**CS221 Group 09**

**Test Specification**

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

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

**Date:** 10/11/15

**Version:** 1.0

**Status:** Release

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2015

**CS221 Group 09**

**Test Specification Contents**

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

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

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

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

**2 TEST SPECIFICATIONS…………………………………………………………………..** 4

3 **REFERENCES**…………………………………………………………………………………. 9

**4 DOCUMENT HISTORY**……………………………………………………………………. 9

**1 INTRODUCTION**

**1.1 Purpose of this document**

The purpose of this document is to ensure that a list of tests are created which specifies in detail each system test to be completed as part of a formal test process, with each test directly referencing the requirements specification [1].

**1.2 Scope**

The Test Specification itself covers all system tests which need to be completed in order to not only identify what parts of the Requirements Specification [1] have not been met, but to also ensure the functionality of final system.

With regards to the intended readers, the Test Specification is an important component when it comes to delivering a system which meets the requirements specification [1] and thus should be referred to by all team members during development.

**1.3 Objectives**

To provide a test specification which has within it a set of reproducible actions to test all of the main functionality of the system.



**3 TEST SPECIFICATION**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Reference | Req Being Tested | Content | Input | Output | Pass Criteria |
| SE-F-001 | FR1 | Regexp that restricts first name format to only latin characters and a hyphen, max 20 characters. | Enter a first name of 20 characters. |  | Data is stored correctly. |
| SE-F-002 | FR1 | Regexp that restricts surname format to only latin characters and a hyphen, max 20 characters. | Enter a surname of 20 characters. |  | Data is stored correctly. |
| SE-F-003 | FR1 | Check to see whether regexp works correctly in name-field. | Enter a first name with a question mark, dot, comma and semicolon . | Error message: “illegal character used!” | System displays error message. |
| SE-F-004 | FR1 | Check to see whether regexp works correctly in surname-field | Enter surname with a question mark, dot, comma and semicolon | Error message: “illegal character used!” | System displays error message |
| SE-F-005 | FR1 | Check to see whether regexp works correctly | Leave name-field empty | Error message:  “No name entered” | System displays error message |
| SE-F-006 | FR1 | Check to see whether regexp works correctly | Leave surname-field empty | Error message:  “No surname entered” | System displays error message |
| SE-F-007 | FR1 | Regexp that restricts email entry to only structurally valid email addresses. Using the regexp from:  http://www.regular-expressions.info/email.html | Enter a valid email. |  | Data is stored correctly. |
| SE-F-008 | FR1 | Check to see whether regexp works correctly | Enter an invalid email address i.e cs221.project.com | Error message: “invalid email address supplied!” | System displays error message. |
| SE-F-009 | FR1 | Check to see whether regexp works correctly | Leave email field empty. | Error message: “No email address entered!” | System displays error message. |
| SE-F-010 | FR2 | The task relation needs to be able to store a title, a team member, start date, completion date, a list of task elements and a status. | Enter valid information for each field. | A task is created. | Data is stored correctly. |
| SE-F-011 | FR2 | The task relation needs to be able to be updated. | Update a task. | A task is updated. | Data is updated correctly. |
| SE-F-012 | FR3 | Site should provide the ability to create a new team member. A check is needed to make sure the form is completed correctly with no fields left blank. Relevant regexps would be required on each field. | Enter the form correctly. | New team member added to list of team members. | Team member stored correctly. |
| SE-F-013 | FR3 | Check to see what happens when a field is entered incorrectly i.e name contains numbers. | Enter a name with a number i.e Steven67 | Error message: “Form entered incorrectly!”. | System displays error message. |
| SE-F-014 | FR3 | Check to see what happens when a field is left blank in the form. | Leave team member email address empty. | Error message: “Fields can’t be left blank!” | System displays error message. |
| SE-F-015 | FR3 | Site should provide the ability to update information about team members. Similarly to adding a team member, checks will be required to make sure the form is entered correctly with no blank fields. Relevant regexps will be required. | Enter the form correctly. | Team member information has been updated on the team team members page. | Team member stored correctly. |
| SE-F-016 | FR3 | Check to see what happens when a field is left blank in the form. | Leave team member name empty. | Error message: “Fields can’t be left blank!” | System displays error message. |
| SE-F-017 | FR3 | Site should provide the ability to delete a team member from the system. Checks should be in place to make sure the user is certain they want to delete the user. | User clicks on delete team member button and clicks yes to the check. | Message: “Team member has been deleted”  Team member is removed from the team member list. | Team member is removed from the system and the message is shown. |
| SE-F-018 | FR3 | Check to see what happens if the user clicks no to the check. | User clicks on delete team member button and clicks no to the check. | Message: “Team member wasn’t deleted.” | System displays message. |
