WOLKITE UNIVERSITY

COLLEGE OF COPMUTING AND INFORMATICS

DEPARTEMENT OF INFORMATION TECHNOLOGY

INDUSTRIAL PROJECT TITLE: WEB BASED JOB SEARCH ENGINE FOR ETHIOPIA

Prepared By: -

GROUP MEMBER	ID
1. ANDUALEM ADDIS	CIR/215/11
2. KETEMAW BAHIRU	CIR/256/11
3. GETASEW LIJALEM	CIR/246/11

Advisor: **Mr. Asifaw.K**

Wolkite University, Wolkite, Ethiopia

June, 8, 2014 E.C

WOLKITE UNIVERSITY

COLLEGE OF COPMUTING AND INFORMATICS

DEPARTEMENT OF INFORMATION TECHNOLOGY

DOCUMENTATION PHASE ONE

TITLE: WEB BASED JOB SEARCH ENGINE FOR ETHIOPIA

Prepared By: -

GROUP MEMBER	ID
1. ANDUALEM ADDIS	CIR/215/11
2. KETEMAW BAHIRU	CIR/256/11
3. GETASEW LIJALEM	CIR/246/11

Advisor: Mr. Asifaw.K

Wolkite University, Wolkite, Ethiopia

June, 8, 2014 E.C

DECLARATION

This is to declare that this project work which is done under the supervision of Asifaw Kiros (MSc.) and having the title WEB BASED JOB SEARCH ENGINE FOR ETHIOPIA the sole contribution of Andualem Addis, Ketemaw Bahiru, and Getasew Lijalem. No part of the project work has been reproduced illegally (copy and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: _____

Group Member:

Student Name:	Signature
1. Andualem Addis	_____
2. Ketemaw Bahiru	_____
3. Getasew Lijalem	_____

APPROVAL SHEET

This is certified that the project entitled WEB BASED JOB SEARCH ENGINE FOR ETHIOPIA will be the documentation is written by:

Name	Signature
1. Andualem Addis	_____
2. Ketemaw Bahiru	_____
3. Getasew Lijalem	_____

A Project Proposal Submitted to the Department of Information Technology of Wolkite University in Partial Fulfilment of Requirements for the Degree of Bachelor of Science in Information Technology

Advisors' Name	Signature	Date
MR. Asfaw Kiros (MSC)	_____	_____

Department Head Name	Signature	Date
_____	_____	_____

Examiner Name	Signature	Date
1, _____	_____	_____
2, _____	_____	_____
3, _____	_____	_____

ACKNOWLEDGMENT

We thank first of almighty God perseverance that help us to reach from start to know. And we have special thanks to Mr. Asfaw Kiros our advisor who guide us to prepare the proposal to our final project of first phase of project. Our cooperation brings us to come up with project title of Development of Job Search Engine for Ethiopian Job Search, we have big thanks our final project proposal for our team for to select title which solve the problem of society as well as country. And we thank Wolkite CCI collage that programmed this final project schedule curriculum to master our knowledge scope of project developing in the case of educational purpose as well as all in our real-life scenario. This program enhances the student scope of solving the problem exists in society as well as world-wide which leads the society to promote the technological globalization to habituative in broad usage and expansion in such like expansion through. So, our acknowledgement is not this much limited we have great thanks.

TABLE OF CONTENT

LIST OF FIGURE

LIST OF TABLE

LIST OF ABRIVATION

SQL……..….structured query language

UML…… …….unified modelling language

CD……..……compact disk

OOSAD…….…..…object-oriented system analysis design

PHP…… ……..……...…….pre-processor hypertext programming

CSS ……..….cascade style sheet

PHP……..pre-processor hypertext programming

HTML…….....…hypertext mark-up language

OOP……..object-oriented programming

OOA……..…object-oriented analysis

OOD …….…….object-oriented design

USB……....universal serial bus

XAMPP…….cross-platform apache MySQL php and Perl

DBMS…….……database management system

GUI…….. …..Graphical user interface

JS……..….Java script

CHAPTER ONE

1. Introduction

A search engine is a web-based software programmed tool that enables users to locate information on the World Wide Web. Search engines send out “web crawlers” or “spiders” (automated computer programs that browse the internet in a methodical and automated manner) to create a copy of all the web pages they have been to so the search engine can then index the pages to create web site listings that facilitate faster searches. A user types a query into the search engine and the search engine then sorts through millions of pages in its database to find a match to that specific query. The search engine then produces the results to your query in a ranked order according to relevancy. Popular examples of search engines are Google, Yahoo!, and MSN Search. Search engines utilize automated software applications (referred to as robots, bots, or spiders) that travel along the Web, following links from page to page, site to site. The information gathered by the spiders is used to create a searchable index of the Web.

Search engines consist three of main parts. Search engine spiders follow links on the web to request pages that are either not yet indexed or have been updated since they were last indexed. These pages are crawled and are added to the search engine index (also known as the catalogue). When you search using a major search engine you are not actually searching the web, but are searching a slightly outdated index of content which roughly represents the content of the web. The third part of a search engine is the search interface and relevancy software. While nobody can guarantee top level positioning in search engine organic results, proper search engine can help. Because the search engines, such as Google, Yahoo!, and Bing, are so important today it is necessary to make each page in a web site conform to the principles of good search engine as much as possible.

It can be big confusing to tell the difference between a job search site and a job search engine, but it's wise to know where the job listings you find online are coming from. A job search site is a website which posts jobs supplied by employers, whereas job search engines scour the web and aggregate job listings from job search site and employer websites. You will find a wider variety of job postings on job search engine

1.1 Background of the Project

A job search engine is a single site where hundreds or even thousands of job postings are published available for job seekers to search. Job search engine however, pull jobs from differentplaces around the web including career sites, employer job listings and other jobs. Because there are thousands of individual job boards it is hard for a job seeker to effectively search for all job openings. Job search engine operate similarly to search engines like Yahoo, Bing and Google pulling job postings from multiple locations. Job search engine are important because searching for a job is time consuming and without an engine requires you to search and locate multiple jobboard for the same job title you require. It's important to note that job boards have suffered asof recent years due to search engine robots penalizing sites that publish content that has also beenpublished at different places across the web which includes job posts.

This is one of the main reasons that job boards have flocked to job search engine sites like those you see below as a qualified candidate traffic source. This is also the reason that job search engine are excited as it offers a ways for jobs from job boards and aggregators to be featured different in search results and thus, improve the likelihood that job seekers will click, select and apply for job postings from them versus traditional career sites. There have been a lot of rumoursover the years of Google entering the job search engine industry and this announcement puts some of those rumours to rest at least temporarily.

1.2 Statement of the Problem

Currently, most of job seekers manually viewing available jobs, or applying for the job at the agency can be reading different Newspapers, job announcements posts on notice board. This way of finding job consumes, time and budget of job seekers and Employers. And also it difficult to get qualified candidate for the companies. Done for which job seekers have to go the agency and check the available jobs at the agency by going to the notice board of agency. And also now a day the company or organization needs a human resource or an employee for announcements and distributing of paper on notice board. And the Employers have workload to enrol the new job seekers (enrolment of job seekers is tedious). And the current job search site provides the job seeker only the information which posts jobs supplied by employers and has to user specific websites to search the job. To solve the above problems, we will develop Job Search Engine Application

1.3 Objectives

1.3.1 General Objective

The general objective of this project is to develop job search engine applications for Job seeker and employees.

1.3.2 Specific Objective

In order to achieve the above general objective, we:

- Design the architecture for the proposed system
- Develop user friendly and interactive system
- To solve the problem of the current system of manual
- To provide user friendly system for stakeholder

1.4 Feasibility Study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. It lets the developer predict the future of the project and the usefulness. By using the gathered information on the requirements for job search engine we do feasibility study from different perspectives. We can see from its technical, operational, time and economical points of view

1.4.1 Economic Feasibility

Our system is economically feasible since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development. The Job seekers also need not to waste their money as travel expenditure to find the jobs. So, the proposed system is economically feasible.

1.4.2 Operational Feasibility

Our system is operational feasible because of that it satisfies the users need and it also offers accurate and efficient system to the users. As well the developed system is simple to interact to user, user friendly with GUI, no more complexes to the first-time user because graphical based interface guided.so that we concluded that our system is operational feasible.

1.4.3. Legal feasibility

The users, so that the project is legally feasible. A project may face legal issues after completionif this factor is not considered at the first stage such as cyber security problems.

1.4.4 Technical Feasibility

Our system is technically feasible since. It is possible to access one user by electronic devices ifthe user has access privilege. In the job search engine job seekers has all the privilege on their account. They can login to the system remotely by using their electronic device that can access internet and access their account. That goes to the proposed system is technically feasible.

We have technical knowledge about:

- PHP to write the code or implementation with XAMPP server.
- SQL to build the database to store the data.
- Unified Modelling Language (UML) model to do analysing and designing in good manner.
- Training on the new system to know how it operates and how to use the computerized system

1.4.5 Time Feasibility

The time given for this project is around six months. First three months for design andthe next three months are for implementation of the designed. We try to finish our designwithin given time. To finish we have schedule based on department schedule, this also help us to finish the design on time give us. According to the data we gathered from the interviews and also from the observation we have concluded that the application istimely feasible.

1.5 Scope and Limitations of Project

1.5.1 Scope of the Project

The scope of our project is that the system works only for applicant who has any electronic device that can access internet. our project or system includes to provide job postings, provide job searching and apply for the job and to maintain job posting details and as well generate various reports about job related, and also provide new job post by job title/by company name/by location and also provide for job search for newly posted and provide to the company to select based on the area of specialization of job seeker.

1.5.2 Limitation of the Project

Our project will have the following limitation: -

- When the job seeker applies for job he/she might uploads wrong/fake CV and information so our system will not control this type of activity. Because of this we cannot include this type of system recognition.

1.6 Significance of the Project

The main purpose of our project is to develop a job search engine application which enables the job seekers to apply for jobs that match their qualifications in an easy, cost effective and timely manner. That means it enables the job seekers to search for jobs through their electronic device anywhere any time. And also, employer can select qualified candidates in an easy, cost effective and timely manner.

1.6.1 Target Beneficiaries of the System

Different parts are going to be benefited from the new system such as:

- Job seekers: - benefited from the proposed system by searching the latest job on his electronic device and decrease his tired of finding posted job on notice board. And minimizes the cost and time of the jobseeker.
- Employer: - save wastage of time to place a paper of latest job on notice board every time and select qualified applicant easily.
- Administrator: - admin can easily manage and control all over the system.

1.7 Benefit (Advantages) Of the Project

It is project benefit that can convert into monetary values. For this project, we have identifiedthe following

Tangible Benefits

- Decrease in company cost
- Reduced Stationary Cost
- Reduction of error.
- The system can facilitate the search of job vacancies
- Reduce the time spend of searching job in different broadsheet and announcement board

Intangible Benefits: -

It is our project benefit that cannot convert into monetary values.

- Enhance reliability
- Increasing the competitiveness of the individual
- More timely, updated, and accurate information
- Better job search finding system
- Improved productivity
- Facilitating information processing of our team
- Faster decision making on the team member.
- Enhance job seeker satisfaction
- Provide secure and reliable service
- Provide precise information.
- Progress job seeker service.

1.8 Methodology of the Project

1.8.1 Requirement Gathering Methods

To develop our application the primary task is understand more about the current Job search System being used to find jobs; we gathered different information from web-based job search applications like and also, we have gathered data using the following techniques.

- Interview: job seekers need fast and reliable service from companies and also the companies need qualified employee. Most job seekers complain that they can't get the job services when that they can't get the job services when they want because of the distance, financial issue, time, effort and lack of information. The system we need to develop will remove all these problems of the job seekers and companies. This interview we find from job seeker in Wolkite university community where they are searching to read the notice board.
- Observation: It is observed most job seekers apply and view job at the agency can be done for which job seekers has to go to the agency and check the available jobs at the agency by reading different Newspapers, job announcements posts on notice board. Generally, job

search engine will remove all this problem of the job seekers and employers.

1.8.2 System Analysis and Design

The method of the system development can be done through OOSAD (Object oriented system analysis and development) during the whole project life cycle. In our project, we will apply the concept of object-oriented system development methodology which categorized in to two phases. These phases are object-oriented analysis and object-oriented design. It increases consistency among analyses, designer implementation and testing. It also allows the reusability of the code which will help to enhance the project in the future.

We consider following object-oriented system has many benefits over structured approach:

- It is easier to develop and maintain.
- It is re usability, extensibility, improves quality, maintainability and manages complexity.

The transition from Object Oriented Analysis (OOA) to Object Oriented Design (OOD) can be done easily because of OOA is resilient to changes as objects are more stable. In general, we will use this object-oriented methodology for the following purposes; Simplicity, Maintainable,Faster Development, Increased Quality.

1.8.3 System Development Model

To develop job-related search engine for Ethiopian we prefer Iterative software development life cycle because of the dynamic and complexity of the requirements of the system. The reasonwe are selecting this life cycle model is:

- It enables us to add a new feature to the system.
- It allows us to find functional or design related flaws as early as possible.
- When we assume a new idea about the system it enables us to add it at the early stage

1.8.4 Development Tools and Technologies

- Software Tools

Microsoft office word 2010: - it is very useful because it takes less time to write and format the text, communicative effectively smart diagram and chart tools, quickly assemble document. By looking its useful properties, we use Microsoft office word to type our project work to get all the above benefits of it.

Power point 2010: -use to present the document in abstract forms. We use it to present our presentation in short and brief way.

EdrawMax application for UML diagrams (for use case diagram, sequence diagram, class diagram, Activity diagram, State chart diagram, component and deployment diagram).

SQL (Structured Query Language): - A language used to interrogate and process data in arelational database. All database systems designed for client/server environments support SQL. SQL commands can be used to interactively work with a database or can be embedded within a programming language to interface to a database. Programming extensions to SQL have turned it into a full-blown database programming language, and all major database management systems support the language. We use PHP, CSS, and java script programming language to develop our system

Hardware Tools

Different hardware used to develop our project

- Computer: -computer is a machine capable of doing many things. We use it to type on it and install all software and programming language. Almost all tasks are done on computer.
- Flash Disk and CD Hardware: - used for the movement of data from one machine to another. We use both of them when we move our data from one machine to another.
- Printer: - to print the document

Front-End Technologies

- HTML: to define the content of web pages
- CSS: to specify the layout of web pages

- JavaScript: to program the behaviour of web pages
- Bootstrap: to develop responsive website

Back-end technologies

- Apache Server: - to compile the sever side scripting language
- SQL DBMS: - server to compile SQL queries and store data
- PHP: - for server-side scripting language

1.9 Budget and Time Schedule of the Project

- ✓ Budget of the Project

Table 1.1: Budget of the Project

Material	Amount	Price(birr)
Hard disk	3	By university
Flash disk (16 GB)	2	400
laptop	1	20,000
Desktop	3	By university
pen	3	30
paper (printing)	Minimum 100 pages	200
Total	……	20,630

1.10 Time Schedule of the Project

Determine how the proposed system accomplished with the given time table. It implies effective time management for the system, and the project should have finished within deadline. So, the team decides to implement and configure the new system on time without any delay.

Table 1.2 :Time schedule of the project

No	Phases						
		20thDec_ 15thJan	20thJan_ 15thFeb	20thFeb_ 18thMar	23thMar_ 27thMay	27thmay_ 5thJun	9thJun
1	Requirement gathering						
2	Requirement analysis						
3	Design						
5	Implementation						
6	Project closure						

1.11 Document Organization

Our project consists of six chapters. Chapter one give description on the project we are working on, that is Job Related Search engine. The main objective of this chapter is to give an introductionabout the system like its scope, objectives, methodologies, etc…. Chapter two focuses more about the system requirement specification or analysis. These chapter includes Description of Existing System, business rules, functional and non-functional requirements, use case diagram, use case documentation, state chart diagram, sequence diagram. In the third chapter the design phase of the proposed system is explained, control and security. Chapter four is about the system design like conceptual modelling, deployment and employment diagram, database design, access. In the fifth chapter deals in testing of the system, Hardware software acquisition, and Installation process of the system. The last sixth chapter about in Conclusions and Recommendations of the project.

1.12 Team Composition

Table 1.3:Team Composition

No	Group member name	Roles in this project
1	Getasew Lijalem	Requirement gathering
2	Ketemaw Bahiru	System Analyzer
3	Andualem Addis	System Designer
4	All together	Testing and Implementation

CHAPTER TWO

2. Description of Existing System

2.1 Introduction of Existing System

Currently, most of job seekers viewing available jobs, or applying for the job at the agency can be done for which job seekers has to go to the agency and check the available jobs at the agency by reading different Newspapers, job announcements posts on notice board. This way of finding job consumes time and finance job seekers and Employers. And also it is difficult to get qualified candidate for the companies. In addition to this, job seekers view and apply for jobs through distinct job search website. But, these sites require extra steps to be accessed.

2.2 Users of Existing System

Current system encompasses different players (actors) to carry out the whole activities.

- Company owner: the person who can have a right to control the employer over all work.
- Employee: is a person who can have the right to post update and edit profile and a person who can place a vacancy paper on notice board in case of manual system.
- Job seeker: a person who can get information about a latest job from notice board

2.3 Major Functions of the Existing System

- Job seekers can apply job, he/she can go to organization
- Job seekers have to go to the agency and check the available jobs at the agency by reading different Newspapers, job announcements posts on notice board.
- Employer can post a job posting a job on notice board.
- Employer can enroll s job seeker putting their full information on a piece of paper.

2.4 Drawbacks of the Existing System

Due to the manual system of job search the job is posted in the notice board by printing on paper and when they want to update the print again in returned manually through long steps.

- Searching and getting available vacancies on notice board are takes long time for job seeker.
- In case of job search website like ethiojob and job search system the job seeker must needs the distinct website in order to use the system. In general, it has no a user-friendly

interface to the user (job seeker) and the pages are difficult to use easily mostly manual essence.

Information problem: It explains the information contents of the existing system in terms of input, process, output and data storage.

2.4.1 Input

- Due to the manually inserting and printing of data it may face to error
- Loss of data on the notice board may occur by weather condition, and by some ugly people.

2.4.2 Output

- Inaccuracy information may be produced
- Poor flow of information between the job seeker and the company.

2.4.3 Data Storage

- Lack of well-organized database
- Data are not easily accessible due to its integration placed in different location
- Difficult to change and edit
- Data redundancy that leads to inconsistency.

2.4.4 Security

- Due to the manual work the input data have no any security and it may erased from the notice board easily.

2.4.5 Efficiency

- The manual system is not efficient to select the appropriate applicant.

2.5 Business Rules of the Existing System

The existing business rule of job search method has its own set of rules and regulations between the job seeker and the company. This rule and regulations must be fulfilled by job seekers to search, view and apply for job posted by the company.

BR1: The job seeker must be free from charges

BR2: The job seeker has been age between 18 and 65

BR3: The Company should be legally licensed

BR4: job seeker must be fulfilled correct criteria

BR5: a job seeker wants to hire another company; he must step down from company

BR6: the job seeker must be fill correct form of his/her personal data

BR7: Employer must post announcement on the website

CHAPTER THREE

3 Proposed System

The job search engine System is web-based application for job seekers. This application contains server containing the database of job seeker and employer; client containing GUI (graphical user interface). And our system will also have detailed information about the job (job search, latest job, categories of jobs). This system enables job seekers to search and view details of a specific job and applies for jobs posted from different employers and will generate and forward announcement and vacancy notifications in return. It wills also shortlist qualified candidates who fulfil the criteria required by the employers.

3.1 Functional Requirements

Functional Requirements are those that refer to the functionality of the system, i.e. what services it will provide to the user. Statements of services the system should provide how the system should react to particular inputs and how the system should behave in particular situations.

- Enable job Seeker and employer to register.
- Enable the job seeker, employer, administrator to login and logout.
- Enable the job seeker to view and update his/her profile.
- Enable the job seeker to receive and view job notification.
- Enable the job seeker to view, post, update his/her resume.
- Enable job seeker to search job, view job detail, apply for job, View latest job, view jobs based on their category.
- Enable the administrator to view employer detail.
- Enable the administrator to approve and reject the employers.
- Enable the employer to post, edit and delete Job.
- Enable the employer to view selected applicant and applicant's detail.
- Enable the employer to View resume.
- Enable employer to update profile

3.2 Non -Functional Requirements

3.2.1 User Interface and Human Factors

The system should have easily understandable interface (users can interact with the system through the user interface easily

A digital product with a clear understanding of its target users can achieve higher user retention and engagement. This feature caters to a smooth user journey through different interface elements. If the latter is consistent, a user will find the needed option fast, therefore the whole experience will be successful and customers will develop usage patterns quickly. Users of the system-involved people with different background, the system should user-friendly window type.

3.2.2 Hardware Consideration

For the development of the system we used the following hardware:

- Printer: For printing documentation
- Highest processor speed and latest CPU
- 4GB RAM
- Hard disc 700MB
- Flash Disk: - used for the movement of data from one machine to another.

3.2.3 Security Issues

- Security becomes crucial issue in the proposed system. A user must login to the system with user name and password. The system should allow login to only authorized users. I.e. users that have previously created account through user name (E-Mail) and password
- The system has two groups of users: The Admin and Limited users (job seekers and employers). The Admin user has full privilege to perform on the system. Whereas limited users in the department can only perform limited operations based on the privilege given by the administrator.
- User must be register before getting user name and password

3.2.4 Performance Consideration

- Our system is going to use efficient way for each task which will make it fast and require less storage.

- Our system is easily accessible to new user
- When companies post new jobs job seekers can found that new job easily and fast way access

3.2.5 Quality Issues

The system, it will be tested, verified and, validated and if the error occurs during the execution time it will give an exception. And, because the user will be involved in testing the quality of the system, it will satisfy the user's needs.

- **Availability**: the system should have to be function at any given time. Which means our proposed system gives service 24 hours per day with maximum response time. The server should be always on to be available.
- **Reliability**: the proposed system will minimize crash during its runtime, since more than

One user could use the system simultaneously. The system should be reliable and matured enough in giving its service.

3.2.6 Error Handling and validation

Our system handles error by showing the message "invalid input "when the user enters invalid input. Generally, if an error occurs, the system will identify the error and notify the user so that he/she can take the appropriate corrections rather than terminating the system and the system must handle the error.

3.3.7 Backup and recovery

The proposed system will have two databases for recovery and backup. One for a copy of the original database used for a backup purpose that is used for recovery if the original database fails.

3.3.8. Physical Environment

The system must be compatible with any environment. But for more feature we recommend that the system to deployed on the cloud that is free from any disaster.

3.3.9. Resource Issues

We develop the system to minimize wastage of resource like reducing human power. In the future, the system use computer, flash, server etc.

3.3.10. Documentation

The new system required full documentation, help contents and tips to allow further maintainability and to support and guide job seekers how to use the new systems.

