



# **ASHESI UNIVERSITY COLLEGE**

## **JOB LISTING APPLICATION FOR DOMESTIC WORKERS**

**APPLIED PROJECT**

B.Sc. Computer Science

**Anne N. Gitau**

**2017**

**ASHESI UNIVERSITY COLLEGE**

# **Job Listing Application for Domestic Workers**

## **Applied Project**

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

**Anne N. Gitau**

**April 2017**

## 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 College.

Supervisor's Signature:

.....

Supervisor's Name:

.....

Date: .....

## **ACKNOWLEDGEMENT**

I take this opportunity to thank all the people who were involved in any way to see the accomplishment of this project. I am eternally grateful to you.

In a special way, I would like to thank my supervisor, Mr David Amatey Sampah who was very consistent in giving feedback and checking on my progress with this project.

I would like to thank my sister Lucy Gitau and my good friend Brian Kangogo who were the major inspiration in taking on with this project.

To my family who gave me support and encouragement throughout this project, I say, 'Thank you'.

I would like to thank the Ashesi community for providing a conducive environment to carry out this project.

Above all I thank God for good health and state of mind throughout this project.

## **ABSTRACT**

In Kenya, youth unemployment rate is 80% of the unemployed population. These are people between the ages of 18 to 35 years. In most households where parents are working, it is very hectic to get a reliable and suitable household employer to take care of their homes when they are away working. Linking the unemployed population to prospective employers in the informal sector is therefore very necessary.

This write up gives details of a job listing user friendly web application, created to connect the two parties (working parents & unemployed youth), easily and reliably. The application builds on techniques that have been employed in existing solutions.

The uniqueness of the application is the fact that it uses google maps to locate vacancies and job seekers who are using the application.

Development and user testing were conducted to evaluate the usability of the application and to make sure that the application meets the listed requirements.

## TABLE OF CONTENTS

|  |      |
|--|------|
| DECLARATION .....                        | i    |
| ACKNOWLEDGEMENT .....                    | ii   |
| ABSTRACT.....                            | iii  |
| LIST OF TABLES .....                     | vii  |
| LIST OF FIGURES .....                    | viii |
| CHAPTER 1: INTRODUCTION .....            | 1    |
| 1.1. Background .....                    | 1    |
| 1.2. Problem Statement .....             | 2    |
| 1.3. Project Motivation.....             | 2    |
| 1.4. Project Objective .....             | 3    |
| 1.5. Assumptions .....                   | 3    |
| 1.6. Wish List .....                     | 4    |
| CHAPTER 2: RELATED WORKS.....            | 5    |
| 2.1. Related Works .....                 | 5    |
| 2.1.1. Indeed.com .....                  | 5    |
| 2.1.2. Bright.com/ LinkedIn .....        | 6    |
| 2.1.3. Kenya great AU Pair.....          | 8    |
| CHAPTER 3: DESIGN AND ARCHITECTURE ..... | 10   |
| 3.1. Requirements.....                   | 10   |

|                                 |  |    |
|---------------------------------|--|----|
| 3.1.1.                          | User Classes.....                        | 10 |
| 3.1.2.                          | System Requirements.....                 | 10 |
| 3.1.3.                          | Functional Requirements .....            | 11 |
| 3.1.4.                          | Non-Functional Requirements .....        | 11 |
| 3.1.5.                          | System Constraints.....                  | 12 |
| 3.2.                            | System Modelling .....                   | 13 |
| 3.2.1.                          | Activity Diagrams .....                  | 13 |
| 3.2.2.                          | User Case Diagrams.....                  | 14 |
| 3.2.3.                          | Sequence Diagrams.....                   | 16 |
| CHAPTER 4: IMPLEMENTATION ..... |  | 18 |
| 4.1.                            | Technology Used for Development .....    | 18 |
| 4.2.                            | Classes and Methods .....                | 18 |
| 4.2.1.                          | Connection Class .....                   | 19 |
| 4.2.2.                          | User Class .....                         | 19 |
| 4.2.3.                          | Vacancy Class .....                      | 20 |
| 4.2.4.                          | Profiles Class .....                     | 20 |
| 4.3.                            | Database Entity Relational Diagram ..... | 21 |
| 4.4.2.                          | Navigation Bar .....                     | 23 |
| 4.4.3.                          | User Sign Up.....                        | 24 |
| 4.4.4.                          | User Login .....                         | 25 |

|  |                                      |    |
|--|--------------------------------------|----|
| 4.4.5.   | Add Vacancy.....                     | 26 |
| 4.4.6.   | Search Vacancy.....                  | 26 |
| 4.4.7.   | View Vacancy.....                    | 26 |
| 4.4.8.   | View Profiles .....                  | 27 |
| 4.4.9.   | Search Job Profile .....             | 27 |
| 4.4.10.  | View on Map.....                     | 28 |
| CHAPTER 5: TESTING.....                        |                                      | 29 |
| 5.1.   | Introduction .....                   | 29 |
| 5.2.   | Development Testing .....            | 29 |
| 5.2.1.   | Unit Testing .....                   | 29 |
| 5.2.2.   | Component Testing.....               | 31 |
| 5.2.3.   | System Testing.....                  | 31 |
| 5.2.4.   | User testing.....                    | 33 |
| 5.3.   | User Test Finding .....              | 33 |
| CHAPTER 6: CONCLUSION AND RECOMMENDATION ..... |                                      | 34 |
| 6.1.   | Conclusion.....                      | 34 |
| 6.2.   | Recommendation and Future Works..... | 34 |
| REFERENCES .....                               |                                      | 36 |



## LIST OF TABLES

|                                   |    |
|-----------------------------------|----|
| Table 4.1: Connection Class ..... | 19 |
| Table 4.2:User class .....        | 19 |
| Table 4.3: Vacancy class.....     | 20 |
| Table 4.4: Job Seeker class ..... | 21 |

## LIST OF FIGURES

|  |    |
|--|----|
| Figure 2.1: Indeed.com job search page .....                     | 5  |
| Figure 2.2: LinkedIn job search page .....                       | 7  |
| Figure 2.3: LinkedIn Job search page (continued).....            | 7  |
| Figure 2.4: Kenyan Great AU Pair page.....                       | 9  |
| Figure 3.1: Activity diagram for vacancy search.....             | 13 |
| Figure 3.2: Activity diagram for job seeker search.....          | 13 |
| Figure 3.3: User Case Diagram for Jobseekers and Employees ..... | 14 |
| Figure 3.4: User case Diagram for Administrator .....            | 15 |
| Figure 3.5: Employer Adding a Vacancy .....                      | 16 |
| Figure 3.6: Job Seeker Adding Profile.....                       | 17 |
| Figure 4.1: ER Diagram for User System.....                      | 22 |
| Figure 4.2: Landing page .....                                   | 23 |
| Figure 4.3: Wide Screen navigation bar .....                     | 23 |
| Figure 4.4: Narrow Navigation Bars.....                          | 24 |
| Figure 4.5: Sign Up.....   | 25 |
| Figure 4.6: Login Interface .....                                | 25 |
| Figure 4.7: Add Vacancy .....                                    | 26 |
| Figure 4.8: Add Job Profile.....                                 | 27 |
| Figure 4.9: Search Job Profile.....                              | 28 |
| Figure 4.10: Opportunities on Map.....                           | 28 |
| Figure 5.1: User class getUser () method.....                    | 30 |
| Figure 5.2: Test class testGetUser () method.....                | 30 |

|                                       |    |
|---------------------------------------|----|
| Figure 5.3: Unit testing results..... | 31 |
| Figure 5.4: input validation.....     | 32 |
| Figure 5.5: user validation .....     | 32 |

# **CHAPTER 1: INTRODUCTION**

## **1.1. Background**

Youth unemployment in third world countries is of major economic and social concern. In a research conducted by the World Bank, 60% of unemployed people in the Sub Saharan Africa are youths below the age of 35 years. In Kenya, youth unemployment rate is as high as 80% of the unemployed population. (World Bank, 2009).

With the Kenya, Free Primary Education (FPE) launched in 2002, followed by the subsidized secondary school education, many Kenyan youth can attend both primary and secondary school. The problem comes in when these youths want to enrol to higher learning institutions. Taking an example from the year 2007, out of 267,000 students who sat for the Kenya Certificate of Secondary Education Examination, 82,000 qualified to join higher learning institutions. Only 10,000 were selected for government sponsorship, 10,000 enrolled on self-paying basis while 5,000 entered the private sector. This left over 200,000 youths neither working nor enrolled in higher learning institutions. (Kenya national Bureau of Statistics, 2014).

While several of these unemployed youths may get into the informal sector to earn a living, a big number of them end up staying idle in the towns and cities and even getting involved in criminal activities. Amongst the jobs in the informal sector that these youths can do includes: nannies, baby sitters, pet sitters, gardeners, watch men and women, care takers and cleaners. It is however difficult for the youth to locate places or homes with vacancies for these kinds of jobs.

It is also difficult for the working population to locate and connect with youths to work for them in this informal job. Most of them take time off from work to travel to the villages to get

people to work for them or in the worst-case scenario, they quit their jobs to take care of their homes.

The project aims to reduce this gap of youth unemployment by coming up with an easy way to help unemployed youths to locate places and homes with informal job vacancy. A web application will be developed to connect informal job seekers to potential employers in an attempt to reduce the unemployment.

## **1.2. Problem Statement**

In this era of technological advancement, a lot of information is accessible through the internet. This has allowed people to get opportunities ranging from school applications, ecommerce and job application and recruitment in the formal sector.

There are however limited online resources for people who want to work in the informal sector. People who want to work as gardeners, nannies, bedsitters or home care takers end up spending a lot of time searching for vacancies from office to office or home to home. This scenario has posed a challenge in the informal sector, where it is difficult and time consuming to find a job vacancy.

