ASHESI UNIVERSITY

SEMPLE (Self Employment Makes People Lives Easier)

A platform to promote self-employment of people while providing avenues for people to get full-time employment in Ghana

Applied Project

B. Sc. Computer Science

Daniel Olukoya Ayomide

2020

ASHESI UNIVERSITY

SEMPLE (Self Employment Makes People Lives Easier)

A platform to promote self-employment of people while providing avenues for people to get full-time employment in Ghana

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi University in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science.

Daniel Olukoya Ayomide

May 2020

DECLARATION

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……

Candidate's Name:

……

Date:

……

I hereby declare that preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied project laid down by Ashesi University.

Supervisor's Signature:

……

Supervisor's Name:

……

Date:

……

Acknowledgment

I would like to first give thanks to God for the wisdom, knowledge, understanding, and strength He granted me to undertake this project. Next, I would like to thank my supervisor, Dr David Ebo Adjepon-Yamoah. I appreciate Dr Adjepon-Yamoah for his patience, time, feedback, and tutoring, long before this project even started. Without him, this project would not have been what it is. I would also like to appreciate Lenry Neequaye for her input and reviews at various points of this project. Special appreciation to my parents, the BRIDGE team and my friends who pushed me throughout this project. I hope and pray that God richly blesses you all, I am more than grateful for the support.

Abstract

The evolution of technology over the years has opened many opportunities in many fields of life. The sectors of our economy, such as health, agriculture, entertainment, have all been improved. This evolution gives a new opportunity to people in marketing themselves. Africa could be said to be behind with some of these technologies, but it is making progress. A sector Africa requires some more technological advancement is the employment and job sector. From the hiring process to the management, to the working styles, to name a few. Ghana, though rich in skilled human resource, still seem to have a relatively high unemployment rate. Reasons including, lack of knowledge on creating a resume, difficulty in the job search process offline and online (for cases where openings are not centralized). This research study seeks to propose the encouragement of self-employment through a web platform, whose features are mainly:

- To aid people to get contract jobs
- To help with outsourcing jobs
- To allow users to leverage their skills online
- To enable the creation of portfolios and resumes with ease
- To provide a job board for users looking to secure full-time positions
- To allow Employers to have a more streamline process when employing

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Background and Problem

The unemployment rate in Ghana by the end of 2019 was 6.78% [30]. Referring to the percentage of people available to work in the country but did not have jobs and or are searching for jobs. This unemployment rate has been on a rise since it was last 4.6% in 2016 [30].

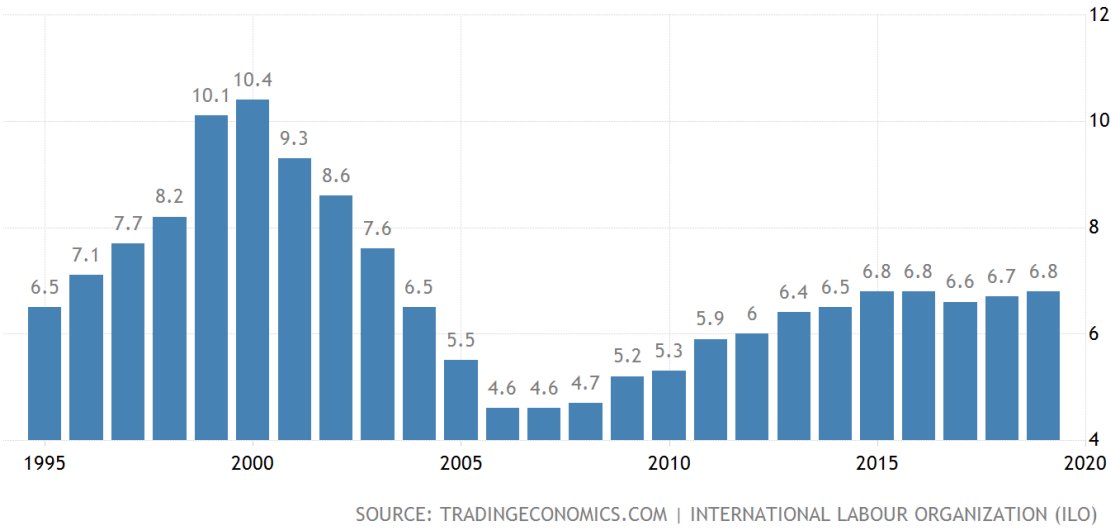

Figure 1. 1 Unemployment growth rate in Ghana

These might seem like low numbers, but the World Bank states that Ghana's labour force is approximately 12,916,874 people as of 2019 [31]. This statistic means that in 2019 there were approximately 1,899,540 unemployed people who could work. These are individuals who have useful skills but are just not getting jobs. This study analyzes how technology in other countries has been able to help their people and economies with employment, mainly with regards to self-employment. Also, this study explores how the implementation of such a platform in Ghana would help with unemployment.

1.2 Motivation

Today the idea of the fissured workplace [17], where companies and individuals get skilled personnel, they need for tasks on a contract basis, is continually growing. From hotels that hire temporary staff to clean, to delivery men and drivers who work indirectly for companies as subcontractors. Through this strategy, companies have been able to reduce costs in developing products while maintaining the quality of their product and brand names. For this reason, there is an opening to provide the unemployed in society with the option of self-employment.

1.2.1 Why Self-Employment?

As defined by the Merriam-Webster dictionary, to be self-employed refers to "earning income directly from one's own business, trade, or profession rather than as a specified salary or wages from an Employer" [28]. "The self-employed are substantially more satisfied with their work than employed persons. We document this relationship for 23 countries and show that the higher job satisfaction can directly be attributed to the greater autonomy that self-employed persons enjoy" [3]. Being self-employed means that a person would not have to work a 9 - 5 jobs because he or she works when they have accepted contracts. Self-employed individuals, whom we will call freelancers, also have the freedom of working location[48]. A freelancer does not have to move to a company workspace day today to work, s/he is free to work wherever and whenever they feel[48]. There is also much freedom that comes with being a freelancer, freedom to choose days to work, freedom in selecting contracts to work on, freedom in work attire, freedom to explore different fields of work at any point in time[48].

1.2.2 Freelancing in America

In 2019 Upwork and the Freelancers Union conducted a study on freelancing in the United States (U.S). This study included an online survey of 6,001 U.S adults who had done paid work in the previous 12 months, freelancing and non-freelancing [16]. The study found that 35% (57 million) of U.S. workforce freelanced in 2019 [16], with an increase in full-time freelancers from 17% in 2014 to 28% [16]. Freelancers contributed nearly $1 trillion to the economy with was approximately 5% of the U.S GDP (Gross Domestic Product) [16].

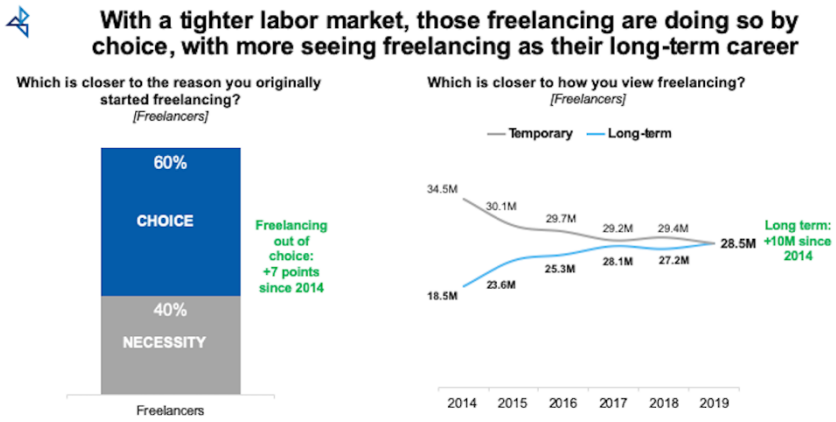

Figure 1. 2 Upwork freelancer analysis 1

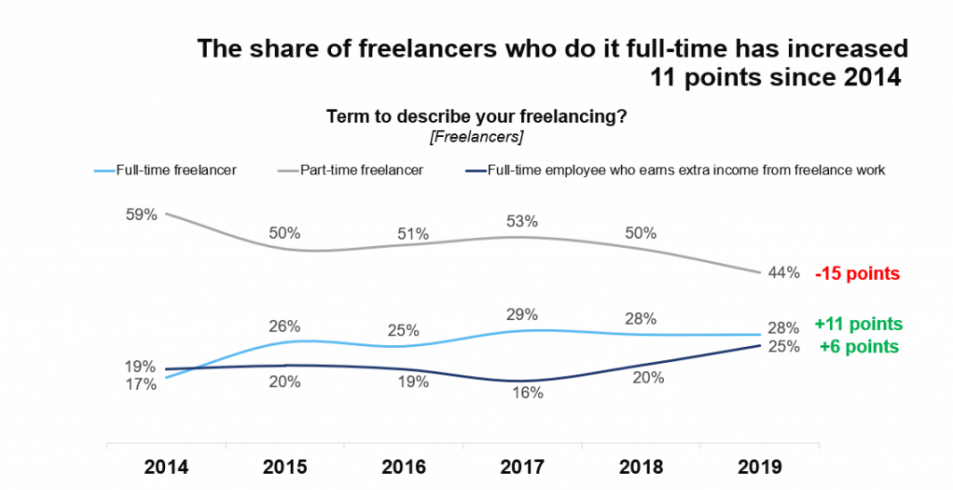

Figure 1. 3 Upwork freelancer analysis 2

Over past years more and more people adopted freelancing as a long-term career, rather than a temporary one. There are more and more people in the U.S adopting full time freelancing and freelancing alongside their full-time jobs [16]. There is also a decline in the number of people who work freelance solely part-time. There are various benefits that the people have reaped from freelancing, with 71% of freelancing respondents saying freelancing impacted their lifestyle, by enabling them to work from anywhere [16]. 46% of freelancing respondents also said freelancing allowed them to the flexibility to work that they could not get from traditional Employers because of personal circumstances [16]. Finally, from this study, 77% of freelancers said technology made finding freelance work more accessible [16], and 59% of non-freelancers said they are likely to do freelance work in the future [16].

1.2.3 Freelancing vs Full-Time Employment

Freelancers though surrounded with all these benefits, still have many issues of their own. Firstly, and mainly, for a full-time freelancer to be entirely comfortable in the position, s/he requires a constant flow of gigs, without these, a freelancer would be practically unemployed [49]. Without a steady stream of jobs, a full-time freelancer would suffer a decline in income which would come with other problems. This problem is associated with the difficulty of finding gigs. Another problem faced by freelancers is with regards to payment, based on how a freelancer goes about negotiations for a gig with a client, s/he might be cheated out of payment [49]. Again, though a freelancer has the choice of which gigs to work on, the compensations are not constant; hence it might be challenging to plan with this uncertainty [49].

Being a wage worker is the ideal form of employment for most people since there are more openings, and it is a more secure form. As a wage worker, one enjoys the security of knowing a specific compensation is coming at the end of a period. Certain benefits that come

with some jobs, benefits like health insurance, transportation bonus, among others. There is also much room for collaboration and help-seeking since most wage workers would be in teams in the company.

Just like freelancers, wage workers also have their fair share of problems with the job type. Most full-time jobs require workers to work a minimum of 8 hours a day in the office, limiting the time they have for any other personal activities. To become a full-time wage worker, one would have to go through the strenuous job search and job application process. This period could take quite some time based on job availability and the individual's skill level.

Platforms like Freelancer [39], Upwork [13] and Fiverr [29] all exist to aid freelancers and outsourcers with their whole interaction. They provide gigs put up but outsourcers to the Freelancer so that interested freelancers could bid for the jobs and then commence upon selection. They also provide chat systems to allow communication between freelancers and outsourcers.

Platforms like LinkedIn [8] and Glassdoor [22], all exist to aid individuals with their job search. They provide job boards of posted available jobs, along with links to application forms or embedded application forms, required by companies to start applications. These platforms remove the stress of moving around, which take time and money, to search for jobs.

1.3 Objective

The objective of this project is to create a platform that would help in the job sector. The goal of this platform would be mainly to promote freelancing as an alternative to the traditional form of full-time employment. In doing this, the hope is that the result would be a

substantial reduction in the unemployment rate in the country, with more hopefully adopting freelancing as a full-time role. The platform would also provide an avenue to accessing full-time jobs in the country, as people would still be interested in that option, and it could be a more straightforward process. Finally, though there are existing platforms with these features, the primary target in this project is to centralize the activities related to getting a full-time job or contract, mainly in Ghana. That is, to ensure that the provided features integrate into a single platform specifically for the Ghanaian society.

1.4 Proposed Solution

SEMPLE which stands for Self-Employment Makes People's Lives Easier. SEMPLE is a platform to centralize activities about employment, be it full time/wage employment or self-employment/freelancing. The platform is developed for three significant roles/users. The Freelancer, the Hirer and the Employer. The hirers would be able to post and manage gigs (contract jobs) on the platform that they need freelancers to complete. Hirers would also be able to pick from packages of freelancers. Employers would be able to post jobs and manage job activities on the platform. The freelancers would be able to request to work on gigs posted by hirers and apply for jobs. They would also be able to set up packages, of their existing skills so hirers could place orders for those skills.