CHAPTER FOUR

4. System Analysis

4.1 Introduction

As mentioned in the first chapter, in this project, the team used an object-oriented system development methodology. In this chapter the major activities performed or identified are: modelling the function of the use case, identifying actors, identifying use case, constructing use case model and use case scenarios and final designing of user interface.

System modelling involves the evaluation of system components in relationship with one another to determine their requirements and how to satisfy them. Some system modelling tools will be employed during the course of this project that will support development tasks, from analysis to design, then to implementation. This will be represented with the use of the sequence diagram, activity diagram, state chart diagram, collaboration diagram and class diagram for the job search engine.

Primary Goals of Search Engines

- Effectiveness (quality): to retrieve the most relevant set of documents for a query Process text and store text statistics to improve relevance in our case it retrieves the relevant of job vacancies from World Wide Web, Efficiency (speed): process queries from users as fast as possible. Specific goals usually fall into the above primary goals Example: handling changing document collections – both an effectiveness issue and an efficiency issue.

How do search engines work?

Crawling is the discovery process in which search engines send out a team of robots (known as crawlers or spiders) to find new and updated content

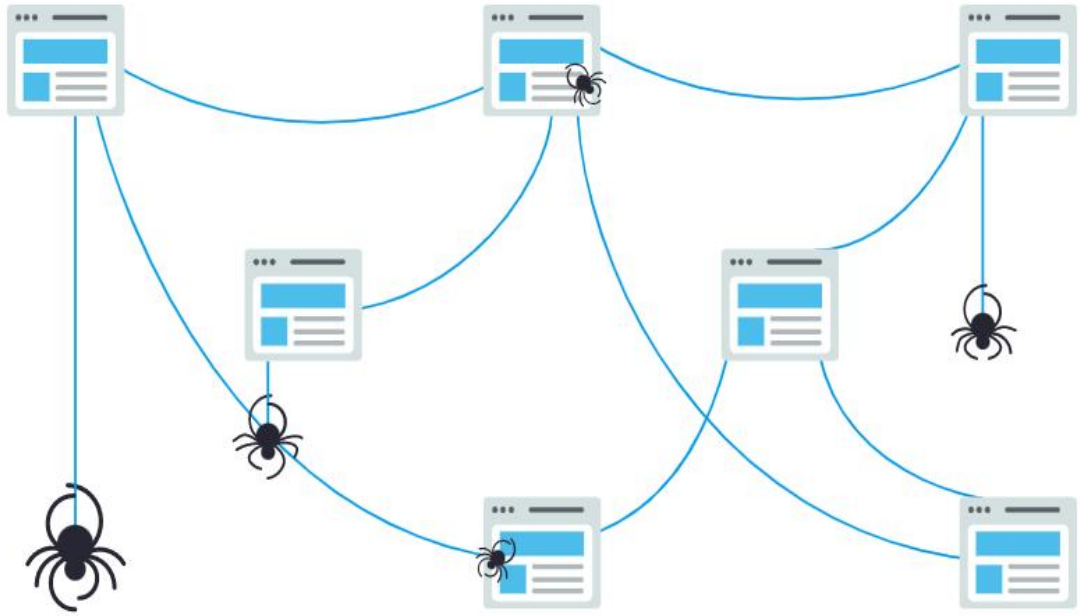


Figure 4.1 Architecture of crawling process

Indexing: How do search engines interpret and store your pages?

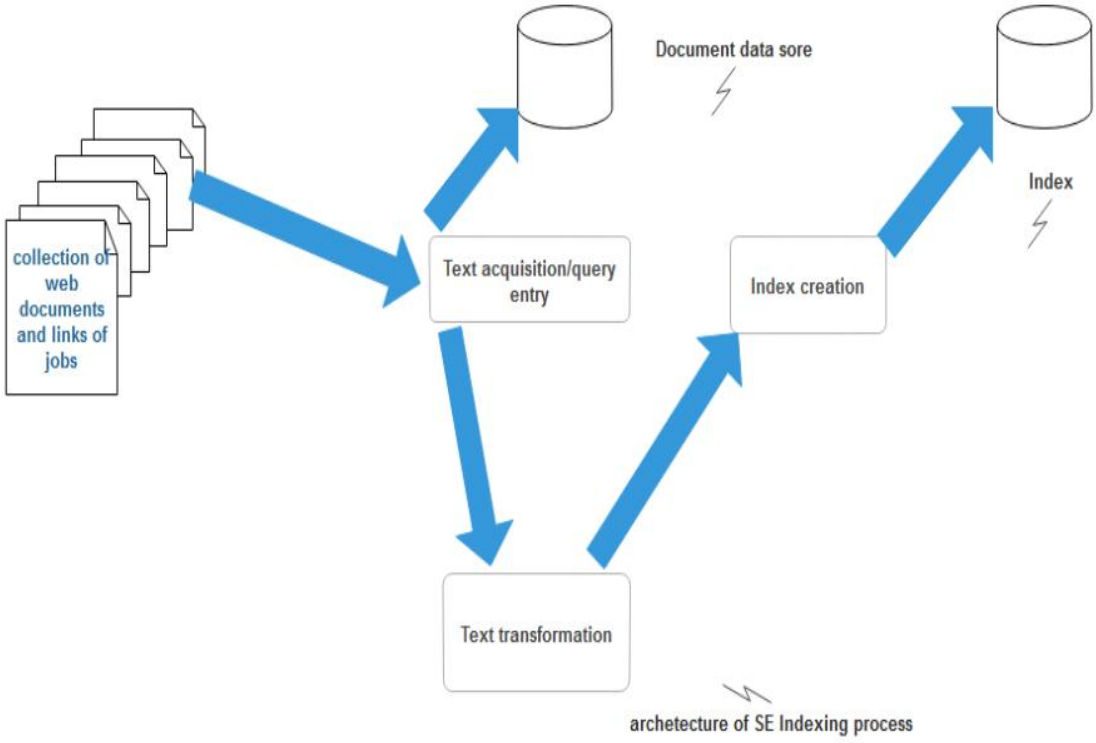


Figure 4.2 Architecture of indexing process

Ranking: How do search engines ensure that when someone types a query into the search bar, they get relevant results in return? That process is known as ranking, or the ordering of search results by most relevant to least relevant to a particular query.

4.2 System Model

4.2.1 Use Case Model

A use case is a sequence of action that provides a measurable value to an actor another way to look at it is that a use case describes a way to which a real world to interacts with the system. An essential use case sometimes called a business the case is simplified, abstract, generalized use case that captures the intention of the user in a technology and implementation independent manner.

The use case models are used to document the behavioural (functional) requirement of a System or the "what "of the system

- A use case describes a sequence of action that provides a measurable value to an actor and draw as a horizontal ellipse.
- An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

4.2.2 Use Case Diagram

Administrator: A person who manages the entire system

Employer: A company that posts job and hire professionals (job seekers).

Job Seeker: a person who find job.

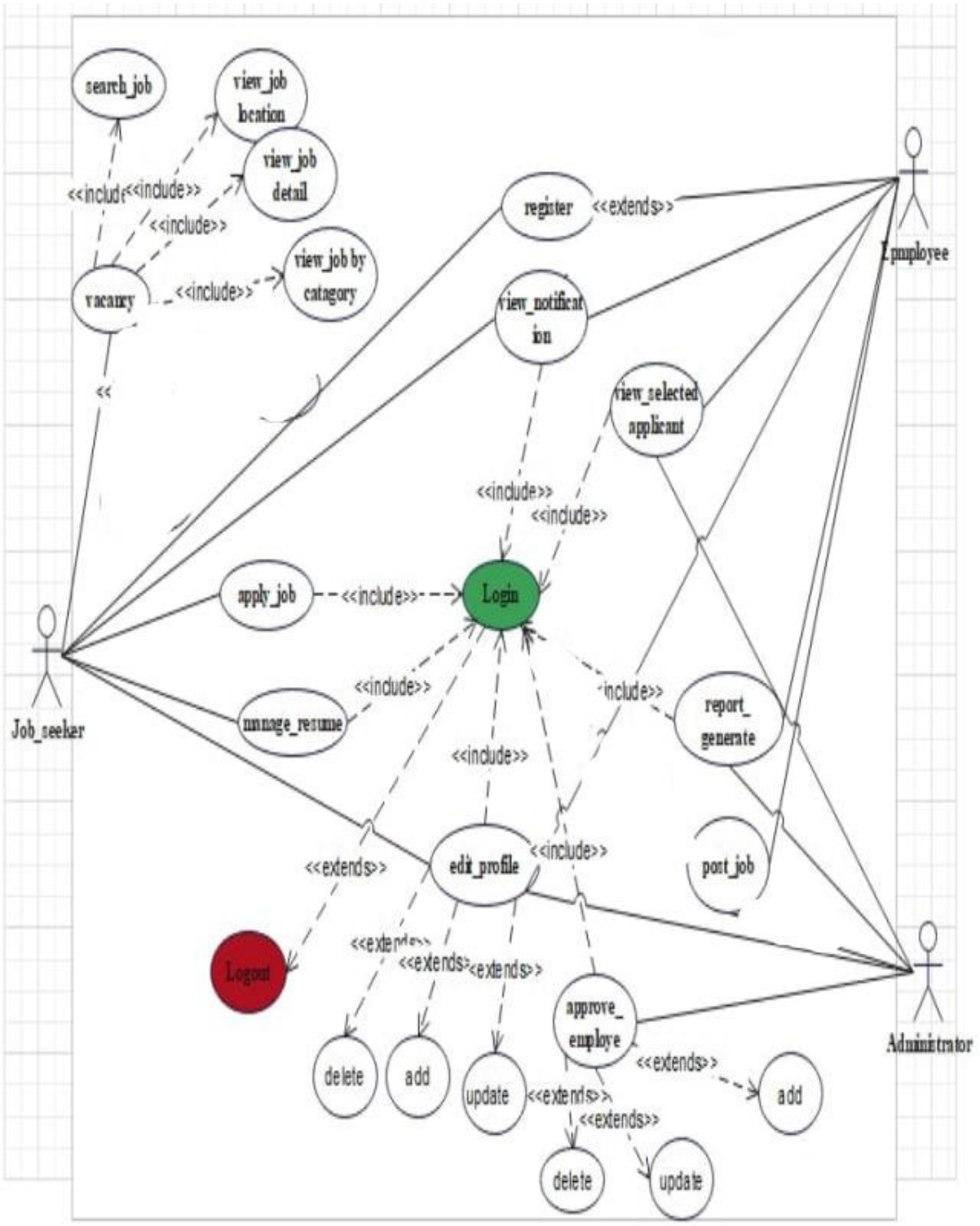


Figure 4.3 Use Case diagram

4.2.3 Use Case Description

Table 4.1:System Use Case description for register

Use Case Name	Register
Actors	Job Seeker, Employer
Purpose	Enable Job Seeker or Employer to register
Overview	The Job Seeker or the Employer registers His/her personal information
Pre-condition	Open application the system displays Home page
Post-condition	The system registers the Job Seeker or Employer
Main Course of Action	
Actor Action	System Action
1 Click Register Button 3. Enters personal information 5. Click save Button 7. Use Case Ends	2. Display registration form 4. Verify all inputs 6. Done
Alternative Course of Action	
4. a the system displays “please insert correct data” message	

Table 4.2. System Use Case description for Login

Use Case Name	Login
Actors	Job Seeker, Employer, Administrator

Purpose	To Authenticate
Overview	The actors enter the required information and the system will authenticate
Pre-condition	Open the application the actors must have an existing account
Post-condition	The system display home page
Main Course of Action	
Actor Action	System Action
1. Enter Username and Password 2. Click Login Button 5. Use Case End	3. Check validity of Username and Password 4. Display home page
Alternative Course of Action	
4a. The system displays “Please Enter User account and Password” message 4b. The system displays “Invalid Username or Password” message	

Table 4.3. System Use Case description for Search Job

Use Case Name Actor	Search Job
Actors	Job Seeker
Purpose	To search a Job
Overview	The Actor enter the name of the Job and the system will display the Job
Pre-condition	Open application

Post-condition	Results of Searched Job display to the Job Seeker
Main Course of Action	
Actor Action	System Action
1. Click Search Job Button 3 Enter Search Criteria 4 Click Search Button Use Case End	2. Display Search Job form 5. Retrieve the Job from the Database 6 Display the Job
Alternative Course of Action	
4a. The system displays “Please Fill the Criteria” message 4b. The system displays “No Job Found” message	

Table 4.4. System Use Case description for View Job Detail

Use Case Name	View Job Detail
Actors	Job Seeker
Purpose	To view detail information about the Job
Overview	The actor enters the name of the job and the system will display detail information of the job
Pre-condition	Should receive notification for available jobs should have searched a particular job
Post-condition	The Job Seeker will view Job detail
Main Course of Action	

Actor Action	System Action
1. Click View Job Detail Button 4. Use Case End	2. Retrieve Job Detail from the database 3. Display Job Detail
Alternative Course of Action	
1a. Click on Back Button	

Table 4.5. System Use Case description for View Latest Jobs

Use Case Name	View Job Detail
Actors	Job Seeker
Purpose	To view latest jobs
Overview	The system displays latest jobs posted
Pre-condition	Open application there should exist some new job offers
Post-condition	Latest jobs are viewed by the Job Seeker
Main Course of Action	
Actor Action	System Action
1 Click Latest Jobs Button 4. Use Case Ends	2 Retrieves latest jobs from database 3 Displays list of recent jobs
Alternative Course of Action	
3a. the system displays “No Recent Job Found” message	

Table 4.6. System Use Case description for view jobs by category

Use Case Name	View Jobs by Category
Actors	Job Seeker
Purpose	To view list of Jobs by their Category
Overview	The actor can view the list of jobs by their category
Pre-condition	Open application there should exist available job offers
Post-condition	Job seekers can View Jobs by Category
Main Course of Action	
Actor Action	System Action
Click on Job by Category Button 4 Click on the specific job Category 7. Use Case End	2. Retrieves Jobs based on their category from database 3. Displays jobs based on their category 5. Retrieve clicked job from database 6. Displays Job Detail
Alternative Course of Action	
3a. the system displays “No Job is found” message	

Table 4.7. System Use Case description applying for a job

Use Case Name	Apply for Job
Actors	Job Seeker
Purpose	Enable Job Seeker to Apply for the Job

Overview	The Job Seeker get Notification for available Jobs to apply that matches with his/her qualification
Pre-condition	Open application there should exist some new job offers
Post-condition	Should receive notifications of new job vacancies
Main Course of Action	
Actor Action	System Action
1. Click on Notification 3. Click on the Job title 5.Click on apply Button 7.Use Case End	2. Retrieve and display list of jobs that are available 4. Displays detail Information about the Job 6. Done
Alternative Course of Action	
5a. Click on Reject Button	

Table 4.8. System Use Case Description for posting resume

Use Case Name	Manage resume
Actors	Job Seeker
Purpose	Enable Job Seeker to post, or edit his/her Resume
Overview	The Job Seeker can update and Post his/her resume
Pre-condition	Login to the system

	View his/her resume
Post-condition	Resume of Job Seeker is posted or updated
Main Course of Action	
Actor Action	System Action
1. Click on Resume Button 3. Select his/her Resume 4. Click on Save Button 7. Use Case End	2. Display Resume page 5. Check the input 6. Done
Alternative Course of Action	
6a. the system displays “Please Select your Resume” message	

Table 4.9. System Use Case description for managing an Account

Use Case Name	Manage Profile
Actors	Employer, Job Seeker, Administrator
Purpose	Enable Job Seeker to Apply for the Job
Overview	The employers or Job Seekers can manage their account
Pre-condition	Login to the system View their profile
Post-condition	Manage account
Main Course of Action	
Actor Action	System Action

1. Click on Manage Profile Button 3. Make Change and Save	2. Retrieve and display the Profile 4. Display profile successfully updated
Alternative Course of Action	
3a. the system display “Cancels” saving message 4a. the system displays “Update Failed” message	

Table 4.10 Use case for approve employer

Use Case Name	Approve employer
Actors	Administrator
Purpose	To approve and reject the employer
Overview	The administrator views registered employers and approve or reject them
Pre-condition	Employers must be on Pending State
Post-condition	Employers are either approved or rejected
Main Course of Action	
Actor Action	System Action
1. Click on Approve Button 3. Click on a specific Employer 5. Click on Save Button	2. Retrieve and display list of employers which are on pending state 4. Display employer detail

Table 4.11. System Use Case description for posting a job

Use Case Name	Post Job

Actors	Employee
Purpose	To enable Employer to Post Jobs
Overview	The employers can post, edit and delete jobs
Pre-condition	Login into the system Employers must be approved
Post-condition	Job is successfully Posted
Main Course of Action	
Actor Action	System Action
1. Click on Post Job button 3. The Employer fills Job details 4. Click on post Button 7. Use Case Ends	2. Displays Post Job page 5. Checks all inputs 6. Done
Alternative Course of Action	
6a. the system displays “Please fill all required fields” message	

Table 4.12. System Use Case description for viewing Notified applicants

Use Case Name	View Eligible Applicants
Actors	Employee
Purpose	Enable the Employer to view eligible applicants from the system
Overview	The Employers can view eligible applicants that fulfils their own company criteria

Pre-condition	The system should shortlist eligible applicants
Post-condition	Eligible applicants are viewed by Employers
Main Course of Action	
Actor Action	System Action
1. Click on View Notified applicants Button 4. Employers view each job seekers detail 5. Click on Save Button 6. Use case end	2. Retrieves list of notified applicants from database 3. Display list of eligible applicants
Alternative Course of Action	
3a. the system displays “Minimum number of applicants” message 4a. Click on View Button 2a. The system Retrieve and display available applicants 4b. Click on Cancel Button	

Table 4.13. System Use Case description for view notification

Use Case Name	View notification
Actors	Job seeker
Purpose	To enable the job seekers to apply or reject for a job
Overview	The System evaluates applicant data based on employer’s criteria The System generates vacancy notification for job seekers that fulfils employers’ criteria The System

	compares applied applicants based on their resume The System generates list of eligible applicants and send it to the employers for approval. The employers check validity of eligible applicants resumes verifying The System generates an announcement notification for those applicants that they got the job.
Pre-condition	The system must be generated notification
Post-condition	The Job seeker apply or reject for a job
Main Course of Action	
Actor Action	System Action
1. Click on view notification button 4.use case end	2. Retrieves generated notification from database 3. Display notification
Alternative Course of Action	
3a.the system generates and displays “no new notification found” message	

4.3 Object Model

4.3.1 Data Dictionary

Table 4.14. Data dictionary for jobseeker table

Field Name	Data type	Description
First name	String	Holds the First name of the job seeker.
Last name	String	Holds the Last name of the jobseeker.

Password	String	Describe the Id used when jobseeker login to the system when he/she perform some action.
Email	String	Describe email used when jobseeker login to the system.

Table 4.15. Data dictionary for Administrator table

Field Name	Data type	Description
First name	String	Holds the First name of the Administrator
Last name	String	Holds the Last name of the Administrator.
Password	String	Describe the Id used when Admin login to the system when he/she perform some action.
Email	String	Describe email used when Admin login to the system.

Table 4.16. Data dictionary for Employer table

Field Name	Data type	Description
Company name	String	Holds the company name of the employer.
Contact name	String	Holds the contact name of the employer.
Password	String	Describe the Id used when employer login to the system when he/she perform some

		action.
Phone Number	String	Describe the phone number of employers.
Company Description	String	Describe the place which job is post by the employer.
Email	String	Describe Email used by Employer login to the system

Table 4.17. Data dictionary for Privilege table

Field Name	Data type	Description
Level	String	Holds the stage of the privilege.
Date	Date	Holds the time line of the privilege.

Table 4.18. Data dictionary for Notification table

Field Name	Data type	Description
Date	Date	Describe the date which notification posts.
Notification Description	String	Describe what type of notification post employer.

Table 4.19. Data dictionary for Resume table

Field Name	Data type	Description
Education Background	String	Describe the education background of job seeker.

Qualification	String	Describe the standard of the job seeker.
Experience	String	Describe the ability of the job seeker.
Language	String	Describe the language job seeker.
Reference	String	Describe the reference of job seeker.

Table 4.20. Data dictionary for job table

Field Name	Data type	Description
Job Title	String	Describe the title of the job.
Job Number	String	Describe the number of the job.
Job Category	String	Describe the job posted when the job seeker view.
Job Location	String	Describe the location of the job.
Posted Date	String	Describe the date of posted job.