## **1.3. Project Motivation**

During my summer vacation, I volunteered at Max Homecare Limited, a company that connects nannies and other domestic workers to prospective employers. The company advertises on social media and interested people would call for invitation to submit documents to the company's office. They would then wait for employers to contact the company looking for a prospective employee. The company will then link up the two and facilitate an interview with

consequent follow ups on hiring. This process takes too long, not less than a month in most cases and was not very effective because sometimes the matches made did not work out well.

Data was mainly recorded on paper which is risky in case of fire or floods. The only online data kept was on excel sheet which was very hectic to search and sort. When the founder of the company left the country to pursue his masters, he left the company with less support, because only he understood fully how the company was intended to function. To keep the company running in his absence, he had to recruit and train more people, a process that was not only time consuming, but also costly.

This motivated the project to develop an online solution that would help job seekers in the informal sector to connect to prospective employers without the need for a third party being actively involved.

#### **1.4. Project Objective**

This project seeks to develop a web application that will help to connect job seekers to potential employers in the informal sector.

The application will allow job seekers to make an online profile. They should be able to search for available vacancies and connecting with prospective employer.

The application will allow needy homes to post job vacancies. They should be able to search through the available job seekers and communicate with the most suitable job seeker for their need.

#### **1.5. Assumptions**

For this project, it is assumed that:

The users can access a computer or a phone

The users have adequate internet connectivity or internet bundles.

The employer does not have any problem employing a stranger to work for him or her.

## **1.6. Wish List**

The application will incorporate google maps plugins, in the same way some transportation application like Uber have utilized it. Once a job seeker logs in to the application either on a computer or a mobile device, they can see the various vacancies around them and using the map, they can easily navigate to these places. The application will also incorporate SMS technology, where a prospective employer can send a message to a job seeker to schedule an appointment with them prior to employment.

Kenyan's internet penetration stands at 54.8% of the population, which translates to 22.3 million internet users. This is relatively a high internet penetration compared to the 40% global internet penetration. (Cabinet Secretary Ministry of Planning and Devolution, 2012). This shows that the application has a chance of reaching many people especially in the urban areas.

In comparison to the existing solutions such as Max Child Care in Kenya and Nanny Connection in South Africa, where a third party or an agent is involved, this application will connect the job seeker directly to the prospective employer and vice versa. This will help to limit overheads that occur because of involving a third party. For instance, several agencies require job seekers registered under their agency to pay some amount of money for the first three months of employment. The employer and the employee will have the liberty to agree on their terms and conditions without the need for third party interference.

## CHAPTER 2: RELATED WORKS

### 2.1. Related Works

This project seeks to develop a web application for listing domestic jobs like nannies, watchmen and women, babysitters, gardeners' etc. There are systems that publishes job listing with different algorithms to match up employers and jobseekers. This section will go through some of the algorithms used to achieve these kinds of match making. Three applications, Bright.com, Indeed.com and Kenya Great AU Pair will be analyzed.

#### 2.1.1. Indeed.com

Indeed.com is intuitive and easy to use. It takes only two inputs: a keyword (Job title or company) and location. These inputs are used to search through the posted positions on the site. The search result page displays a list of matching jobs ranked by relevance. This search engine is one of the primary ways job seekers find employment mainly in the United States of America. Indeed.com has however gone beyond this search method to adding recommendation algorithms to give better matches to both job seekers and employers. (Appan, 2016).

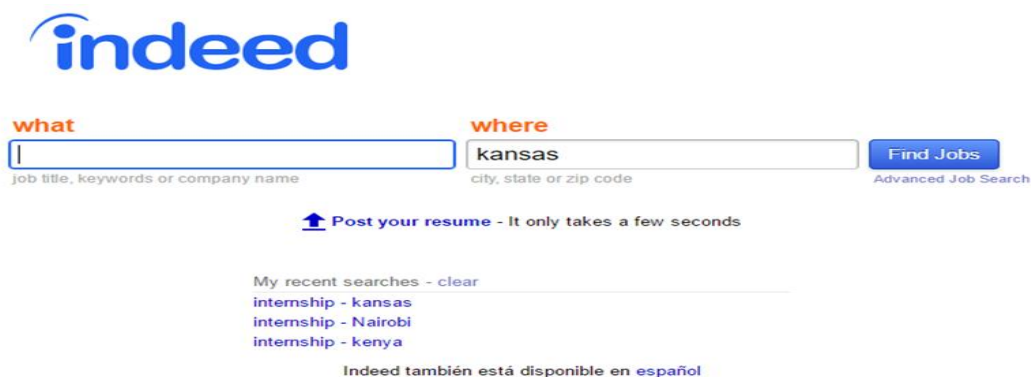


Figure 2.1: Indeed.com job search page



Given a set of job seekers and employers, indeed.com can match them up according to their needs. There are two main approaches that indeed.com uses to make these matches. These are content-based approach and behavior based approach. They both have their pros and cons but the project will focus on how they work. (Appan, 2016).

For content based approach, the algorithm mainly relies on the data provided by both the job seeker and the employer. It uses the keywords of the job description and those uploaded by the jobseeker and the more similar these keywords are, the better a match is made. (Appan, 2016)

On the other hand, behavior based algorithm leverages on user behavior. The more active jobseekers are on the sight the better recommended they are to employers. This approach however suffers from a cold start problem. That is, new users take a while to adapt and may not be very active. With little user activity, it is generally hard to generate good recommendation base. Content based approach is therefore the preferred algorithm. (Appan, 2016).

Merging these two approaches would produce a great system that would recommend jobs to seekers or introduce seekers to employers based on the content or data they present while signing up. (Appan, 2016)

### **2.1.2. Bright.com/ LinkedIn**

Bright.com was launched in 2011. It later partnered with LinkedIn in 2014. Bright.com works in many ways like a dating sight, as it uses data and algorithms to match candidates up with potential jobs. The common approach is using common interests presented by jobseekers and employers to make matches. (Giang, 2013).

**Let recruiters know you're open**  
Share that you're open to new opportunities and let your next job find you.

☒ On

**What kind of jobs are you considering?**

**Title**  
[+ Add title](#)

**Job type**  
☐ Full-time 
 ☐ Part-time 
 ☐ Contract 
 ☐ Internship 
 ☐ Remote 
 ☐ Freelance

**When can you start?**  
Anytime

**Introduction** (300 characters or less)  
What else should recruiters know?

Figure 2.2: LinkedIn job search page

**Get more relevant job recommendations**  
Help us see past your profile details so we can surface the most relevant jobs to you..

**What locations would you like to see jobs in?**  
[+ Add location](#)

**Which experience levels are you interested in?** Training to Executive

Training   Entry Level   Experienced Hire   Manager   Director   Executive

**What industries are you considering?**

- ✓ Entertainment
- ✓ Motion Pictures and Film
- ✓ Broadcast Media
- ✓ Museums and Institutions
- ✓ Fine Art
- ✓ Performing Arts
- ✓ Marketing and Advertising
- ✓ Newspapers
- ✓ Publishing
- ✓ Printing
- ✓ Information Services
- ✓ Libraries
- ✓ Design
- ✓ Writing and Editing
- ✓ Computer Games
- ✓ Arts and Crafts
- ✓ Online Media
- ✓ Media Production
- ✓ Animation
- ✓ Photography
- ✓ Graphic Design
- [+ Add industry](#)

Suggested based on your experience

**What size company would you like to work for?** 201 to 10,001+ employees

Figure 2.3: LinkedIn Job search page (continued)


Unlike Indeed.com that make use of keyword and location to filter out preferences, Bright.com incorporates a range of variables to make the matches. These variables include education, prior employment, and skills listed to infer other skills that may not be listed.

Bright.com uses an algorithm that recognizes pattern from specific employer while making the matches. For instance, the algorithm would know that company, say Company A, likes to hire people who attended a certain university or worked within a particular industry, and typically does not hire people from a different university or company. The algorithm analyzes these patterns to predict the kind of jobseekers to recommend to the employer. This algorithm may become problematic because it filters out qualified people who may have not attended a university, but would have been an excellent candidate for the job in question. (Giang, 2013)

### **2.1.3. Kenya great AU Pair**

Kenya Great AU Pair Job Service is an online domestic job pairing service that uses advance tools and security systems to pair job seekers and job givers. It is common in European countries and recently, some African countries such as Kenya have started using it. Most of the job seeker using Kenya Great AU Pair are between the age of 19 and 30 years. Through this platform, job seekers find domestic jobs as nannies, babysitters, pet sitters, housekeepers, tutors and senior care.

The platform does the sorting and matching based on criteria provided by the clients to make best matches and help the jobseeker and job giver to connect. Studying the architecture used on these applications will be very beneficial for this project. This Project work looks to improve the functionality by adding google maps for easier matching.


**GREATAUPAIR** Where care providers and families click.™  
 Au Pairs, Nannies, Babysitters, Pet Sitters, Housekeepers, Tutors and Senior Care.

Welcome!  
[Register Now](#) or [Sign in](#)  
[Support](#) | [Blog](#) | [Home](#)

[Find Jobs](#)
[Hire](#)
[J-1 Au Pairs](#)
[Join Free](#)
[My Account](#)


[All Jobs](#)
[Au Pair](#)
[Nanny](#)
[Babysitter](#)
[Pet Sitter](#)
[Housekeeper](#)
[Tutor](#)
[Senior Care](#)
[Personal Assistant](#)
[FastFind](#)

### Kenya Au Pair Jobs Available. Get hired for Local Part-time or Full-Time Au Pair Jobs in Kenya. Find great Au Pair Jobs worldwide.

Click a state/province below, to easily find all part-time and full-time Au Pairs available in that state/province. Find Au Pairs that meet your specific requirements. Review Au Pairs and apply for free.

GreatAuPair helps people just like you find great Au Pairs. We give you the most advanced search tools and security systems available to make it safe and easy to find Au Pairs with the specific traits you want. Whether you're seeking a part-time, live-out job or perhaps you want a full-time, live-in job... Either way, if you want great Au Pairs, we've got you covered.