1.5 Related Works

This chapter explores and discusses various implementations of freelance platforms and job boards used by freelancers, hirers, wage workers/outsourcers and job searchers. How these technologies help their users achieve their goals would also be explored. Proper understanding of how these platforms effectively guide users in completing their goals, is necessary to reimplement or improve upon such guides.

1.5.1 LinkedIn

LinkedIn [7] simply put a social media network that promotes career building/development and professional networking. Some features of LinkedIn include LinkedIn jobs, where hirers post job vacancies, and anyone signed onto the platform could apply. Along with the provision of the job, LinkedIn also allows its users to save jobs they might like to apply for later, so as not to go through the whole searching process again. LinkedIn ensures that it notifies individual users on updates in previous searches. LinkedIn set it ist's user profile pages such that it would represent a resume and used for applications.

1.5.2 Glassdoor

Glassdoor [40] is a recruitment software. Its features are similar to but not limited to that of LinkedIn Jobs. With Glassdoor, job recommendations are made based on the specified skills of the user. Also, when the need job openings that fall in line with a user's skills, a notification is sent to the user, so they could apply. Rating provided for companies inform applications decision. Glassdoor also provides the option to compare salary options of companies for make a better-informed decision.

1.5.3 Freelancer

Freelancer [39] is a freelance platform that connects people who need to get some task completed to people who are willing to get such task completed at a fee. The process for working on Freelancer is that when a job is posted freelancers bid for the project providing invoices and proof of work/portfolios, then the hirers select a candidate from the pool. Also, a is available chat area for communication between the two parties.

1.5.4 Fiverr

Fiverr [29], on the other hand, works such that Freelancer set up their profiles along with packages for the services they wish to provide. In doing this, hirers have to scan through the available freelancers, and select whom they believe can best execute the job they need done.

From the analysis of the related works, several concepts like the chat room for interactions, notifications and others included/modified to fit the SEMPLE system to meet user's needs better.

Chapter 2: Requirement Analysis

This chapter seeks to describe the various functionalities found on the SEMPLE platform. Alongside this, the different types of users and their interactions are explored.

2.1 Project Scope

SEMPLE seeks to connect three main factors for the development of the platform, Self-Employment, Full-Time Employment and CV generation. SEMPLE is a web application that seeks to promote self-employment, whiles providing an avenue to get a full- time employment position. Through the platform, people would be able to generate CV (PDF and online) to help their application processes. The following would be employed to build the system:

- Web Application. The platform would be a web application to start with so that the users of the platform would be able to access the platform anytime on any device. Later on, the platform would be made into a native mobile application, to allow ease of access on users preferred devices.
- REST(Representational State Transfer) API(Application Programming Interface)[19] are used on the web application. The REST API[19] would serve all the data stored in the database to the application when called. Also, for future scaling into a mobile application using a REST API[19] would allow for this with ease—data from the database of the application and for interactions with external API.
- Payment Integration. Since the platform is built in the local Ghanaian setting, avenue to make payment with local platforms (MoMo). International's payment avenues like Visa and MasterCard would also be made available to allow international transactions.

2.2 User Type Description

2.2.1 Hirer

A hirer on SEMPLE refers to a user whose main goal on the platform is to get a freelancer to help them complete a task, which is achieved in two ways. The Hirer could either post a gig (contract job) or place an order for freelancers' package. When a gig is posted, the Hirer is able to select who s/he want to undertake the task and proceed to interactions on selection. When an order is placed for a package, the Hirer waits for the Freelancer to approve or decline an order before moving to the next stage.

2.2.2 Freelancer

A freelance user is on the platform to get gigs, jobs or post a package based on skills they have. Which regards to getting gigs on the platform, the freelance would be able to scan through gigs posted by the Hirer. Upon identifying a gig, they like, they could send a proposal for why they should be selected to complete the task. The Freelancer would also be able to apply for jobs posted by the Employers on the platform.

2.2.3 Employer

This user role is from entities/bodies that are looking to employ freelancers on the platform. They would be able to post jobs that would reflect in the Freelancer's job boards for applications. They would be able to analyze these applications on the platform when they come in and give adequate responses based on saved replies

2.3 User Case Scenarios

In order to fish out the functionalities of the application, various use cases for different types of users expected on the platform were created. This allowed for proper analysis of features needed by different individuals in different situations

- John is a freelance designer who gets jobs only when he is referred by a friend/previous hirer who knows his works. He is looking for a way to access more jobs without relying on referrals solely or leaving his home
- Ama is a recent university graduate looking for a way to get a job without having to move from location to location to place her CV. She would also like to know which jobs she fits best for and apply for those first.
- Agatha is an Instagram based makeup artist and hairstylist; she is looking for a way to get more clients outside Instagram. Her reason being that here post easily gets lost in the wide variety of content people post on Instagram. Her pricing is also not being clearly displayed to possible customers.
- Overloaded with work at his design firm, Michael is beginning to fear he would miss deadlines and that the quality of his work would reduce. He, therefore, needs a way to outsource some of the jobs he has and still ensure quality and timeliness
- Bridge is a digital agency form by a group of Ashesi students. Upon finishing school, they realize that for the type of projects they are managing and receiving, a bigger team would be required. They need a way to get new employees based on their abilities and not just their educational background. They would also want to know which applicants a great fit is based on selected criteria.

2.4 Functional Requirements

- **REQ-001**

  Users should be able to make payments via local/international options. Unlike other international freelance platforms, SEMPLE is developed for Ghana, hence the integrations of payment services that would make it easier for bother hirers to make, receive and withdraw payments.

- **REQ-002**

  Users should be able to register and login into their accounts. The various user types on the platform should be able to log into their accounts with ease, through a single portal. Be it an Employer, Hirer or Freelancer; there should be a single portal to process and redirect the users to their various destinations.

- **REQ-003**

  Users should be able to post, edit and view jobs on the job board. Creation of job post on the Employer side is made short and straightforward, still ensuring that the details provided allow for best-fit analysis. The Employer should be able to manage the job post and the applications to that job.

- **REQ-004**

  Users should be able to apply for job positions on the job board. On the freelance side, once the profile section is appropriately set up, there would be no need to fill in specific values, as the system would do that with the updated values on the profile.

Extra information provided to apply, would be from what the Employer requires in their form.

- **REQ-005**

  Users should be able to post, edit and view gigs on the freelance section. When a hirer posts a gig s/he should be able to manage the activities related to that bid. The description of the bid and the selection from the pool of bid applicants.

- **REQ-006**

  Users should be able to bid for gigs on the freelance section. In the Freelancer's section, when a gig is posted on the gig board, the freelancers should be able to make bids for the role posted.

- **REQ-007**

  Users should be able to hire freelancers for specific role types (full-time, part-time, among others). Specification on the current type of job should be provided on each job post to ensure that freelancers know the types of jobs they are applying.

- **REQ-008**

  Users should be able to select the type of job they would like recommendations on the job board. The platform should have a recommender system to recommend jobs and freelancers for positions. The platform should be able to recommend jobs to freelancers that match their skill set best. On the Employer side, there should be a provision of recommendation for freelancers that best fit the position from the application pool.

- **REQ-009**

  Ease in Identifying possible Employers/Employees. With the structure and analysis provided by the SEMPLE for both Employers and freelance applicants, knowing which Employer or applicant is seamless. For freelancers on the job board, there would be a tag by the jobs for which they fit best for, and there would also be a filtering option to allow only those jobs to be displayed. On the Employer side, after setting up a job post, based on the selected criteria, the Employer would also be able to filter to see which applicants best fit their search.

- **REQ-010**

  Provision of both an online and offline portfolio and CV, respectively. As stated, based on the data provided by freelancers in their profile section, their resume would be generated using the provided template. At any point in time, the Freelancer could download this updated CV, for anything the wish to use it for outside the platform. The link to their public profile could also be shared to which would serve as their online portfolio with all their skills, experience, gigs and reviews for better assessment.

2.5 Non-Functional Requirements

- **SPD-01**

  Page load should be less than ten seconds. To better the user experience on the platform, the various pages should be made to load up or be mounted in less than ten seconds.

- **SPD-02**

  Images and files should be stored in such a way that access over the application would not be difficult. When images and others file are stored on the platform, this should be done such that storage capacity is not overburdened and

- **SCL-01**

  The platform should be built in a scalable, reusable manner. The platform should be built such that scaling is not a problem, such that maintaining and making addition and integrations in the future could be done with ease.

- **UX-01**

  User interaction to complete tasks should be simplified. When users are on the platforms, the process of completing tasks should not be too long, to slow them reaching their goal.

- **UX-02**

  Longer forms should be broken down into sections, and progress should be provided to let the user know their status. Doing this would help with the user experience on the application significantly. Many forms would be filled on the platform for various activities. Hence these should not be made overly long.

- **UX-03**

  Page cleanliness should be ensured. Page cleanliness should be ensured because with the amount of data flowing into the application. Users need a clean well-structured page to be able to identify the different data form on the various pages.

- **SAF-01**

  Since cash transactions would be occurring over the platform, payment and withdrawal transactions should be made as secure as possible. This has to be ensured so as not to have mix-ups in funds and errors when transacting.

- **SAF-02**

  SEMPLE is a multi-user platform, where different users have different sections, and the different sections have different functionalities. There are no crosses in the roles; hence there should be Access Level Control. This will ensure that users do not get unauthorized access.

- **SAF-03**

  Validations must be done on both the client and server-side to ensure the integrity of the data being passed through the platform. Validation of form, requests, among others must be implemented wherever necessary.

Chapter 3: System Architectural Design

SEMPLE, as determined from the requirements gathered and analyzed, would be a web application. In future, the platform would be scaled into a native mobile application and or a progressive web application to provide users choices for convenience. However, before jumping into the development stage of the web application, the system architecture has to be decided to know the structure of the web application as well as the models to be used.

When a web platform executes, there is two main types of code that are run, namely the server-side code that handles all the requests and parses all the data the user passes to the server. Furthermore, the client-side code runs in the browser displaying the different views and manipulating user inputs and providing validations.

3.1 Type of web application architecture

For the development SEMPLE, based on the number of users expected to use the system simultaneously, a serverless architecture was selected. With the serverless architecture, the backend or the data layer of the platform would be developed and managed separately from the front-end component of the platform.

Using this architecture would provide many benefits in the long term, as this approach would allow for scalability. This is because when the server-side is developed independently of the client-side, integration with newer systems is more convenient. Also, upgrading either side could be done independently, preventing errors on the server side from affecting the client-side and vice versa. In the future SEMPLE is envisioned to be a progressive web application and or native mobile application. Having the backend component separated from

the front-end component would all other platform integrate with the same backend, so as to not rebuild and transfer data to the server-side of the mobile application.

The serverless architecture was also selected to provide a form of security. While consuming serverless services, a layer of authentication is implemented to ensure that only authorized personnel could access data. In this process, there would be two-level or authentication and authorization. The user would have to be authenticated to use the client-side features and also be granted access to make requests to the server. Validations here would also occur on both the client-side and the server-side ensuring more security on the platform.

For the implementation of the serverless side of the system, Laravel [1] would be used to create the REST API's [19] that would be used to control the flow of data through the SEMPLE platform.

3.2 Model View Controller

For the development of the system, the Model View Controller [32] structure would be implemented. By virtue of using the PHP (Hypertext Preprocessor) Laravel [1], this structure would have to be followed as that is what was used to develop the framework. With this, the view, the data layer and the logic would all be handled/written separately. This is to ensure modularity in code, efficiency and ease when maintaining and updating.

The model defines the data structures, with regard to the database of SEMPLE. The model handles what the tables would be, what columns they would need, their relationships and various functionalities that would be needed when working with data on the platform. The view is what the user sees and what they interact with to get their overall goal completed.

When the user interacts with the view, the controller is what sets up the logic that would be required to make what the user sends make sense.

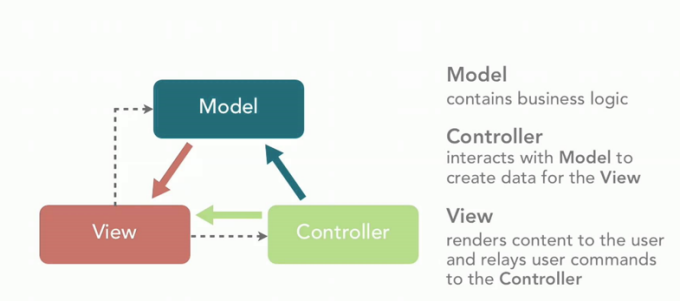

Figure 3. 1 Model View Controller

In relation, to SEMPLE the model would be implemented in Laravel [1] with an Object-Oriented approach, where the database and its tables and functionalities would be represented in forms of classes that would interact with the controller before being parsed to the user on the view. For the model seeders and factories would be used to help with the creation of dummy data that would be used when testing the platform as compared to manually populating the data. The view of the platform would be implemented as a Single Page Application (SPA) [12] with a framework known as Vue JS [27]. Vue Js [27] was selected as it natively integrates best with Laravel, as it is inbuilt into its structure. Also, with Vue JS [27], it would be a seamless connection between the Laravel [1] models and controllers. This connection would make validations and querying data much more convenient. Finally, the controller would contain all the logic of the application that would be used for notifications, emails, middleware checks, among many other functionalities in the application.

