**Updated README file (i.e. guidelines for system’s configuration and installation, list of files included, changelog, detected bugs, test cases, traceability matrix)**

**• Documents (e.g. user stories, personas, motivational model) to be exported as PDF and added to code repository**

**(docs/). Test cases to be exported and added to code repository (tests/)**

**• Description of key algorithms (e.g. calculating the service price or any algorithms that are unusual) + mechanism of**

**importing data from 3rd party systems.**

Authentication - (Passport.js, Bcrypt.js)

We use bcrypt.js to hash user passwords before storing them in the database. It takes care of hashing the strings, comparing plain strings with hashes, and appending salt.

When a user attempts to login, their request will either be sent to LocalStrategy or GoogleStrategy depending on their selected login method. If bcryprt.compare() authorizes the request, the browser will be designated a session cookie, and the user's object will be returned and stored in the request object.

**• Description of key classes and the application’s layers (this simply makes it easier for the new team to find the place that requires changes).**

1. User Interface:

* The user sends a HTTP request to the express application

1. Domain:

* The express app hands the request to specific router
* The router interprets the request and makes a call to the relevant controller function
* The controller then executes the function, and potentially makes a call to the database through a mongoose model

1. Database

* The database returns a model object to the controller which passes the object into a view template for rending

1. Domain/User Interface

* This view template is then returned to the client and loaded into the browser

**Future Adaptations**

1. Controllers
   1. User Controller: describes the application logic concerning user management.
      1. Ie. Signup/Login and Editing user model details
   2. Upload Controller: describes the application logic concerning the uploading of user documents.
      1. Ie. Image uploading, PDF uploading
   3. Blog Controller: describes the application logic concerning blog management.
      1. Ie. Posting, Editing, Deleting
   4. Comment Controller: describes the application logic concerning comment management.
      1. Ie. Posting, Editing, Deleting, Liking
   5. Experience Controller: describes the application logic concerning user experience objects.
      1. Ie. Adding, Editing, Deleting
   6. Education Controller: describes the application logic concerning user education objects.
      1. Ie. Adding, Editing, Deleting
   7. Skill Controller: describes the application logic concerning user skill objects.
      1. Ie. Adding, Editing, Deleting
2. Models
3. User
   1. Fundamental model that stores all relevant information of a particular user
   2. All other models are referenced as attributes
4. Blog
   1. Model that stores information relevant to a single blog post created a particular user
   2. References the Comment model as an attribute
5. Comment
   1. Model that stores information relevant to a single comment made on a blog
   2. References the User and Blog model as attributes
6. Document
   1. Model that stores information relevant to a single document uploaded by a user
   2. Document types include pdfs, and jpg/png
   3. Ie. resume, profile picture, background picture, etc.
   4. References the User model as an attribute
7. Education
   1. Model that stores information relevant to a single education artefact belonging to a user
   2. References the User model as an attribute
8. Experience
   1. Model that stores information relevant to a single experience artefact belonging to a user
   2. References the User model as an attribute
9. Links
   1. Model that stores information relevant to a single link artefact belonging to a user
   2. Ie. resume, facebook, twitter, linkedin, etc
   3. References the User model as an attribute
10. Skill
    1. Model that stores information relevant to a single skill artefact belonging to a user
    2. References the User model as an attribute

**• Description of the database structure (additionally, in case of a relational database, a description of their correlations should also be included; this can be presented on ERP - Entity Relationship Diagram).**

**• Deployment guidelines (i.e. how to deploy the source code, database and run the project on a new server, any necessary administrator/test customer login credentials, access to code repositories, databases and servers as well**

**• Any licensing agreements that need to be considered**