Sign up free. Search free. Connect free. GreatAuPair offers the easiest, most direct way to find great Au Pairs locally and worldwide. Safe. Secure. Privacy Assured. Start Now!



#### Find Au Pairs


- Guided video tour
- Founders' Video
- Customer Videos
- GreatAuPair customer reviews

#### Find Caregivers

#### Au Pairs

[View All](#)

[an outgoing loving and carin...](#)




Nairobi, Kenya  
Live-in, Full Time  
Feb 2017 - Apr 2017  
[View aupair](#)

#### Au Pair Jobs

[View All](#)

[English native Au Pair to he...](#)



Nyeri, Kenya  
Live-in, Full Time  
Jul 2016 - Jul 2016  
[View aupair](#)

Figure 2.4: Kenyan Great AU Pair page

## **CHAPTER 3: DESIGN AND ARCHITECTURE**

### **3.1. Requirements**

#### **3.1.1. User Classes**

There are three user classes for this application namely; employers, job seekers and the system administrator.

- i. Employers—these are people who have vacancies in their homes. They can add vacancies on the application, seeking for nannies, babysitters, caretakers etc.
- ii. Job seekers—these are individuals seeking employment in the informal sector to work as nannies, babysitters, cooks, watchmen etc.
- iii. System administrator—this is the person in charge of maintaining the application and managing all privileges of the system.

#### **3.1.2. System Requirements**

The domestic job listing application will display all vacancies posted and a list of job seekers on the landing page. The application will also give the user the option to view the vacancies and jobseekers on a map showing their geographical location. The application will also provide interfaces to get started as an employer to add a vacancy or as a job seeker to add a job profile.

To add a vacancy or a profile, the user is first required to register and login to the application. For registration, the user will provide; full name, email, username and a password. The application will allow a user to login only if they have a registered account.

Application data will be stored in a database. This include the geolocation of the user and the coordinates. A map will be used to display the location of users.

The application will allow the user to edit or delete the details of a post on the application.

The application will send an automated email to a job seeker once an employee indicate interest.

The application does the same to the employer if a job seeker clicked on the 'interested' button on a vacancy.

### **3.1.3. Functional Requirements**

- i. The application allows all user to view vacancies and job seekers profile.
- ii. The application allows a new user to register and login.
- iii. Only registered users can post a vacancy or a job seeker profile.
- iv. Users can only edit job vacancy and/or profile in their name.
- v. The application allows users to pin their location on a map.
- vi. An admin can add, edit, and delete any information in the application.
- vii. Users can search for vacancies or available job seekers.

### **3.1.4. Non-Functional Requirements**

#### *I. Usability and learnability*

This application should be easy to learn. It should have clear and simple instruction to ensure easy navigation even for new users. Menus and buttons should also be clearly label and positioned.

#### *II. Dependability*

The application should delivers as expected without fail all times.

### III. *Scalability*

The server hosting the application should be running always. It should be able to support many users accessing the application at the same time.

### IV. *Maintainability*

This application should be easy to maintain and improve upon as time goes on. This should be possible without having to shut down the software.

### V. *Security*

All information provided by users should be secured from unauthorized access. Users' passwords are encrypted using MD5 before storing them in the database. Also, SQL queries are rendered as prepared statement to avoid MySQL injection.

#### **3.1.5. System Constraints**

- i. The application requires Internet connection to work.
- ii. The application has a mapping system that can only run on GPS enabled phones or computers.
- iii. The users can only alter information they posted. Only an administrator can access all the functionalities of the application.

## 3.2. System Modelling

### 3.2.1. Activity Diagrams

Below are activity diagrams for user searching and posting an opportunity on the application.

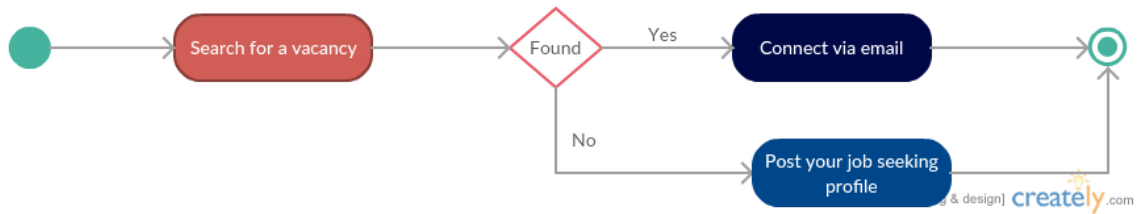


Figure 3.1: Activity diagram for vacancy search

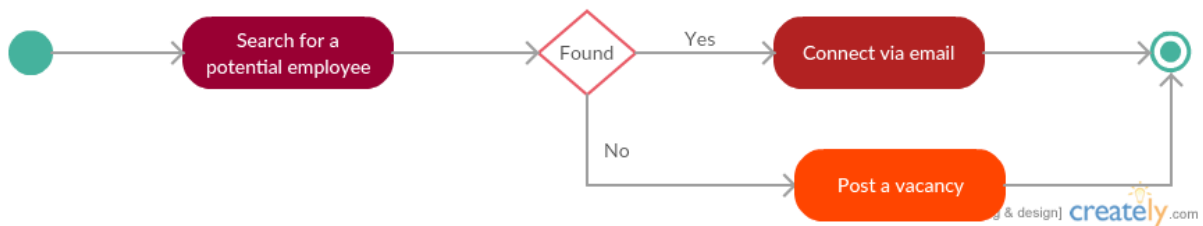


Figure 3.2: Activity diagram for job seeker search



### 3.2.2. User Case Diagrams

Below is the use case diagram for Employer and Job Seeker.

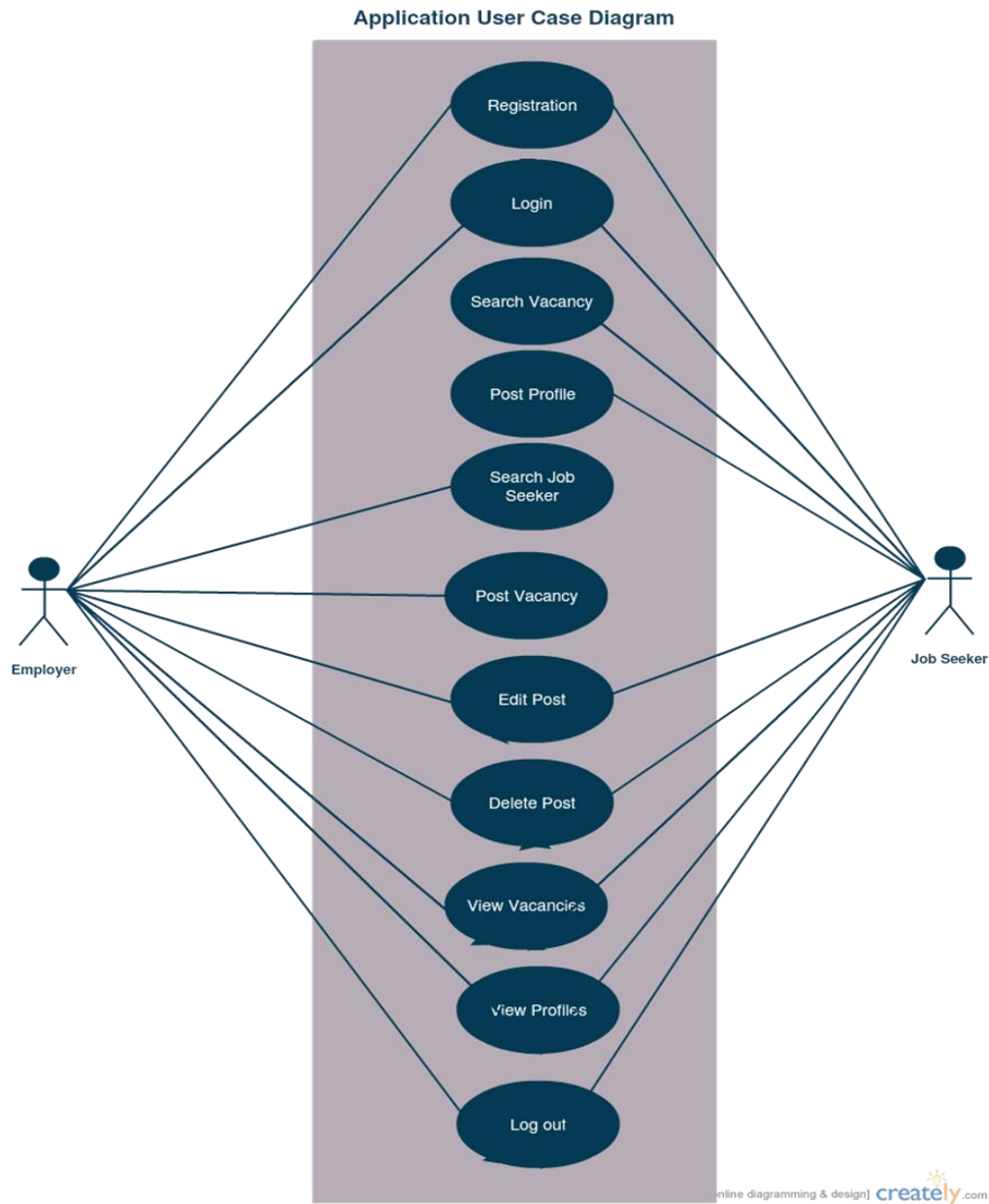
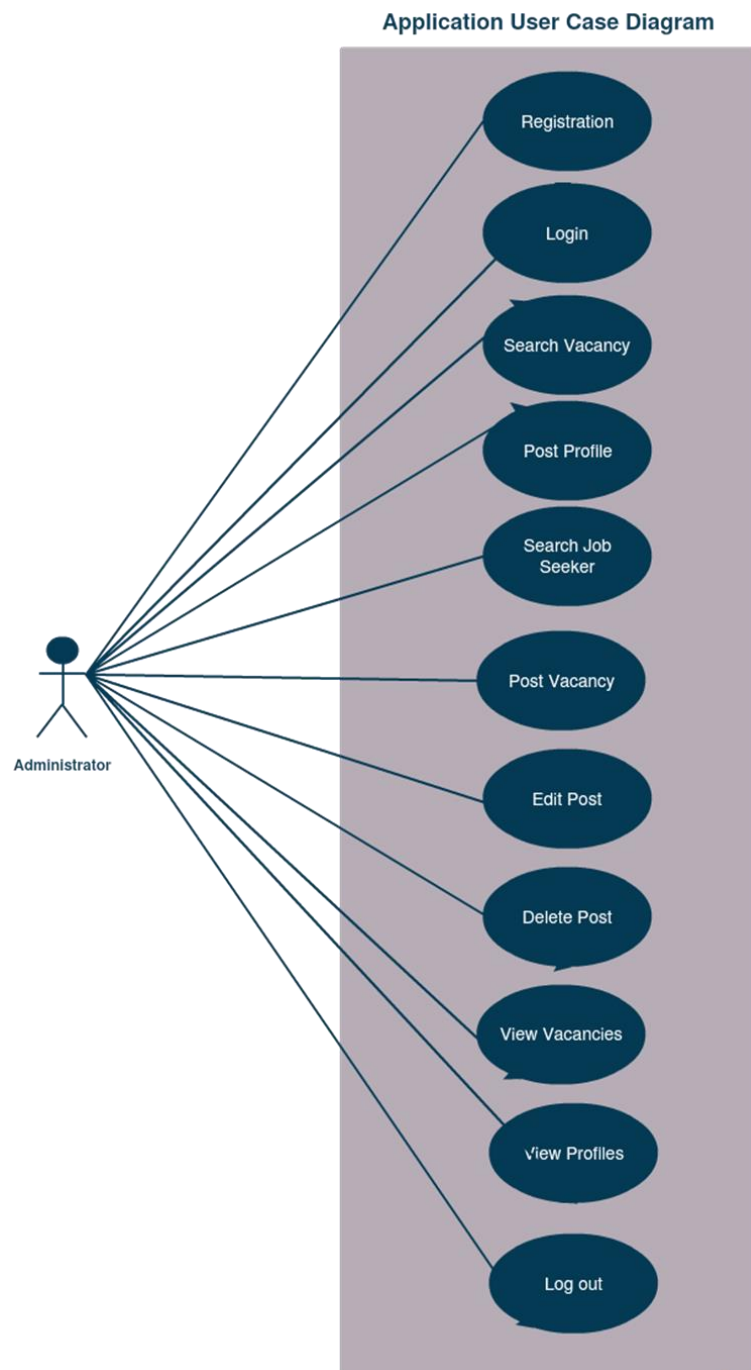