3.3 Database Design

The SEMPLE database consists of thirty (30) normalized table. The structuring of these tables allowed for the easy query of data within a table and from an external table that is related.

Table 3.1 Database Design

Table	Description
Users	This table store all the information of the various users of the platform. Their names, emails, encrypted passwords, phone numbers and roles mainly. Other field included the date the user was created, the last date an update is made to a user and the date their email was verified.
Categories	This table store name of the various categories of jobs and gigs on the platform. For example, Design, Software, Lifestyle, among others.
Skills	This table stores the names of skills related to the various categories on the platform. For example, front end development, back end development, graphic design, UI/UX design among others

Jobs	This table store the base information of every job posted by employers on the platform. The name, the category, the description, the country and state the job is located, the seniority level and employment type of the job.
Gigs	This table stores all the information on the gigs that have been posted by hirers on the platform. It stores the name of the gig, the category of the gig, its description, the price range for the gig, the duration and the status of the gig.
Job_skills	This table store the skills related to every job posted. This allows for matching freelancers who would fit best for available positions.
Gig_skills	This table stores skills associated with each gig. This would allow matching them to freelancers who would be best equipped to take on the task.
User_skills	This stores the skills associated with each freelancer on the platform.
Reviews	This table keeps all the reviews for gigs, hirers and freelancers on the platform.

Saved	This table stores information on the saved data on the SEMPLE. Data saved includes gigs, packages, jobs and freelancers.
Schools	This table store information on schools that freelancers have attended.
Work_history	This table stores the work history of freelancers on SEMPLE. It keeps track of every job a freelancer has done the platform.
Portfolio	This table keeps information on the various works a freelancer has done and uploaded to the platform.
Packages	This table keeps the information on the posted packages of a freelancer
Notification	For every notification on the platform, the information is stored here. The status of seen and unseen is set to every notification stored.
Links	Links to social media platforms and external websites are stored in this table.
Job_Form	When posting a job on the platform, an employer specifies the additional information that they would need from the applicant. The structure of the form is stored in this table

Employment_history	Previous employment information of freelancers on the platform are stored in this table
Education	The educational history of freelancers on the platform are stored on this table.

3.4 Layered Architecture

To ensure that the platform is developed in a modular and scalable manner, a layered architecture is employed. This allows for proper separation of concerns that would improve debugging of errors, integration of external API's and ensure code modularity among others. The layers are separated into four layers namely the Presentation layer, the Business logic layer, the Data Access Layer and the Data Layer.

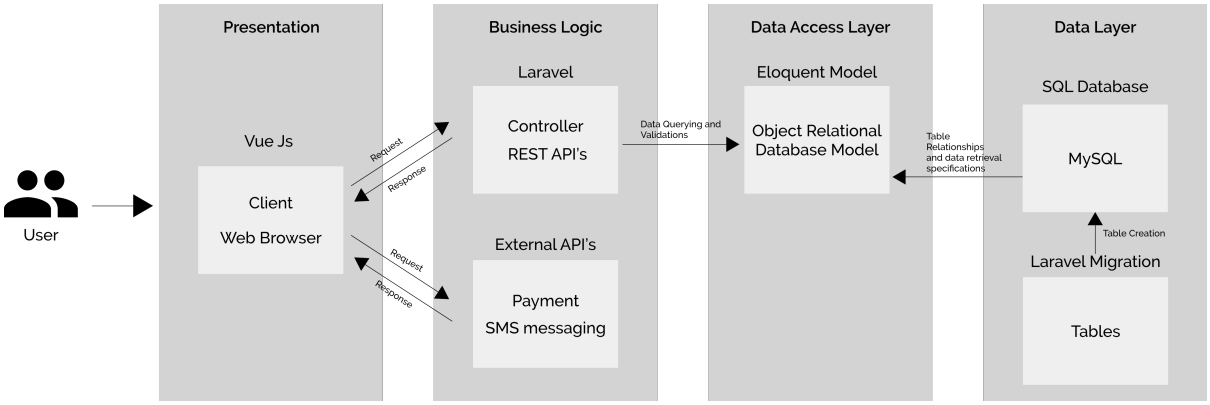

Figure 3. 2 SEMPLE layered architecture

In the data layer the selected choice of database is MySQL[45], which is a relational database. Using Laravel [1] to develop the system, the idea of migrations is employed. Migrations [46] are used to create database table directly in the Laravel [1] codebase. This allows for ease when sharing with team member, to enable them to get up to date with the

latest database table designs. Hence, in SEMPLE, migrations were used to create all thirty tables.

In the Data Access Layer, the retrieval of data from the database layer occurs. Laravel [1] uses Object Oriented models to represent tables in the database. These models allow for various database structure specifications. In SEMPLE models are used to retrieve and insert data. Other uses on the platform include specification of which columns of the tables/models are fillable by the user. This specification provides an additional layer of security to the database. Another use of models in SEMPLE is specify which columns of a table should be retrievable, that is which data in columns of a table should be displayed when a query is made. Models mostly helped with regards to relationships. To relate tables, Laravel [1] has inbuilt functionalities that make this more convenient. These are used in the models relate users to gigs, jobs, skills, bids among others without writing queries.

The Business Logic is the backbone of the platform. It links the client side to the database or external functionalities. The Business Logic for SEMPLE is made to be a REST API's [19], to which the client side makes requests. This layer is sectioned into the REST API's [19] built into the platform and those that are consumed from external sources. The payment and messaging functionalities come from third party services, hence, to use them the client make requests directly to those services. The internal REST API's [19] are developed to perform Create, Read, Update and Delete (CRUD) operations on the database. Whenever any of these functionalities need to be performed, the client would make a request to routes specified for the Laravel controllers.

The presentation layer is what the user sees when performing operations. SEMPLE is built as a web application, using Vue JS [27], and would be server in a web browser. Vue JS [27] is a Single Page Application framework that allows for better usability by providing more responsiveness and less/no reloads. With Vue JS [27] the routing logic of the

application is separated from the view components and the imports of user interface (UI) components used in the application are also separated. The user interface components of the platform are also developed as components to allow for reusability in various scenarios. Components including the gig and job cards used in the Gig and Job Board. To interact with the data layer of the platform, the front-end components make requests to various REST API's [19] made available by the developed controllers. For external services like messaging and payment, third party API's are used by making request to the available endpoints.

Chapter 4: Implementation

4.1 System Description

SEMPLE (Self Employment Makes People's Lives Easier) platform to centralize activities pertaining to employment, be it full time/wage employment or self-employment or freelancing. Because of this reason, the platform is segmented by three major roles the Freelancer, the Hirer and the Employer. These three roles determined the implementation of the various sections of the application.

4.2 Decision Making

In the development of this platform, various factors need to be considered before the implementation of the system begun. Firstly, the user experience needed to be very efficient and straightforward, such that anyone who came onto the platform could easily find their way about and complete their goal with ease. The speed when using the platform, which in some ways fell under user experience, was also considered as this would hasten the process of various goal completion on the platform. The security factor was another major factor that had to be considered as this platform would link various individuals from all sides of the country, and also handle their transactions. All the mentioned factors, among many others, influenced the choices in design, colours, user flows, payment systems, programming languages, frameworks, and libraries, among others.

4.3 Technology Stack

The decision with regards to which technology stack to use was the first significant decision. Based on this selection, further decisions on the speed of the application, speed of development, ease of integration, ease of maintenance, security factors, deployment, testing, among others would be made. The selection was between developing this platform using vanilla code [42], where the code would be written in its plain/basic form or developing the

platform using a framework [42] with made use of various libraries. All decisions made had one major underlying factor, scale.

Table 4. 1 Use of vanilla code vs frameworks

Factor	**Vanilla Code**	**Framework & Libraries**
Application Speed	The speed of an application using vanilla code would remain relatively fast, based on the development style/structure of the programmer. For example, querying the database for information could cause the speed of an application to deteriorate, based on how a programmer does this, that would inform the application speed in that regard.	The speed of an application using a framework would also be fast based on how the developer structures the code. With frameworks, though, measures have been put in place to ensure that this is the case as certain conventions are provided out of the box to be used to ensure a faster running program. With regards to querying the database, server-side frameworks provided various APIs to ease the connection to and querying of data from the database.
Development Speed	Vanilla code is slower compared to a framework,	Frameworks provide various functions, styles, among

	as the programmer is now inventing the wheel. The functions, styles, resources, among others, that would be used by the application, would have to be implemented from scratch by the developer. For example, CSS could be written from scratch when developing an application.	others, that would make the development process easier as generic content is already made available. Bootstrap, for example, is an HTML, JS, CSS libraries that allow for the creation of reusable components without writing any code.
Ease of Integration	Based on what is being integrated, this process could be quite easy or difficult. Integrating jQuery, for example, could be easy as compared to integrating some advanced text fields.	Integrations using framework could also vary based on what is being integrated. Since frameworks are already a collection of libraries, adding unto the already available ones is not mostly a daunting task

Ease of maintenance	The maintenance of vanilla code is based on the structure used to develop the project. If the project is developed such that the components could be re-used through the applications, then that would be a plus. Extra work would have to be put in to make sure this is possible in the long run.	Frameworks are designed and maintained by external bodies. Hence most of the load is taken off the development. The developer would have to maintain the parts they create and leave the rest to the developers of libraries/APIs used in the application.
Security	The security layer for an application built using vanilla code could be quite tricky as all things such as validations, and vulnerability assessments would have to be done and maintained by the developer.	The security aspects, when using frameworks and libraries, are mostly handled by developers whose APIs/libraries are being integrated. Frameworks also provide more accessible ways of doing things like validations, authentication, among others.
Reusability	The reusability of code here again is based on the	The structure is put in place by the frameworks to make

	structure the developer chooses to follow for development.	the code more reusable. But conventions are also available to allow extension of this factor by the developer
Testing	To test vanilla code, external resources, like PHP Unit, would have to be brought in to test the functionality of the application.	Some frameworks have inbuilt test resources, like PHP Unit, that would make it easier for the developer to test the application
Debugging	Debugging applications here could be a daunting task, as the developer would have to trace the errors occurring with minimal support from the browser.	Debugging applications with some frameworks is an easier task. Also, the location of code errors is easier with solutions are suggested sometimes.
Deployment	Deployment difficulty of vanilla code is based on the programming languages used, libraries used, and the hosting service is.	Deployment difficulty is based on the framework used, the host being deployed to, and the libraries that were integrated.

After analyzing all these factors, a blend of the two was chosen as the way to move forward, leaning more towards the framework. The vanilla part of the process being the CSS

(Cascading Style Sheet)/styling portion as this would allow for customization of the project to a new site, rather than using generic templates or using predefined CSS like Bootstrap. The next decision that had to make after picking a blend was which frameworks to use. The decision on the would determine the way in also determine various factors, but that would lean more based on preference and functionality available. Considering the type of user flow/experience that the user is expected to have, a Single Page Application (SPA) [12] was decided to be best. A SPA [12] compared to a Multi-Page Application (MPA) [12] has more of an advantage in this system because of the following reasons:

- The overall user experience is hugely boosted because in a SPA [12], there are no reloads required to make the application function. The page is served one, and then the JavaScript is used to parse the data that is flowing through the application at all points.
- For a SPA [12] to work effectively, it serves its data by making requests to a Representational State Transfer (REST) Application Programming Interface (API) [19]. REST APIs [19] serve as the waiter in a diner who serves its clients whatever they request from the menu, in this case, the database. Because of a SPAs [12] dependency on REST APIs [19], it is easier to move a platform from the web to a mobile application as mobile applications also require REST API's [19] to serve data.
- Debugging the request flowing through the application is also more manageable in a single page application is also easier when comparing to an MPA [12]. This is because since the application is served on one single page, all the network activity could be tracked on one page. Whereas with an MPA [12], it is more difficult to track as each request reloads and starts a new page, reloading network activity.

