**Group 17 – Test Specification**

Authors: Alexander Webb (ajw21), Frederick Kranich (frk2), George Vinters (ghv), Peter George (peg4), Jose Iznardo (jov2), Jack Thomson (jat36)

Date: 08/11/2015

Version: 2.1

Status: Release

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University

Contents

[1 Introduction 3](#_Toc435029326)

[1.1 Purpose of this Document 3](#_Toc435029327)

[1.2 Scope 3](#_Toc435029328)

[1.3 Objectives 3](#_Toc435029329)

[2 Tests 3](#_Toc435029330)

[2.1 System Tests 3](#_Toc435029331)

[2.2 Tests for the Database 4](#_Toc435029332)

[2.3 Tests for Desktop Client and Subsystem Tests for Desktop Client and Database 6](#_Toc435029333)

[2.4 Tests for Web 8](#_Toc435029334)

[2.5 Subsystem Tests for Web and Database 10](#_Toc435029335)

[3 References 12](#_Toc435029336)

[4 Document History 12](#_Toc435029337)

# 1 Introduction

## 1.1 Purpose of this Document

The purpose of this document is to list the tests designed to be performed with the projects desktop and web applications along with their details.

## 1.2 Scope

The document will detail the tests that the various applications and platform elements will be subjected to. These tests are designed to identify any issues, not fix them. This document should conform to standards laid out in SE.QA.06[1].

## 1.3 Objectives

The objectives of this document are as follows:

* Clearly lay out the design and purpose of all tests.
* Allow the development/testing team to be able to utilize and find relevance in the information gathered with them.

# 2 Tests

## 2.1 System Tests

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Ref** | **Req Tested** | **Test context** | **Input** | **Output** | **Expected Result** |
| SE\_17\_T\_01 | FR4 | Manager adds a new task. | Manager creates and adds the task | Appears under the correct users application and shows correct information | For both the user and the manager the task are visible |
| SE\_17\_T\_02 | FR5 | Manager edits the a task | Manager changes some values in a task | The task updates correctly for the users and they will be able to view it on the application | For both the user and the manager the changes are visible |
| SE\_17\_T\_03 | FR10 | User updates the task | The user edits a value in the task | The manager can see the update on the website and the team members can see the changes on the application | For both the user and the manager the changes are visible |
| SE\_17\_T\_04 | FR8a | User drops out of connection | The connection is removed from the pc | The application should be able to run as it did before and once connection is resumed the program will connect and update the database with both changes | The user should be able to see the tasks in TaskerCLI |
| SE\_17\_T\_05 | FR5 | Manager relieves a user of a task | Manager removes user in the interface | The user will no longer be able to see the task and will be removed from their application. | The user should no longer be able to see the task |
| SE\_17\_T\_06 | FR9 | While a user is out of connection the task is updated by the manager | The connection is removed then the manager updates the task | When the user connects back to the internet the task will be updated as expected | When connection is resumed the user will be able to see the changes |
| SE\_17\_T\_07 | FR9 | While the user is out of connection they update the task | The connection is removed then the user edits a value in the task | When the user connects back to the internet the task will be updated so every user can see the newly updated task | When connection is resumed both the user and the manager will be able to see the changes |
| SE\_17\_T\_08 | FR9 | While the user is disconnected if the manager updates the task but so does the user | The connection is removed then both the user and manager edit values in the task | The manager overrides the changes made by the user | When connection is resumed both the manager and the user will see the changes made by the manager |