Figure 3.3: User Case Diagram for Jobseekers and Employees

Below is the use case diagram for Administrator.



*Figure 3.4: User case Diagram for Administrator*

### 3.2.3. Sequence Diagrams

Below is a sequence diagram for an employer adding a vacancy.

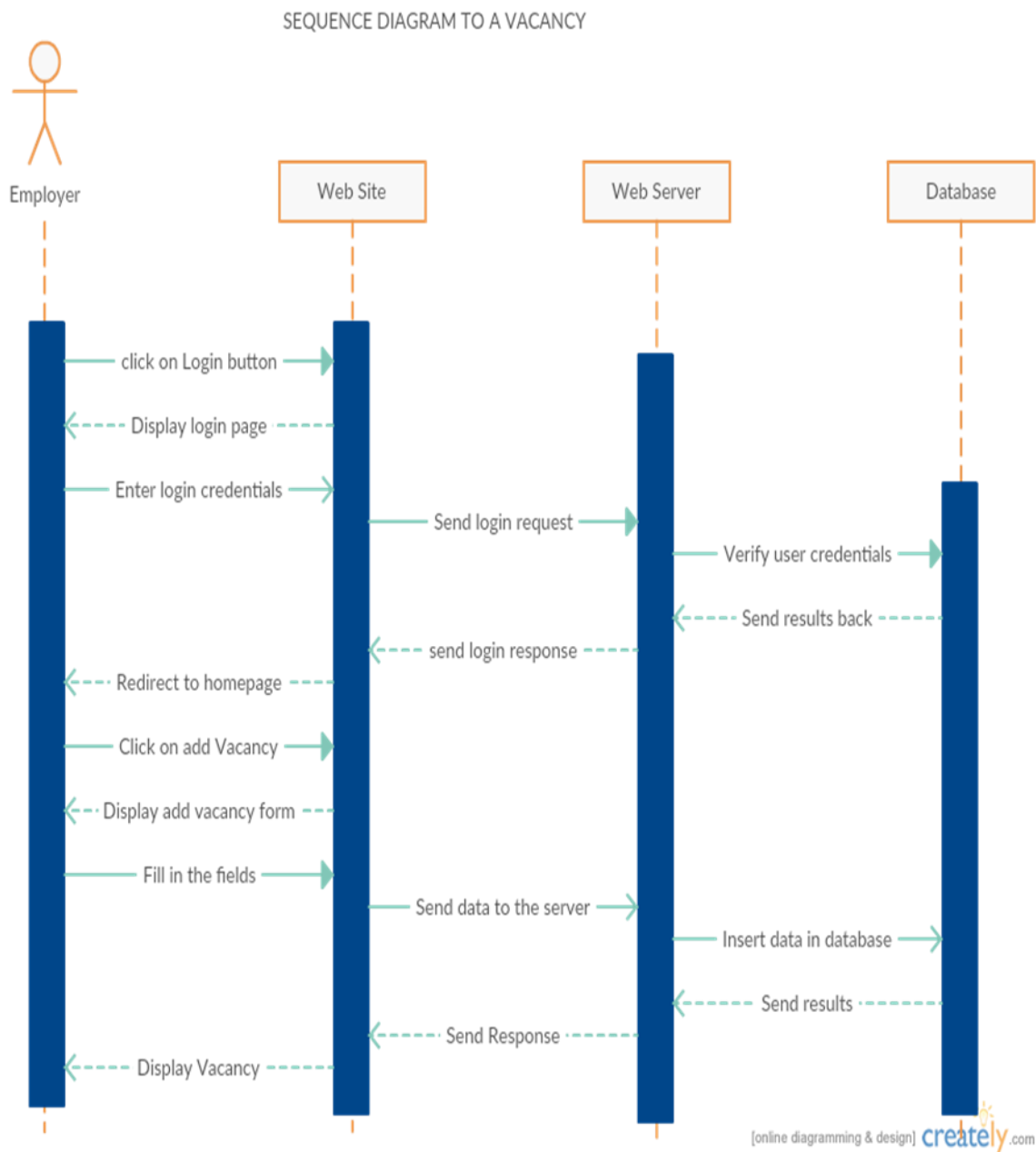


Figure 3.5: Employer Adding a Vacancy

Below is a sequence diagram for a job seeker adding a profile.