After making all these considerations, Vue Js [27] was picked as the SPA [12] framework to be used for this application. Most mainstream SPAs [12] available (React JS [37]) are

quite similar, with regards to speed, scalability, rendering, data bindings, integrations, among others. Vue JS [27] was selected because of its flatter learning curve when compared to the other mainstream SPA frameworks. Another reason for the selection of Vue JS [27] was the ease of integration with the selection of the server-side framework, Laravel [1]. Laravel is a PHP framework with an advanced architecture that still ensure simplicity in its syntax and ease in its usage [1]. Laravel [1] is considered one of the best PHP frameworks available in the developer community. Laravel [1] could be used to build a variety of products ranging from small to large scale products. Laravel [1] makes the following easy to ensure in an application:

- REST API [19] development and integration are easier in this case where Laravel [1] would be built with Vue JS [27]. Because the REST APIs [19] are developed on the same code base, and Laravel [1] handles the integration, communication is easier and faster.
- Deployment of the application is also made easier and less costly. This is because when compared to other frameworks where the client-side and server-side are implemented separately, there need to deploy and maintain the two separate code bases might become an issue.
- Laravel [1] supports the Model View Controller (MVC) [32] pattern of creation of software. As this is the structure defined in the architecture to be used in the development of this application
- Laravel [1] enhance security in an application, providing CSRF protection, more straightforward validations, inbuilt authentication flow, all password encryption, among others that would be very necessary for this application.
- Laravel [1] also makes integration of external services easier with its Command Line Interface (CLI) provision. In this application, this would come very handy as various

libraries could be imported from the developer community to implement payment, sentiment analysis, among many others.

4.4 A few other Libraries, Tools and Resources used

- Laravel Sentiment analysis [2]

This library, developed by Antoine Augusti, allows data to be passed to a controller to parse any data and provide sentiment on it. This library API provides functions to review if provided text is neutral, positive, negative, is a decision, and even tell scores on the levels.

- Moment Js [26]

This library provides functionalities to parse date value quickly. With this library, the dates (22/03/2020 12:33:45 :01) stored in the database could be parsed into forms like “a few seconds ago”, “2 days ago,” all in one line.

- Sweet Alert [35]

This provided functional to integrate various modals, toasts and other alert dialogs could be integrated into the application. These alerts are used, during validations, notifications, among others, quite quickly because of this library.

- Postman [33]

To test the REST API’s developed for the application, Postman, was the resources to test and make sure they functioned properly. It provided details on the status code, headers and body of a response to clients.

- Vue2Editor [14]

The Vue2Editor is a library used for creating advanced text areas/fields. With this type of text field, text formatting, images, links, lists among others could all be included with ease. This was used for the job description section when an Employer is creating a job post. With this, the Employer posting a job is able to style the job to their taste and

preference. Also, this library allows for parsing Hypertext Markup Language (HTML) and display them in the browser.

Chapter 5: Testing

The process of testing is done to fish out errors that might have occurred on the server-side or at the client-side. Testing is also done to ensure that all the requirements that were defined have been met by the applications. The Systems Sciences Institute at IBM found that "the cost to fix an error found after product release was 4 to 5 times more than one uncovered during design, and up to 100 times more than one identified in the maintenance phase" [23]. In a system like SEMPLE, where users would be mostly unknown, constant, widely scoped testing is required to ensure great user experiences and prevent malicious attacks.

In testing SEMPLE, the various test approaches were used. The purpose of each test case was to ensure that all requirement specifications were met. For each functional requirement, different tests were performed to ensure proper implementation.

5.1 Functional Requirement Tests

- **REQ-001**

  The payment system used on in SEMPLE is served through the Mazzuma [44] Application Programming Interface (API) which provides mobile money services. Two major tests were performed, with regards to payment (deposit/withdrawal). The first was to ensure that the parameters (price, payers id, among others) were all appropriately validated. This test ensured that in every transaction, only the right fields are processed. In the event that invalid parameters are passed, an error is sent to the application, and the payment is not processed.

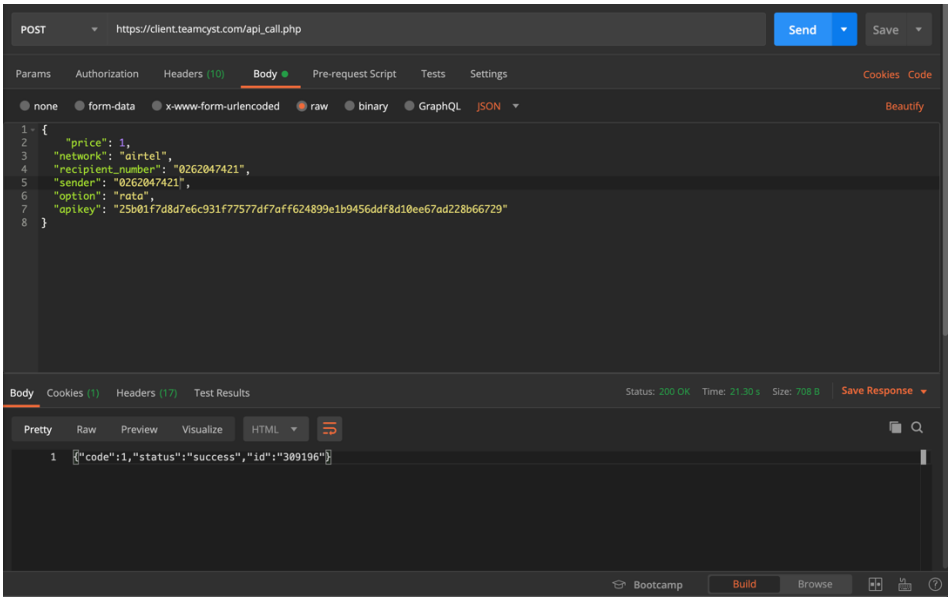

Figure 5. 1 Testing payment API with valid parameters

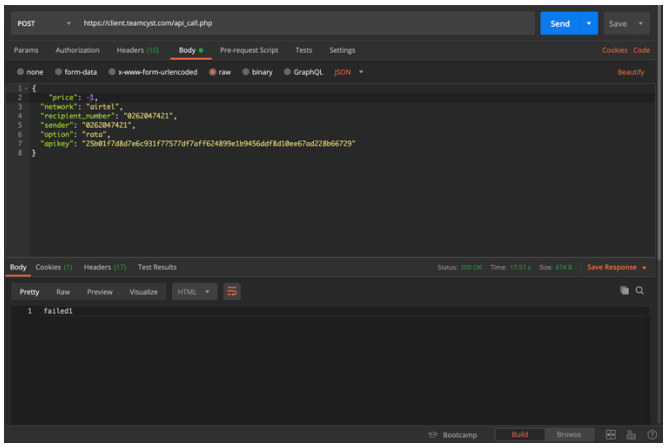

Figure 5. 2 Testing payment API with invalid parameters

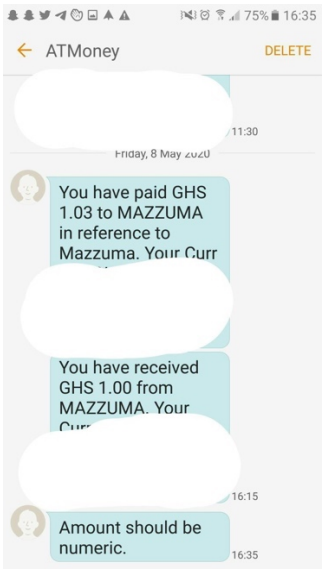

Figure 5. 3 Payment API SMS response

- **REQ-002**

  User logins were regressively tested. To access the application, one would have to create an account first and then log into the created account. Upon creation of an account, a path for using the application would be selected. For every login, the redirection was tested to ensure that the user ended up at the correct role destination.

Freelancers have to be directed to the Gig board, hirers to the hirer homepage and Employers to the Employer dashboard.