## 2.2 Tests for the Database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Test Ref* | *Req being Tested* | *Test Content* | *Input* | *Output* | *Pass Criteria* |
| SE\_17\_T\_09 | FR1 | Check system can take first name | Legal chars into fields | Required sql statement | Characters entered added correctly to database |
| SE\_17\_T\_10 | FR1 | Check system can take second name | Legal chars into field | Required sql statement for database addition | Characters entered correctly added to database |
| SE\_17\_T\_11 | FR1 | Check error is thrown when illegal chars are added to first name | Add illegal chars into fields | Error statement indicating problem | Error thrown when illegal chars entered, not added to database |
| SE\_17\_T\_12 | FR1 | Check error is thrown when illegal chars are added to second name | Add illegal chars into fields | Error statement indicating problem | Error thrown when illegal chars entered, not entered to database |
| SE\_17\_T\_13 | FR1 | Check system accepts legal email address input | Email address added including only legal chars. | Required sql statement produced for database addition. | Email correctly added into database. |
| SE\_17\_T\_14 | FR1 | Check email field rejects illegal chars | Email containing illegal chars | Error statement indicating problem | Error thrown when illegal characters are used. |
| SE\_17\_T\_15 | FR1 | Delete from first name. | Delete command to database | Remove statement sent to database | Removal of data from first name |
| SE\_17\_T\_16 | FR1 | Delete from second name. | Delete command to database | Remove statement sent to database | Removal of data from second name |
| SE\_17\_T\_17 | FR1 | Delete from email. | Delete command to database | Remove statement sent to database | Removal of required data from email section. |
| SE\_17\_T\_18 | FR2 | Added new title | Add title with legal chars | Sql statement produced for database | New title added correctly |
| SE\_17\_T\_19 | FR2 | Add new title with illegal chars | Add title with illegal chars | Error message with reason | Error message is thrown |
| SE\_17\_T\_20 | FR2 | Add new team member | Add team member | Sql statement for database produced | Team member added to correct team |
| SE\_17\_T\_21 | FR2 | Remove new team member | Command for team member removal | Designated team member removed | Team member successful removed from team |
| SE\_17\_T\_22 | FR2 | Start date addition | Start date through calendar | Start date assigned to task | Start date successfully added. |
| SE\_17\_T\_23 | FR2 | Completion date added | Completion date added through calendar | Completion date added to task using sql statement | Completion date successfully added. |
| SE\_17\_T\_24 | FR2 | Addition of task elements | Task element description | Sql statement adding description to the database. | Task element added to the database correctly. |
| SE\_17\_T\_25 | FR2 | Assignment of team member to a task element. | Desired team member and task. | Assignment for team member to specific task. | Correct team member is added to correct task. |
| SE\_17\_T\_26 | FR2 | Assignment of a non-existent team member to a task | A team member not in database assigned to task. | Error message detailing what has gone wrong. | Error message is correctly thrown. Database doesn’t change. |
| SE\_17\_T\_27 | FR2 | Assignment of three values to a task. | Required values to single task. | Change of task status | Task status successfully changed to desired result. |

## 2.3 Tests for Desktop Client and Subsystem Tests for Desktop Client and Database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Ref** | **Req. being tested** | **Test Content** | **Input** | **Output** | **Pass Criteria** |
| SE\_17\_T\_28 | FR8 & FR8A | User login with username | User is to enter details of a created member of the team | The program should accept the user’s details and act correctly upon them. | The user must be accepted and the correct data selected according to the details entered. |
| SE\_17\_T\_29 | Users log in information retrieved from the database | Program started | Program loads and accesses the database for the user information | The program loads and correctly retrieves the user data and saves it to the machine for use later on |
| SE\_17\_T\_30 | User login with the mail | User is to enter details of a created member of the team | The program should accept the user’s details and act correctly upon them. | The user must be accepted and the correct data selected according to the details entered. |
| SE\_17\_T\_31 | Store locally the log in details to speed-up the login | The program is started | The program starts and retrieves all data from the database that it needs. | The program should get data from the database and store it in the local filesystem. |
| SE\_17\_T\_32 | Download task for the identified user and store locally | The program is started and the user’s information is given. | The program retrieves the required data and stores it correctly. | The data is obtained from the database about that user and stored correctly in the local file store. |
| SE\_17\_T\_33 | FR9 | Synchronized the list of task stored locally | The program has been running for more than 10 minutes | The program automatically goes to the server and checks for updates to the data and retrieves new data from the system. | The system successfully synchronises the data and updates the locally stored data. |
| SE\_17\_T\_34 | Check that all the tasks have status allocated or complete for the identified user | Task list from memory or database | Task list should be displayed on the screen and each task should have allocated or complete status next to it. | The task list is displayed and each task hs the appropriate task status of either allocated or complete. |
| SE\_17\_T\_35 | Additions on taskSRV take priority | The TaskerMAN should be loaded on one machine with the TaskerCLI on another and the team should input a new task to the same user that is being viewed on the TaskerCLI | The TaskerCLI should synchronise and update the list with the new tasks as a priority being displayed first before the old tasks. | The system should successfully accept the new task and display the new task on the desktop interface and should be displayed so that the user can see the new tasks at the top of the task list. |
| SE\_17\_T\_36 | Deletions on taskSRV take priority | The TaskerMAN should be loaded on one machine with the TaskerCLI on another and the team should delete (through abandonment) a task that belongs to the same user that is being viewed on the TaskerCLI | The system should sync and the task list should be updated to represent that the task has been abandoned (deleted). | The system correctly abandons the correct task and the task has been deleted from the task list through the status of abandoned. |
| SE\_17\_T\_37 | Local updates take priority | The user will change one of the tasks from allocated to complete. | The system should update the local file store and the database (if connection is possible, if not then should update the database from local store at earliest opportunity). | The system updates the local store and database (if possible) with the task alteration from allocated to complete. |
| SE\_17\_T\_38 | FR10 | Select a task | The user should select a task that is displayed to them | The task should be displayed in more detail and shows the details of the task such as comments and task description. | The system opens the correct task and displays the correct information on that task. |
| SE\_17\_T\_39 | Edit task steps/comments | An existing task should be edited so that it has a new comment with in it. The comment then should be edited so that it has some more description about the task in it. | The task should be updated and the comment thread updated with the new comment. The task should then be updated with the new additions to the comment. | The system accepts the new task comment and the update to it whilst displaying them correctly. |
| SE\_17\_T\_40 | Change the task status to completed | A task will be selected from the list and will have its status updated to complete. | The system should update the local copy and the database copy along with correctly showing the status in the task list. | The system accepts that the task is completed and updates all the necessary areas such as the local file store, database and task list being displayed. |
| SE\_17\_T\_41 | FR11 | Synch at start-up | The program will be started. | The program should access the database and gain the information and store it locally. | The program will get the correct data and store it in a local copy on the local file store. |
| SE\_17\_T\_42 | Synch every 5 minutes | The program will run for 5 minutes. | The program should use threads to then automatically sync with the data in the database. | The system should sync with the database after 5 minutes and should correctly update the local file store. |