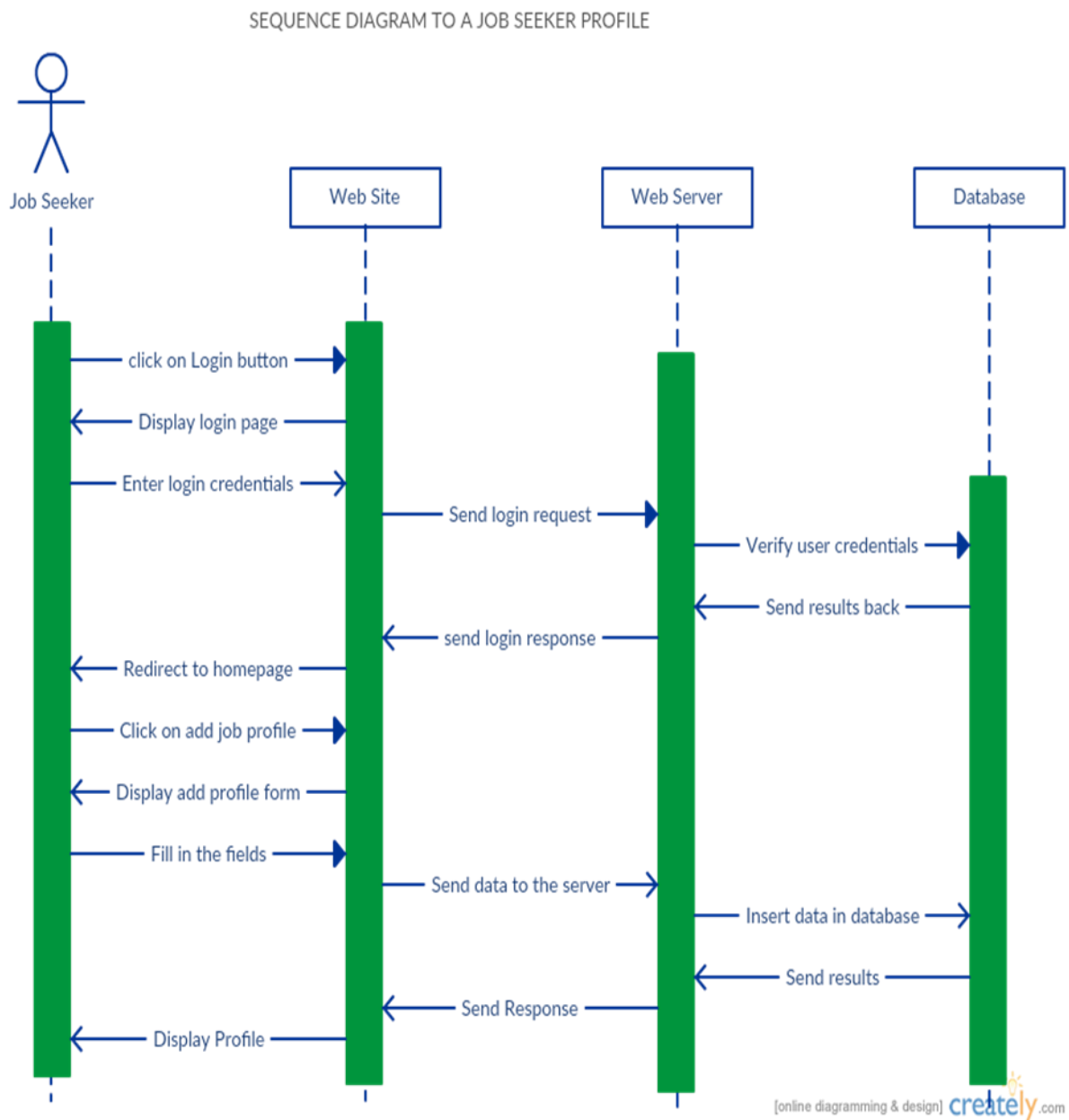


Figure 3.6: Job Seeker Adding Profile

## **CHAPTER 4: IMPLEMENTATION**

### **4.1. Technology Used for Development**

The interface of the application is built on bootstrap framework. This is due to the flexibility and responsiveness of the framework to different devices and screen sizes. Bootstrap framework has a great gridding system which makes it easy to arrange content on the web application. It has base styling for most Hypertext Markup Language elements such as buttons and forms. Another reason is, Bootstrap framework bundles JavaScript plugins. Components such as pop ups, and drop downs are made interactive with the numerous JavaScript plugins in the bootstrap package.

The backend of the application is written in Hypertext Preprocessor (PHP). PHP is a general purpose open source scripting language. PHP was used because it has a global acceptance. Statistically, about 20 million websites and a million web servers are using PHP (d4 Business Solutions, 2017). This has set a high standard on the dependability of PHP language. PHP is cross-platform, meaning it can be used on different operating systems. PHP community also offers great support to developers. This makes it easy to use the language while receiving continual support.

NetBeans and brackets interactive developments (IDEs) was used, due to ease of use and indentation. With Model View Controller (MVC) architecture, code isolation was made possible. This means that a change in one tier would not affect the rest of the code. To store all the data, MySQL database system was used.

### **4.2. Classes and Methods**

The tables below show the backend classes and functions in each class.

#### 4.2.1. Connection Class

This class does the connection to database. It holds the server name, username, password and the database used for the entire application. All the other classes require this class in order to connect to the database.

*Table 4.1: Connection Class*

| <b><i>Method name</i></b> | <b><i>Function</i></b>                    |
|---------------------------|---|
| <i>connect()</i>          | <i>Establishes connection to database</i> |
| <i>query()</i>            | <i>Execute query</i>                      |
| <i>fetch()</i>            | <i>Fetches results of the query</i>       |

#### 4.2.2. User Class

The user class contains methods to validate all users of the application. It does the authentication of users.

*Table 4.2: User class*

| <b><i>Method name</i></b> | <b><i>Function</i></b>  |
|---------------------------|---|
| <i>login()</i>            | <i>Validating the user's username and password to allow logging in.</i> |
| <i>addUser()</i>          | <i>Adding users details into the database during user registration.</i> |
| <i>changePassword()</i>   | <i>Change the user's password given the user's username and email.</i>  |
| <i>allUser()</i>          | <i>Return all users in the user table</i>                               |

|                     |                                       |
|---------------------|---------------------------------------|
| <i>getUser()</i>    | <i>Get a user given a username</i>    |
| <i>deleteUser()</i> | <i>Delete a user given a username</i> |

#### 4.2.3. Vacancy Class

This class contains methods involved in posting and altering vacancies on the application.

*Table 4.3: Vacancy class*

| <b><i>Method name</i></b> | <b><i>Function</i></b>  |
|---------------------------|---|
| <i>addVacancy()</i>       | <i>Add a new vacancy into the vacancy table with all the attributes of a vacancy.</i> |
| <i>editVacancy()</i>      | <i>Edit the properties vacancy stored in the vacancy table.</i>                       |
| <i>allVacancies()</i>     | <i>Return all the vacancies in the database.</i>                                      |
| <i>getVacancy()</i>       | <i>Get a vacancy given the its id</i>   |
| <i>searchVacancy()</i>    | <i>Search through the vacancy table for a specified job title</i>                     |
| <i>deleteVacancy()</i>    | <i>Delete a vacancy from the database</i>   |

#### 4.2.4. Profiles Class

This class contains methods involved in posting and altering seeker profile on the application.

Table 4.4: Job Seeker class

| <b>Method name</b>       | <b>Function</b>  |
|--------------------------|--|
| <i>addJobSeeker()</i>    | <i>Add a profile into the jobseeker table in the database, providing all the attribute of the user adding the profile.</i> |
| <i>editJobSeeker()</i>   | <i>Edit a profile in the database</i>  |
| <i>allJobSeekers()</i>   | <i>Return all the profiles added in the jobseeker table in the database</i>  |
| <i>Search()</i>          | <i>Search through the jobseekers table for a specified job title.</i>  |
| <i>getJobSeeker()</i>    | <i>Get a profile from the database</i>   |
| <i>deleteJobSeeker()</i> | <i>Delete a profile from the database</i>  |

### 4.3. Database Entity Relational Diagram

The application is implemented with MySQL database. There are four main entities namely, user, vacancy, jobseeker and location.

The user table consists of credentials necessary for user registration and logging in. these are; user id, full name, email address, username, password and a timestamp to keep record of when each user registered.

The vacancy table holds the details of a job vacancy. Its attributes are; vacancy id, name of the user posting the vacancy, job title, job description, contact information, location, timestamp of when the vacancy was posted and job status (open or closed).



The jobseeker table holds profiles of people seeking employment. The attributes of this entity are; profile id, job seeker's name, job title, their experience, gender, age, image, address and contact information.

The location table hold geographical coordinates of vacancies and profiles.

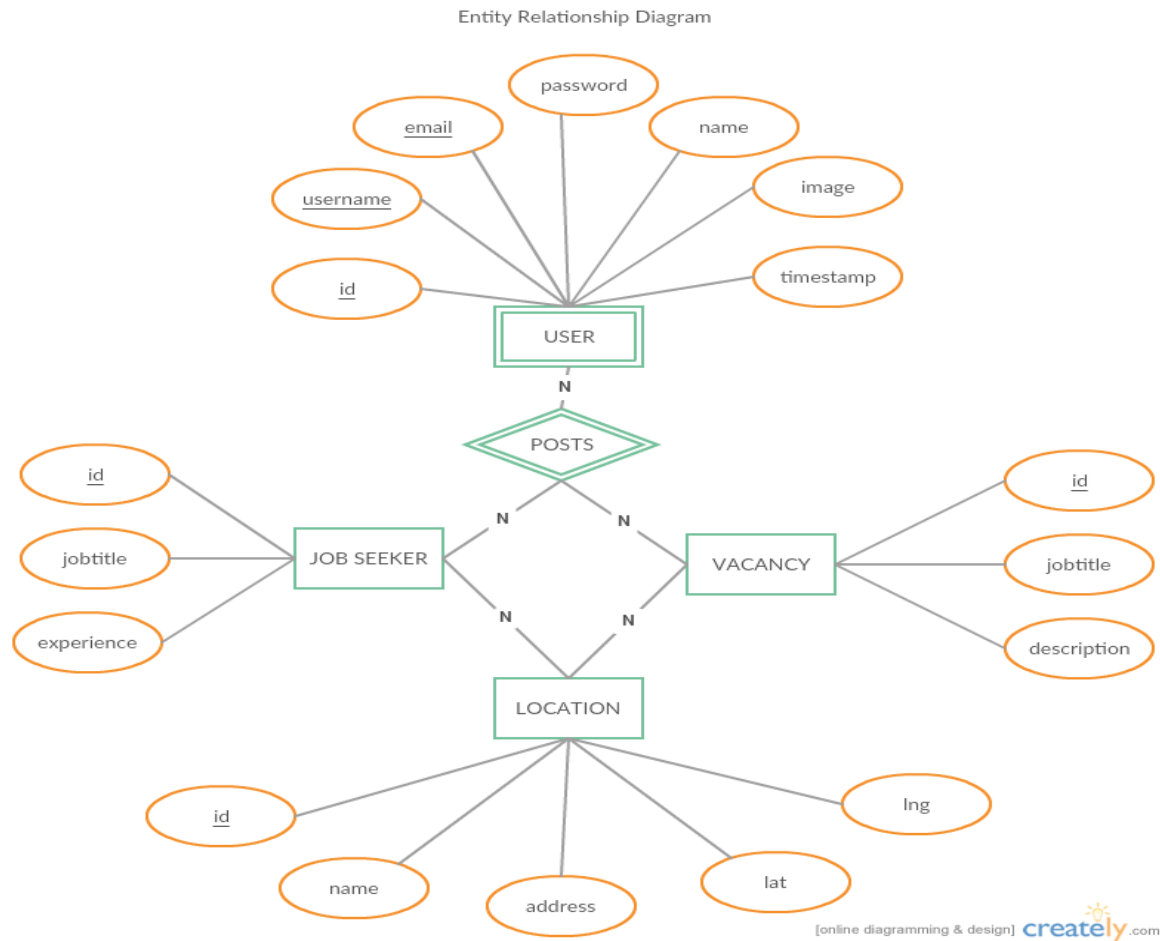


Figure 4.1: ER Diagram for User System

## 4.4.Interfaces

### 4.4.1. Landing Page

The application has an easy to navigate landing page. It provides the user with necessary instructions of what the application is about.



Figure 4.2: Landing page

#### 4.4.2. Navigation Bar

The navigation bar directs the user to various pages/functionalities of the application. It is very responsive depending on the device or screen size. Below is the navigation bar for a wide screen and a narrow screen.