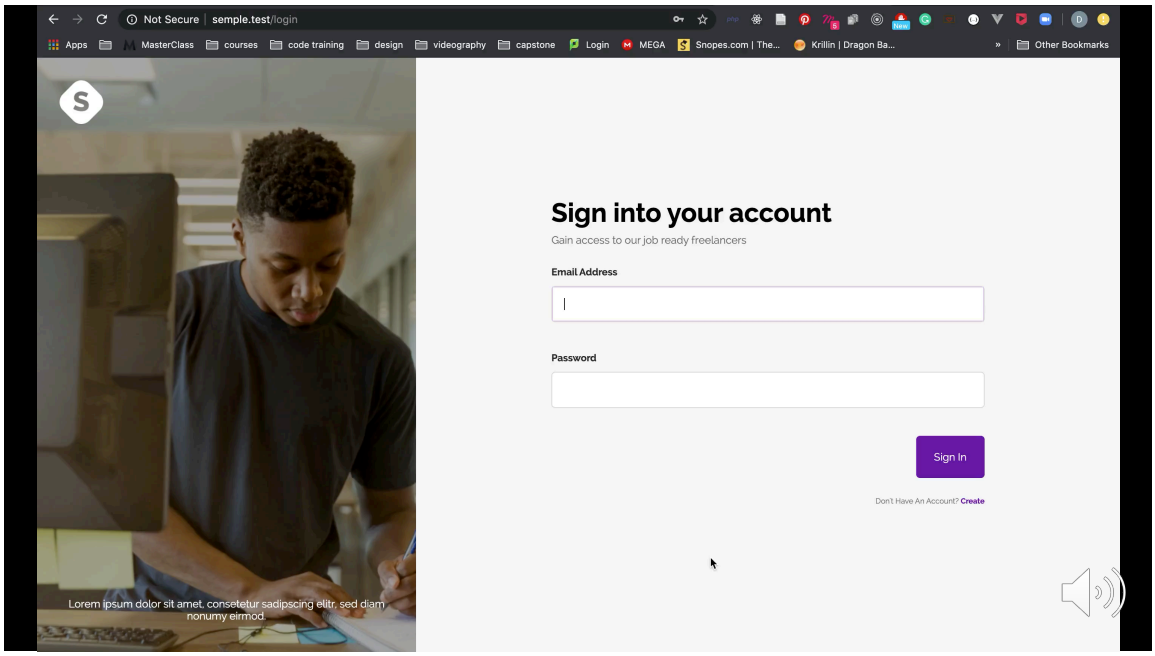

Figure 5. 4 SEMPLE login screen

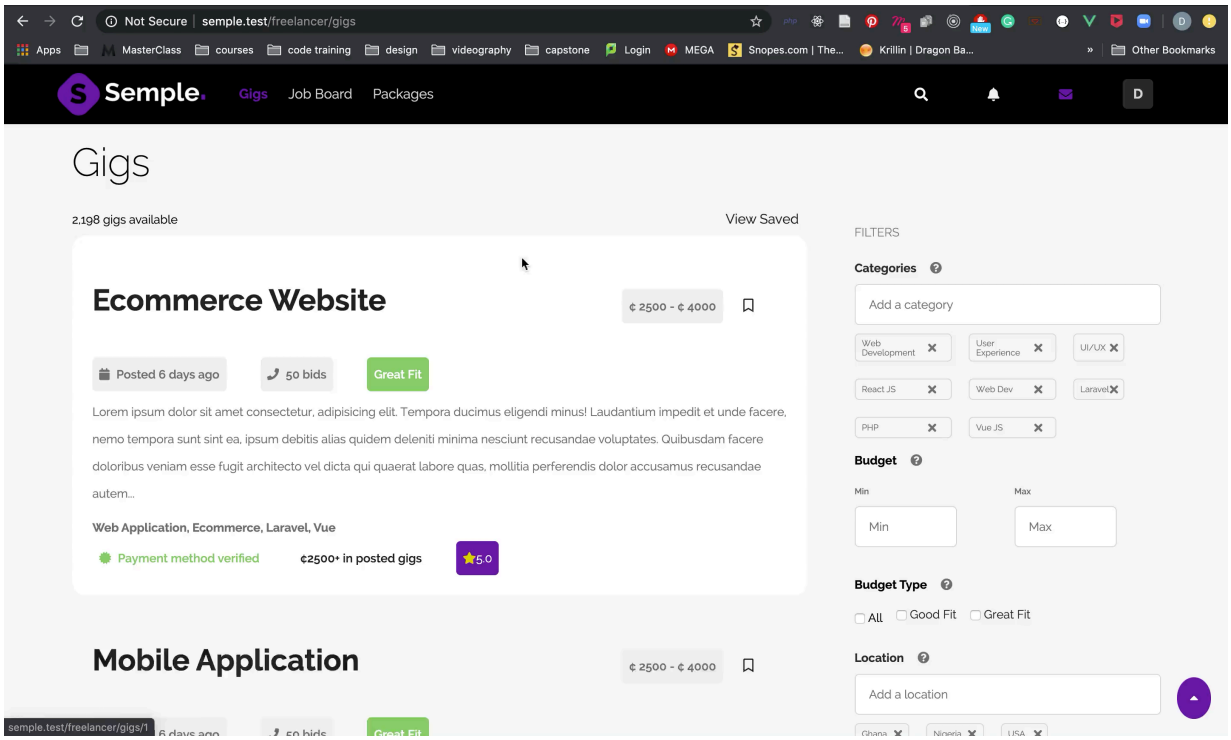

Figure 5. 5 SEMPLE freelancer gig board

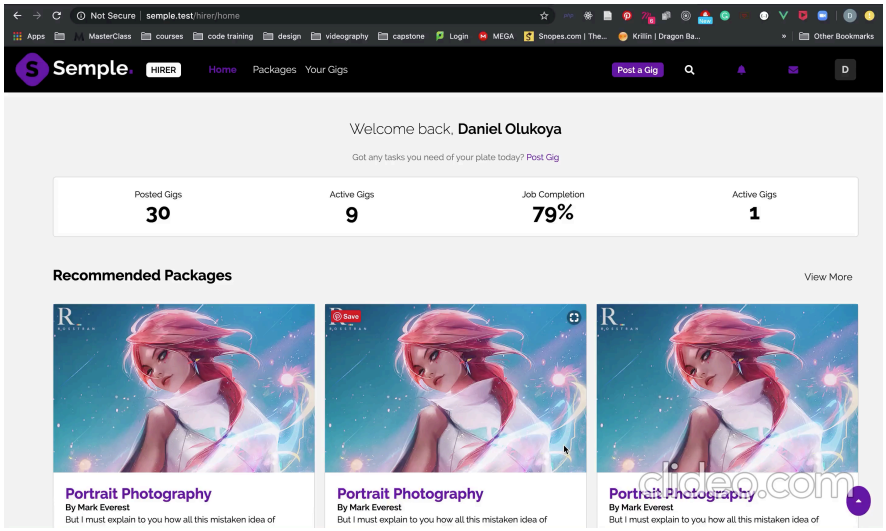

Figure 5. 6 SEMPLE Hirer home page

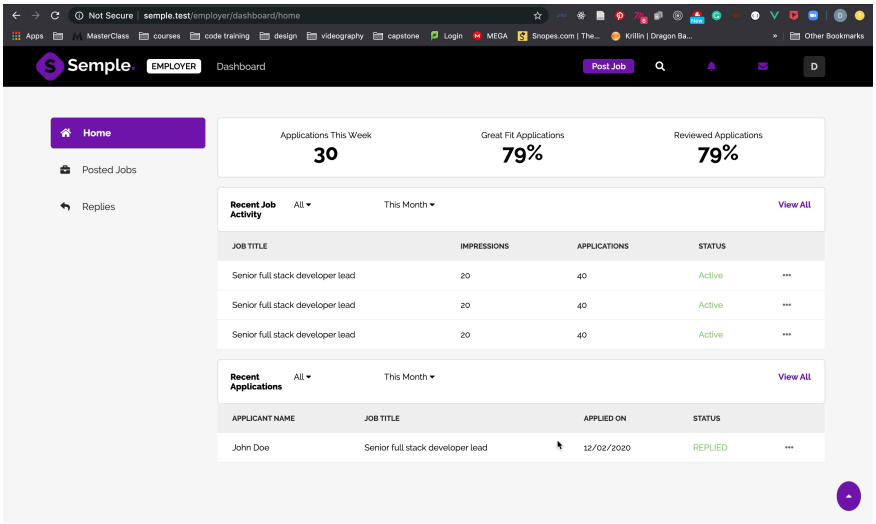

Figure 5. 7 SEMPLE Employer dashboard

- **REQ-003**

The Employers post job was also tested, ensuring that when a job is posted, it is displayed on the job board and in the Employers posted jobs. The form was validated to ensure that the data being passed through is valid.

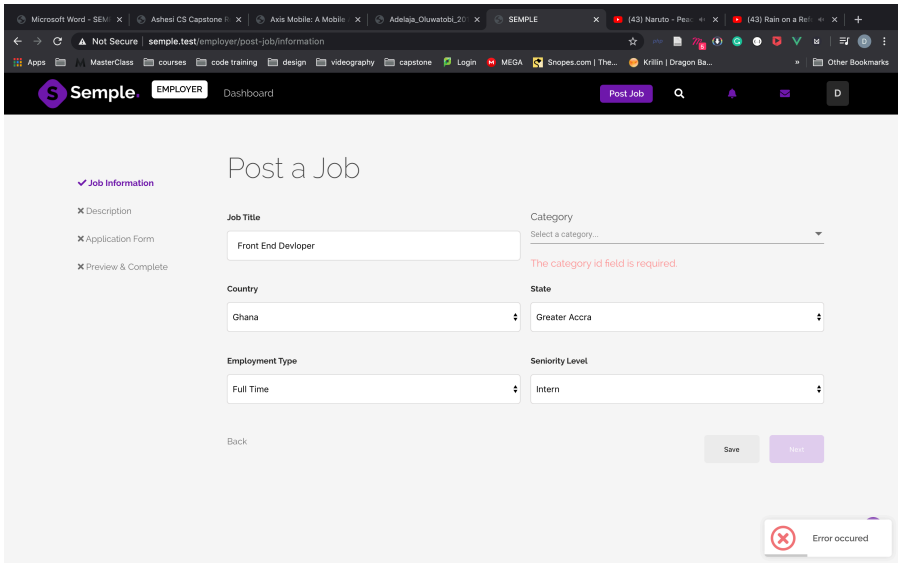

Figure 5. 8 SEMPLE Employer job post

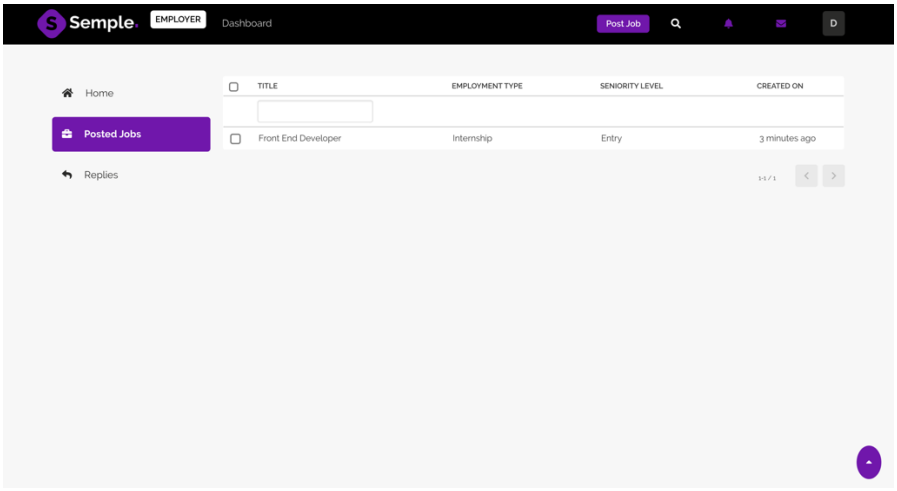

Figure 5.8 Employer posted jobs

- **REQ-004**

To test that freelancers would be able to see the posted jobs, a freelance test account was created to check if the posted jobs are displayed in the Freelancer's section on the job board. The details were also tested with the values stored in the database, ensuring freelancers see the correct details for a selected job and then apply for said job.

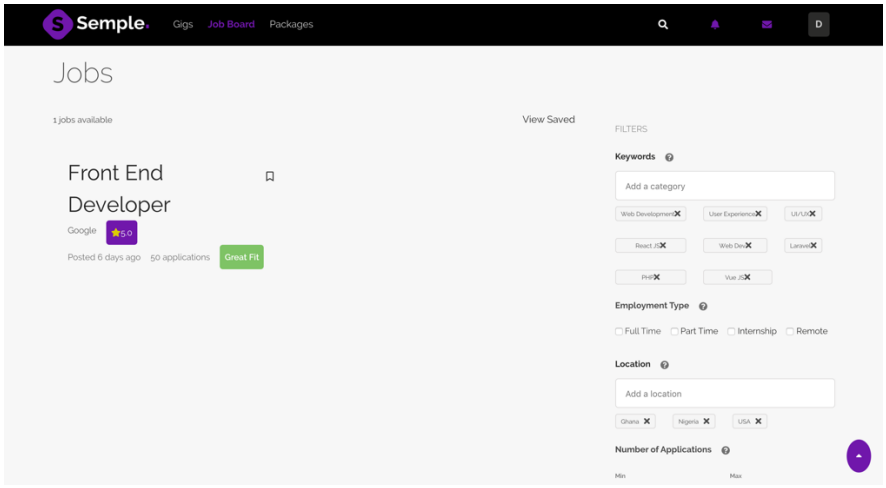

Figure 5. 9 SEMPLE Posted job on job board

- **REQ-004**

In the hirer section, a form was provided for posting gigs (contract jobs) to the gig board. When the Hirer makes a gig post, it should reflect on the Freelancer's gig board so bids could be placed by the freelancers. This was tested to ensure that the hirers' inputs were validated, to ensure that the posts appeared on the gig board with correct details.

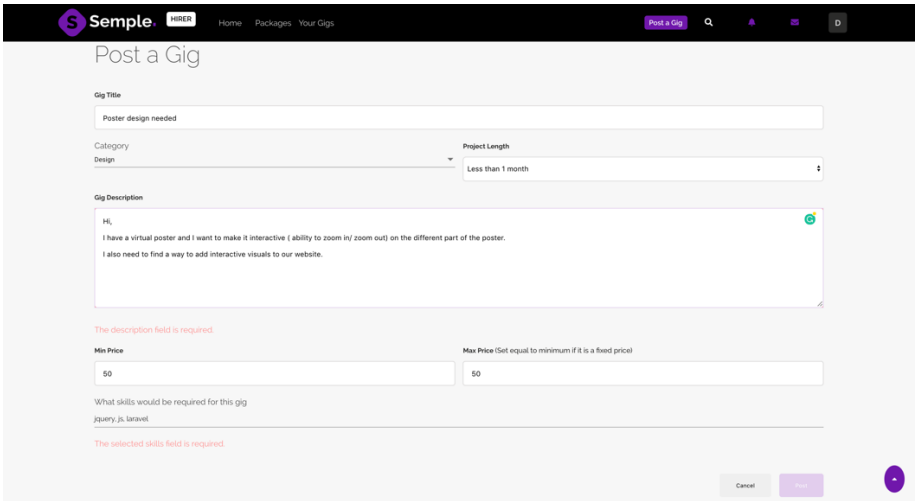

Figure 5. 10 SEMPLE Hirer gig posting form

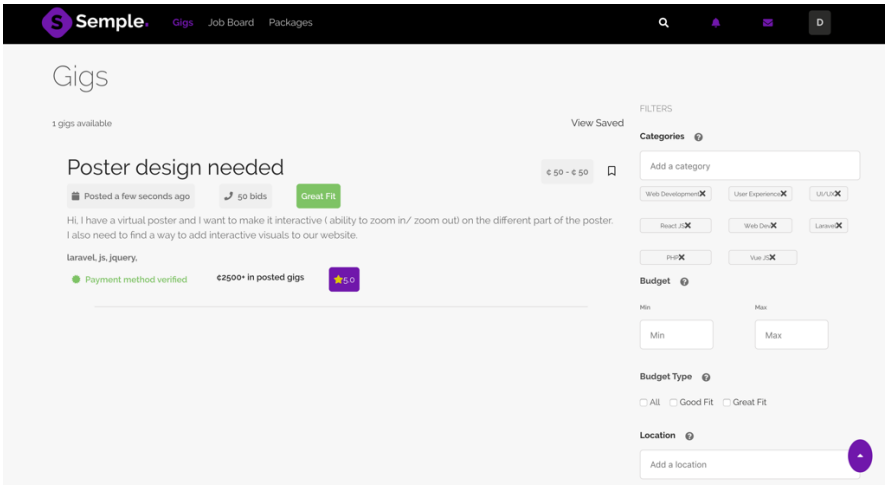

Figure 5. 11 SEMPLE posted gig on gig board

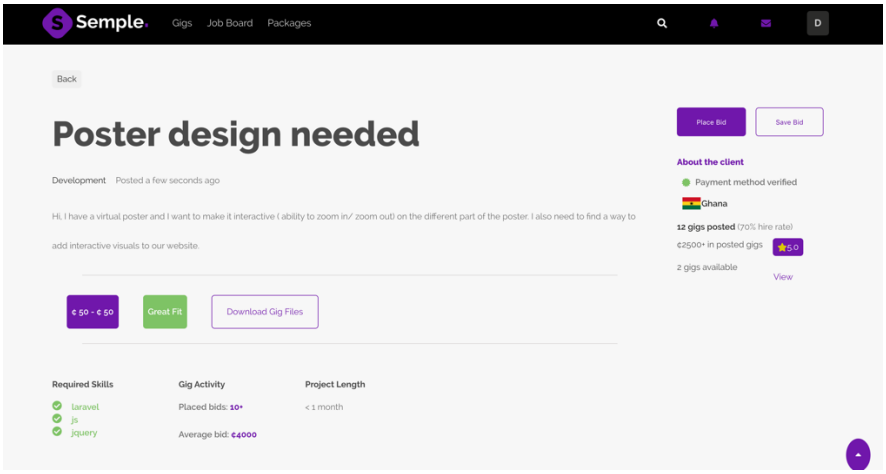

Figure 5. 12 SEMPLE posted gig details

- **REQ-005**

Freelancers are able to place bids for gigs they feel they could complete. The feature has to be tested to ensure that hirers would receive responses from freelancers placing their bids. When a freelancer places a bid, it is also displayed in the hirers section for review and selection.

