**Detailed Design**

**CV Creation**

**COP4331C-0002, Fall 2017**

Team Name and Number: Team 15

Team Members:

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

**Contents of this Document**

Design Issues

Detailed Design Information

Trace of Requirements to Design

**Detailed Design Issues**

**Reliability**

As previously stated, it was determined that there were almost no perceivable issues in terms of reliability. Since the importance of full functionality was emphasized, we planned development around incremental changes to the final product that allow for full vetting of each service and function as we add them to the website, rather than assembling all features at once and attempting to find the inconsistencies by testing all of the components at once.

**Reusability**

Reusability of the product is a key function. Since users will ideally be generating multiple CV’s for several applications, requiring the users to input an entire CV’s worth of information every time they wanted to generate a new one would be extremely inefficient for them. The complications arose in storing and replicating this information; our original plan for the project did not involve storage of this sort of information, as it would require additional server space to hold (as this sort of information is not conducive to using a database). Once we revised the functionality of the project, we prioritized storing the users’ information to accelerate the process of generating CV’s to increase the quality of product while increasing user retention.

**Maintainability**

Planning for maintenance and upkeep of our project drove the strategy and coding standards that our team agreed on; the ability of programmers to understand the components in a quick and effective manner was of the upmost importance to us. As such, our programming focuses on communication of functionality and modules through thorough comments, common-sense functions and variables naming methodologies, and modular, independent methods that separate non-overlapping functionalities.

**Testability**

The blueprints of our prototypes were largely influenced on the testability of functionality in a linear fashion, almost in an order that mirrors the end user experience. There are few distinct components to our project, which allows us to create simpler, incremental prototypes that test a small amount of new features at a time. Due to this, we can test and perfect one complete suite of features while another part of the team is preparing the next section for implementation and testing.

**Security**

Security was in our minds before the planning process even began; knowing that we would be handling user information, the team immediately knew not only that we had to protect this sensitive data, but how to protect it. We are including input sanitization specifically to prevent malicious users from gaining access to other users’ information by attempting to inject code into our input fields. In addition, we’re encrypting and hashing user information so that if unauthorized users *do* get access to our databases, the information will be useless because they won’t have the decryption key to obtain sensitive information.

**Safety**

Safety, as previously stated, was an extension of security. We knew what information would be necessary for the service provided to users, so we ensured that no other information would reach anything we host if the wrong people gain access to said data. A large portion of security is only requiring what’s necessary, and keeping other information out of the system entirely.

**Prototypes**

**Prototype Alpha**

The first prototype is simple: it is a login/registration prompt to test connecting to our user database. We can attempt to register a user with the database, where we enter information that is either found to already exist (at which point the registration is rejected), or that the information is available (at which point the information is added to the database). Once successful, we can log in with the new information, and be taken to a “Welcome” page that just gives users the option to sign out.

**Prototype Beta**

The following prototype is a fleshing-out of functionality. It contains the functionality of the first prototype, but once logged in, the user will be brought to a page containing text boxes and a photo submission button for the user to enter their information for the CV, which users will be able to save. At this time, this prototype is still in development.

**Trace of Requirements to Design**

<In this section, provide a trace of each requirement in your SRS to the design: for each requirement, indicate which module(s) will fulfill that requirement?>

**Functional Requirements**

* Accept template documents
  + Template.html
* Store user information, including email addresses, work history, contact information, non-work qualifications and education
  + N/A
* Encrypt user information
  + N/A

**Interface Requirements**

* User enters login information
  + Login.html
* User attempts to register an account
  + AccountCreation.html
* User inputs name, work experience, volunteering experience, contact information and education as raw text into pre-generated text boxes
  + User.html, CV.html
* User inputs photo in PNG or JPG format into the image upload section
  + User.html, CV.html
* System outputs PDF file on request
  + CV.html

**Physical Environment Requirements**

* 256 MB RAM
  + N/A
* 400 MHz CPU
  + N/A

**User and Human Factors Requirements**

* The system should be usable for any user who has read the documentation of the service.
  + User.html
* System will prevent misuse of the document’s reformatting by setting a maximum width to each text box, preventing information from not appearing on the page.
  + User.html, CV.html

**Documentation Requirements**

* Online documentation, should be comprehensible by any user with more than one month experience with using computers
  + User.html

**Data Requirements**

* User information, in raw text
  + User.html, CV.html
* Template documents, in PDF format
  + Template.html
* User photos, in JPG/PNG format
  + User.html, CV.html

**Resource Requirements**

* Personnel experienced in database management
  + N/A

**Security Requirements**

* All stored user information must be encrypted
  + N/A

**Quality Assurance Requirements**

* System response in <2 s
  + Login.html, AccountCreation.html, User.html, Template.html, CV.html
* Website available 16 hours/day on average
  + N/A