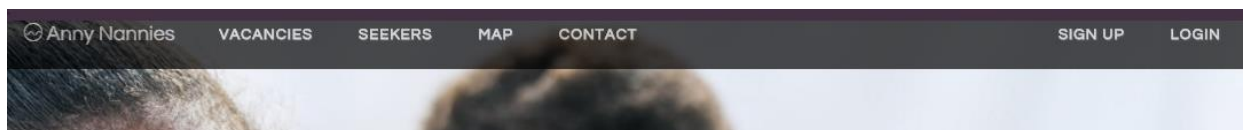
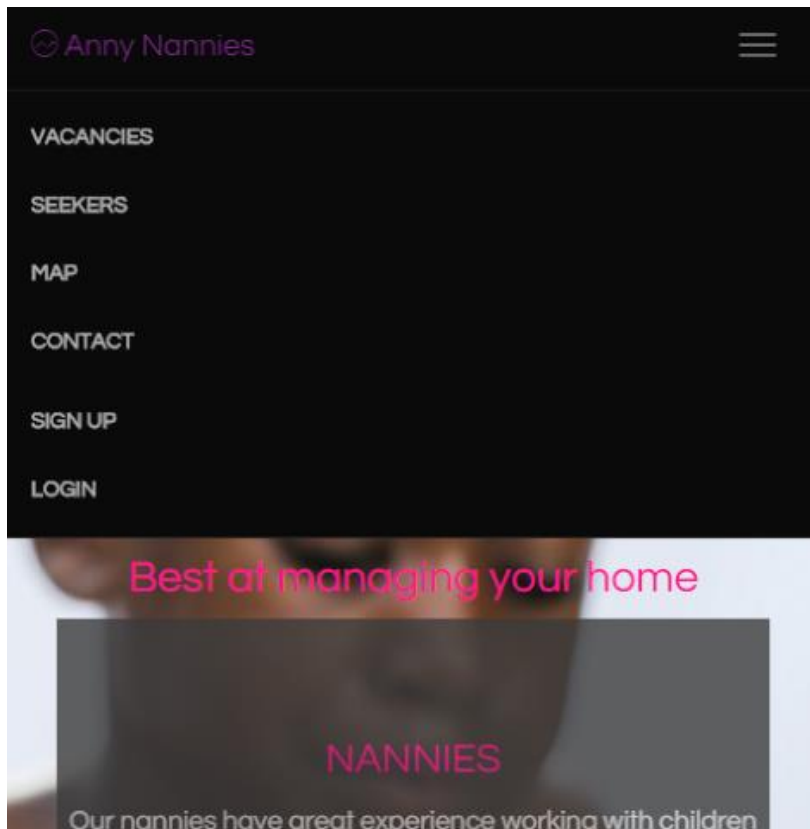


Figure 4.3: Wide Screen navigation bar



*Figure 4.4: Narrow Navigation Bars*

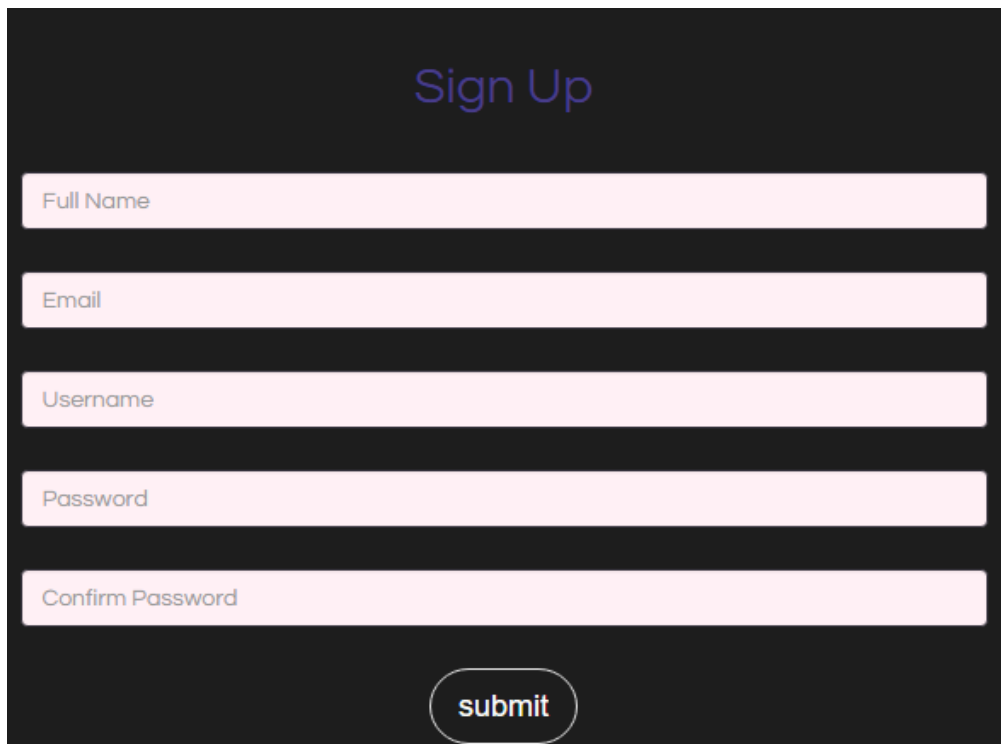
The user can view vacancies, seekers profiles, a map of vacancies and the feedback section.

The user can also click the sign up or login links provided on the navigation bar.

#### **4.4.3. User Sign Up**

The sign-up interface pops up after the sign-up button is clicked. It has fields required to register a new user. Clicking submit button transfers the details entered in the fields to the database.

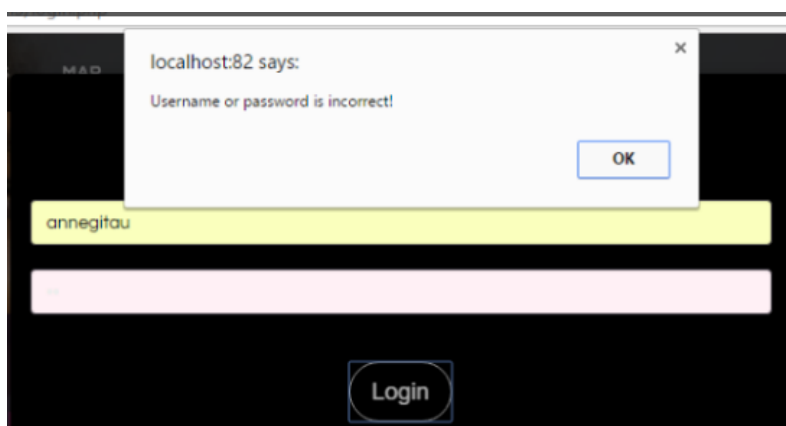
Clicking outside the pop up will automatically close the signup interface.

A dark-themed sign-up form with five light pink input fields. The fields are labeled 'Full Name', 'Email', 'Username', 'Password', and 'Confirm Password'. Below the fields is a rounded rectangular button labeled 'submit'.

*Figure 4.5: Sign Up*

#### **4.4.4. User Login**

The login interface is also a pop up that appears when the login button is clicked. It requires the user to provide correct username and password. Data authentication is done to only allow users with the right credentials.



*Figure 4.6: Login Interface*

## Add Vacancy

A user can add a vacancy only when they have registered and logged into the application. The user can add vacancy by completing (input data) the fields on the interface and pressing the submit button. The data entered on this interface is stored in the vacancy and location tables in the database.