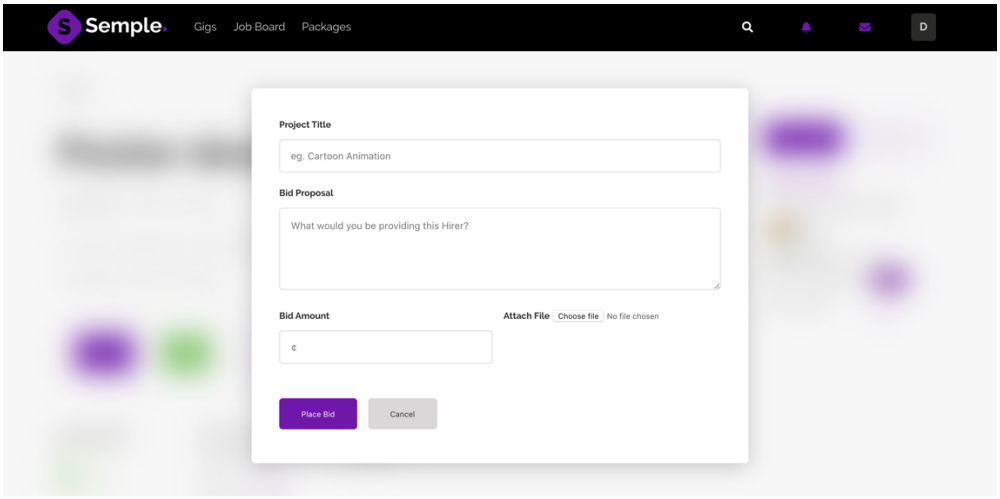

Figure 5. 13 SEMPLE Freelancer bid placement form

- **REQ-009**

  To make the application process for jobs and on the platform more straightforward, the Freelancer's profile is structured such that the data could be passed to a pdf document. The freelancer profile section is also structured as such so that with a link to their profile page, they are CV information could be accessed online. This page was tested to make sure the correct data is being displayed at all times.

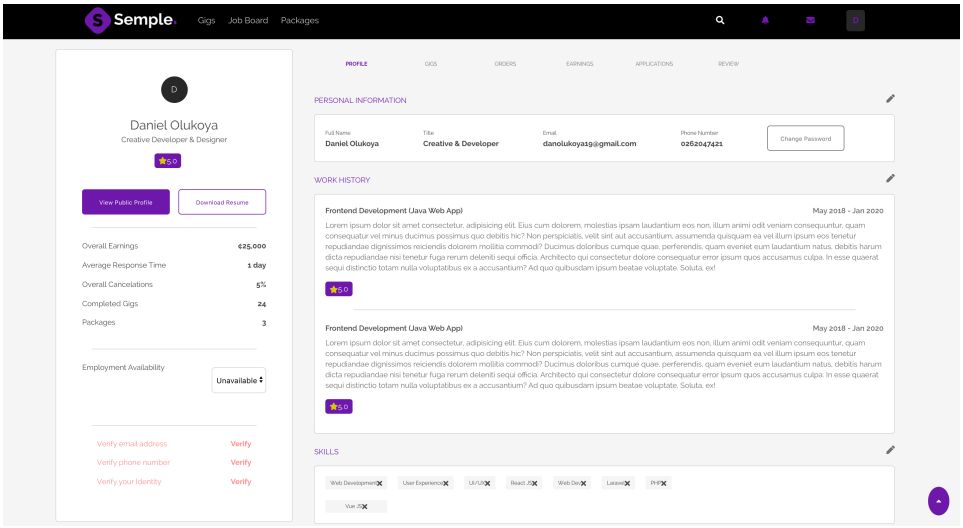

Figure 5. 14 SEMPLE Freelancer profile section

Chapter 6: Conclusion

This chapter sheds light on how SEMPLE aids people of different roles with the job search process. Also, the chapter explores the various limitations that were faced whiles developing this platform, recommendations for improvement and future works that would all help make SEMPLE much better than it currently is.

The main aim of SEMPLE (Self Employment Makes People's Lives Easier) is to promote self-employment of people, though providing avenues for people to get full time/wage employment. Individuals on the platform seeking to be freelancers could do so in two ways. Firstly, as freelancers relying on posted gigs to get jobs, this process has been made quite smooth, with the provision of filters to better searches and tags that form of gigs that people fit for best. The other way to freelance on the platform is by packages. Packages allow freelancers who have existing skill-based businesses off the platform, to show these to hirers looking for a specific person to undertake a task for them based on the Freelancer's terms.

6.1 Recommendations for Future works

- Migration to mobile application

As the platform was in the beginning stages. It was decided that it would be developed as a web application to first test out the idea. As a web application, it would be more accessible to a larger group with ease. Upon reaching a stable usage flow in future, the platform would be integrated with a mobile application, to allow better accessibility for the users.

- Making API Open for integration into an external application

The REST API's [19] for some functionality would be made open source, to allow freelancers, advertiser, developers, among others, for their various integrations and sourcing. Thing like analytics on the data collected, salaries per industry, among others, would be made available to the public with a trademark SEMPLE.

- Including Salary based filtering

With a large amount of data, a filtering system based on the average salaries of the various industries would be created. This would allow the freelancers to search for their job opportunities by salaries. This analysis would be done based on the data collected on the platform and not out, to enable authenticity.

- Provision of Skills section

A skills section of the content generated by freelancers and tutors on the platform would be created. Both free and paid. This would, in turn, help the freelancers have another mode of income through teaching. Also, it would allow freelancers to have a relatable base for learning, right there on the platform.

- Integration of more local payment options for various countries

This system is meant to make Self EMployment in People's Lives Easier (SEMPLE). The "P" is People nor Ghanaians only. SEMPLE would be grown to fit into the payment options of various local societies access to the internet. This would break that major factor if receiving funds. The platform is also being made to be lightweight and usage with lower signals and power doing various load tests.

- Developed mainly for the Ghanaian society

The system could be said to have been built only for the Ghanaian society. This was mainly because of the accessibility to other countries payment systems without specific details. If able to reach the appeal of an external audience, then more work would be put into penetrating such areas.

- Cross Role Activity

Currently, the roles on the platform are static, in that this is no cross-role activity for any user. Making provision for this would significantly improve accessibility on the platform, as long as the data integrity is kept. With the way the system is built now the user would have to

create multiple accounts to access the different roles, though there could be cases these roll overlap.

- Integrate image to text for verification and cv upload

Currently, every Freelancer would have to enter in all the values to fill their profile. However, with the integration of existing, maintained algorithms, this could be done. The upload would allow ease in populating data that the user already has. Image verification would allow ease for the admin. This is because it would take less time to verify users.

- Provision of resources for ensuring confidentiality on certain gigs/contract jobs.

In future, hirers would be provided the ability to attach document to the gigs they post so as to ensure confidentiality of such projects. Freelancers who are selected to work on such projects would also be required to provide an e-signature to bind them to the agreement with the hirer.

References

[1].~ Anthonylomagno. 2017. Laravel – The Open Source PHP Framework *OSD600*. Retrieved April 13, 2020 from https://anthonylomagno.wordpress.com/2017/09/11/laravel-the-open-source-php-framwork/

[2].Antoine Augusti. 2020. *AntoineAugusti/laravel-sentiment-analysis*. Retrieved May 7, 2020 from https://github.com/AntoineAugusti/laravel-sentiment-analysis

[3].Matthias Benz and Bruno S. Frey. 2004. *The Value of Autonomy: Evidence from the Self-Employed in 23 Countries*. Social Science Research Network, Rochester, NY. DOI:https://doi.org/10.2139/ssrn.475140

[4].Brian. 2020. Best Automation Testing Tools for 2020 (Top 10 reviews). *Medium*. Retrieved April 13, 2020 from https://medium.com/@briananderson2209/best-automation-testing-tools-for-2018-top-10-reviews-8a4a19f664d2

[5].William Craig. 2014. Working with JavaScript Dates Using Moment.js. *WebFX Blog*. Retrieved May 7, 2020 from https://www.webfx.com/blog/web-design/javascript-dates-moment-js/

[6].Maya E. Dollarhide. Self Employment Gives Flexibility and Freedom with Responsibility. *Investopedia*. Retrieved May 1, 2020 from https://www.investopedia.com/terms/s/self-employed.asp

[7].Randy Duermyer. Find a Work At Home Job on LinkedIn. *The Balance Small Business*. Retrieved May 2, 2020 from https://www.thebalancesmb.com/using-linkedin-to-find-a-work-at-home-job-1794574

[8].Randy Duermyer. How LinkedIn Works and Helps Your Business or Career. *The Balance Small Business*. Retrieved May 2, 2020 from https://www.thebalancesmb.com/introduction-to-linkedin-1794572

[9]. Nick Georgeandis. 2020. Fiverr review: Learn the pros and cons May 2020 | finder.com. *finder US*. Retrieved May 2, 2020 from https://www.finder.com/fiverr

[10]. Gregory Hundley. 2001. Why and When Are the Self-Employed More Satisfied with Their Work? *Industrial Relations: A Journal of Economy and Society*. (2001).

[11]. Jean Murray, MBA, Ph.D. Independent Contractor or Employee - What's the Difference? *The Balance Small Business*. Retrieved May 2, 2020 from https://www.thebalancesmb.com/independent-contractor-or-employee-what-s-the-difference-397912

[12]. Neoteric. 2018. Single-page application vs. multiple-page application. *Medium*. Retrieved April 13, 2020 from https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58

[13]. Fulltime Nomad. 2017. What is Upwork & How to Make Money With It? *Fulltime Nomad*. Retrieved May 2, 2020 from https://www.fulltimenomad.com/what-is-upwork/

[14]. David Royer. 2020. *davidroyer/vue2-editor*. Retrieved May 7, 2020 from https://github.com/davidroyer/vue2-editor

[15]. Benny says. 2018. What is Laravel and Why You Should Learn it? *LaraShout*. Retrieved April 13, 2020 from https://www.larashout.com/what-is-laravel-and-why-you-should-learn-it

[16]. Upwork. Freelancing in America: 2019. Retrieved May 2, 2020 from https://www.slideshare.net/upwork/freelancing-in-america-2019/1

[17]. David Weil. 2014. *The Fissured Workplace: Why Work Became So Bad for So Many and What Can Be Done to Improve It*. Harvard University Press, Cambridge, Massachusetts.

[18]. David Weil. 2017. How to Make Employment Fair in an Age of Contracting and Temp Work. *Harvard Business Review*. Retrieved May 1, 2020 from https://hbr.org/2017/03/making-employment-a-fair-deal-in-the-age-of-contracting-subcontracting-and-temp-work

[19]. 100AD. Understanding And Using REST APIs. *Smashing Magazine*. Retrieved April 13, 2020 from https://www.smashingmagazine.com/2018/01/understanding-using-rest-api/

[20]. 2015. Software Testing - Definition, Types, Methods, Approaches. *Software Testing Material*. Retrieved April 13, 2020 from https://www.softwaretestingmaterial.com/software-testing/

[21]. 2016. 6 Types of Self-Employed, or Freelance, Flexible Workers. *FlexJobs Job Search Tips and Blog*. Retrieved May 1, 2020 from https://www.flexjobs.com/blog/post/types-self-employed-freelance-flexible-workers/

[22]. 2017. What is Glassdoor ? | Jobboardfinder News. *Jobboard Finder News*. Retrieved May 2, 2020 from https://www.jobboardfinder.com/news/what-is-glassdoor/

[23]. 2017. A 6-Step Guide to Web Application Testing - Usersnap. *Usersnap - Blog for collecting visual feedback*. Retrieved April 13, 2020 from https://usersnap.com/blog/web-application-testing/

[24]. 2017. The 7 Common Types of Software Testing! - Usersnap. *Usersnap - Blog for collecting visual feedback*. Retrieved April 13, 2020 from https://usersnap.com/blog/software-testing-basics/

[25]. 2017. Software Testing | Basics. *GeeksforGeeks*. Retrieved April 13, 2020 from https://www.geeksforgeeks.org/software-testing-basics/

[26]. 2020. *moment/moment*. Moment.js. Retrieved May 7, 2020 from https://github.com/moment/moment

[27]. Comparison with Other Frameworks — Vue.js. Retrieved April 13, 2020 from https://vuejs.org/v2/guide/comparison.html

[28]. Definition of SELF-EMPLOYED. Retrieved May 1, 2020 from https://www.merriam-webster.com/dictionary/self-employed

[29]. Fiverr - Freelance Services Marketplace for Businesses. *Fiverr.com*. Retrieved May 2, 2020 from https://www.fiverr.com/?utm_source=51512&utm_medium=cx_affiliate&utm_campaign=&cxd_token=51512_3662768_36052dd56ddef535c7e2201ae8ce3f9d&show_join=true

[30]. Ghana Unemployment Rate | 1991-2019 Data | 2020-2022 Forecast | Historical | Chart. Retrieved May 2, 2020 from https://tradingeconomics.com/ghana/unemployment-rate

[31]. Labor force, total - Ghana | data. Retrieved May 2, 2020 from https://data.worldbank.org/indicator/SL.TLF.TOTL.IN?locations=GH

