**Test Plan**

**CV Creation**

**COP4331C-0002, Fall 2017**

Team Name: Team 15

Team Members:

* Conor Forgey
* Alex Acevedo
* Elizabeth Maspoch
* Zaina Aljallad
* Anthony Hinnant

**Contents of this Document**

Overall Objective for Software Test Activity

Description of Test Environment

Overall Stopping Criteria

Description of Individual Test Cases

Appendices

**Overall Objective for Software Test Activity**

Our strategy is to primarily test the functional aspects of our project to ensure that our application meets the minimum requirements for the service. We expect to determine the errors in functionality, if any (e.g. major components simply not functioning properly, unexpected outcomes from different inputs).

**Description of Test Environment**

Apple iMac

* 1 TB storage
* 8 GB DDR3 RAM
* Quad-core 2.7 GHz Intel i5-4570R Processor
* macOS Sierra 10.12.6

**Stopping Criteria**

The stopping criteria revolves around testing until the errors interfere with progress on the functionality. Minor failures (cosmetic errors, results that don’t affect dependencies) will be documented as they are encountered, but the testing will continue. Once an error is encountered that improperly influences a dependency, the effect on the dependency will be evaluated; if we deem the error as too critical, such that it will cause further errors if testing continues, testing will stop until the encountered error is resolved.

If no errors are determined for a functional test and all outlined permutations of it, we will continue to the next available test.

For our team, “good enough” means that the project is functionally sound; functional components work as expected. Trivial errors that don’t affect the functionality itself are not preferred, but are acceptable for this definition.

**Description of Individual Test Cases**

* Test Objective: Create an Account (REG)
* Test Description: We will determine a username that is known to not exist in the user database (as well as an email address), and assign a password to accompany it. The test will involve navigating to the “Login/Registration” page, entering the new username, an email address, and the password into the respective text boxes.
  + Username: username[X], where X = a sequential number not previously used for testing, e.g. username1 -> username2 -> username3.
  + Password: password[X], where X = a sequential number not previously used for testing, e.g. password1 -> password2 -> password3.
  + Email: email[X]@mailinator.com, where X = a sequential number not previously used for testing, e.g. email1 -> email2 -> email3
    - Mailinator is a web service that allows you to enter a username, then check the inbox of the given email address, essentially giving us an infinite amount of email addresses for testing without the need to register with a website.
* Test Conditions: See Test Environment
* Expected Results: that user is successfully registered to the website. The username/password/email are encrypted and stored in the database, and entering their username and password into the “Login” prompt will give us access to that user’s account.
* Test Objective: Log In (LOG)
* Test Description: we will enter a username and password that currently exists in the database into the “Log In” prompt.
  + Username: username
  + Password: password1
* Test Conditions: See Test Environment
* Expected Results: access granted to that user’s account, which would allow us to enter information, select a template, etc.
* Test Objective: Fail to Create Account A (REGA)
* Test Description: We will attempt to create an account with a username that already exists in the database. The account will be entered into the database prior to testing.
  + Username: username
  + Email: [failtestA@mailinator.com](mailto:failtestA@mailinator.com)
  + Password: password
* Test Conditions: See Test Environment
* Expected Results: the account will not be added to the database, and a prompt will appear notifying us that we failed to register the account.
* Test Objective: Fail to Create Account B (REGB)
* Test Description: We will attempt to create an account with an email address that already exists in the database. The account will be entered into the database prior to testing.
  + Username: usernameA
  + Email Address: [failtestB@mailinator.com](mailto:failtestB@mailinator.com)
  + Password: password
* Test Conditions: See Test Environment
* Expected Results: the account will not be added to the database, and a prompt will appear notifying us that we failed to register the account.
* Test Objective: Fail to Create Account C (REGC)
* Test Description: We will attempt to register with a blank username and password field.
* Test Conditions: See Test Environment
* Expected Results: registration will fail, and a user account won’t be generated.
* Test Objective: Fail to Log In A (LOGA)
* Test Description: We will enter an account name that doesn’t exist in the database, in conjunction with a password, into the “Log In” prompt.
  + Username: usernameZ
  + Password: password
* Test Conditions: See Test Environment
* Expected Results: login will fail, and we won’t gain access to a user’s account.
* Test Objective: Fail to Log In B (LOGB)
* Test Description: We will enter an account name that exists in the database, in conjunction with an incorrect password, into the “Log In” prompt.
  + Username: username
  + Password: passwordA
* Test Conditions: See Test Environment
* Expected Results: login will fail, and we won’t gain access to a user’s account.
* Test Objective: Fail to Log In C(LOGC)
* Test Description: We will attempt to log in with a blank username and password field.
* Test Conditions: See Test Environment
* Expected Results: login will fail, and we won’t gain access to a user’s account (or the admin account).
* Test Objective: Enter Information (INFO)
* Test Description: we will navigate to the section of the website that allows us to enter user information, and enter sample text into each text box (Education, Work Experience, Other Skills & Qualifications), and attempt to save it to the user’s account.
  + Education:

University of Central Florida: 1 year

Stanford University: 1 year

* + Work Experience:

McDonald’s: 6 days

Lazy Moon: 5 years

* + Other Skills:

Underwater Basket Weaving

Making Ropes

* Test Conditions: See Test Environment
* Expected Results: the information will be stored on our server, tied to that account.
* Test Objective: Upload Photo (UP)
* Test Description: upload a pre-determined photo in the “Upload Photo” section for entering information.
* Test Conditions: See Test Environment
* Expected Results: photo is uploaded to the server and tied to the user’s account.
* Test Objective: Upload Wrong Photo (UPB)
* Test Description: attempt to upload a non-JPG, non-PNG file in the “Upload Photo” section for entering information.
* Test Conditions: See Test Environment
* Expected Results: file fails to upload.
* Test Objective: Upload Template (UT)
* Test Description: navigate to the “Upload Template” prompt, and upload a pre-determined template file to the server.
* Test Conditions: See Test Environment
* Expected Results: template is uploaded to the server, and is available for the user to select as a template.
* Test Objective: Upload Wrong Template (UTB)
* Test Description: attempt to upload a non-PDF file with the “Upload Template” prompt.
* Test Conditions: See Test Environment
* Expected Results: the file will be rejected, and not uploaded to the server.
* Test Objective: Select Template (ST)
* Test Description: select a template that is available for use. This test will be repeated for each available template, starting with the first template, then second, third, etc.
* Test Conditions: See Test Environment
* Expected Results: the selected template will be highlighted, and the “Generate CV” button will now be available for user interaction.
* Test Objective: Generate CV (GCVA)
* Test Description: enter user information, select a template file, and press the “Generate CV” button to generate an editable CV.
* Test Conditions: See Test Environment
* Expected Results: we will be navigated to a new page that displays the CV template with editable text boxes formatted into it.
* Test Objective: Generate Blank CV (GCVB)
* Test Description: leaving the “Enter Information” text boxes blank, we will select a template file, and press the “Generate CV” button to generate an editable CV with no user information.
* Test Conditions: See Test Environment
* Expected Results: we will be navigated to a new page that displays the CV template with editable text boxes formatted into it, but each text box should be blank.
* Test Objective: Edit CV (ECV)
* Test Description: select a text box in the CV. Then, we will add the letter ‘A’, delete it, and then repeat the process with a line break. Then, we will delete the first word entered into each text box, and re-add it. Repeat this process for each text box in the CV.
* Test Conditions: See Test Environment
* Expected Results: the text box will accept the inputs and properly re-format as the information is added.
* Test Objective: Download CV (DCV)
* Test Description: we will download the completed document.
* Test Conditions: See Test Environment
* Expected Results: the document will be encrypted and downloaded to our test computer.

**Trace of Individual Test Cases to Requirements**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | # | CV1 | CV2 | CV3 | CV4 | CV5 | CV6 | CV7 | CV8 | CV9 | CV10 | CV11 | CV12 | CV13 | CV14 | CV15 |
|  | 70 | 4 | 4 | 4 | 4 | 3 | 1 | 3 | 2 | 2 | 2 | 1 | 1 | 2 | 1 | 1 |
|  | 35 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REG | 3 | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |
| LOG | 1 |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |
| REGA | 3 | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |
| REGB | 3 | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |
| REGC | 3 | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |
| LOGA | 2 |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |
| LOGB | 2 |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |
| LOGC | 2 |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |
| INFO | 3 |  |  |  |  |  | X | X |  |  | X |  |  |  |  |  |
| UP | 2 |  |  |  |  |  |  | X |  |  | X |  |  |  |  |  |
| UPB | 1 |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |
| UT | 2 |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |
| UTB | 2 |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |
| ST | 2 |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |
| GCV | 1 |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |
| GCVB | 1 |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |
| ECV | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |
| DCV | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | # | DB1 | DB2 | DB3 | DB4 | DB5 | DB6 | DB7 | DB8 | DB9 | DB10 | DB11 | DB12 | DB13 | DB14 | DB15 |
| # | 68 | 4 | 4 | 4 | 4 | 4 | 4 | 1 | 0 | 1 | 1 | 2 | 2 | 1 | 1 | 1 |
|  | 34 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REG | 3 | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |
| LOG | 3 |  |  |  | X | X | X |  |  |  |  |  |  |  |  |  |
| REGA | 3 | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |
| REGB | 3 | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |
| REGC | 3 | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |
| LOGA | 3 |  |  |  | X | X | X |  |  |  |  |  |  |  |  |  |
| LOGB | 3 |  |  |  | X | X | X |  |  |  |  |  |  |  |  |  |
| LOGC | 3 |  |  |  | X | X | X |  |  |  |  |  |  |  |  |  |
| INFO | 1 |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |
| UP | 1 |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |
| UPB | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UT | 1 |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |
| UTB | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ST | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| GCV | 2 |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |
| GCVB | 2 |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |
| ECV | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| DCV | 3 |  |  |  |  |  |  |  |  |  |  |  |  | X | X | X |

[Note: upon review of the functional requirements and their feasibility following the submission of the SRS, the team agreed to remove the function provided by DB8, as noted by the lack of dependency for DB8.]