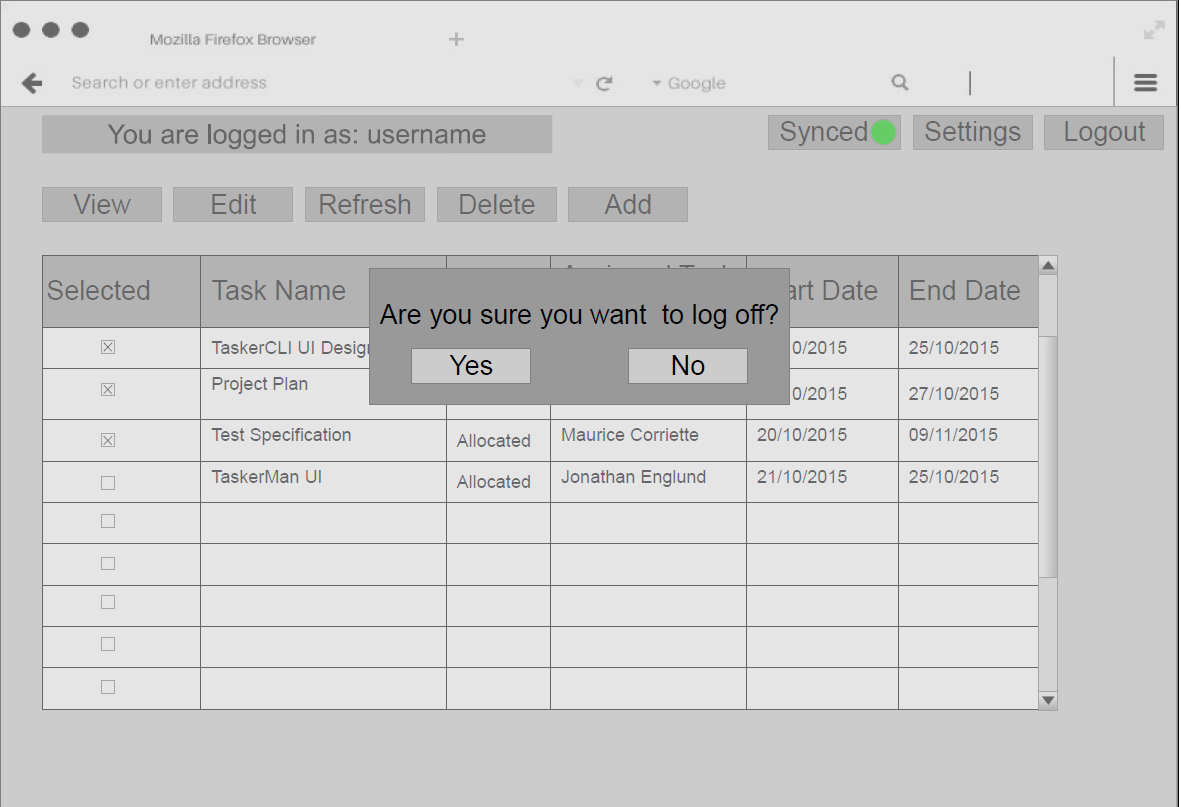
* In visual likeness to the ‘View Task’ screen, except blank where information can be inserted.
* Will use validation to ensure only correct/meaningful data can be entered.

### Connection Setting Screen



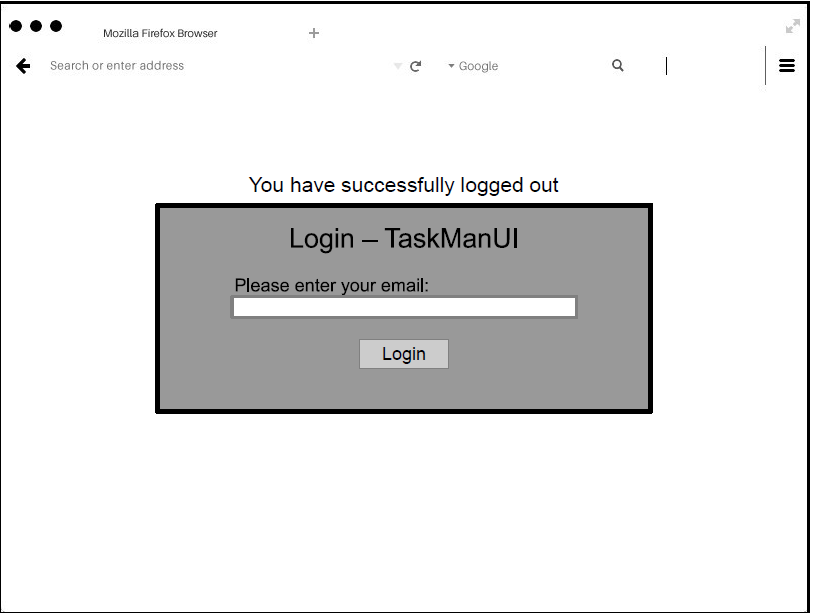
* Shows current synchronisation/connection status to *TaskerSRV.*
* Connection settings can be modified, but as this is a web based client, it is hidden behind an administration password wall and can be only modified by administrators.

### Log Out Prompt



* Prompts the user whether or not they wish to logoff. Choosing ‘No’ returns control to the user. Choosing ‘Yes’ directs the user to the ‘Logout Successful’ screen.
* The user cannot dismiss this prompt by clicking outside of its boundaries.

### Logout Successful



* User has been successfully logged out and is informed that they must login again to use the system.

references

|  |  |
| --- | --- |
| [1] | Tutorialspoint, “TutorialsPoint JUnit Environment Setup,” TutorialsPoint, 27 10 2015. [Online]. Available: http://www.tutorialspoint.com/junit/junit\_environment\_setup.htm. [Accessed 27 10 2015]. |
| [2] | 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]. |
| [3] | 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]. |
| [4] | 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]. |
| [5] | N. W. Hardy, C. J. Price and B. P. Tiddeman, *SE.QA.05 A - Design Specification Standards,* Aberystwyth University: Software Engineering Group Project, 2015. |
| [6] | Pixeden, “Firefox Web Browser Mockup Template,” CorruptedDevelopment, 26 08 2014. [Online]. Available: http://corrupteddevelopment.com/firefox-web-browser-mockup-template/. [Accessed 26 10 2015]. |

APPENDICES

APPENDIX A – Java Version



APPENDIX B – PHP Version



APPENDIX C – Apache Information



APPENDIX D – Linux information



APPENDIX E – MySQL Version



DOCUMENT HISTORY

| *Version* | *CCF No.* | *Date* | *Changes made to document* | *Changed by* |
| --- | --- | --- | --- | --- |
| 1.0 | N/A | 28/10/15 | Original version | OLE4 |
| 1.1 | N/A | 29/10/15 | Fixed template issue and fixed inconsistent tense usage | DAF5 |