[32]. MVC: Model, View, Controller. *Codecademy*. Retrieved April 21, 2020 from https://www.codecademy.com/articles/mvc

[33]. Postman | The Collaboration Platform for API Development. *Postman*. Retrieved May 7, 2020 from https://www.postman.com/

[34]. Report: Freelancing and the Economy in 2019. *Press, News & Media Coverage*. Retrieved May 2, 2020 from https://www.upwork.com/press/economics/freelancing-and-the-economy-in-2019/

[35]. sweetalert. *npm*. Retrieved May 7, 2020 from https://www.npmjs.com/package/sweetalert

[36]. SweetAlert2. Retrieved May 7, 2020 from https://sweetalert2.github.io/

[37]. Vue vs React in 2020: Which Framework to Choose and When. Retrieved April 13, 2020 from https://www.monterail.com/blog/vue-vs-react-2020

[38]. What is contract employment. *BDC*. Retrieved May 1, 2020 from https://www.bdc.ca/en/articles-tools/entrepreneur-toolkit/templates-business-guides/glossary/pages/contract-employment.aspx

[39]. What is Freelancer? | General | Freelancer Support. Retrieved May 2, 2020 from https://www.freelancer.com/support/General/what-is-freelancer

[40]. What Is Glassdoor And Why Should A Recruiter Use It? *Business 2 Community*. Retrieved May 2, 2020 from https://www.business2community.com/human-resources/glassdoor-recruiter-use-01133909

[41]. What is the gig economy? - Change Recruitment. Retrieved May 1, 2020 from https://www.changerecruitmentgroup.com/knowledge-centre/what-is-the-gig-economy

[42]. What is vanilla? - Definition from WhatIs.com. *WhatIs.com*. Retrieved April 12, 2020 from https://whatis.techtarget.com/definition/vanilla

[43]. Why do people choose frameworks over vanilla JS? Retrieved April 12, 2020 from /why-do-people-choose-frameworks-over-vanilla-js/

[44]. 2019. Ghanaian mobile money app Mazzuma plotting move into crypto. *Disrupt Africa*. Retrieved May 10, 2020 from https://disrupt-africa.com/2019/07/ghanaian-mobile-money-app-mazzuma-plotting-move-into-crypto/

[45]. What is MySQL Tutorial. *Website Tutorials - Find Out How to Use the Most Popular Web Apps*. Retrieved May 11, 2020 from https://www.siteground.com/tutorials/php-mysql/mysql/

[46]. What Are Migrations In Laravel? – Vegibit. Retrieved May 11, 2020 from https://vegibit.com/what-are-migrations-in-laravel/

[47]. Eric Kollegger. 2018. What is Axios.js and why should I care? *Medium*. Retrieved May 11, 2020 from https://medium.com/@MinimalGhost/what-is-axios-js-and-why-should-i-care-7eb72b111dc0

[48]. 2019. 5 Benefits of freelancing - Self-employment. *Skillvalue*. Retrieved May 29, 2020 from https://skillvalue.com/en/blog/tech-skills/5-benefits-of-freelancing/

[49]. Jessica Howington. 2019. The Pros and Cons of Freelance Jobs. *FlexJobs Job Search Tips and Blog*. Retrieved May 29, 2020 from https://www.flexjobs.com/blog/post/pros-and-cons-of-freelance-jobs/

Appendix

A.1 Flow Charts

Based on the different user scenarios defined during the requirement analysis, various user flows for their various interactions with the platform were generated. This was done to guide better the flow of development of pages and interactions of the various users of the platform.

- John is a freelance designer who gets jobs only when he is referred by a friend/previous hirer who knows his works. He is looking for a way to get more jobs without relying on referrals solely or leaving his home.

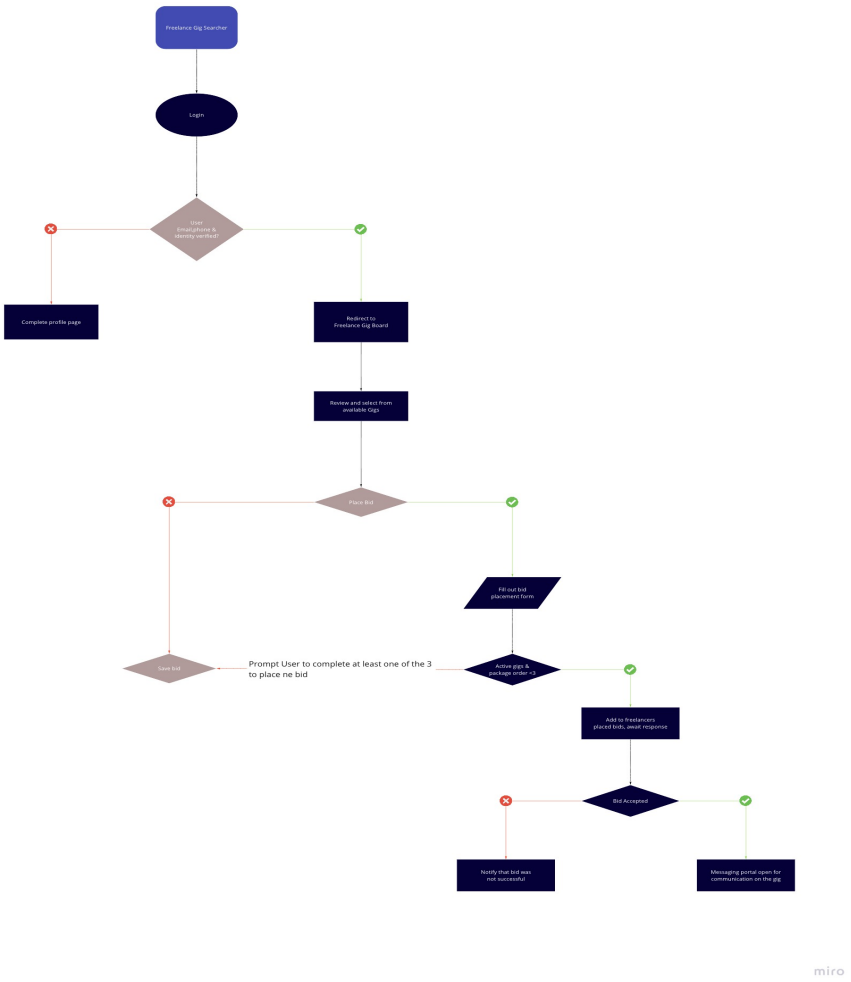

Figure A. 1 User scenario one flow chart

- Ama is a recent university graduate looking for a way to get a job without having to move from location to location to place her CV.

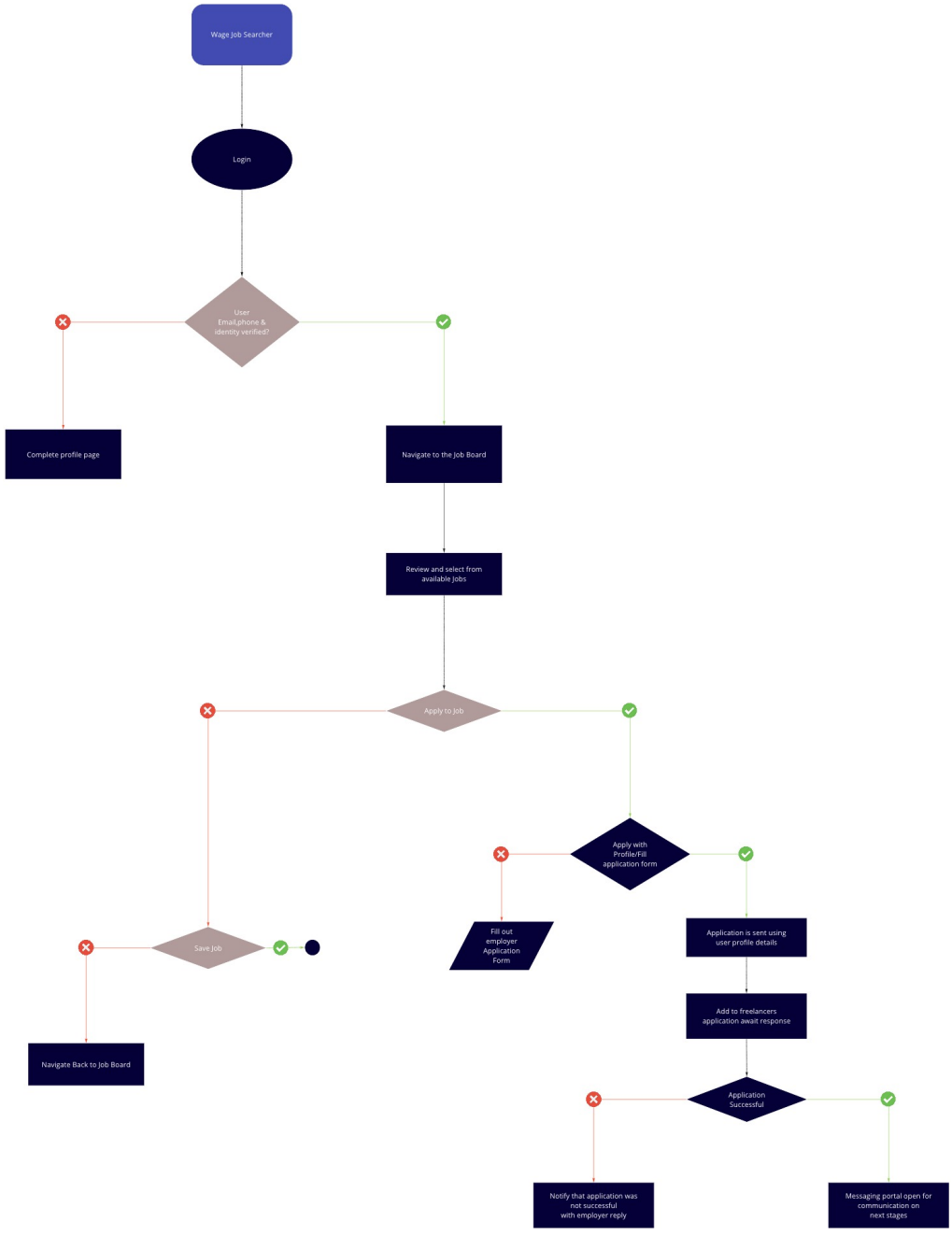

Figure A. 2 User scenario two flow chart

- Agatha is an Instagram based makeup artist and hairstylist. She is looking for a way to get more clients off Instagram. Her reason being that here post easily gets lost in the wide variety of content people post on Instagram, and her pricing also not being clear to possible customers

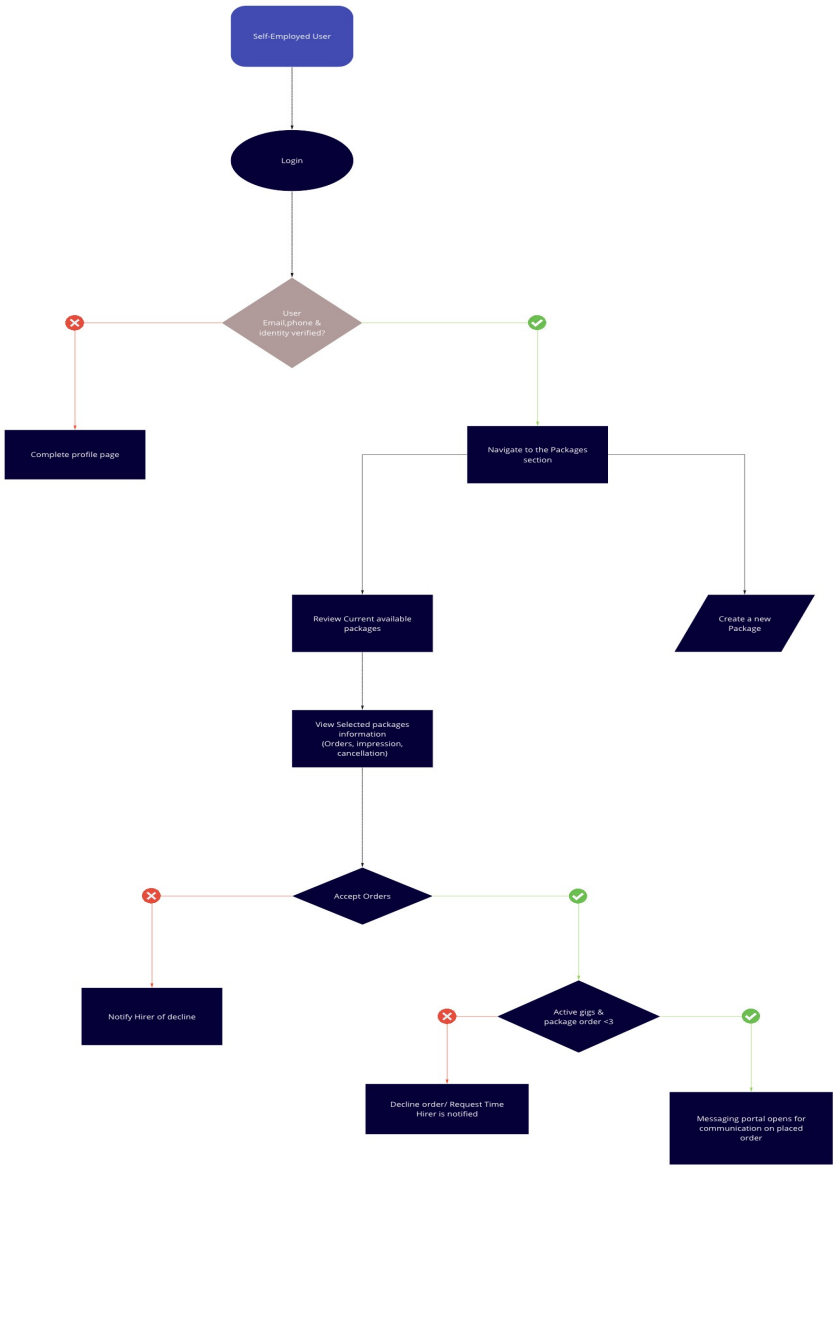

Figure A. 3 User scenario three flow chart

- Overloaded with work at his design firm, Michael is beginning to fear he would miss deadlines and that the quality of his work would reduce. He, therefore, needs a way to outsource so of the jobs he has and still ensure quality and timeliness