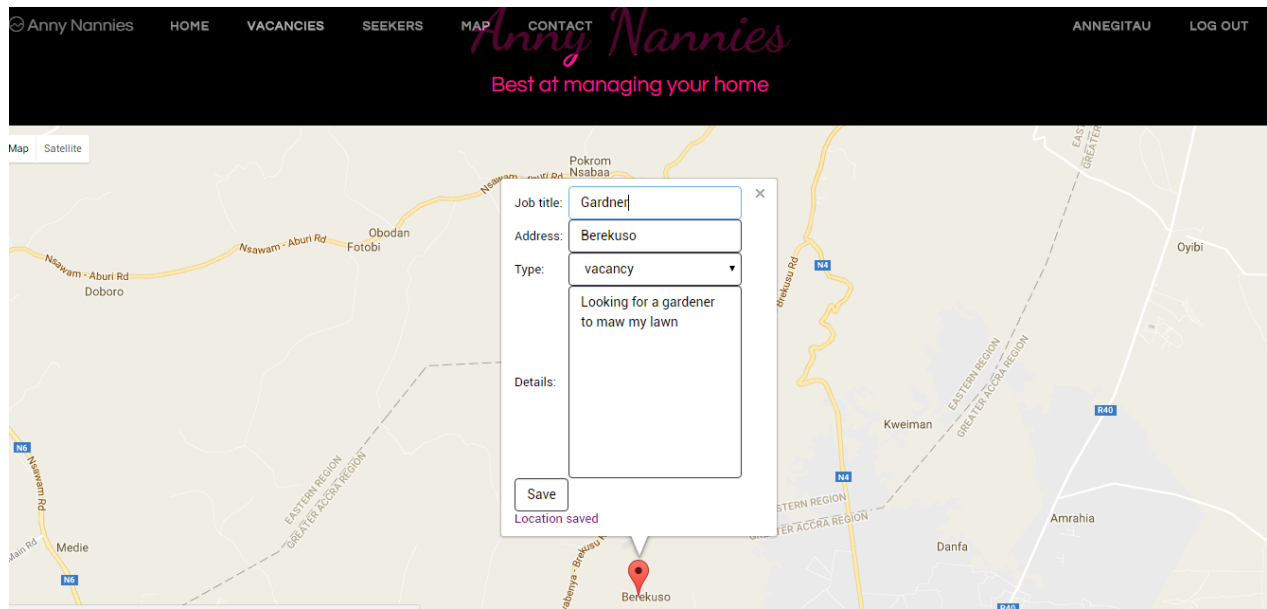


Figure 4.7: Add Vacancy

### 4.4.5. Search Vacancy

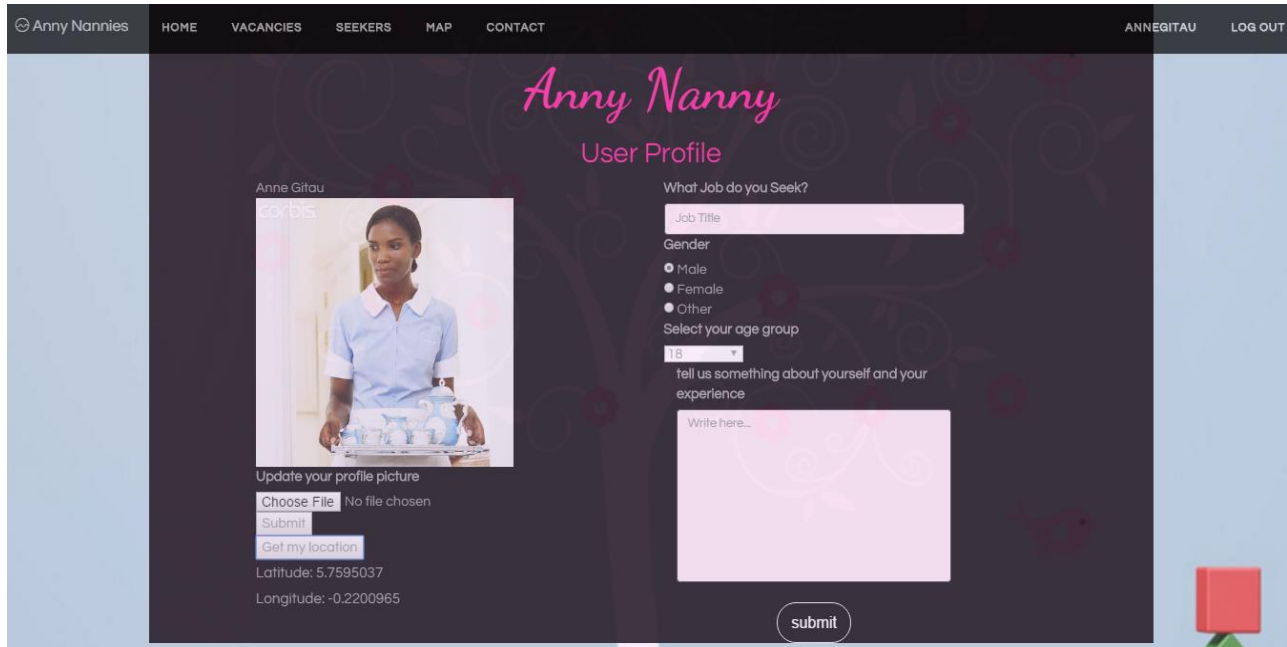
User can search through the available vacancies by providing a job title and pressing

### 4.4.6. View Vacancy

The user can view the available vacancies whether they have signed in.

## Add Profile

The user can add their profile if they are seeking employment. The figure below show the various fields required to add a job profile.



The screenshot shows the 'Anny Nanny' User Profile page. The header includes a logo and navigation links: HOME, VACANCIES, SEEKERS, MAP, CONTACT. The user is logged in as 'ANNEGITAU' and can click 'LOG OUT'. The main heading is 'Anny Nanny User Profile'. On the left, there is a profile picture of a woman in a blue uniform, with the name 'Anne Gitau' and 'carbis' below it. Below the picture is a section to 'Update your profile picture' with a 'Choose File' button (showing 'No file chosen'), a 'Submit' button, and a 'Get my location' button. Below this, the current location is shown: 'Latitude: 5.7595037' and 'Longitude: -0.2200965'. On the right, there is a section titled 'What Job do you Seek?' with a 'Job Title' text input field. Below this are radio buttons for 'Gender' (Male, Female, Other) and a 'Select your age group' dropdown menu (showing '18'). Below these is a text area labeled 'tell us something about yourself and your experience' with a 'Write here...' placeholder. A 'submit' button is at the bottom right of the form.

Figure 4.8: Add Job Profile

### 4.4.7. View Profiles

The user can view the job profiles of job seekers whether they are signed up on the application.

### 4.4.8. Search Job Profile

The user can search through the jobseeker table by providing a job title and pressing search.



Figure 4.9: Search Job Profile

#### 4.4.9. View on Map

The user should be able to view the available vacancies and job profiles on a map. This interface gets details from the location table to plot coordinates of vacancies and job profiles.

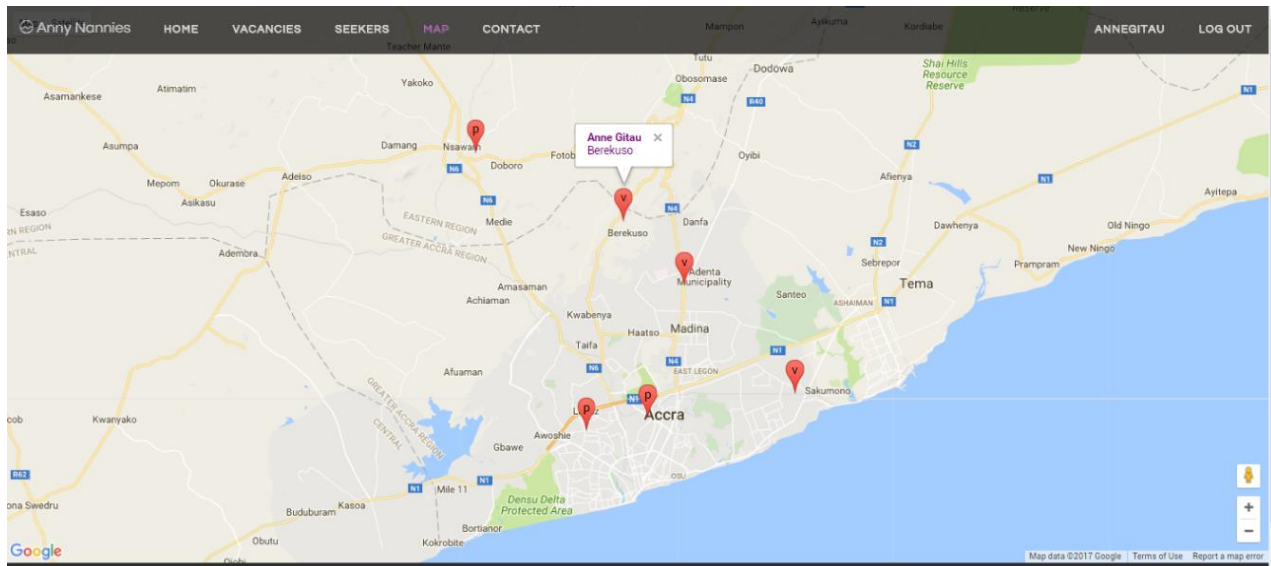


Figure 4.10: Opportunities on Map

## **CHAPTER 5: TESTING**

### **5.1. Introduction**

Testing is the process where the application is tried to see whether it meets the set requirements both functional and non-functional. Testing checks whether the software developed meets functionalities expected and user satisfaction.

There are two major types of testing;

- a. Development Testing
- b. User testing

### **5.2. Development Testing**

This testing is done by the developer of a software with the aim of handling, finding and dealing with bugs found on the system. It is done in three levels namely;

- i. Unit testing
- ii. Component testing
- iii. System testing

#### **5.2.1. Unit Testing**

This involved testing methods in the classes used to develop a software. For this application, PHPUnit test is used to check the functioning of the methods in the various classes. PHPUnit is essential in testing methods of classes written in PHP language. I created a test class called `testClass ()` which extends `PHPUnit_Framework_TestCase` to test the various methods in my classes.



The following figures shows the user class getUser () method and the test class testGetUser(). Added, is a screen shot of the PHPUnit test from command prompt.

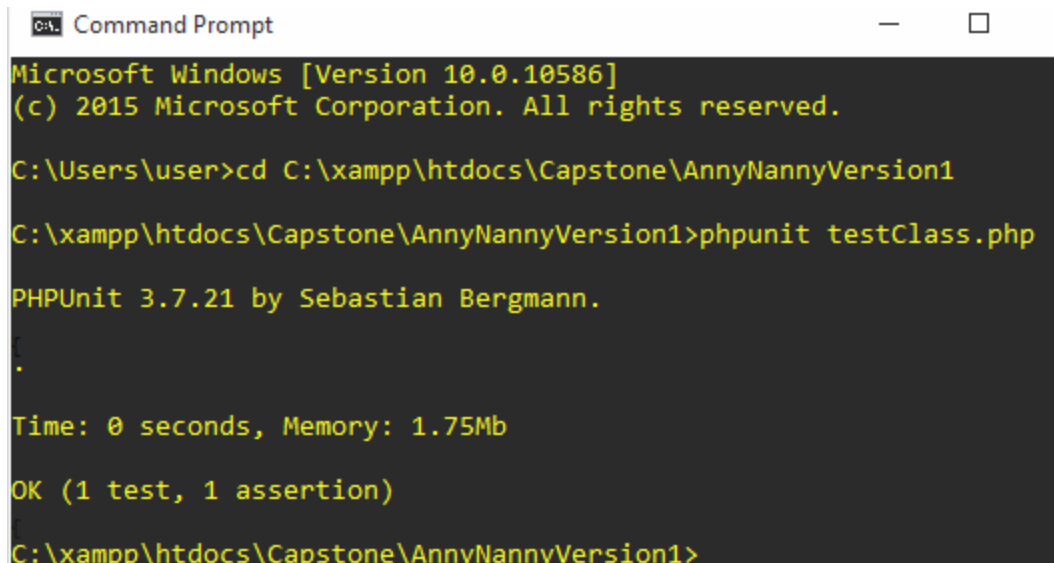
```
<?php
/**
 * @author Anne Gitau
 * @Version 1.0.0
 */
include_once("connection.php");
/**
 * User class
 */
class user extends connection{
    /**
     * @method to get a user from the user table in the database
     */
    function getUser($username){
        $strQuery="select * from user where username='$username'";
        return $this->query($strQuery);
    }
}
```