## 2.4 Tests for Web

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Ref** | **Req. being tested** | **Test Content** | **Input** | **Output** | **Pass Criteria** |
| SE\_17\_T\_43 | FR3 | Check to see if the user can view the team members | Data for multiple team members | A list of all team members | The list correctly shows all members, with all data being shown. |
| SE\_17\_T\_44 | FR3 | Check to see if the user can update a team member | Existing team member data and new valid data to be entered in replace. | New data is stored and overwrites the old data. | The new data must be shown instead of the old data and must be replace the old data at the source. |
| SE\_17\_T\_45 | FR3 | Check to see if the user cannot enter incorrectly update a team member | Existing team member data and new invalid data | The new data must not be accepted and the old data is kept and shown as the current data | The new data is rejected and appropriate error messages are displayed and the old data is kept as current. |
| SE\_17\_T\_46 | FR3 | Check to see if a member can be deleted | Existing team member | Team member deleted | The team member must be removed from the system. |
| SE\_17\_T\_47 | FR4 | Check that the can add a new task and allocate it correctly | New task data of a task: valid data. The team member that the task is being allocated to. | The task is accepted and is correctly assigned to the correct member. | The system must show the task correctly assigned to the correct team member. |
| SE\_17\_T\_48 | FR4 | Check that the user cannot add a new invalid task and make sure that it is not allocated. | New task data of a task: invalid data. The team member it is to be allocated to. | The system rejects the new task and cannot allocate it to the team member | The system must show appropriate error messages and the system must not allow the task to be created or allocated. |
| SE\_17\_T\_49 | FR5 | Check if a task can be reallocated correctly | 2 team members with a valid task that is to be reallocated. | The system will show the task belonging to a new person. | The system should correctly reallocate the task to the new member specified. |
| SE\_17\_T\_50 | FR5 | Check to see if a task cannot be reallocated to non-existent team member | Team member with a valid task and a team member that does not exist in the system. | The system should stop the reallocation as it cannot find the member. | The system should provide appropriate error messages and must not allow the task to be reallocated. |
| SE\_17\_T\_51 | FR6 | Check to see if a task can be abandoned | A valid task belonging to a team member | The system should show that the task has been abandoned | The system shows that the task is abandoned and is reflected correctly in the rest of the system. |
| SE\_17\_T\_52 | FR7 | Check to see if there is a list of tasks that is viewable for each member | A team member with a collection of tasks (valid tasks) | The system should show the tasks that team member has. | The system shows the team member what tasks they have and that they are only theirs. |
| SE\_17\_T\_53 | FR7 | Check to see if the details of the tasks in the list are correct and all details are present | A team member with a collection of tasks (valid tasks) | The system should show all the correct details and all details should be correct. | The team member’s tasks must be shown correctly and all details must be present. |