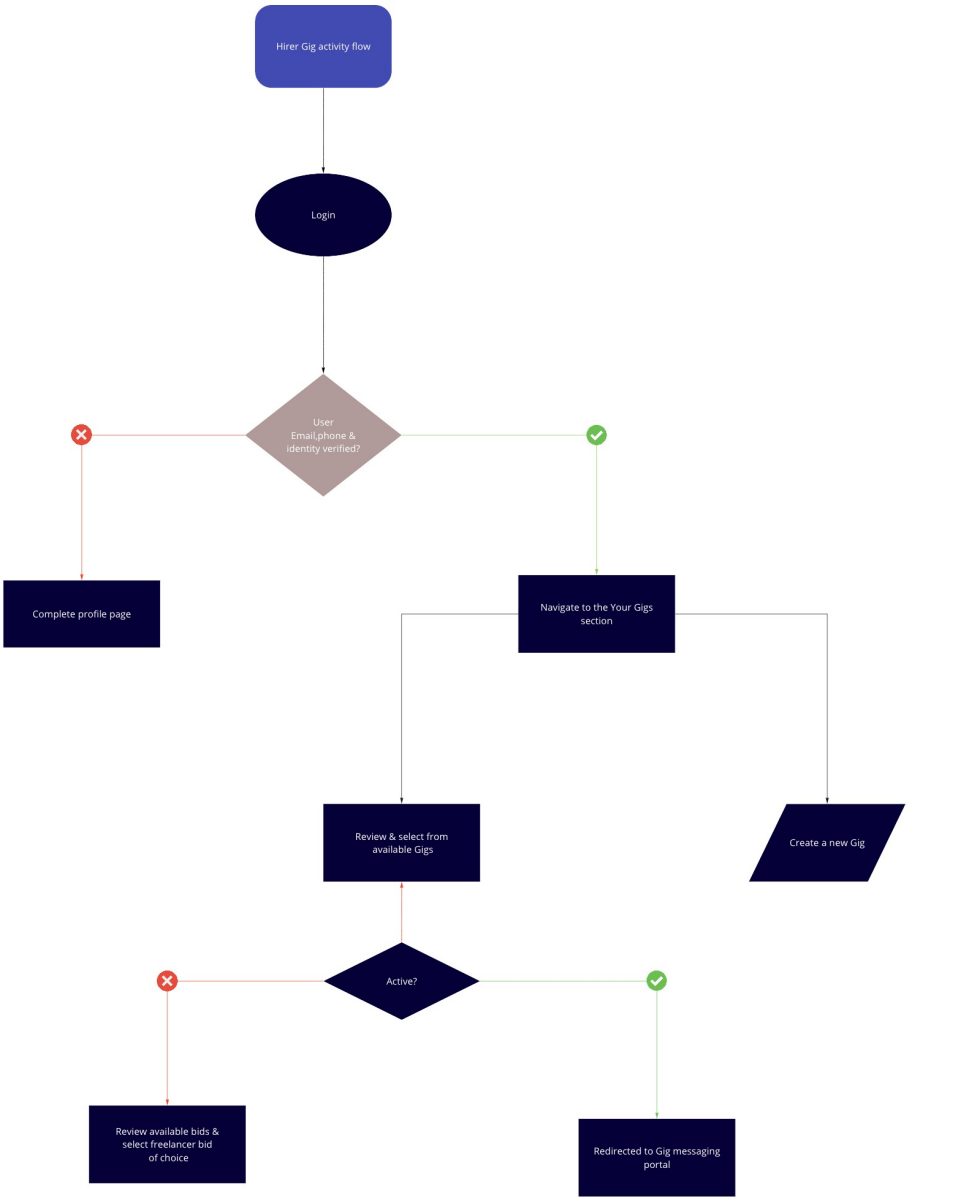

Figure A. 4 User scenario four flow chart

- Bridge is a digital agency form by a group of Ashesi students. Upon finishing school, they realize that for the type of projects they are managing and receiving, a bigger team would be required. They need a way to get new employees based on their abilities and not just their educational background.

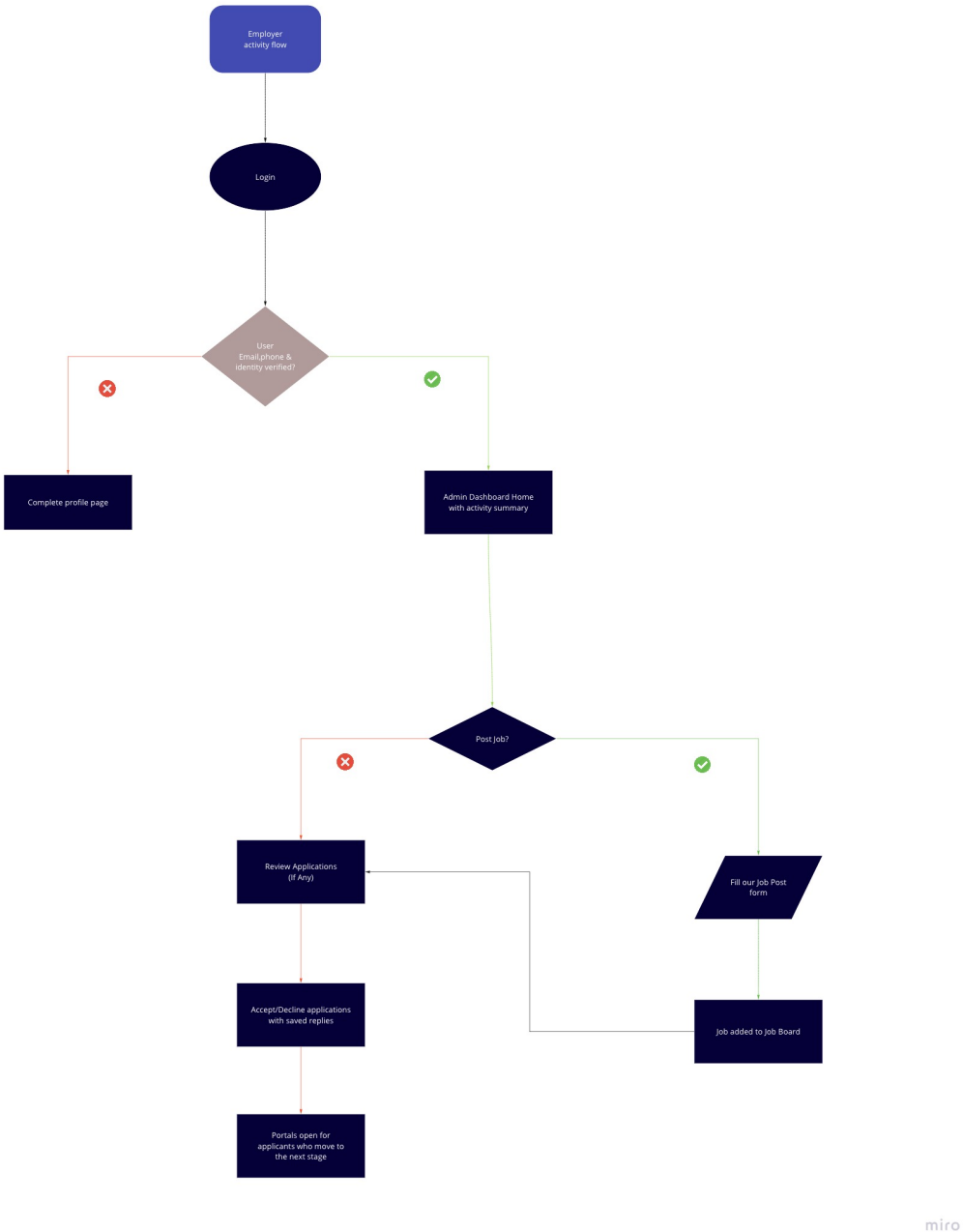

Figure A. 5 User scenario five flow chart

A.2 Entity Relational Diagram

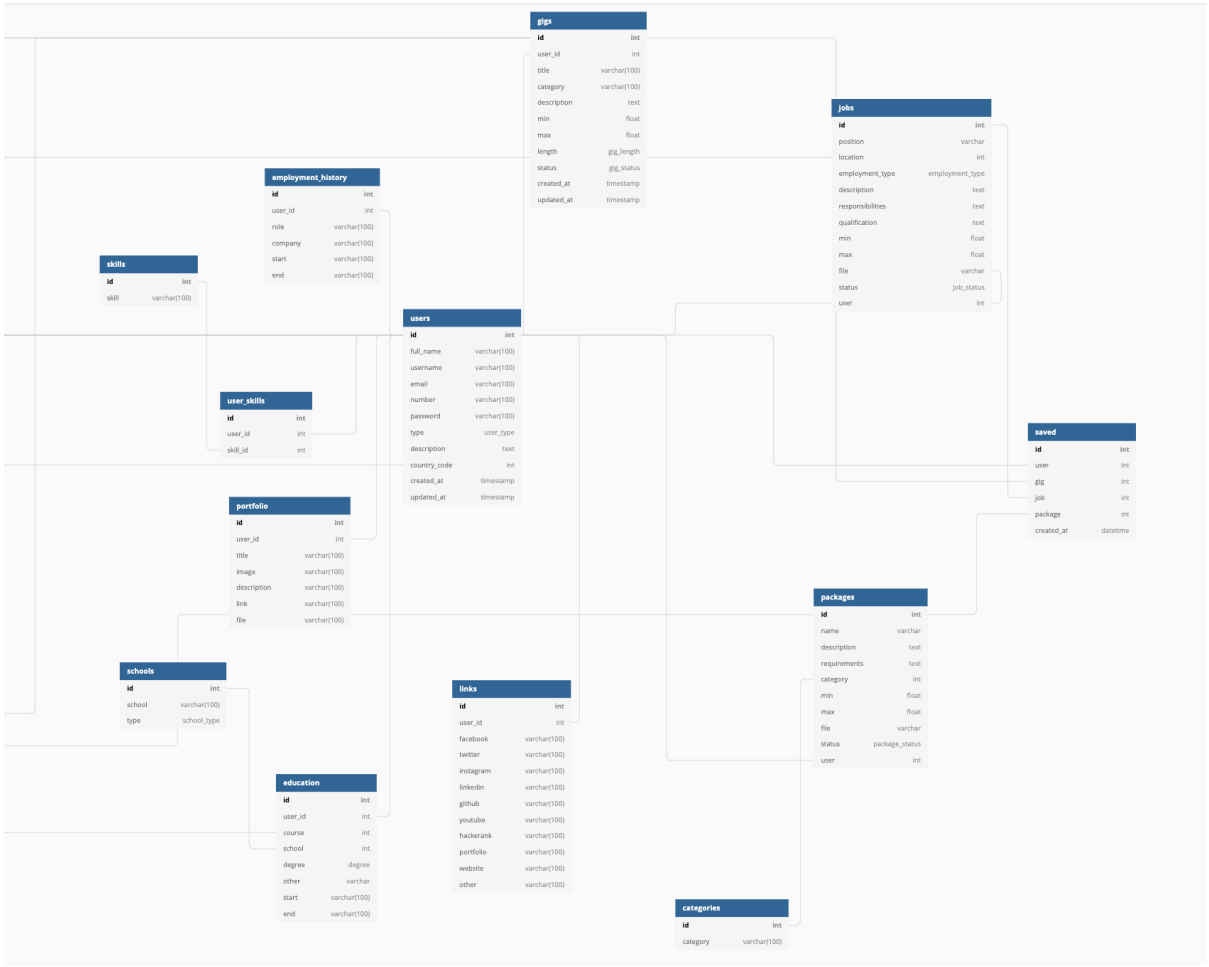

Figure A. 6 Major database tables