Salary	String	Describe how much money be needed
Job Description	String	Describe detail about the job

4.3.2 Data Dictionary Analysis Level Class Diagram (Conceptual Modelling)

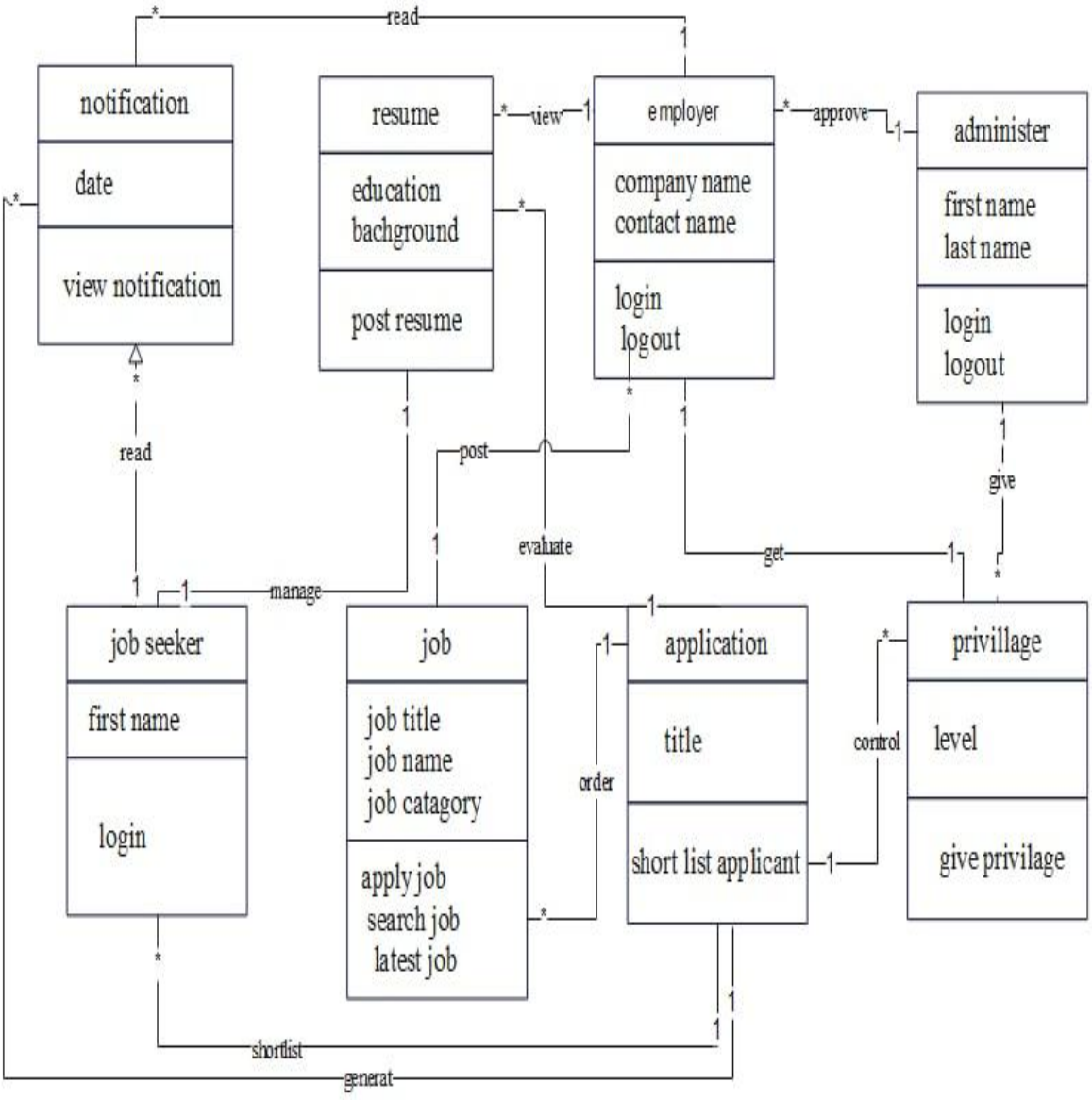


Figure 4.4 Analysis Level Class Diagram (Conceptual Modelling)

