Test Cases and Test Plan

Spartan Course Analysis & Matching

Version 2.0 - 5/16/17

San José State University Charles W. Davidson College of Engineering

Mark Casapao
Christian Timbol
Preyrna Yadav
Hiep Nguyen

Revisions

Version	Primary Authors	Description of Version	Date Completed
v1.0 draft	Mark Casapao Christian Timbol Preyrna Yadav Hiep Nguyen	An outline or foundation of the document	4/17/17
v1.0 final	Mark Casapao Christian Timbol Preyrna Yadav Hiep Nguyen	A finalized and confirmed iteration of the document	5/7/17
v2.0 final	Mark Casapao Christian Timbol Preyrna Yadav Hiep Nguyen	A refined iteration of the document. All docs are now interconnected.	5/16/17

Table of Contents

1.0 Introduction

The purpose of this document is to inform the reader of how SCAM's features and functionalities will be set up, evaluated, and examined in all of the testing phase. Mainly a listing of all the test cases as well as a log that tracks of the test results will be kept in check and updated as more variations in the program may be attended.

1.1 System Overview

The following subcategories will consist of the method of approach that will be used in testing the software, and is subject to change as the project is further developed. The Test Approach describes of what process the features of SCAM will undergo and explains the different phases of the evaluation. The traceability matrix then follows as a up-to-date table that will refresh the reader of how the test cases tie back to the other documents.

1.2 Test Approach

The test cases will undergo three different phases. These phases are, evaluating, success, and failure. Evaluating is a term to describe that the test case is still under extensive evaluation and must finished and updated as soon as possible. Success and failure is a notification to whether the test case is already evaluated, with the addition that failure requires the software to get another round of implementation. Further in the document, the resources required will be described in greater detail (2.3).

1.3 Requirements Traceability Matrix

Functional Requirements	Design Specifications	Test Cases	Implementation
2.2.1.1 Registration - User must be able to	1.1 Users register via login page	3.1.1 Registration is tested via BBT	1.1 Executed as the default screen

register SCAM account.		- working	
2.2.1.2 Logging In - User must be able to log in using the newly generated account.	1.2 User is capable of logging in by entering fields via LoginPage	3.1.2 Login is tested via BBT and unit tests - working	1.2 Executed as the default screen
2.2.2.1 Edit Profile - Fill in blanks with necessary information.	2.1 ProfilePage provides an option to edit profiles	3.2.1 Profile is tested via BBT and unit tests - working	2.1 Executed at the Profile Page
2.2.2.2 Delete Info - Privacy options	2.2 ProfilePage's options provides a feature to hide all information necessary	3.2.2 Hiding is tested via BBT and unit tests - working	2.2 Executed at the Profile Page
2.2.2.3 Delete Account - Option to delete account	2.3 Users can delete accounts via ProfilePage	3.2.3 Deletion is tested via unit tests - working	2.3 Executed at the Profile Page
2.2.2.4 View Profile - User must be able to view their own profile.	2.4 Users are able to see their profile via ProfilePage	3.2.4 Viewing is tested via BBT - working	2.4 Executed at the Profile Page
2.2.3.1 View Schedule - Visual schedule	3.1 A schedule is displayed through SchedulePage	3.3.1 Viewing is tested via BBT - working	3.1 Executed in the Wall Page
2.2.3.2 Edit Schedule - Add/drop/modify class	3.2 SchedulePage provides options to edit class schedules	3.3.2 Editing is tested via unit tests and BBT - working	3.2 Executed in the Wall Page
2.2.3.3 Feedback - User must be able to contribute information.	3.3 Users are able to rate and review classes	3.3.3 Feedback is tested via unit tests - working	3.3 Executed in the Wall Page
2.2.4.1 View Inbox - access inbox page from the menu	4.1 Inbox is provided in SocialPage	3.4.1 Inbox is tested via unit tests and BBT - working	4.1 Executed using the menu in the corner of the screen
2.2.4.2 Send/Delete Messages - send their own messages, delete old messages.	4.2 Messages are sent back and forth between inboxes	3.4.2 Msg is tested via unit tests - working	4.2 Executed using the menu in the corner of the screen
2.2.4.3 Search Users - find users via search bar.	4.3 A search bar is provided within the SocialPage	3.4.3 Searching is tested via BBT - working	4.3 Executed using the menu in the corner of the screen

2.2.4.4 Add User to Friends - add user to friends	4.4 Every user has a friends list paired with him/her	3.4.4 Adding is tested via BBT and unit tests - working	4.4 Executed using the menu in the corner of the screen
2.2.4.5 View User Profile/Schedule -view contacts' profile and schedules	4.5 Users can view the profiles of other users they search up in the search bar	3.4.5 Viewing is tested via BBT - working	4.5 Executed using the menu in the corner of the screen
2.2.4.6 View Notifications - view any recent events around the user's network	4.6 Users can view their notifications via the notifications list	3.4.6 Notifications were tested via BBT - working	4.6 Executed using the menu in the corner of the screen

2.0 Test Plan

SCAM will have many of its features evaluated, but note that some features may be impossible or aren't explicitly getting evaluated through the utilization of test cases. The following features listed below will then be categorized to bring additional context to just what is exactly getting evaluated.