| SE-F-019 | FR4 | Site should provide the ability to create new tasks. Checks should be in place to make sure the task is valid. | User clicks on new task button and adds a valid title. | Task should be added to the list of tasks. | Task should be stored correctly. |
| SE-F-020 | FR4 | Check to see what happens if the user enters an invalid task title. | User clicks on new task button and adds an invalid title i.e a title with non-latin characters. | Error message: “Please provide a valid title! Only latin characters are permitted.” | System displays error message. |
| SE-F-021 | FR4 | Site should provide the ability to add information about tasks. Checks should be in place to make sure the form is entered correctly. | User clicks on edit task and fills in form correctly. | Task information has been updated. | Task should be updated correctly. |
| SE-F-022 | FR4 | Check to see what happens if the user enters invalid information. | User clicks on edit task and fills in form incorrectly. | Error message: “Please make sure the form is filled correctly.” | System displays error message. |
| SE-F-023 | FR4 | Site should provide the ability to allocate a team member from a list of members to a task. | User allocates a team member. | Task is allocated to team member. | Task should be updated correctly. |
| SE-F-024 | FR5 | Site should provide the ability to re-allocate a task to a different team member. | User clicks on edit task and selects a new user from the list of team members. | Task is allocated to team member. | Task should be updated correctly. |
| SE-F-025 | FR6 | Site should provide the ability to set a task as abandoned. | User clicks on edit tasks and changes the task status to abandoned. | Task is shown to be abandoned. | Task should be updated correctly. |
| SE-F-026 | FR7 | Site should provide the ability to show a list of tasks, with their titles and allocated team members. | User clicks on the task page. | A list of tasks is shown with their titles and allocated team members. | System displays list of tasks with their titles and allocated team members. |
| SE-F-027 | FR7 | Site should list these tasks by expected completion date. | User clicks on the task page. | A list of tasks are shown with their titles and allocated team members, sorted by expected completion date. | System displays a list of tasks sorted by expected completion date. |
| SE-F-028 | FR7 | Site should provide the ability to filter tasks by task status and/or allocated team member. | User clicks on a status from a drop down list on the task page. | A list of tasks are shown with their titles and allocated team members for the selected status. | System displays all tasks for the selected status. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SE-F-029 | FR8 | Test whether an incorrect email address is able to log in. | Enter a random email address that isn’t stored in the database. | Message: “Invalid email address”. | Error message is shown to the user. |
| SE-F-030 | FR8 | Test whether the email-text field is empty | Click on the login button without any input in the text field | Message “Please enter an email address” | Error message is shown to the user. |
| SE-F-031 | FR8 | Test whether the password-text field is empty | Click on the login button without any input in the text field | Message “Please enter a password” | Error message is shown to the user. |
| SE-F-032 | FR8 | Test whether an incorrect password is entered | Password which is not same as one paired with login | Message “Wrong password” | Enter message is shown the user. |
| SE-F-033 | FR8 | Check to see whether incorrect input is acknowledged. | Enter random string of chars and ints/symbols. | User isn’t logged in and message is displayed:”Incorrect input, enter valid email address. | Message is displayed and user isn’t allowed further, |
| SE-F-034 | FR8 | Check to see whether ‘remember me’ script works | Tick ‘remember me’ box | box ticked | On next app launch email and censored password are already filled in appropriate fields |
| SE-F-035 | FR8 | Test whether the user’s tasks have been stored on their home directory. | Check user's home directory for task file. | Task is displayed in user profile. | Task is displayed. |
| SE-F-036 | FR9 | Test whether local tasks gets synchronised with the external database | New local tasks | The new tasks are on server | Use local database |
| SE-F-037 | FR9 | Test whether new tasks on server gets synchronised with taskerCLI | New tasks in database | The new tasks are on taskerCLI | Error message is shown to the user. |
| SE-F-038 | FR9 | Test whether tasks that are marked as abandoned or complete does not show in taskerCLI | Tasks are marked as abandoned or complete in database | No tasks are marked as abandoned or complete in taskerCLI | Delete tasks marked as abandoned or complete in taskerCLI |
| SE-F-039 | FR10 | Test whether the task is shown to the user after click in list | User clicks on a task in a list of tasks | The field that shows the task does now show the task which the user clicked on | Reload list of tasks |
| SE-F-040 | FR10 | Test whether the user can add comments about the progress | User clicks on the button that adds the comment | The local and external database does now contain that comment | Add comment to local and external database |
| SE-F-041 | FR10 | Test whether the user can change status to completed | The user clicks on the “complete” button | The task status is changed to “complete” in local and external database | Update the task in the databases to “complete” |

**3 REFERENCES**

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

[2] QA Document SE.QA.06 - Test Procedure Standards

**4 DOCUMENT HISTORY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | CCF# | Date | Change Description | Changed By |
| 1.0 | N/A | 02/11/15 | Document Layout Added | J.D |
| 1.1 | N/A | 10/11/15 | Formatted Document for Release | J.D |