4.4 Dynamic Model

4.4.1 Sequence Diagram

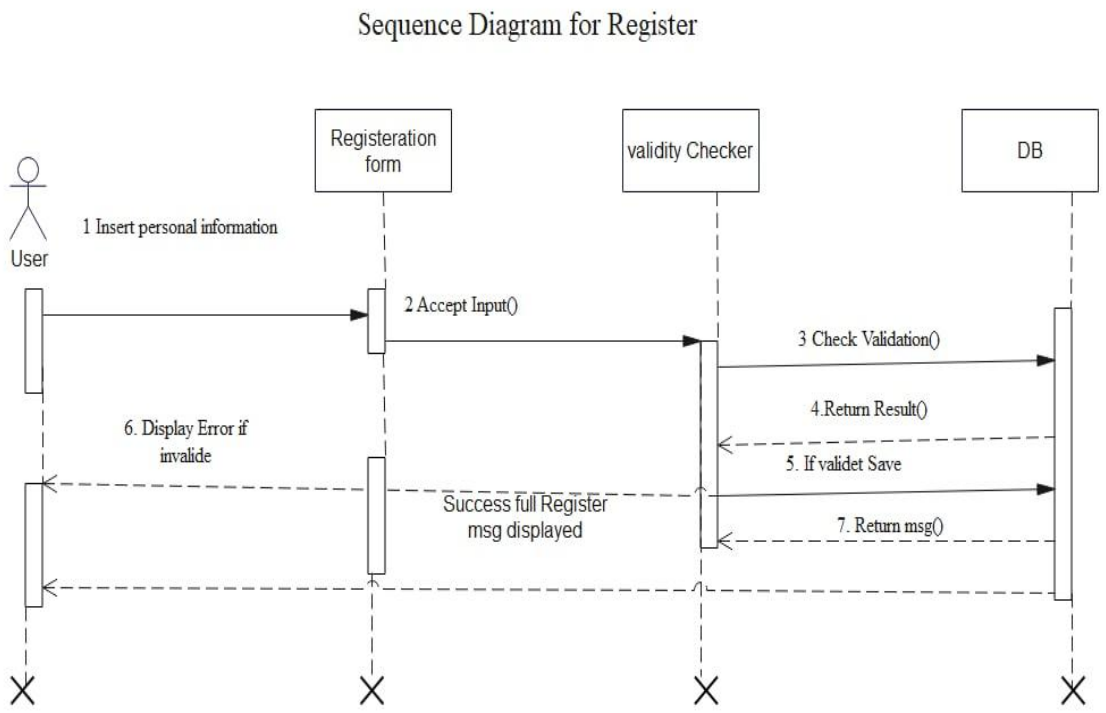


Figure 4.5 Sequence Diagram for register

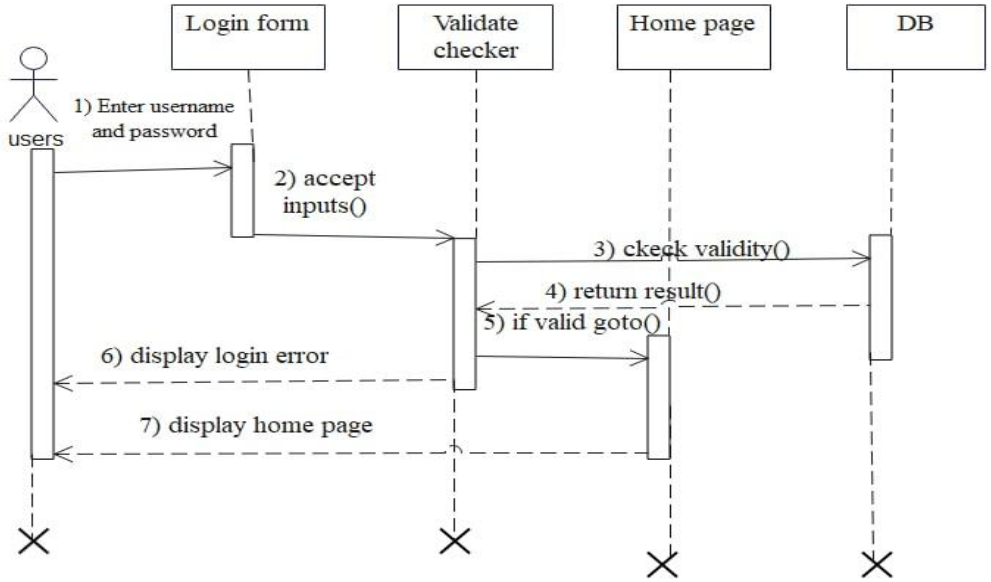


Figure 4..6 Sequence Diagram for login

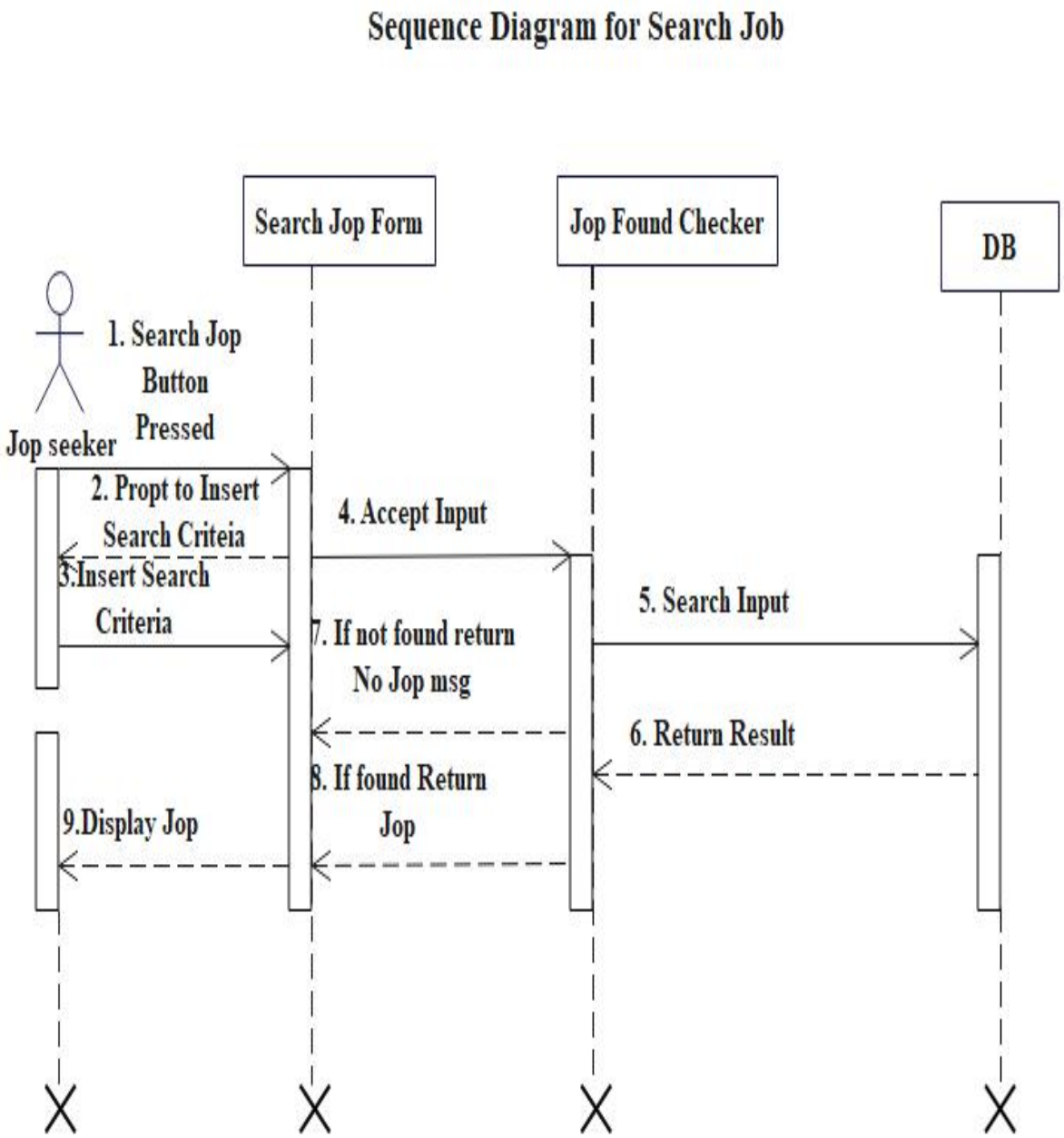


Figure 4.7 Sequence Diagram for search job

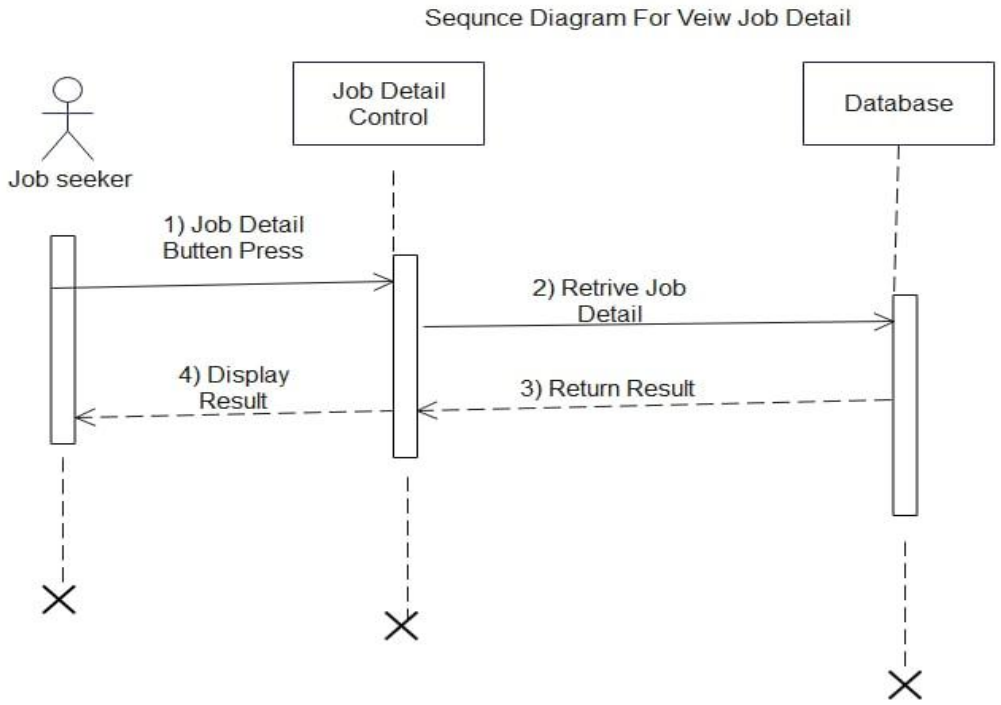


Figure 4.8 Sequence Diagram for view job detail

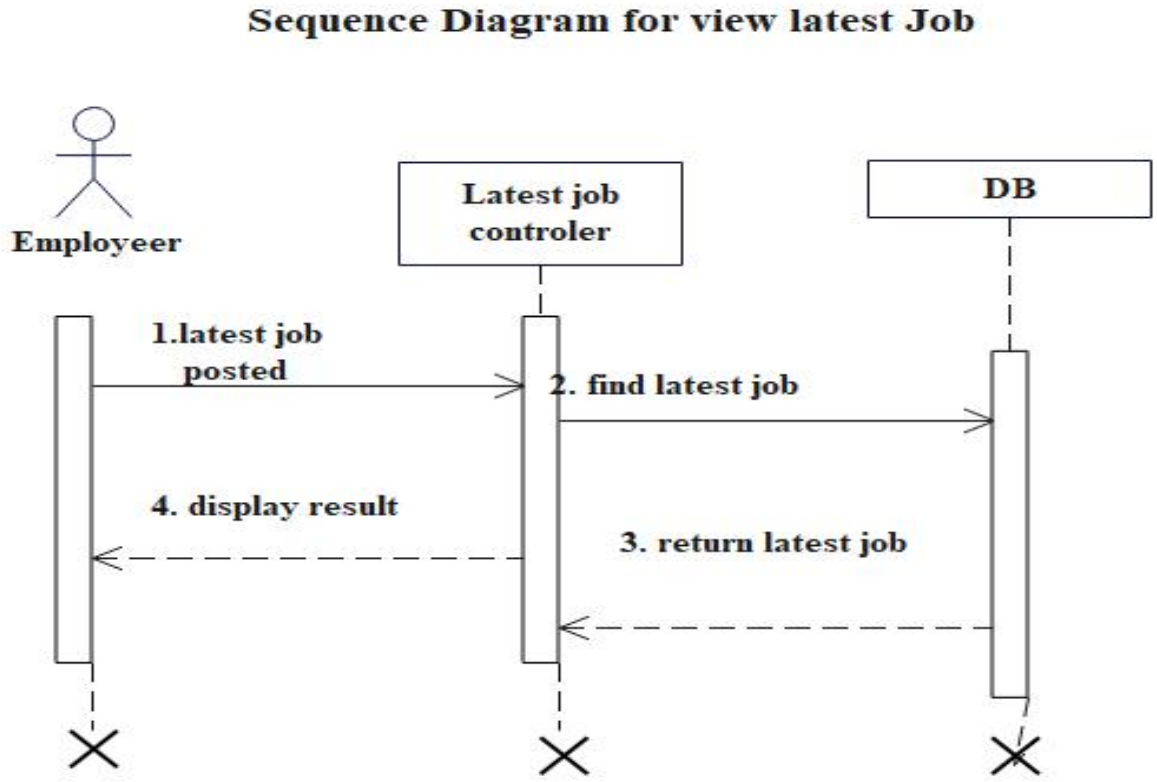


Figure 4.9 Sequence Diagram for view latest job

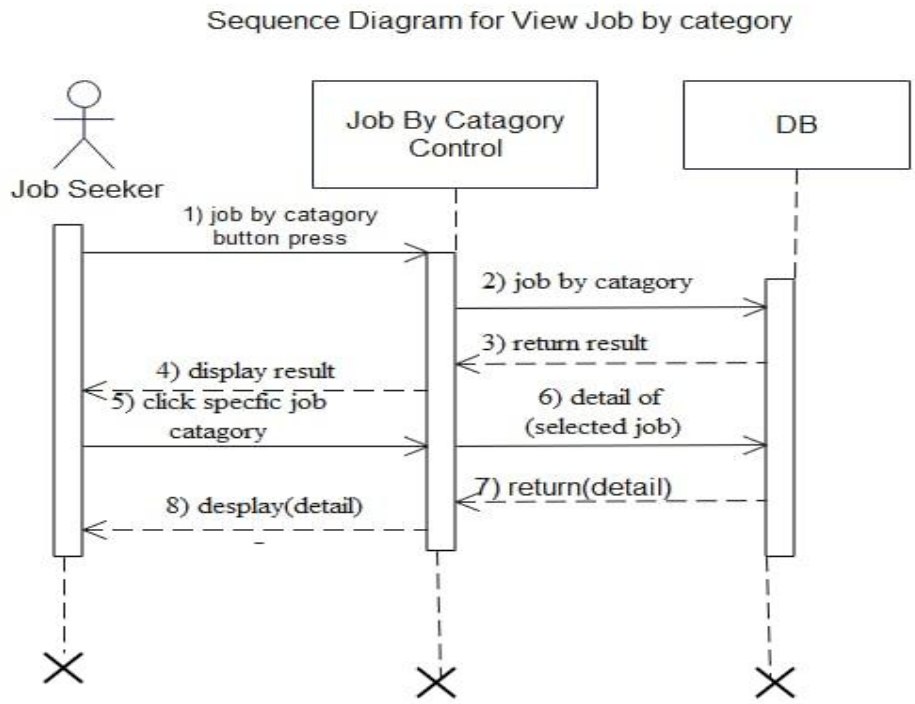


Figure 4.10 Sequence Diagram for view job by category

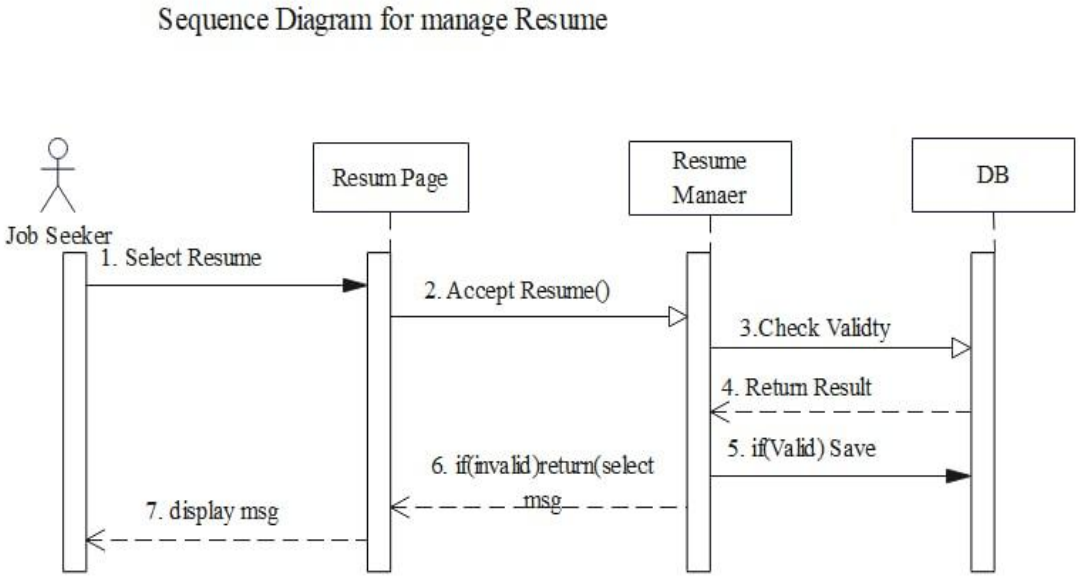


Figure 4.11 Sequence Diagram for manage resume

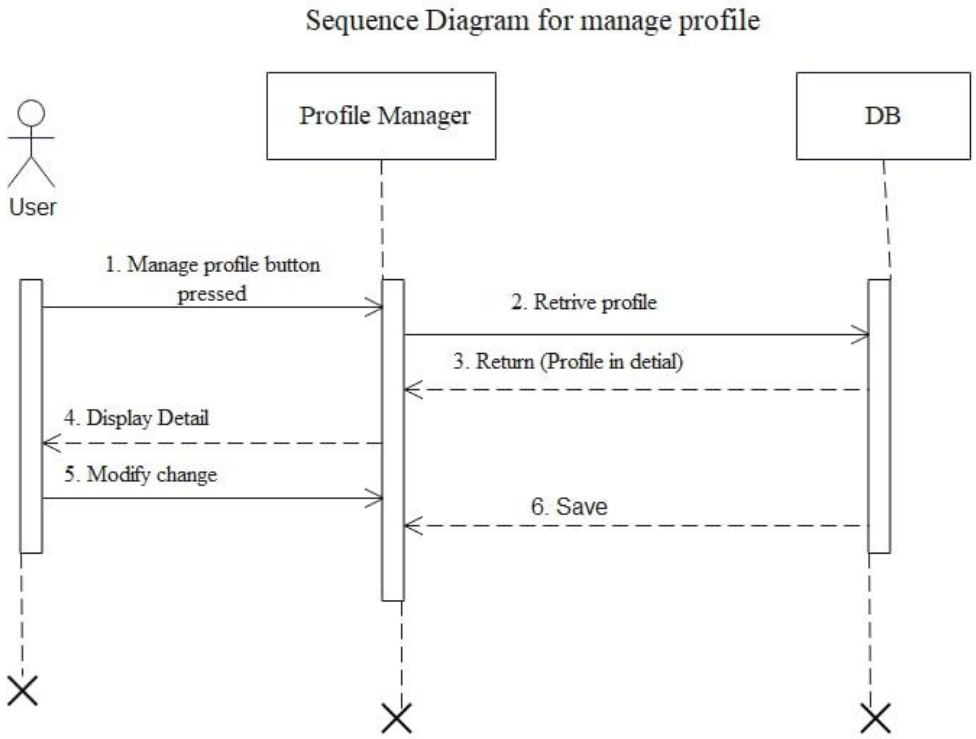


Figure 4.12 Sequence Diagram for manage profile

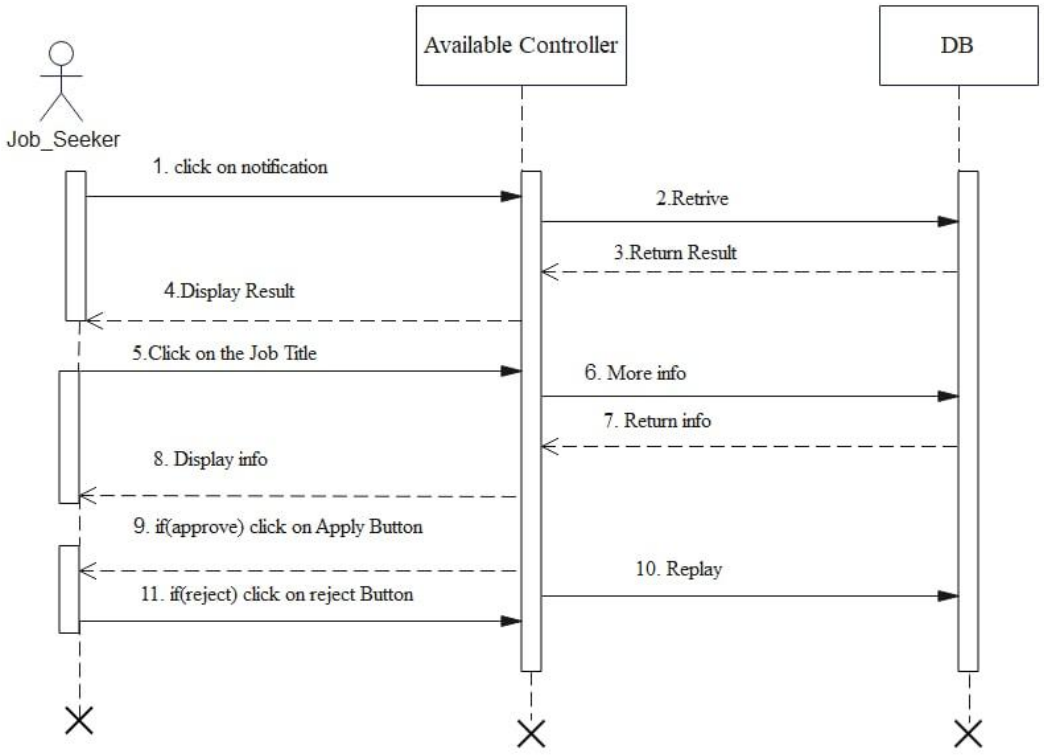


Figure 4.13 Sequence Diagram for apply job

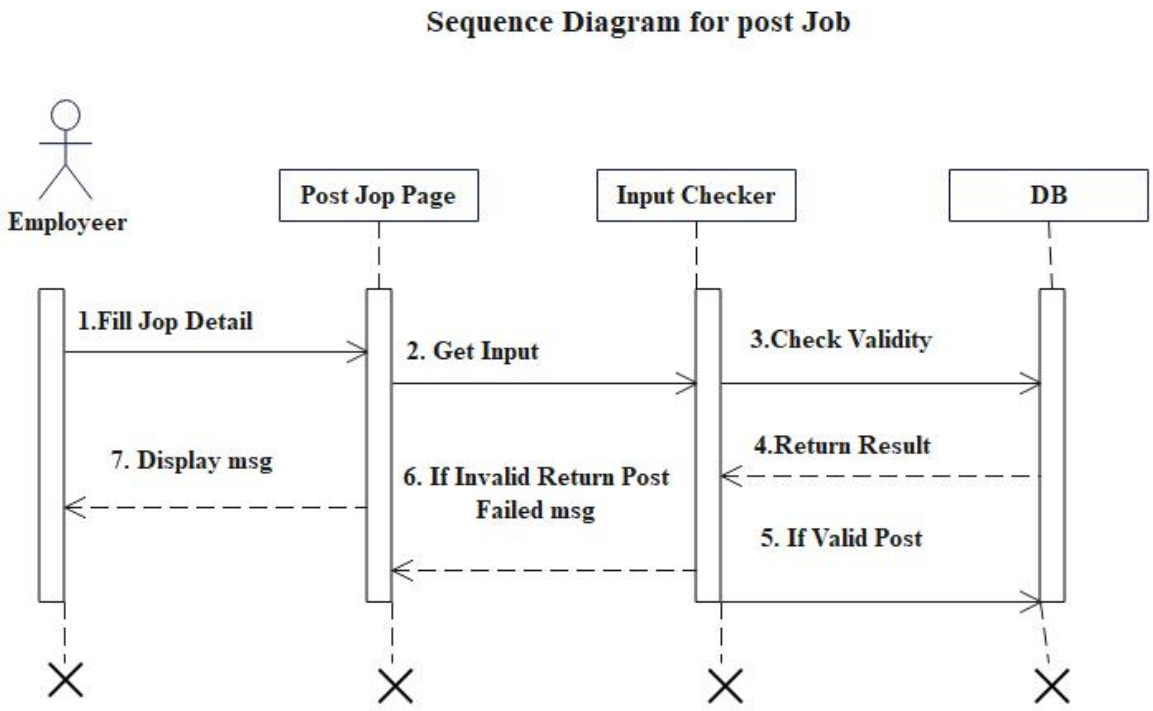


Figure 4.1: Sequence Diagram for post job

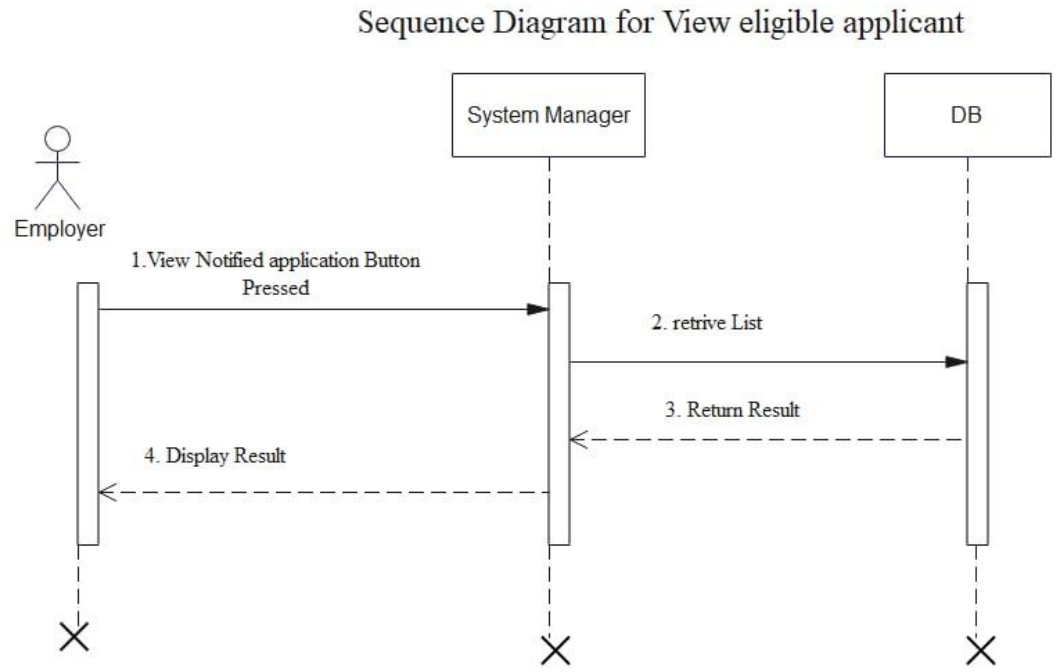


Figure 4.2: Seq4ence Diagram for view eligible applicant

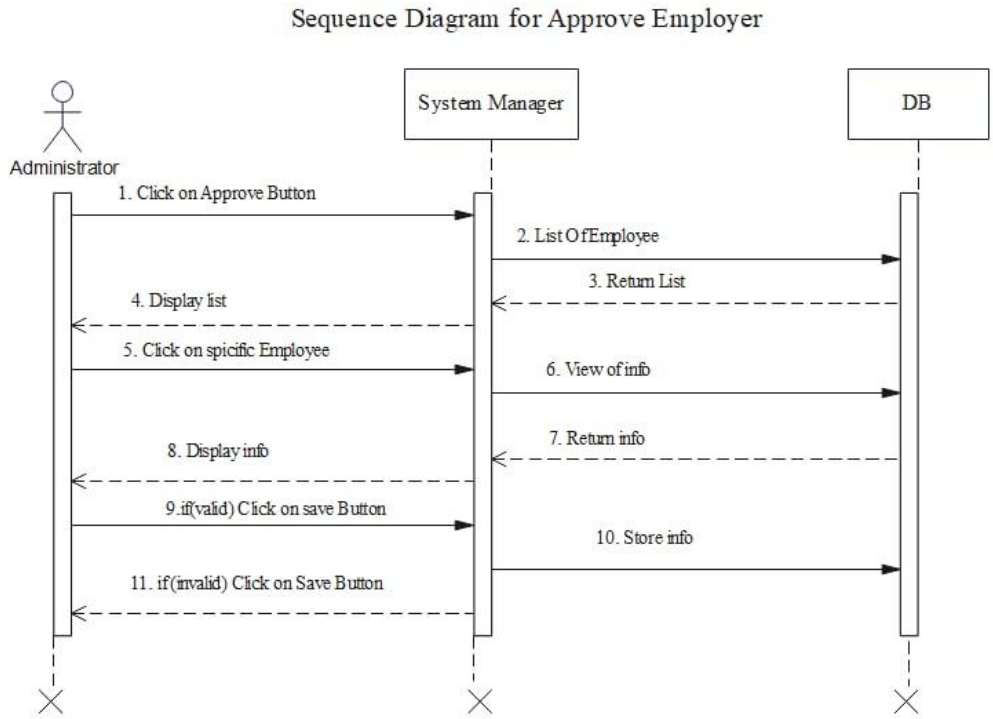


Figure 4.3: Sequence Diagram for approve employer

4.4.2 Acivity Diagram

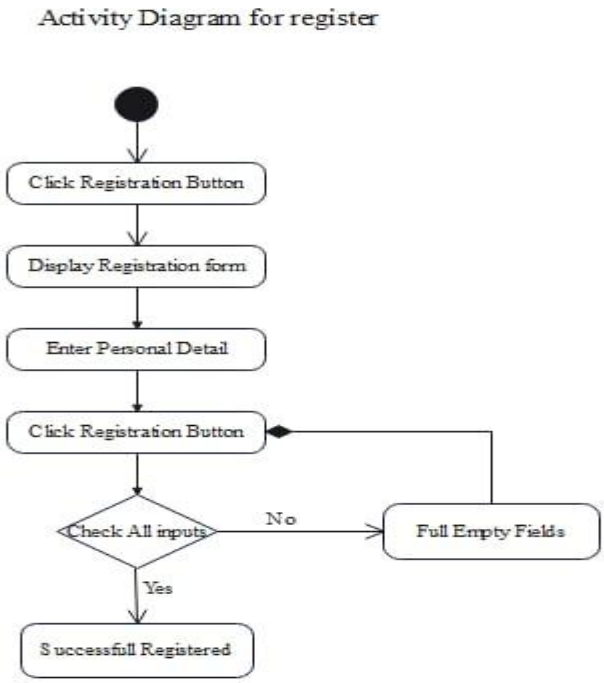


Figure 4.4: Activity Diagram for register

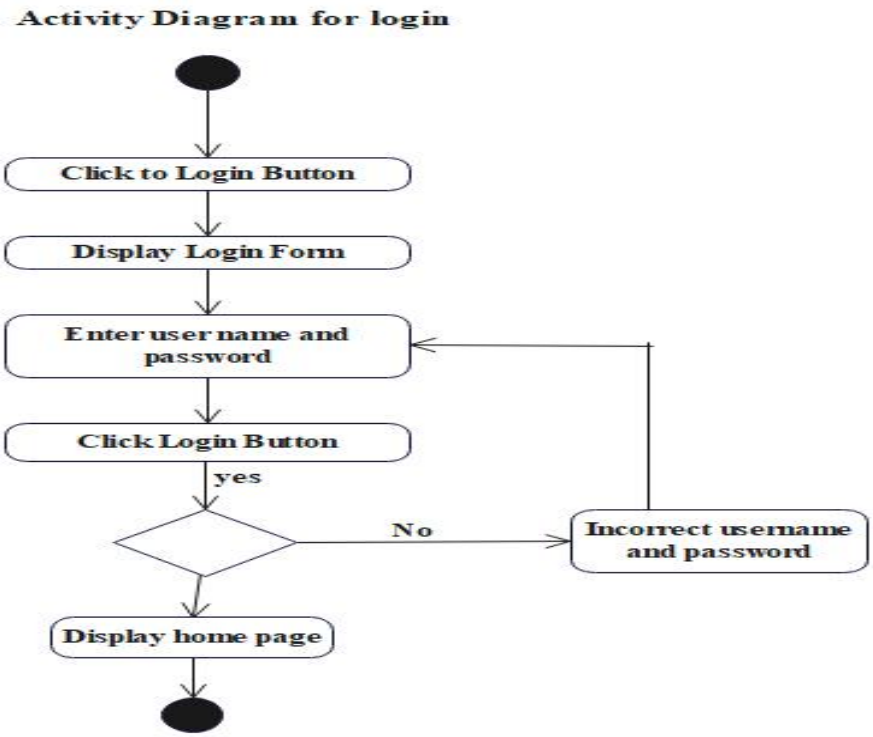


Figure 4.5: Activity Diagram for login

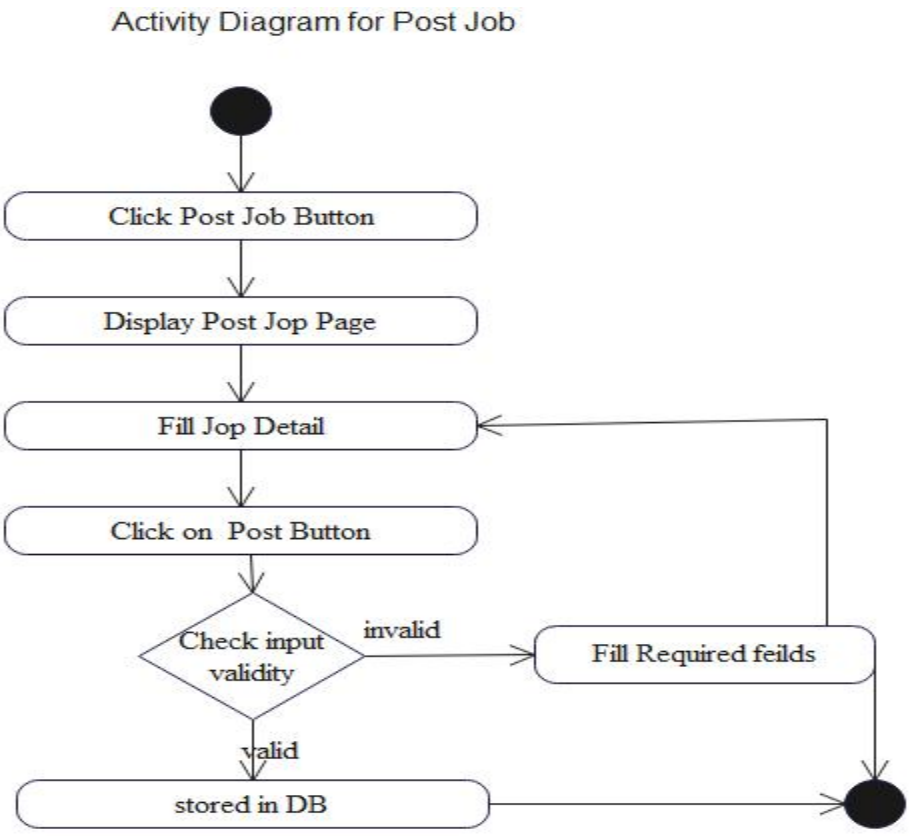


Figure 4.6: Activity diagrams for post job

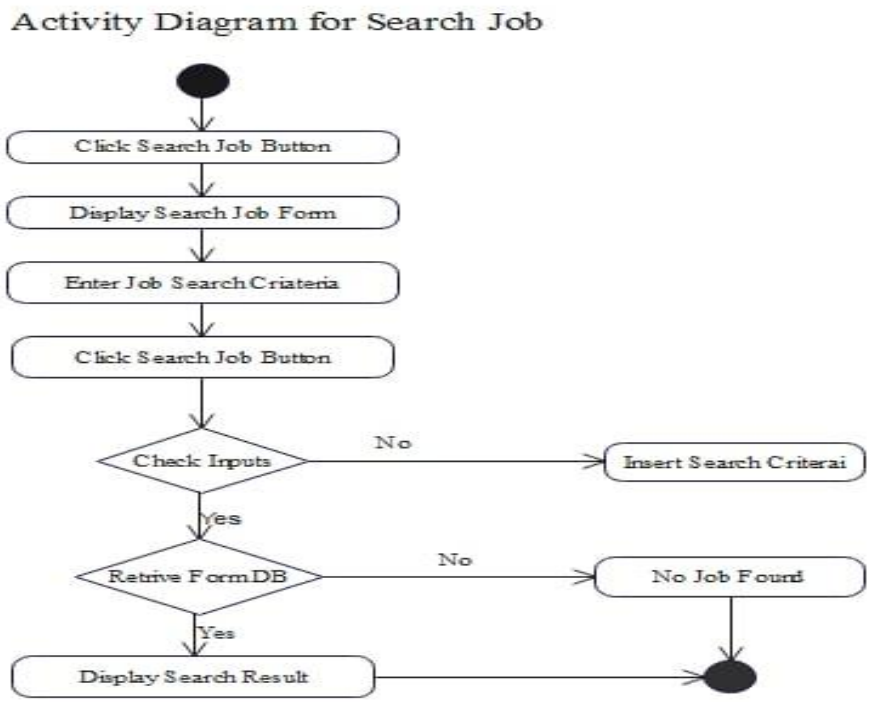


Figure 4.7: Activity Diagram for search job

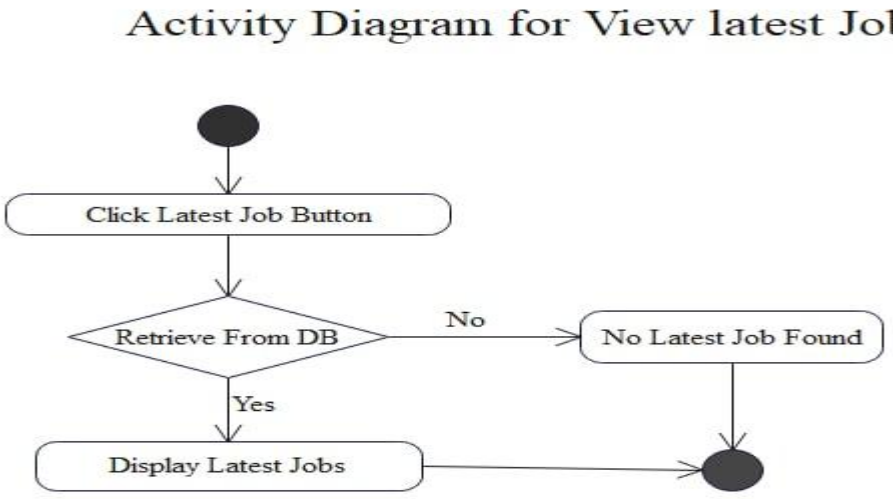


Figure 4.9: Activity Diagram for view latest job

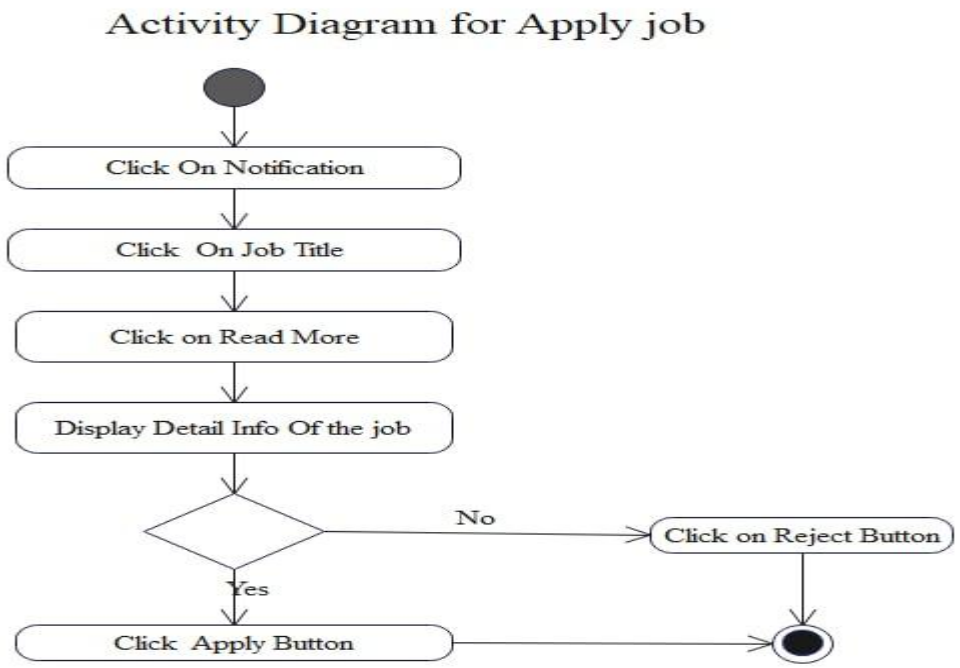


Figure 4.10: Activity Diagram for apply job

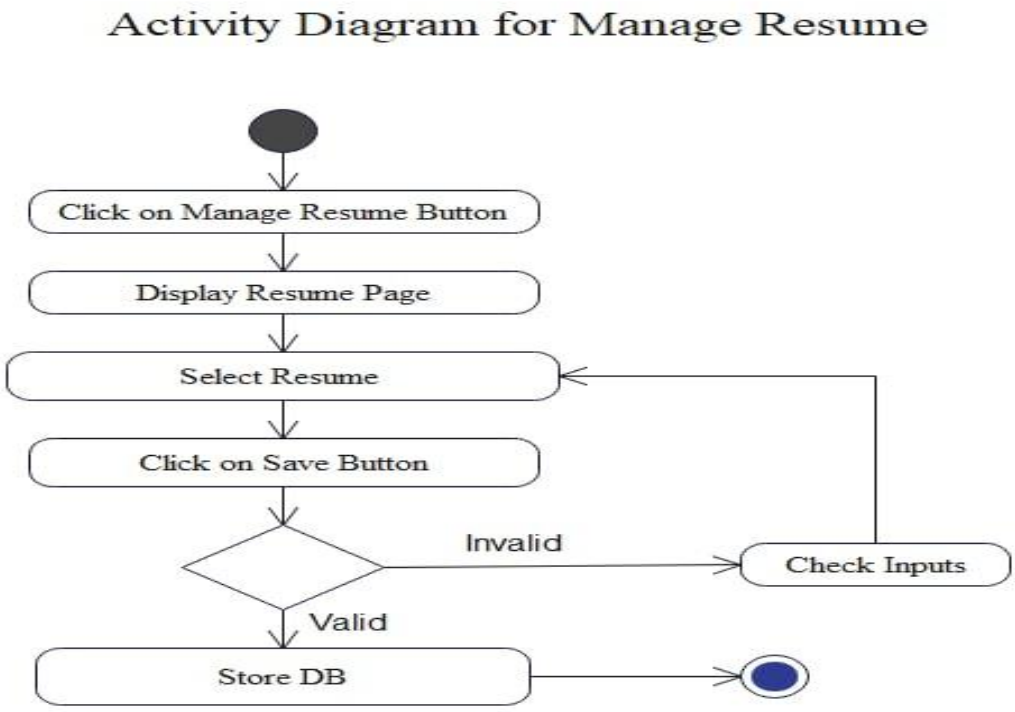


Figure 4.11: Activity Diagram for manage resume

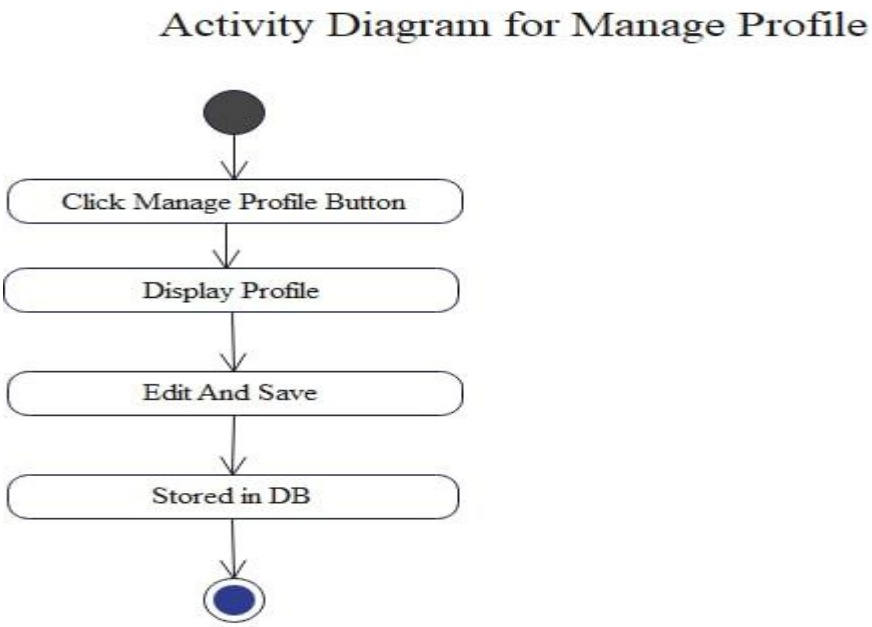


Figure 4.12: Activity Diagram for manage profile

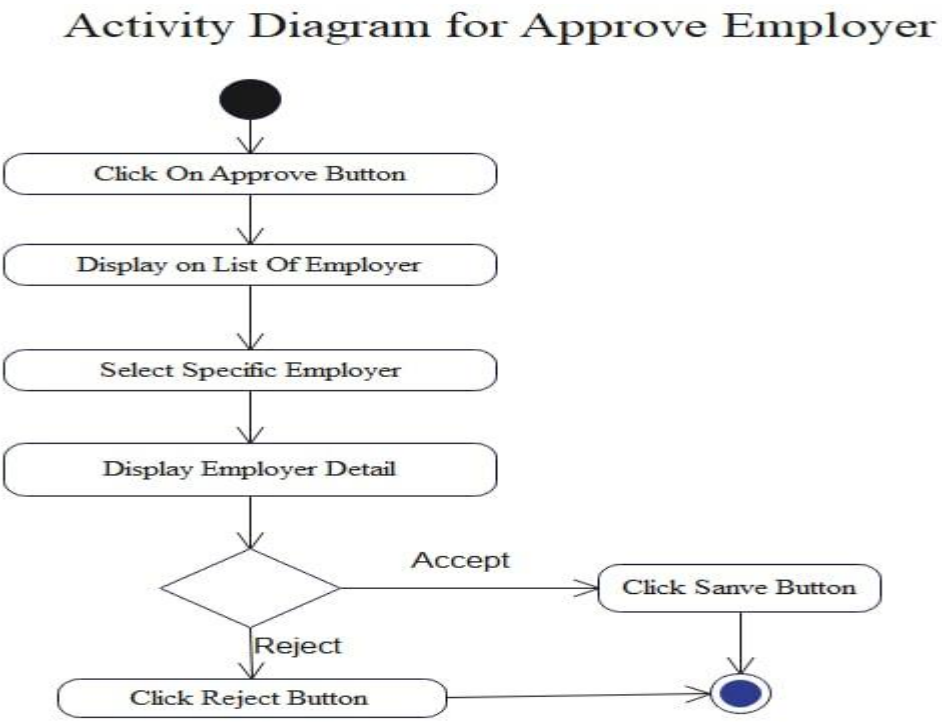


Figure 4.13:Activity Diagram for approve employee

4.4.3 State chart Diagram

Figure 0.14: State Chart Diagram for post job

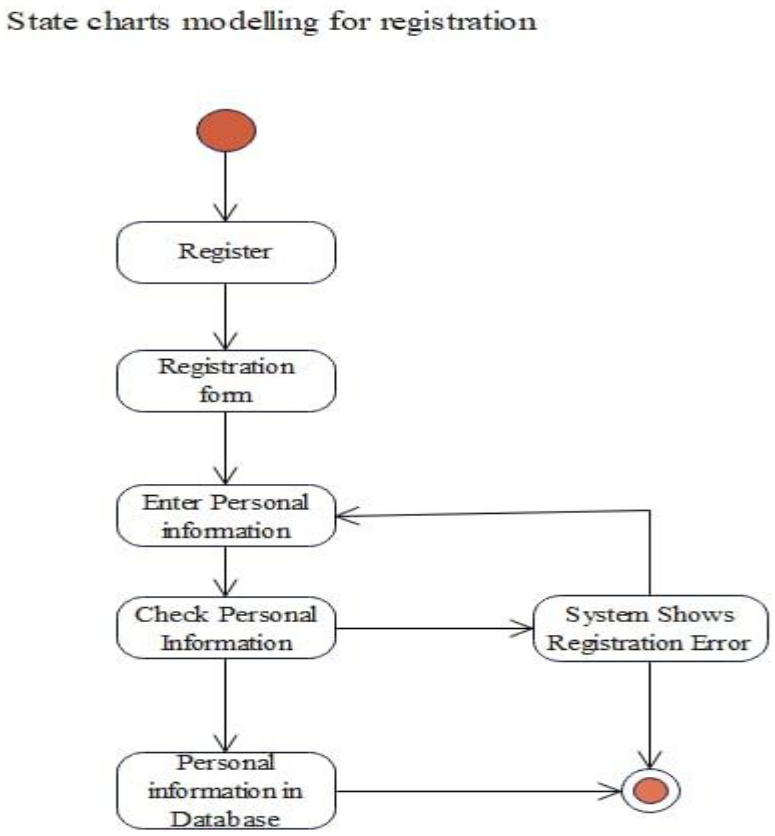


Figure 4.15: State chart modelling for registration

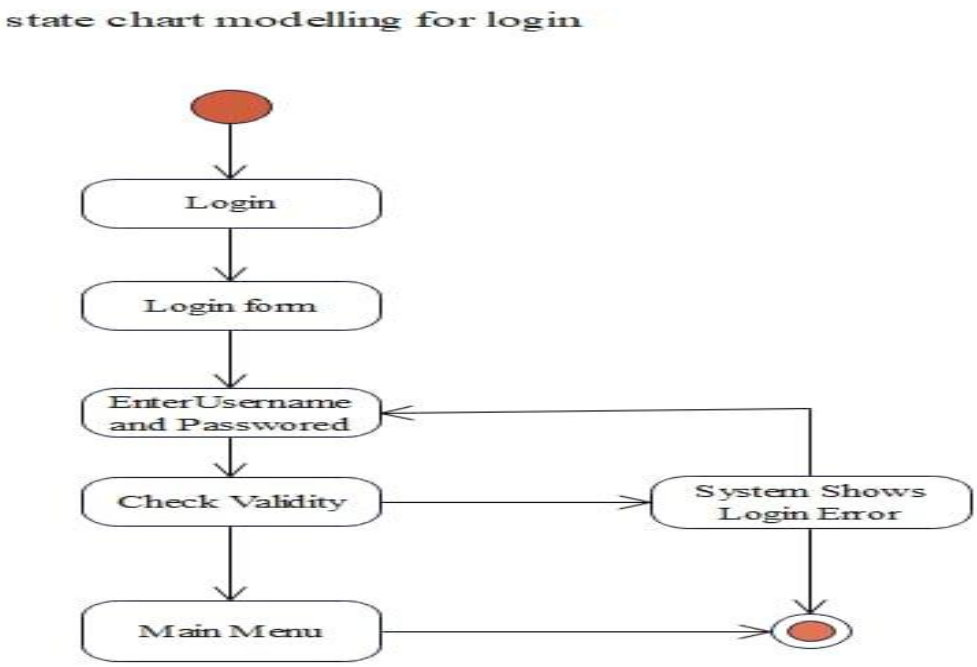


Figure 4.16: State chart modelling for latest job