2.1 Features to be Tested

All of the feasible features of SCAM will be thoroughly tested and labeled properly. The traceability matrix should summarize most of the functional requirements that will be worked in great detail throughout the Test Cases section (3.0). For the sake of summarization, the five main categories will be explained briefly. In the login screen, the user must be capable of creating an account, retrieving forgotten account information, and able to log into their profiles. In the profile page, the user should be able to edit and maintain their personal features and have the option to delete or hide their account. For the two schedule pages, current schedule and future schedule, the user must be able to create and manipulate a schedule that they have / will have.

These schedules must also provide interactive features (with some exceptions to the future schedule as the user may or may not acquire it) such as rating a class, providing a review, and viewing additional information about a particular course they're interested in. From the inbox page, the user shall be able to see and interact with their own personal inbox, their following contacts list, and a search bar that will enable them to preview the profiles of fellow users.

2.2 Testing Tools and Environment

The program will extensively be tested through the use of several IDEs in terms of software and is run on both PC and Mac platforms. All personnel that work on the project will run each of the test cases separately on their own individual devices, and possibly even run early-user tests.

3.0 Test Cases

Requirements	Input	Test	Expected output	Actual output
3.1.1 Registration	- Valid SJSU email - Valid password	Step 1: Click "register account" Step 2: Enter SJSU email Step 3: Enter password Step 4: Click "submit"	View login screen	Success
3.1.2 Logging in	- Registered account with valid username and password	Step 1: Enter username Step 2: Enter password Step 3: Click "enter"	Enter into student portal, and view home screen	Success

3.2.1 Edit profile	- Valid profile associated with registered account	Step 1: Click “edit profile” Step 2: Edit whatever information needed Step 3: Click “save”	View the edited profile.	Success
3.2.2 Delete information	-Registered account with valid username and password	Step 1: After being in edit mode, delete/hide whatever information. Step 2: Click “save”	view Edited profile	Success
3.2.3 Delete account	- Registered account with valid username and password	Step 1: Click “delete account” Step 2: Enter answers for security questions Step 3: Click “confirm”	Account deleted	Success
3.2.4 View profile	- Registered account with valid username and password	Step 1: Click “home”	View profile	Success
3.3.1 Edit schedule	- Registered account with valid username and password	Step 1: Click status button Step 2: Add/delete classes i.e. statuses Step 4: Click the post button	View schedule	Success
3.3.2 View schedule	- Some type of schedule	Step 1: Click profile	View schedule	Success

3.3.3 Feedback	- Class	Step 1: Click “comment” on a status containing a class Step 2: Give a teacher and/or class review Step 3: Click “save”	Updated information on class	Success
3.4.1 View inbox	- Registered account with valid username and password	Step 1: Click “view inbox”	Inbox page	Success
3.4.2 Send messages	- Registered account with valid username and password	Step 1: Click “view inbox” Step 2: Click “send message” Step 3: Pick contact and write a message Step 4: Click “send”	-Message sent -View inbox	Success
3.4.3 Delete messages	- Some kind of message	Step 1: Click “view inbox” Step 2: Select a message Step 3: Click “delete message” Step 4: Click “yes” or “no” for confirmation	-Message deleted -View inbox	Success
3.4.4 Search users	- Registered account with valid username and password	Step 1: Click on search bar Step 2: Type user’s name Step 3: Click “enter” Step 4: Click on intended user	View user’s profile (depending on privacy settings)	Success

3.4.5 Add user to contacts	- Intended user	Step 1: Click “send friend request” on user’s profile	If user accepts, get notification.	Success
3.4.6 View user profile/schedule	- Intended user	Step 1: Click on user’s profile	View profile	Success
3.4.7 View notifications	- Registered user with valid username and password	Step 1: Click on notifications	View notif	Success

4.0 Additional Material

This particular section incorporates another, more precise and accurate, version of the cost of estimation. There is more detail and deeper analyzation put into getting a smoother estimate for the cost of the project.

4.1 Project Plan

The software development of this project was via Javascript and HTML, and PHP to a MySQL database. Using the Deliverables and our own capabilities, our web app turned out to be not exactly like the Deliverables, but it did complete the main functions.

4.2 Cost of Estimation

Using elements of COCOMO II, the cost for this software project has been estimated. In Figure 1, the main workers and their estimated cost per hour and per day are calculated. Using this, in Figure 2, the project’s total cost may be estimated. As seen, the Design and Development Effort reflect most of the costs of this software project. In total, the project would cost around $14,560.

Cost Units		
Workers	**per hour**	**per days**
Architect/Designer	50	400
Developer (Blended rate for Senior and Junior Developer)	30	240
Testing Lead	50	400
Tester	30	240
Technical Writing Cost	30	240
On-site Manager	150	1200

Figure 1: Workers' estimated cost

Total Project Cost			
Category	**Effort (Days) Estimated**	**Effort (Days) to Quote**	**Cost**
Requirements/Design Effort	11.4	11.0	$4,400
Development Effort	20.5	21.0	$5,040
Test Planning	3.7	4.0	$1,600
Testing Effort	4.6	5.0	$1,200
Documentation Effort	2.3	3.0	$720
Deployment Effort	0.5	1.0	$400
On-site Management Effort	0.5	1.0	$1,200
Total	**43.4**	**46.0**	**$14,560**

Figure 2: Total estimated cost for project