## 2.5 Subsystem Tests for Web and Database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Ref | Req bring tested | Test content | Input | Output | Pass Criteria |
| SE\_17\_T\_54 | FR3 | Check that adding a new team member updates the database | Input the new members email address, name and any other relevant details | The new team member's data is sent to the database | The database is updated with the info about the new team member |
| SE\_17\_T\_55 | FR3 | Newly created team members can be accessed from the web application | Create a new team member filling out all the required information | The web application displays the newly created team member | Newly created team members can be viewed on the website application |
| SE\_17\_T\_56 | FR3 | Editing existing team members updates the database files | Edit the information of an existing team member | The information on the database is updated | The database should now reflect the information on the website application |
| SE\_17\_T\_57 | FR3 | Website application does not allow illegal characters to be sent to the database | Create a new team member using illegal characters such as & \ / ' | Error message is displayed warning of illegal character being used | System displays "illegal characters" error |
| SE\_17\_T\_58 | FR3 | Website application does not allow illegal characters to be sent to the database | Edit an existing team member using illegal characters such as & \ / ' | Error message is displayed warning of illegal characters being used | System displays "illegal characters" error |
| SE\_17\_T\_59 | FR3 | Deleting a team member prompts a confirmation message | Delete an existing team member | Confirmation message is displayed and team member is not yet removed | System displays "are you sure" confirmation message |
| SE\_17\_T\_60 | FR3 | Deleting a team member removes that team member from existing tasks | Remove a team member who is assigned to a task | Team member is removed from any tasks on the database | Team member is no longer associated with any tasks on the database |
| SE\_17\_T\_61 | FR3 | Selecting "Yes" at team member deletion confirmation removes the team member from the database | After selecting to delete a team member, select "Yes" when the confirmation message is displayed | Confirmation message is no longer displayed and the selected team member is deleted | Selected team member is removed from the database |
| SE\_17\_T\_62 | FR3 | Selecting "No" at team member deletion confirmation does not remove the team member from the database | After selecting to delete a team member, select "No" when the confirmation message is displayed | Confirmation message is no longer displayed and the selected team member is not deleted | Selected team member is not removed from the database |
| SE\_17\_T\_63 | FR4 | Newly created tasks are sent to the database | Create a new task filling in the required information | The task data is sent to the database | Database now includes the newly created task |
| SE\_17\_T\_64 | FR4 | Website application does not allow illegal characters to be sent to the database | Create a new task using illegal characters such as & \ / ' | Error message is displayed warning of illegal characters being used | System displays "illegal characters" error |
| SE\_17\_T\_65 | FR4 | Assigning a team member to a task updates the database | Assign a team member to a task | Task now has the selected team member assigned to it | Database is updated showing that the team member is assigned to the task |
| SE\_17\_T\_66 | FR5 | Reallocating a task updates the database | Reallocate an existing task to a different team member | The task is now allocated to the selected team member | The database updates to show that the new team member is allocated to the task |
| SE\_17\_T\_67 | FR5 | Reallocated tasks must include at least one team member | Reallocate a task so that no team members are assigned | The task cannot be reallocated without at least one team member being assigned to the task | System display error message |
| SE\_17\_T\_68 | FR7/4 | Web application can view a list of created tasks | Create a new task filling in the required information | All created tasks are sent to the web application to be viewed | Information of tasks is correctly sent from the database to the web application |
| SE\_17\_T\_69 | FR6 | Abandoned tasks are removed from the database | Choose to abandon a task | The task is removed from the web application | The selected task is removed from the database |
| SE\_17\_T\_70 | FR7 | Viewing a task receives the task information from the database | Select a task a choose to view information on the tasks | Information is received from the database and more details about the task can be viewed | All information for the selected task can be viewed |

# 3 References

[1] Software Engineering Group Projects: Test Procedure Standards

# 4 Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | CCF No. | Date | Changes made to document | Changed by |
| 1.0 | N/A | 8/11/2015 | Document created by Frederik Kranich | FRK2 |
| 1.1 | N/A | 11/11/2015 | Added the tests I didn’t get before the initial creation of the document | FRK2 |
| 2.0 | N/A | 11/11/2015 | Final additions to tests made | AJW21 |
| 2.1 | N/A | 11/11/2015 | Test references added. | AJW21 |