Figure 0.17: State chart modelling for search job

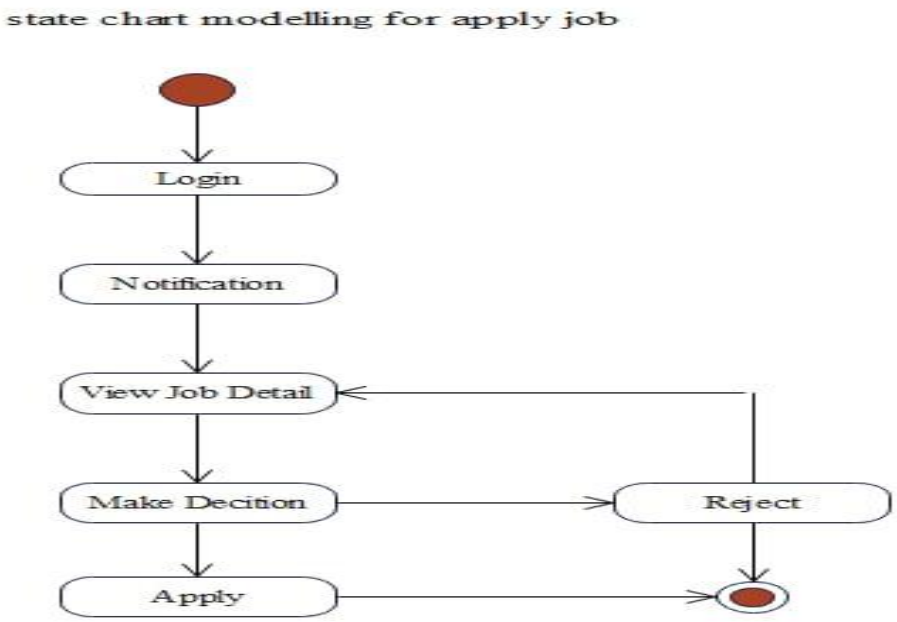


Figure 4.18: State chart modelling for apply job

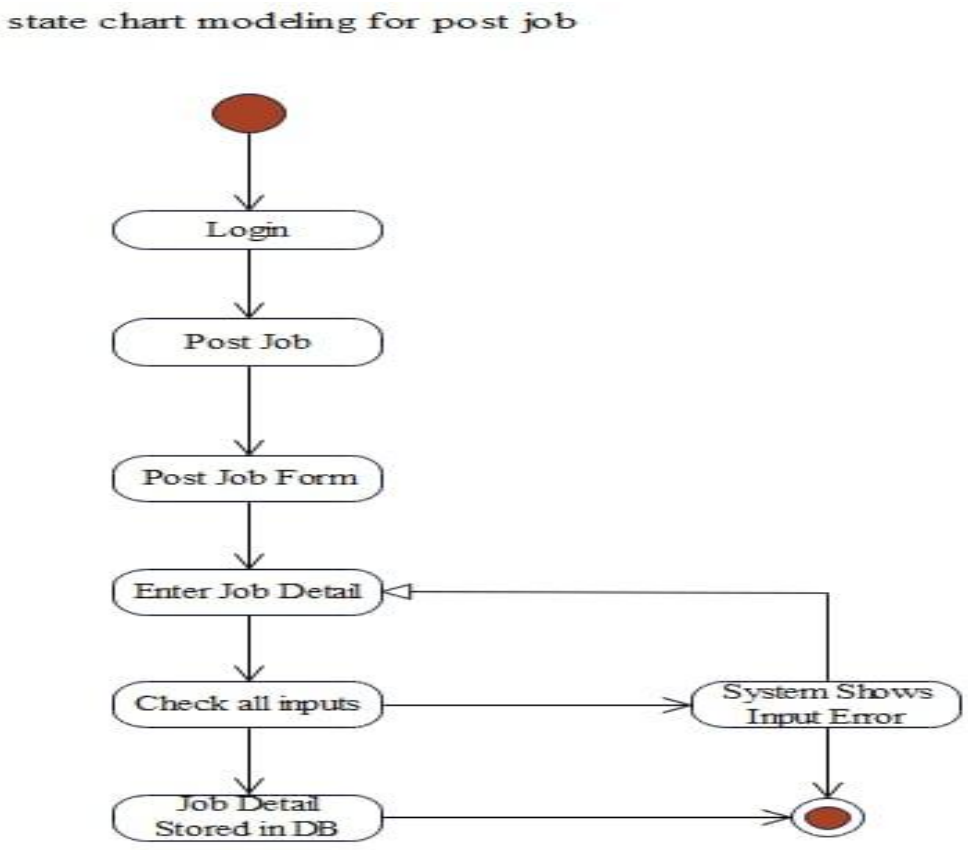


Figure 4.19: State chart modelling for post job

CHAPTER FIVE

5. System Design

5.1 Design Goals

As the popularity and complexity of Internet search engines increase, the design, development and maintenance of large, complex web-based Information Retrieval (WIR) systems become a challenge. The difficulty of designing a WIR system is compounded by information overload triggered from various different information sources. From a standpoint of the search engine users, it is more usable for the WIR to provide a single search point to multiple databases. To tackle this issue, we present the design and implementation of a job search component for the job search Engine. The job search Engine allows the user to search heterogeneous, multiple Meta data of job related vacancy from multiple job site of Ethiopia with one command. Our performance benchmark tests show that the job search Engine is scalable and usable.

System design goal for Job Search Engine. In this phase we illustrate design goal of the project, proposed software architecture, subsystem decomposition, component diagram, deployment modelling, persistence modelling, access control and security.

It describes object design trade-offs made by developers, guidelines they followed for subsystem interfaces, the decomposition of subsystems into packages and classes, and the class interfaces. The object design document (ODD) is used to exchange interface information among teams and as a reference during testing.

Conceptual design involves a series of trade-off decisions among significant parameters such as operating speeds, memory size, power, and I/O bandwidth - to obtain a compromise design which best meets the performance requirements. Those factors which can be used to evaluate the design trade-goals (usually on a qualitative basis) include:

- Maintainability
- Compatibility

Maintainability: Since our project is developed by php, Repair should be readily accomplished during ground operation, and if inflight maintenance is desired, this should be specified as a design requirement. Generally, maintenance procedure is to remove bad components or subsystems from the system and replace them with another component.