Figure 5.1: User class getUser () method

```
<?php
/**
 * @author Anne Gitau
 * @Version 1.0.0
 * Test Class for all classes
 */
include "user.php";
class testClass extends PHPUnit_Framework_TestCase{
    /**
     * @method to test the getUser() Method
     * @param no parameters
     * @returns boolean
     */
    public function testGetUser(){
        $anne = new user();
        $username = "annegitau";

        $this->AssertTrue($anne->getUser($));
    }
}
```

Figure 5.2: Test class testGetUser () method



```
CA: Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\user>cd C:\xampp\htdocs\Capstone\AnnyNannyVersion1

C:\xampp\htdocs\Capstone\AnnyNannyVersion1>phpunit testClass.php

PHPUnit 3.7.21 by Sebastian Bergmann.

.

Time: 0 seconds, Memory: 1.75Mb

OK (1 test, 1 assertion)

C:\xampp\htdocs\Capstone\AnnyNannyVersion1>
```

*Figure 5.3: Unit testing results*

### **5.2.2. Component Testing**

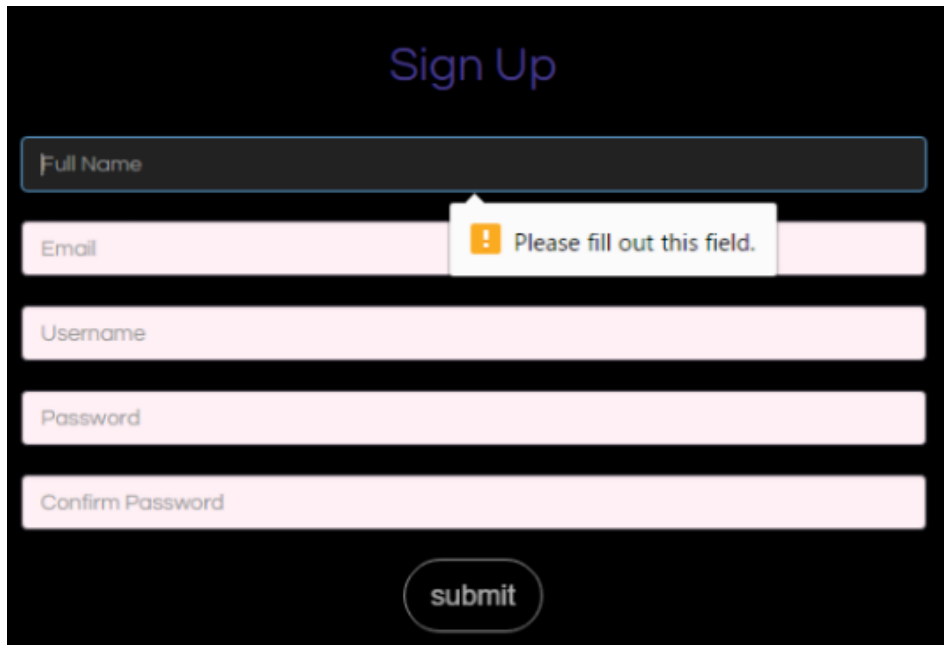
After testing the methods in each object class independently, component testing is then conducted. This is ensuring that the different classes function properly independently to make up a component. The connection class is required for the other classes to function, a bug in this class will therefore affect the other classes.

### **5.2.3. System Testing**

This involves combining the different components of the system that were developed independently. It is focused on making sure that there is easy and smooth interaction between components to ensure the running of the entire system.

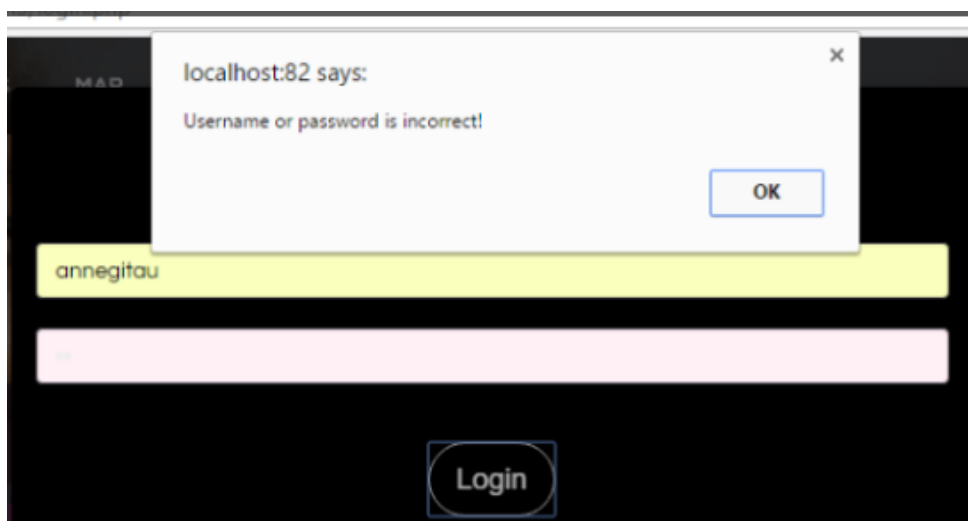
In this job listing application, the user must sign in before they can add a vacancy or a profile online. The user therefore interacts with the user class first before interacting with the vacancy and profile classes.

An example of system test is when signing up, the system cannot sign me up if any field is left empty. Also, when logging, only valid users, validated from the database can log into the system. Below are figures to show user validation.



A screenshot of a 'Sign Up' form on a dark background. The form has five input fields: 'Full Name', 'Email', 'Username', 'Password', and 'Confirm Password'. The 'Email' field is highlighted with a red border, and a red tooltip with an exclamation mark icon and the text 'Please fill out this field.' is pointing to it. Below the fields is a rounded 'submit' button.

Figure 5.4: input validation



A screenshot of a login form on a dark background. The form has two input fields: a yellow 'Username' field containing 'annegitau' and a red 'Password' field. Below the fields is a rounded 'Login' button. A white error dialog box is open in the center, titled 'localhost:82 says:' and containing the message 'Username or password is incorrect!'. The dialog has an 'OK' button.

Figure 5.5: user validation

#### **5.2.4. User testing**

User testing involves allowing a user who is unfamiliar with the system, to test functionalities and satisfaction. The user identifies any loopholes in the system that may have been overlooked during development stage. The purpose of this testing is to get feedback on whether the application meets the specifications.

User testing conducted on this application included 10 random Ashesi University students, to test for feedback.

#### **5.3. User Test Finding**

After conducting user testing, the following came up;

- i. Users were familiar with the navigation bar and how it changes depending on the screen size.
- ii. Buttons were clear and well labelled making it easy to navigate through the application.
- iii. The Map did not load for most of the users due to poor internet connection.
- iv. The marker showing the location of the user did not show on the map unless the user refreshed the application.
- v. Most users found it difficult to terminate sign up process, since there is no cancel button. Four users later realized that clicking outside the signup pop up would exit them automatically.

## **CHAPTER 6: CONCLUSION AND RECOMMENDATION**

### **6.1. Conclusion**

Inspiration to take on this project came after observing many working mothers going through a lot of trouble to find suitable people to take care of their homes whilst at work. The most moving moment was when one of them had to let go of a job promotion which involved travelling, because she did not have anyone to take care of her children when she is away.

The project explored strengths in coding and project management skills. Extensive research was conducted on the different way of implementing functionalities.

Courses such as Mobile Web Development, Web Technologies, Software Engineering and Database Management Systems offered in Ashesi University also played a big role in executing this project.

The application met most of the requirements set in the earlier chapters. Including: signing up, logging in, adding a vacancy or a profile, viewing these opportunities on a map and contacting the person of interest.

### **6.2.Recommendation and Future Works**

This application can be improved by adding the following functionalities.

- i. Implementing the administrator dashboard –having a working administrator dashboard will be helpful in manipulating the data in the application by an authorized administrator. The administrator should be able to perform all function on the application including deleting, editing viewing and adding posts.
- ii. Applying on the map info window—this will allow a user to connect to a job seeker or an employer by clicking on markers on a map.

- iii. A validation summary during signup—having a summary of the requirements of the different field during signup will help to inform the user what is expected of them. This will include, password length and format.
- iv. Checking login attempts—checking and limiting the number of login attempts will help in keeping unauthorized users at bay.

## REFERENCES

- Appan, P. (2016, April 25). *Algorithms and architecture for job recommendations*. Retrieved from O'Reilly: <https://www.oreilly.com/ideas/algorithms-and-architecture-for-job-recommendations>
- Cabinet Secretary Ministry of Planning and Devolution. (2012). *2015 Economic Survey Report Highlights*. Nairobi.
- d4 Business Solutions. (2017, April 14). *Our Technologies*. Retrieved from d4 Business Solutions Ltd : <http://www.d4solutions.com/index.php/our-technologies-side-menu>
- Giang, V. (2013, 10 25). *Why New Hiring Algorithms Are More Efficient — Even If They Filter Out Qualified Candidates*. Retrieved from Business Insider: <http://www.businessinsider.com/why-its-ok-that-employers-filter-out-qualified-candidates-2013-10>
- Kenya national Bureau of Statistics. (2014). *Student Enrollment in Public Universities: 2007-08 to 2012-13*. Nairobi: Humanitarian Data Exchange.
- World Bank. (2009). *Youth and Employment in Africa: The Potential, the Problem, the Promise*. Washington, DC: World Bank.