Compatibility: Since our project is developed by php. It can easily run in any web browser without any problem.

The primary goal of search engine

- Effectiveness (quality): to retrieve the most relevant set of documents for a query
- ✓ Process text and store text statistics to improve relevance
- Efficiency (speed): process queries from users as fast as possible
- ✓ Use specialized data structures
- Specific goals usually fall into the above primary goals
- ✓ Example: handling changing document collections
- ✓ both an effectiveness issue and an efficiency issue

5.2 Proposed System Architecture

5.2.1 Subsystem Decomposition and Description

Sub-system decomposition is the processes of breaking down the system into sub-systems, and each sub-systems analysis separately. The main aim is to minimize the complexity of the system. We decompose a system into simpler parts, called subsystems, which are made of a number of solution domain classes. This subsystem performs the following tasks.

- Administrator subsystem
    - Manage users
    - Approve Employer
- Employers
    - Post Job
    - View eligible Applicant
    - View Notification
    - Generate Report
- Job seeker
    - View job detail
    - Search job
    - View latest job
    - Apply job
    - Manage resume

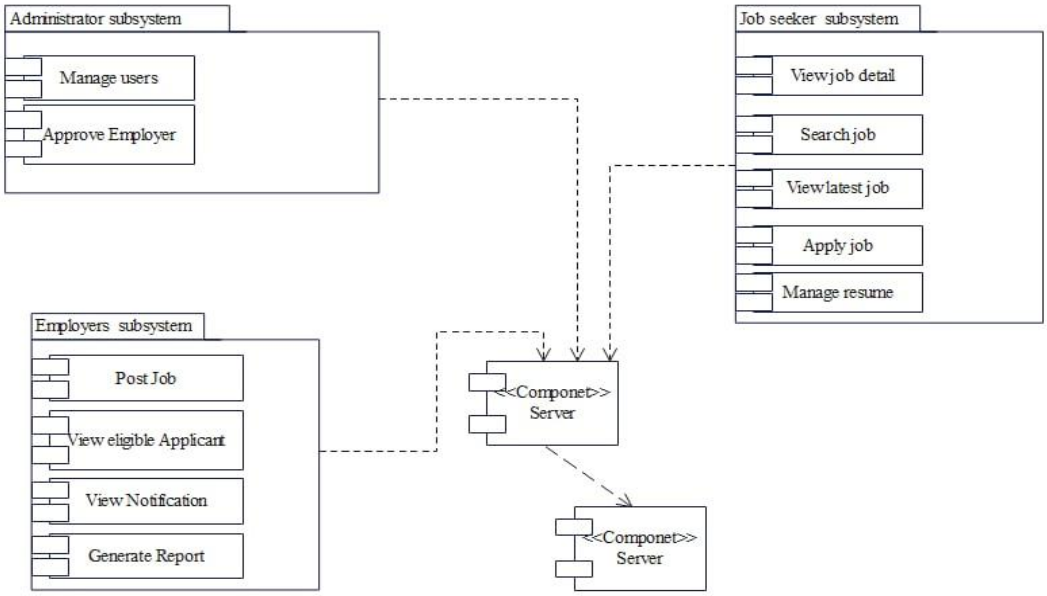


Figure 5.1: Subsystem Decomposition and Description

5.2.2 Component Diagram

Component diagrams show how the physical components of a system are organized. And also shows which component or objects will be accessed by whom and what type of security infrastructures it is using.

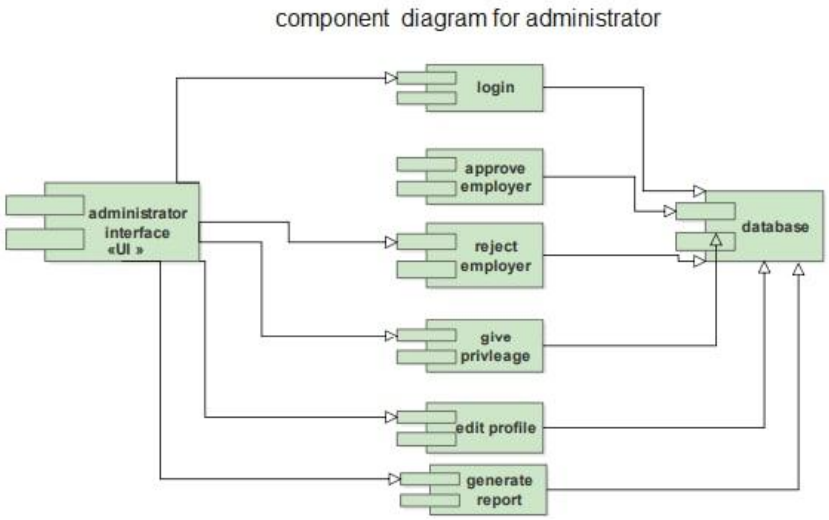


Figure 5.2: Component Diagram for Administer

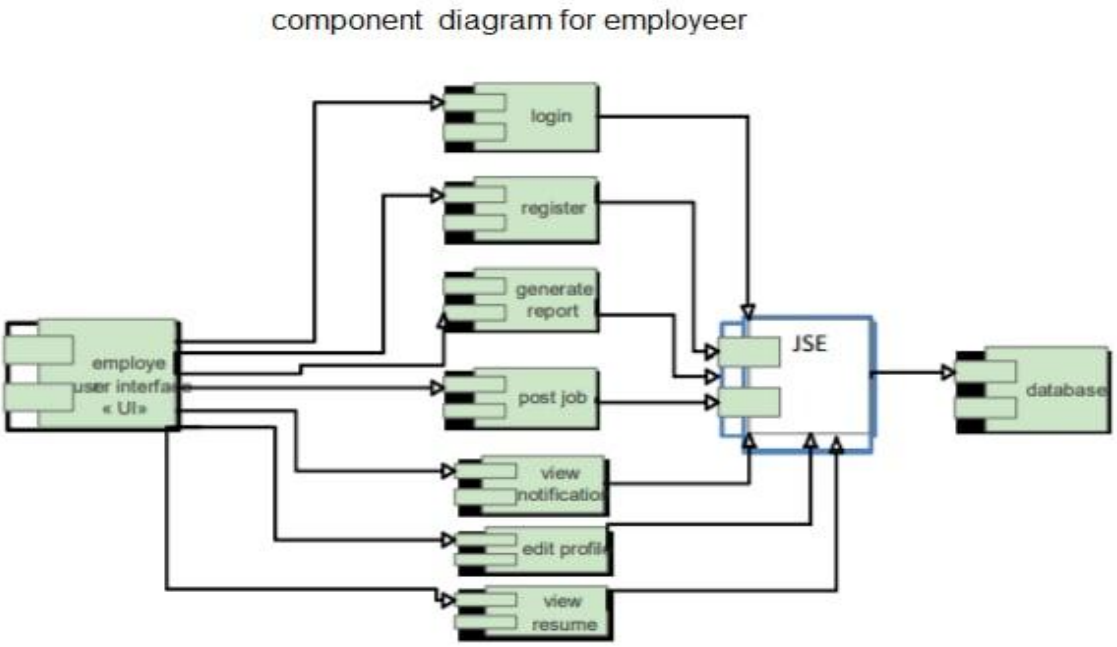


Figure 5.3: Component Diagram for Employer

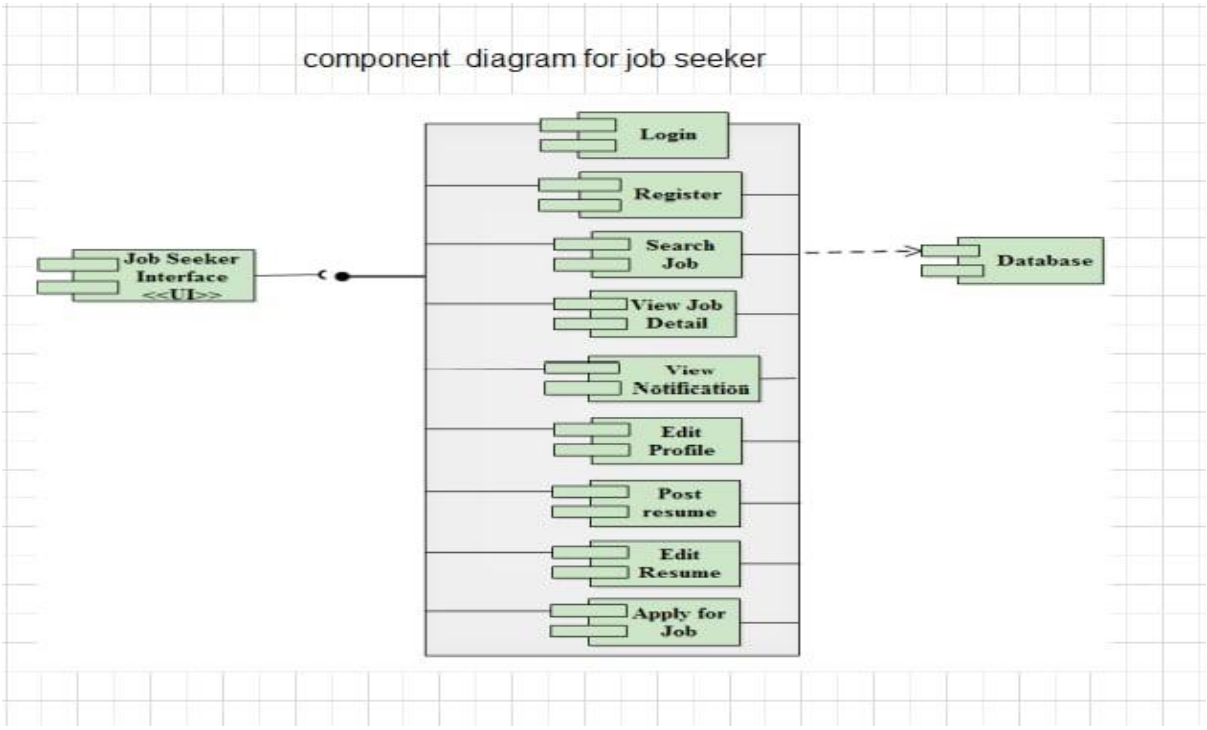


Figure 5.4: Component Diagram for Jobseeker

5.2.3 Hardware/Software Mapping

Deployment modelling is used to show the hardware of the system, the software that is installed in the hardware and also shows how the software and the hardware components work together.

The purpose of deployment diagrams can be described as:

- Visualize hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe run time processing nodes.
- To model the hardware topology of a system.
- To model embedded system.
- To model hardware details for a client/server system.
- To model hardware details of a distributed application.
- Forward and reverse engineering

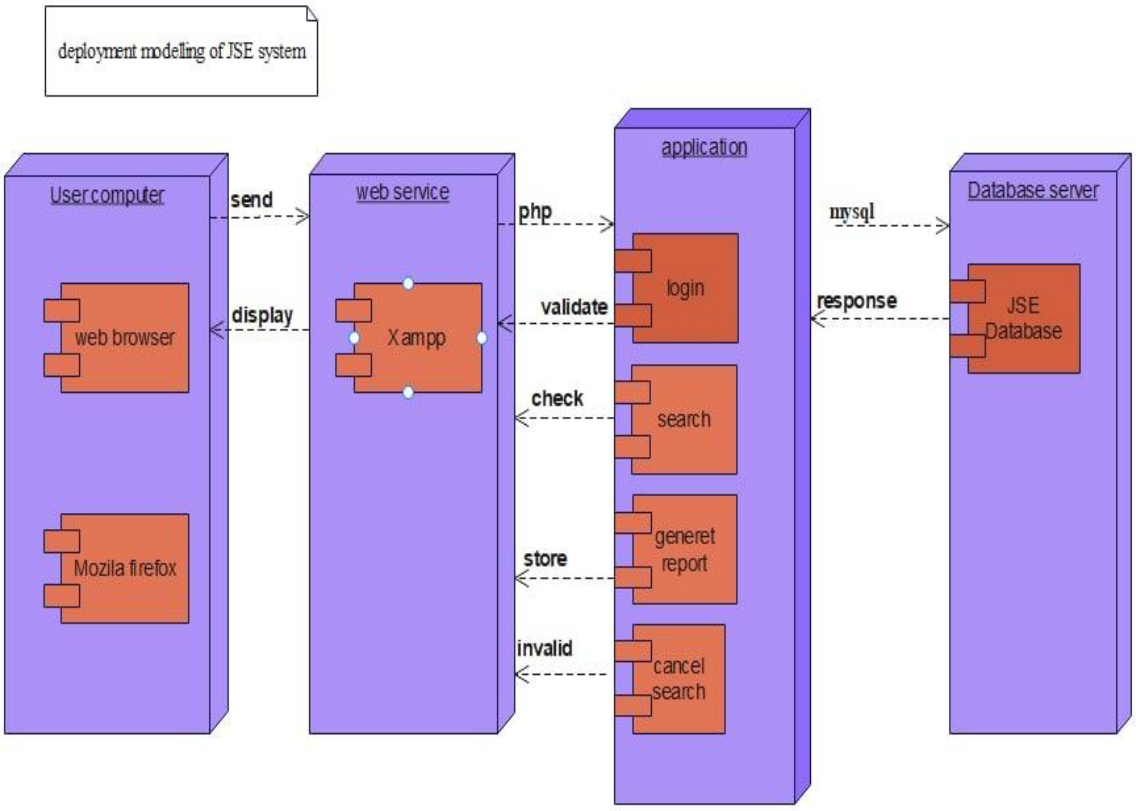


Figure 4.5: deployment modelling of JSE system

5.2.4 Detailed Class Diagram

Class diagrams in the unified modelling language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.

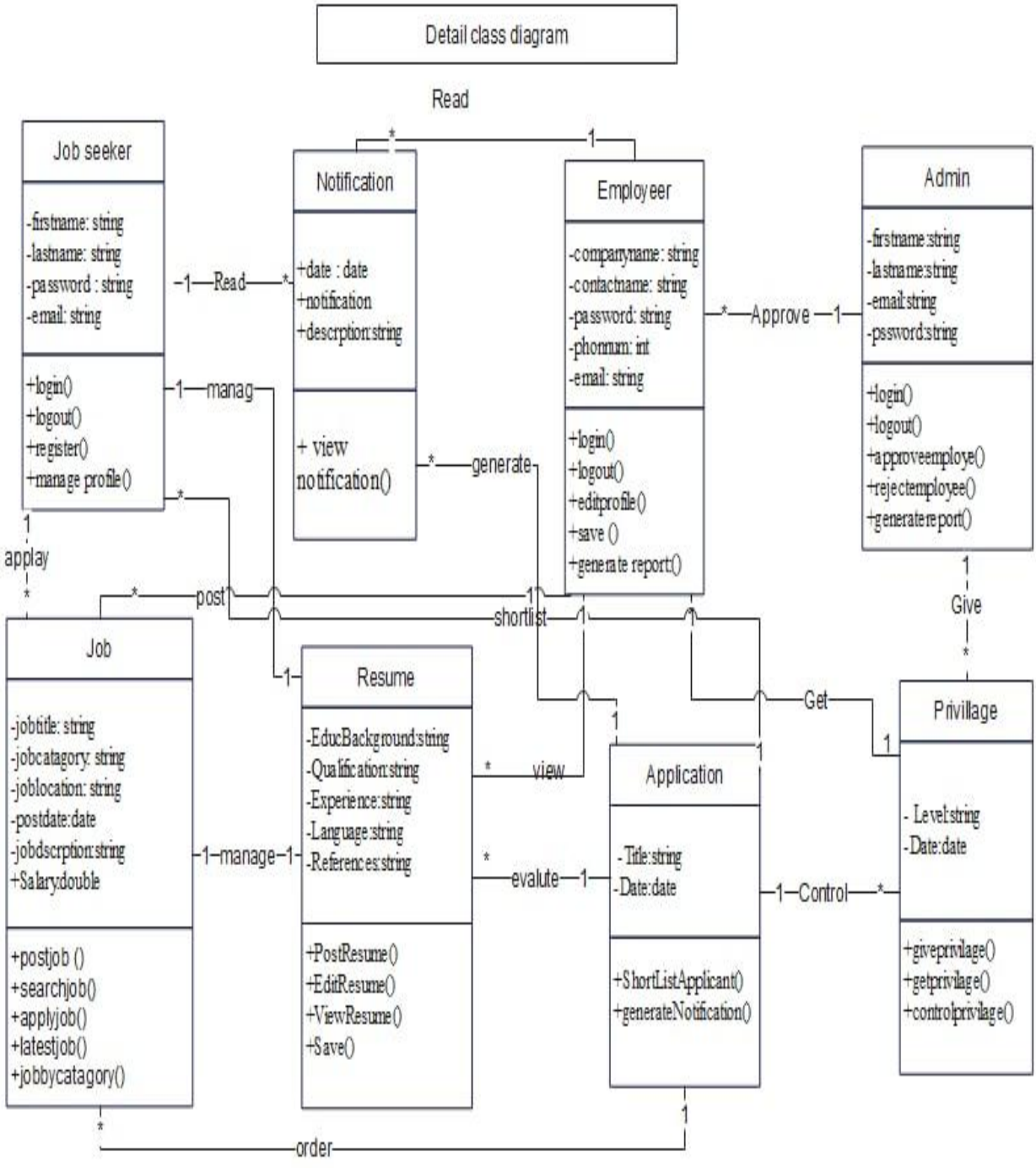
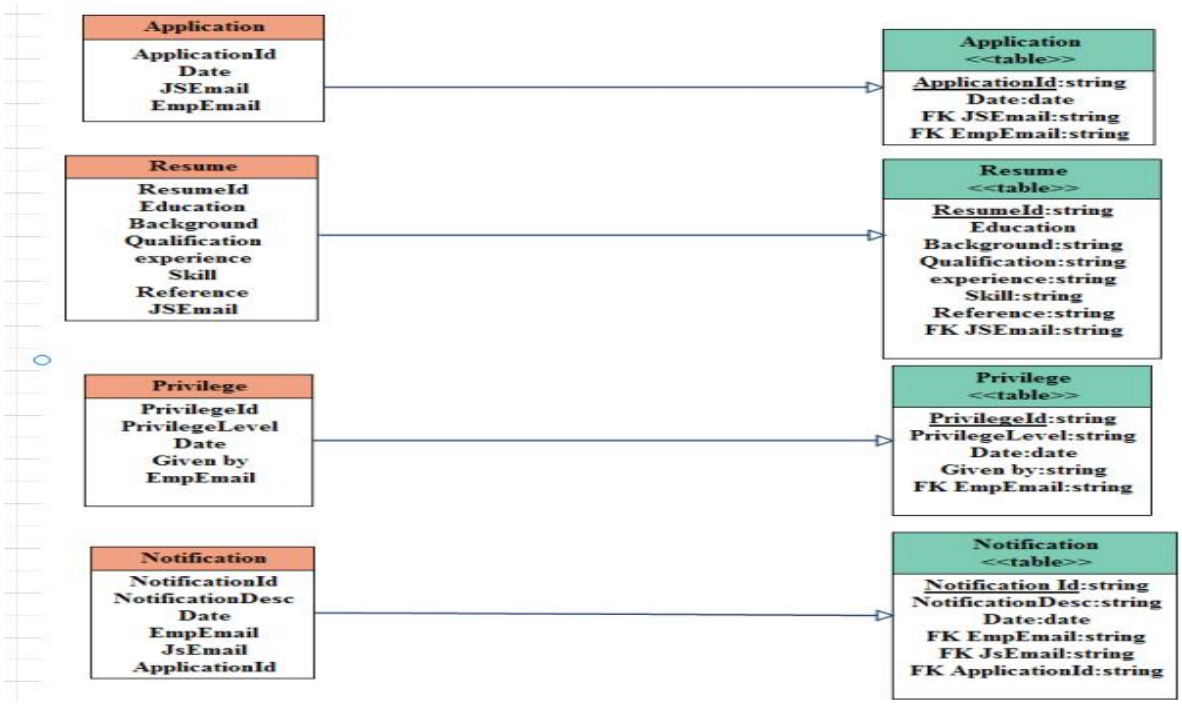


Figure 5.6: Detailed Class Diagram

5.4.5 Persistent Data Management

Persistence modelling allows the programs in the system to share and do operations consistently. In order to store and access data persistently we identified the major tables that will be implemented on the application. The Persistent model mapping and relationship among the identified tables of each are shown as follow



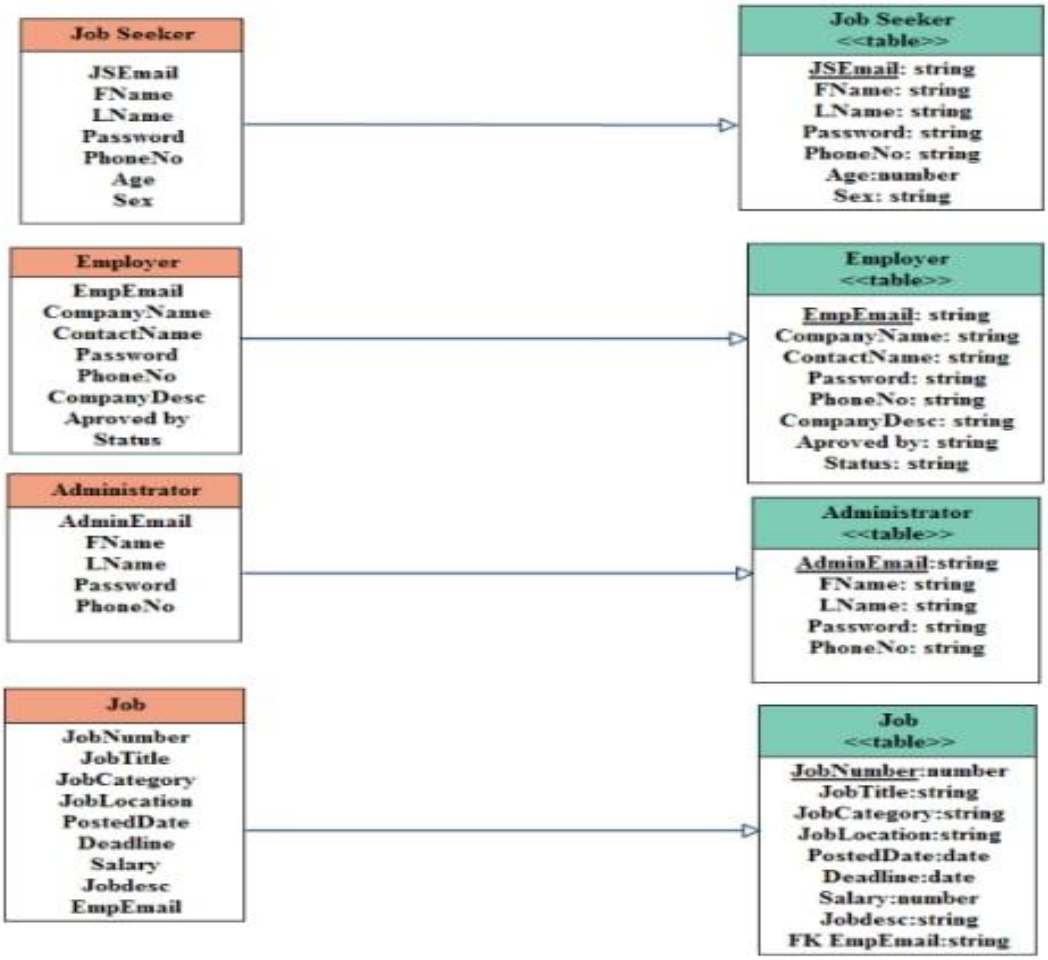


Figure 5.7: Persistent model mapping of Job Search Engine

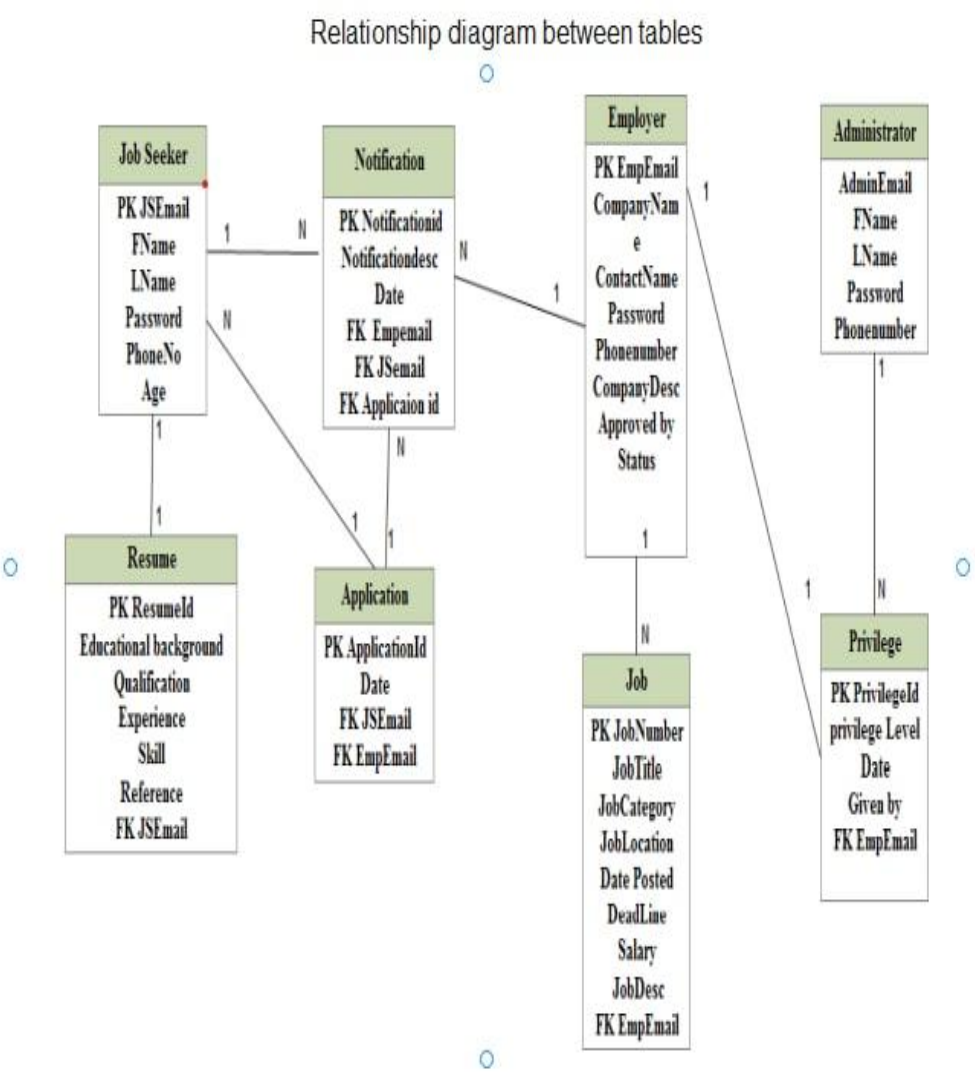


Figure 5.8: Relationship between tables

5.2.6 Access Control and Security

Access control and security describes security issues, such as the selection of an authentication mechanism, the use of encryption, and the management of keys. Access controls are ways used to prevent unauthorized access of resources and used to achieve security goals i.e.

- Confidentiality: data need to be hidden from unauthorized access,
- Integrity: protected from unauthorized change,
- Availability: the right person should access the right thing.

Table 5.1: Access Control and Security

Privilege	Actor		
	Administrator	Job Seeker	Employer
Login	✓	✓	✓
Register		✓	✓
Search Job		✓	
Apply for Job		✓	
Manage Resume		✓	
Manage Profile	✓	✓	✓
Post Job			✓
Approve Employer	✓		
View Eligible Applicant			✓
View Notification		✓	✓
Generate report	✓		✓
View Job By Location		✓	
View Job Detail		✓	
View Latest Job		✓	
View Job By Category		✓	

5.3 Packages

This section describes the decomposition of subsystems into packages and the file organization of the code. This includes an overview of each package, its dependencies with

other packages, and its expected usage. Package is an organized and functionality-based set of related interfaces and classes. Packages organize classes that belong to the same category or provide similar functionality. And security management package.

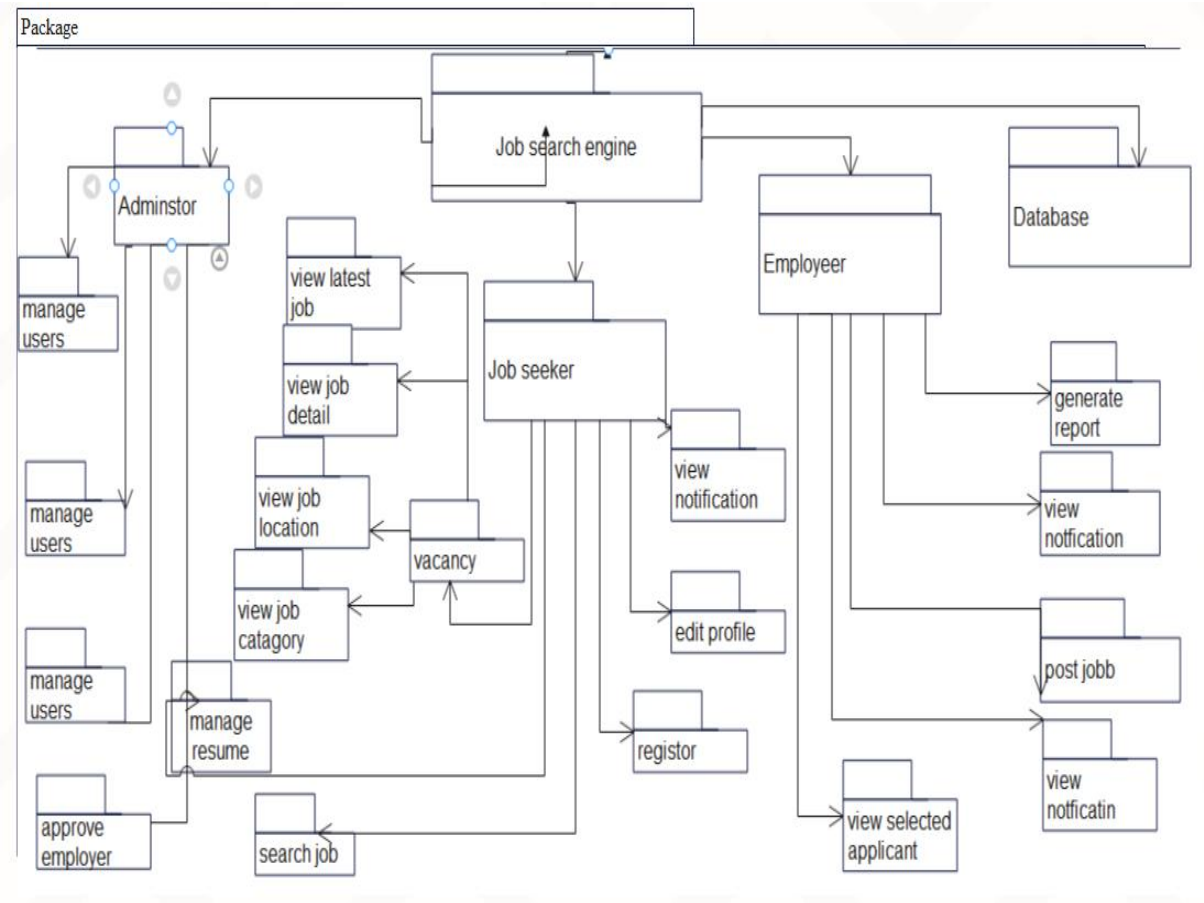


Figure 5.9: package diagram

5.4 Algorithm Design

Algorithm for login

1. Get Username and password
2. If user name in database is equal to the entered username and the password is equal to the entered password
3. Then login successful
4. Else login failed
5. End if get username and password

   If file exist then

   Read password from database

If file. Password==entered password

Login successful

Else

Print “incorrect username or password”

End IF

Else

Print “incorrect username or password”

End

Algorithm for register job seeker

1. Go to home
2. Click on jobseeker link
3. Registration form is display
4. Fill job seeker personal data
5. Click on submit button

IF(valid input)

Display “successfully register message”

ELSE

Display” invalid input message”

ENDIF

5.5 User Interface Design

The goal of our system user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-cantered design) so in our project we have designed sample of the user interfaces that increase the user experience.

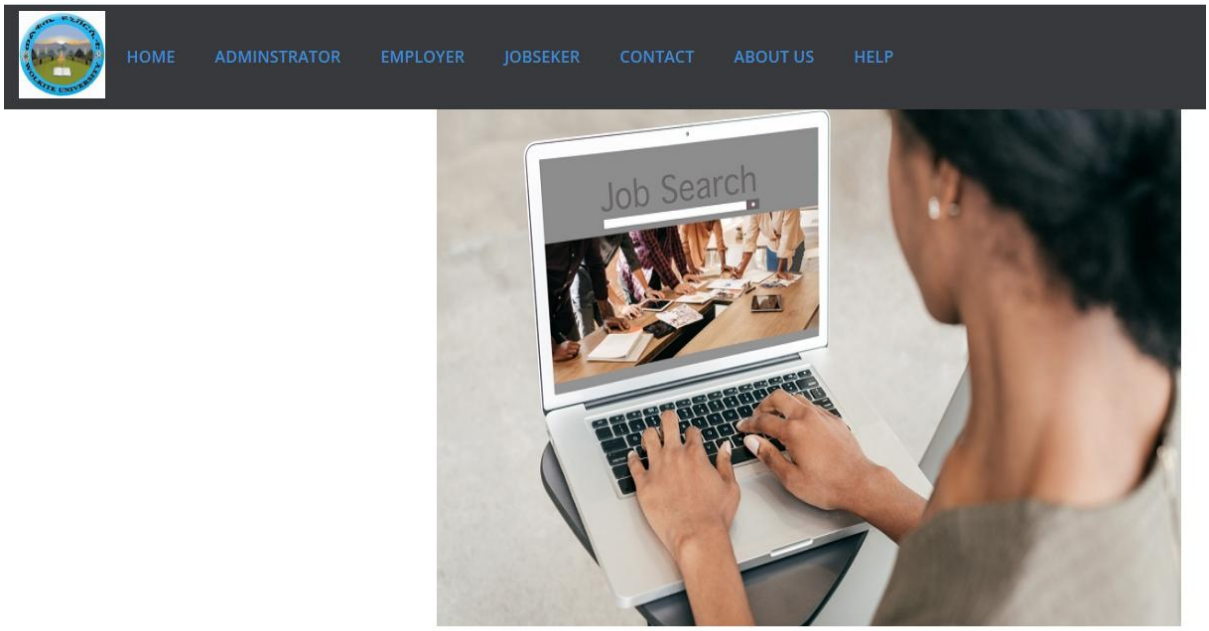


Figure 5.10: User Interface for Homepage

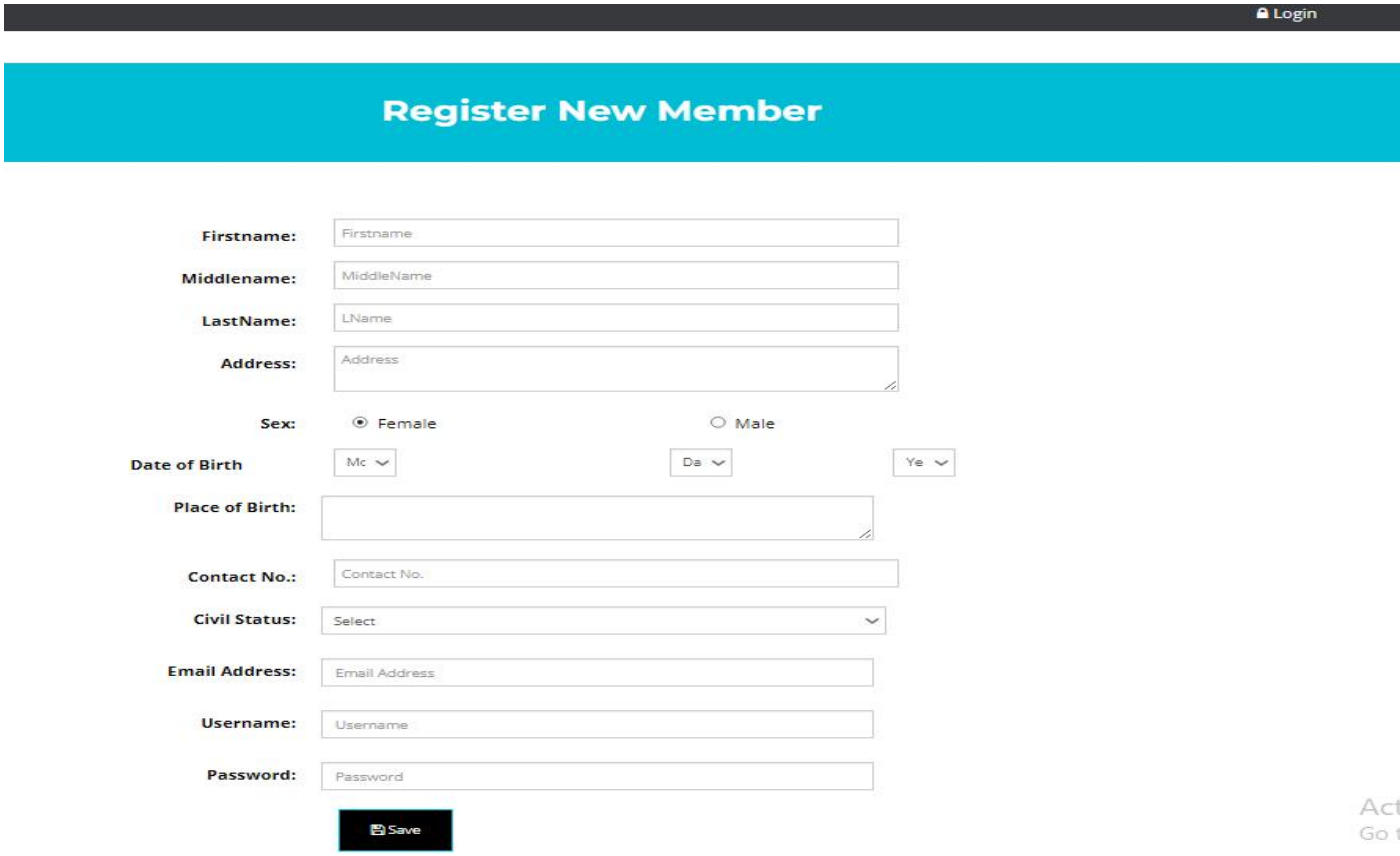


Figure 5.11: User interface for registration of JobSeker

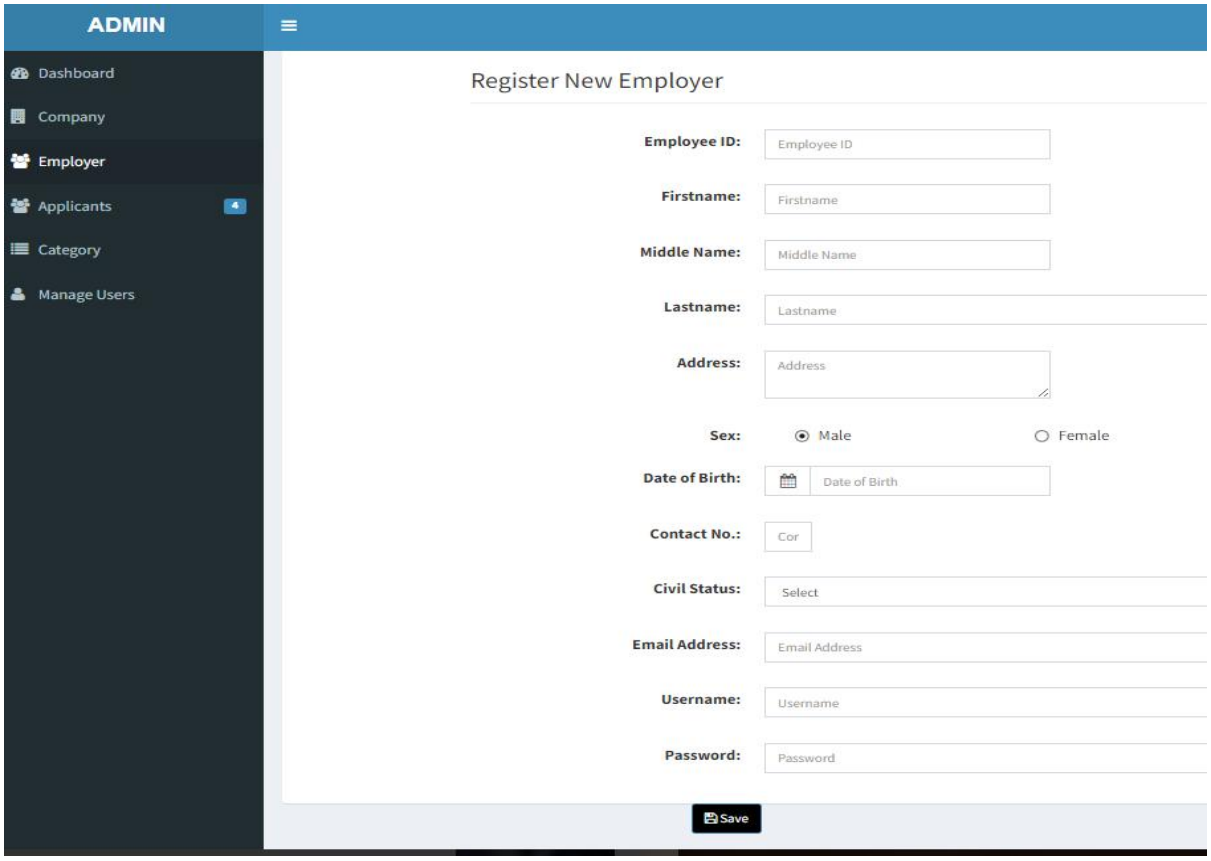


figure 5.12 User interface for registration of employer

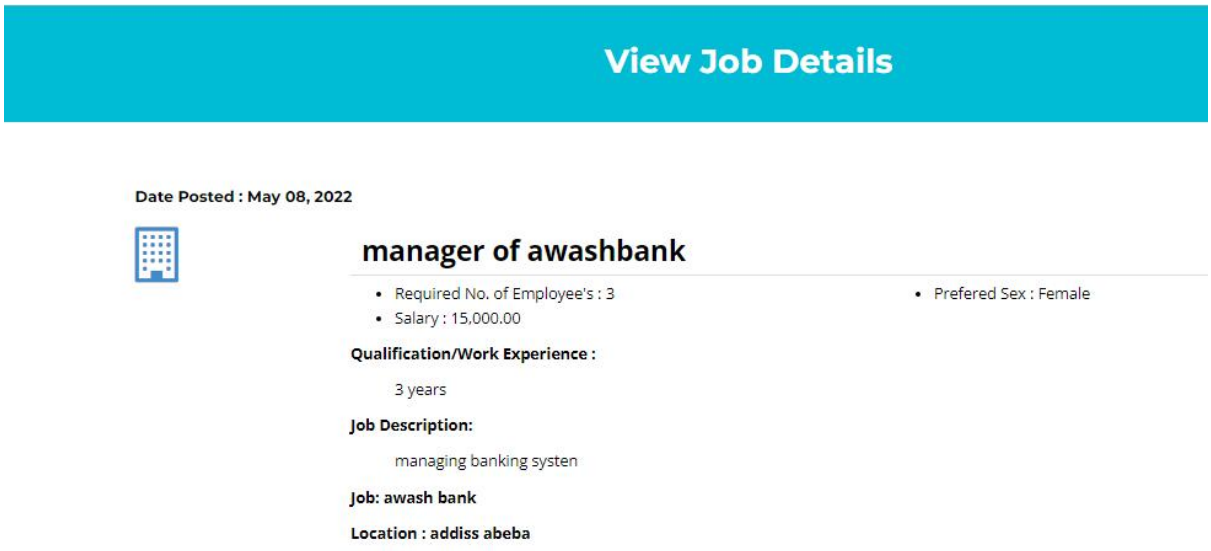


figure 5.13 User interface for view job Details

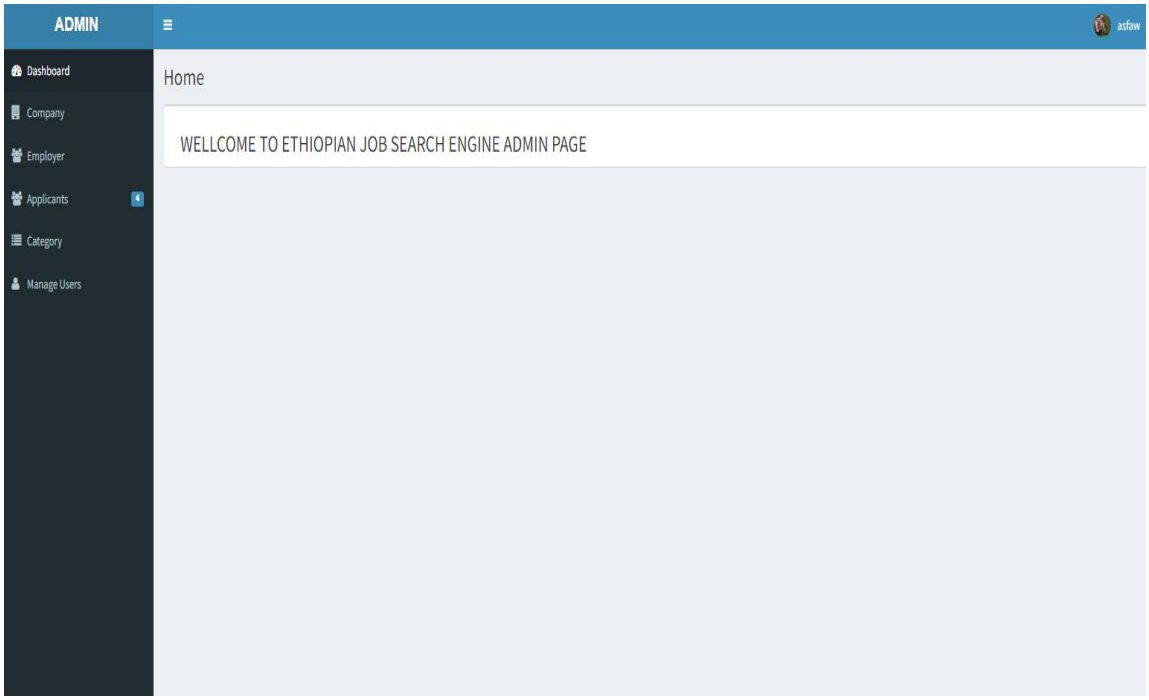


figure 5.14: User interface for admin page

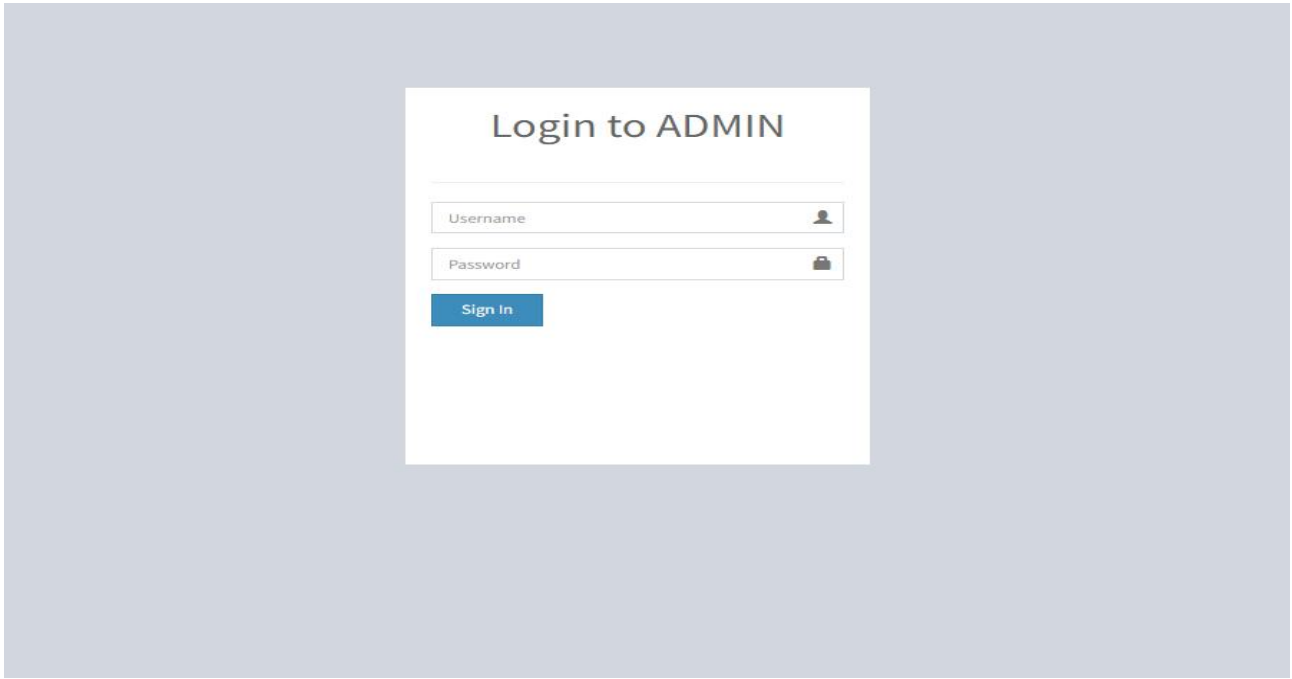


figure 5.15: User interface for admin login

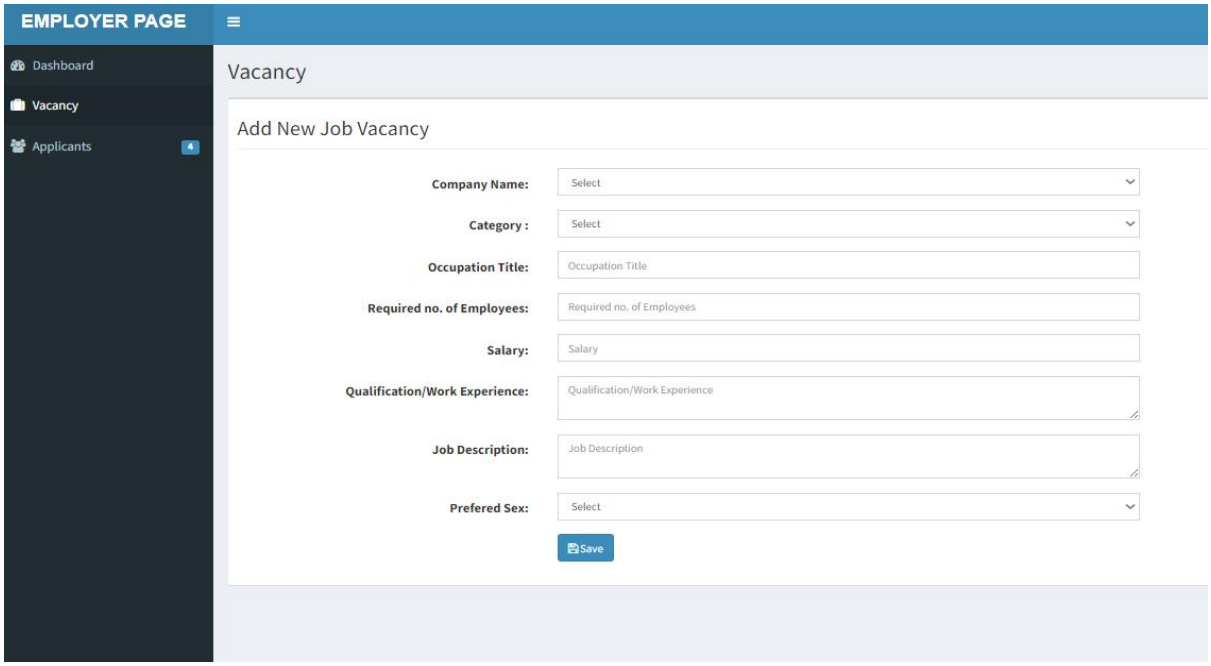


figure 5.16: User interface for add vacancy

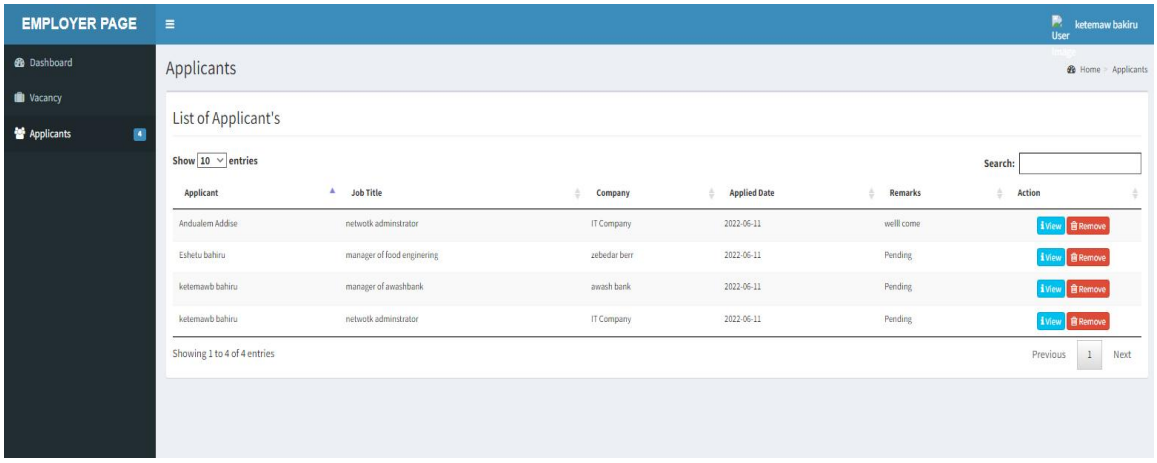


figure 5.17: User interface for list of applicants

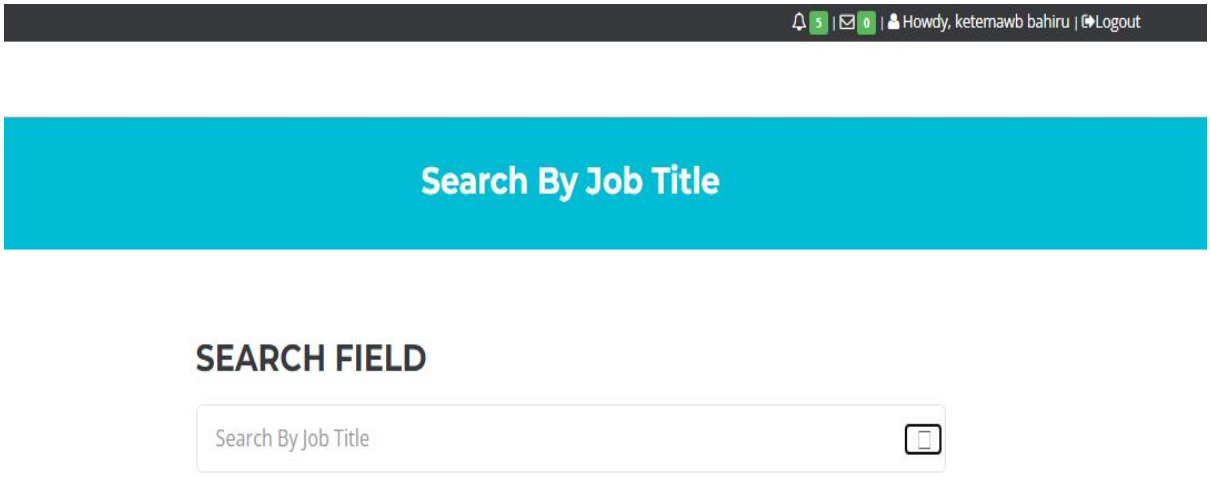


figure 5.18: User interface for search job

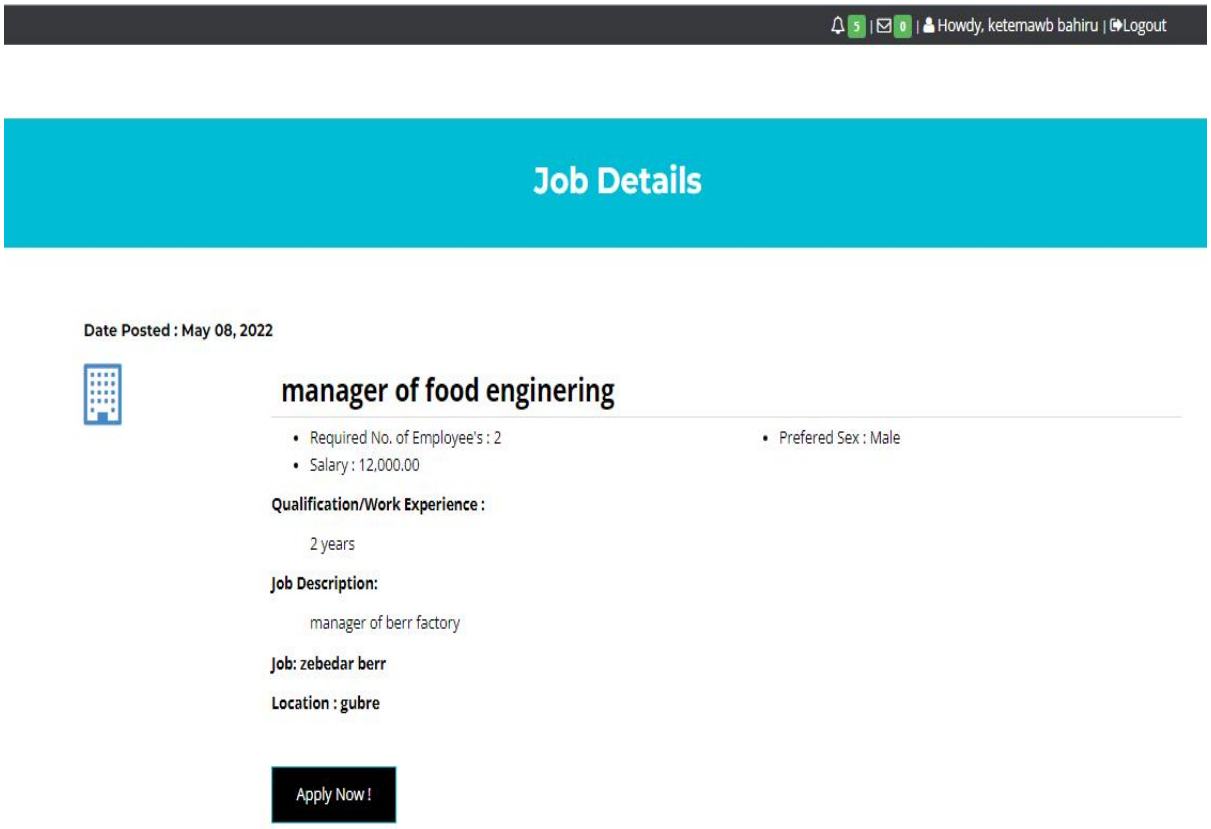


figure 5.19: User interface for apply job

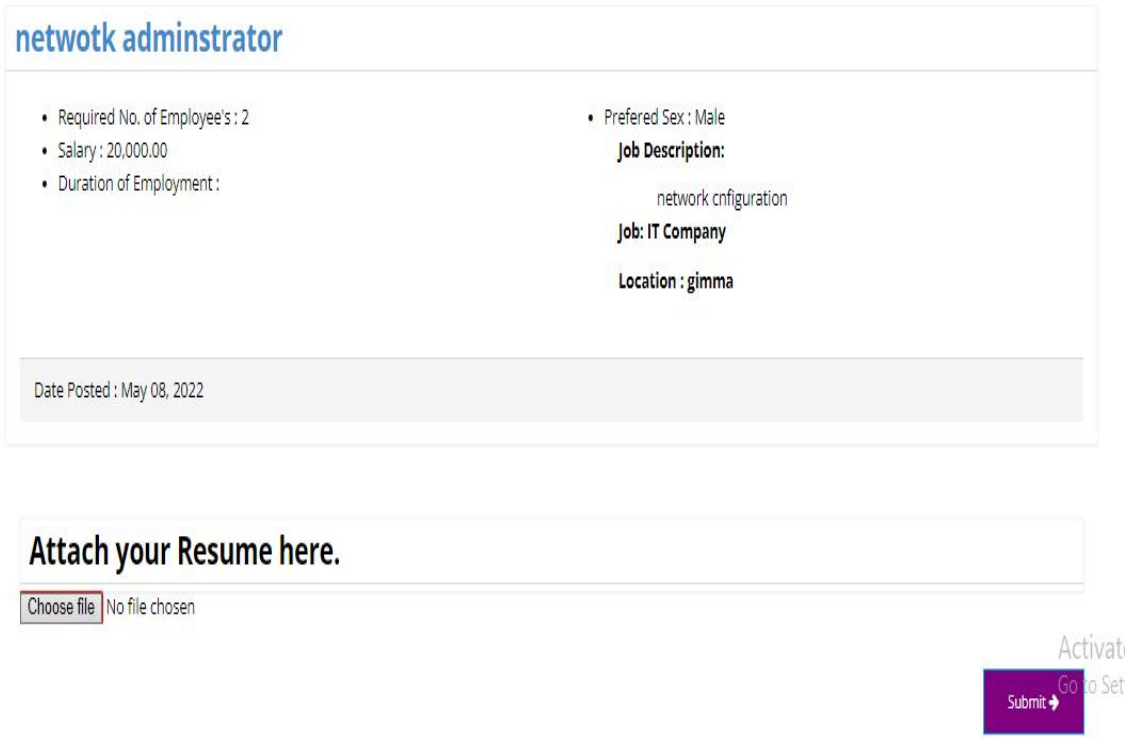


figure 5.20: User interface for manage resume

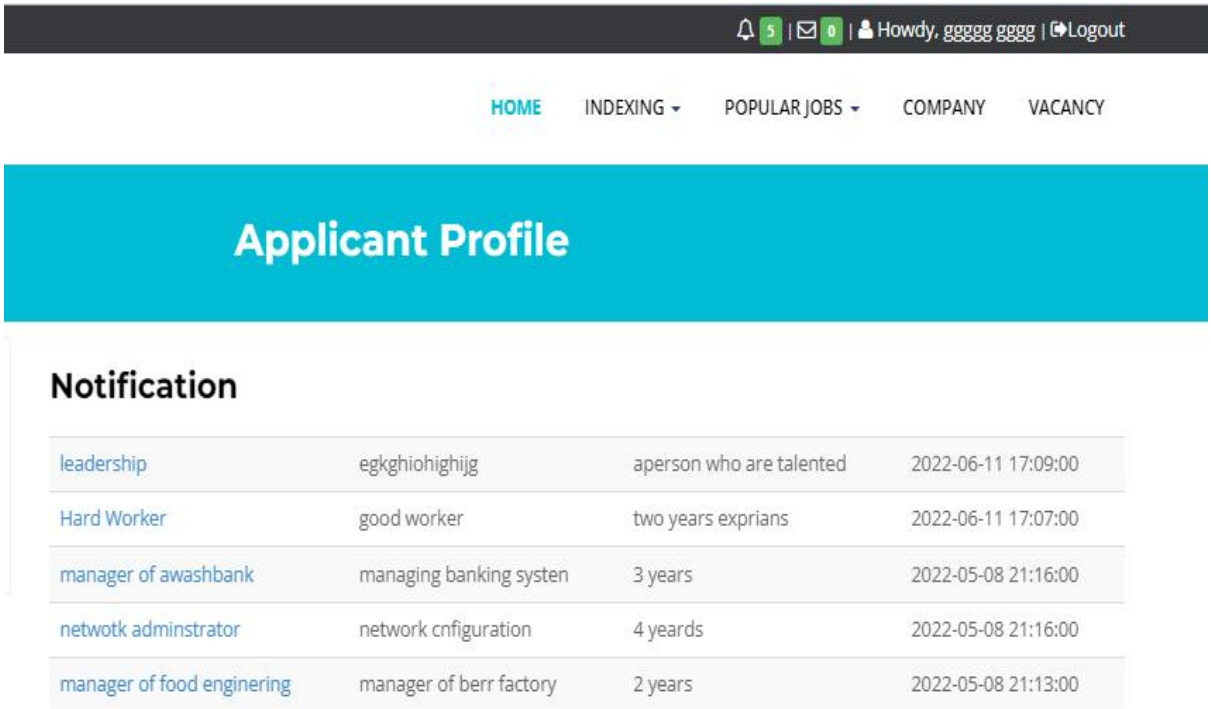


figure 5.21: User interface for view notification

CHAPTER SIX

6. Implementation and Testing

6.1 Testing

Developing software is a complex process. No matter how hard we try to eliminate all faults simply by going through the phases of requirements, design and implementation, however through good practice we can make sure that most series fault does not occur in the first place.

6.2 Testing Tools and Environment

Testing tools are important for the success of testing phase and naturally the success of product. In our unit testing phase Sublime Text3, Google Chrome and XAMPP are used.

Sublime text3:- is used to edit PHP, HTML, JavaScript and CSS code.

XAMPP: - is an open source tool used to handle the administration of MYSQL database. It allows user to view, modify, add and delete tables and their records in database. This tool will be used in order to check the correctness of the database and database related part of the project.

6.3 Unit Testing

It is done at the source or code level for language-specific programming errors such as bad syntax, logic errors or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the program's correctness.

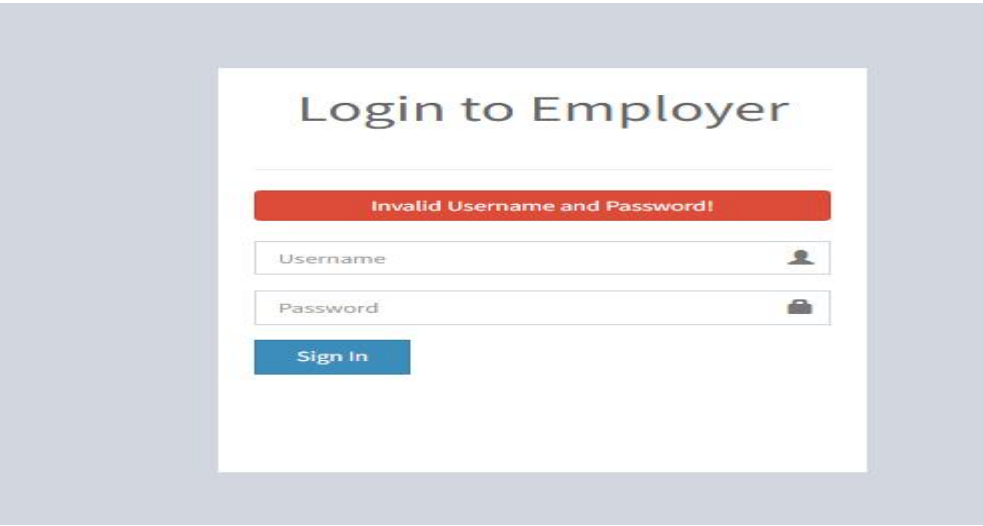


Figure 6. 1 Unit Testing

6.4 Integration Testing

We have the specific permissions related to each user type (authorization) and authentication mechanism. Our integration testing procedure is given below.

- Firstly, applicant can manage resume and send to employer
- Then the employer can view his resume and send response to the applicants

6.5 Implementation of User Interface

```
// advanced search

<style type="text/css">

#content {

min-height: 500px;

}

#content .panel {

padding: 10px;

}

.panel-body label {

font-size: 20px;

}

.panel-body input {

font-size: 15px;

}

.panel-body > .row {

margin-bottom:10px;}

</style>

<form action="index.php?q=result&searchfor=advancesearch" method="POST">

<section id="content">
```

```
<div class="container content">

<div class="col-sm-2"></div>

<div class="col-sm-8">

<div class="panel">

<div class="panel-header"></div>

<div class="panel-body">

<div class="row">

<div class="col-sm-12 search1">

<label class="col-sm-3">SEARCH:</label>

<div class="col-sm-9">

<input class="form-control" type="" name="SEARCH" placeholder="Search For">

</div>

<div class="row">

<div class="col-sm-12 search1">

<label class="col-sm-3">COMPANY:</label>

<div class="col-sm-9">

<select class="form-control" name="COMPANY">

<option value="">All</option>

<?php

$sql = "SELECT * FROM tblcompany";

$mydb->setQuery($sql);

$res = $mydb->loadResultList();

foreach ($res as $row) {
```

```
echo '<option>'.$row->COMPANYNAME.'</option>';

}

?>

</select>

</div>

<div class="row">

<div class="col-sm-12 search1">

<label class="col-sm-3">CATEGORY:</label>

<div class="col-sm-9">

<select class="form-control" name="CATEGORY">

<option value="">All</option>

<?php

$sql = "SELECT * FROM `tblcategory`";

$mydb->setQuery($sql);

$res = $mydb->loadResultList();

foreach ($res as $row) {

echo '<option>'.$row->CATEGORY.'</option>';}?>

</select>

</div>

<div class="row">

<div class="col-sm-12 search1">

<label class="col-sm-3"></label>

<div class="col-sm-9">
```

```
<input type="submit" name="submit" class="btn btn-success">
</div>
<div class="col-sm-2"></div>
</div>
</section>
</form>
// admin login
<?php
require_once("../include/initialize.php");
 ?>
  <?php
 // login confirmation
  if(isset($_SESSION['ADMIN_USERID'])){
    redirect(web_root."admin/index.php");
  }
  ?>
 <!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>ADMIN | Log in</title>
  <!-- Tell the browser to be responsive to screen width -->
```

```
  <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" name="viewport">
  <!-- Bootstrap 3.3.5 -->
  <link rel="stylesheet" href="<?php echo web_root;?>bootstrap/css/bootstrap.min.css">
  <!-- Font Awesome -->
  <link rel="stylesheet" href="<?php echo web_root;?>plugins/font-awesome/css/font-awesome.min.css">
  <!-- Theme style -->
  <link rel="stylesheet" href="<?php echo web_root;?>dist/css/AdminLTE.min.css">
  <!-- iCheck -->
  <link rel="stylesheet" href="<?php echo web_root;?>plugins/iCheck/square/blue.css">
  <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
  <!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
  <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
  <![endif]-->
</head>
<body class="hold-transition login-page">
<div class="login-box">
  <div class="login-box-body" style="min-height: 400px;">
    <h1 class="login-box-msg">Login to ADMIN</h1>
    <hr/>
```

```
<p><?php check_message(); ?></p>
<form action="" method="post">
 <div class="form-group has-feedback">
  <input type="text" class="form-control" placeholder="Username" name="user_email">
  <span class="glyphicon glyphicon-user form-control-feedback"></span>
 </div>
 <div class="form-group has-feedback">
  <input type="password" class="form-control" placeholder="Password" name="user_pass">
  <span class="glyphicon glyphicon-lock form-control-feedback"></span>
 </div>
  <div class="row">
  <!-- <div class="col-xs-8"> -->
  <!--   <div class="checkbox icheck">
     <label>
      <input type="checkbox"> Remember Me
     </label>
    </div>
   <!-- /.col -->
  <div class="col-xs-4">
   <button type="submit" name="btnLogin" class="btn btn-primary btn-block btn-flat">Sign In</button>
  </div>
```

```
    </form>
  </div>
  <!-- /.login-box-body -->
</div>
<!-- /.login-box -->
<?php
if(isset($_POST['btnLogin'])){
  $email = trim($_POST['user_email']);
  $upass  = trim($_POST['user_pass']);
  $h_upass = sha1($upass);
   if ($email == '' OR $upass == '') {
      message("Invalid Username and Password!", "error");
      redirect("login.php");

    } else {
  //it creates a new objects of member
    $user = new User();
    //make use of the static function, and we passed to parameters
    $res = $user->userAuthentication($email, $h_upass);
    // if ($res==true) {
    //    message("WELLCOME TO ".$_SESSION['ROLE']."","success");
      if ($_SESSION['ROLE']=='Administrator'){
        $_SESSION['ADMIN_USERID'] = $_SESSION['USERID'];
```

```
        $_SESSION['ADMIN_FULLNAME'] = $_SESSION['FULLNAME'] ;
        $_SESSION['ADMIN_USERNAME'] =$_SESSION['USERNAME'];
        $_SESSION['ADMIN_ROLE'] = $_SESSION['ROLE'];
        $_SESSION['ADMIN_PICLOCATION'] = $_SESSION['PICLOCATION'];

        unset( $_SESSION['USERID'] );
        unset( $_SESSION['FULLNAME'] );
        unset( $_SESSION['USERNAME'] );
        unset( $_SESSION['PASS'] );
        unset( $_SESSION['ROLE'] );
        unset($_SESSION['PICLOCATION']);

         redirect(web_root."admin/index.php");  } }
 }?>
<script src="<?php echo web_root;?>plugins/jQuery/jQuery-2.1.4.min.js"></script>
<script src="<?php echo web_root;?>bootstrap/js/bootstrap.min.js"></script>
<script src="<?php echo web_root;?>plugins/iCheck/icheck.min.js"></script>
<script>
  $(function () {
    $('input').iCheck({
      checkboxClass: 'icheckbox_square-blue',
      radioClass: 'iradio_square-blue',
      increaseArea: '20%' // optional
```

```
    });
  });
</script>
</body>
</html>
  // Sample Home code
  <section id="banner" style=" height: 500px;">
  <!-- Slider -->
       <div id="main-slider" class="flexslider" style="background-color:white;">
          <ul class="slides">
            <li>
              <img  src="<?php echo web_root;?>plugins/home-plugins/img/slides/good.jpg" alt="" style=" width: 600px; height: 500px;  margin-left: 400px;background-color: white;" />
              <div class="flex-caption" </div>
            </li>
            <li>
              <img src="<?php echo web_root; ?>plugins/home-plugins/img/slides/bad.jpg" alt="" style=" width: 600px; height: 500px; margin-left: 400px; background-color: white;"/>
              <div class="flex-caption"</div>
            </li>
          </ul>
       </div>
      </section>
```

```
<!-- end slider -->
<section id="content"
<div class="container">
    <div class="row">
   <div class="col-md-12">
    <div class="aligncenter"><h2 class="aligncenter">list of Company</h2></div>
    <br/>
   </div>
  </div>
  <?php
   $sql = "SELECT * FROM `tblcompany`";
   $mydb->setQuery($sql);
   $comp = $mydb->loadResultList();
   foreach ($comp as $company ) {?>
        <div class="col-sm-4 info-blocks">
           <i class="icon-info-blocks fa fa-building-o"></i>
           <div class="info-blocks-in">
              <h3><?php echo $company->COMPANYNAME;?></h3>
              <!-- <p><?php echo $company->COMPANYMISSION;?></p> -->
              <p>Address :<?php echo $company->COMPANYADDRESS;?></p>
              <p>Contact No. :<?php echo $company->COMPANYCONTACTNO;?></p>
           </div>
        </div>
```

```
  <?php } ?>
 </div>
 </section>
  </div>
 </div>
    <?php
require_once ("include/initialize.php");
$action = (isset($_GET['action']) && $_GET['action'] != '') ? $_GET['action'] : '';
switch ($action) {
        case 'submitapplication' :
        doSubmitApplication();
        break;
        case 'register' :
        doRegister();
        break;
        case 'login' :
        doLogin();
        break;
        }
function doSubmitApplication() {
        global $mydb;
                $jobid  = $_GET['JOBID'];
                $autonum = New Autonumber();
```

```
$applicantid = $autonum->set_autonumber('APPLICANT');

$autonum = New Autonumber();

$fileid = $autonum->set_autonumber('FILEID');

@$picture = UploadImage();

@$location = "photos/". $picture ;

if ($picture=="") {

    # code...

redirect(web_root."index.php?q=apply&job=".$jobid."&view=personalinfo");

}else{

if (isset($_SESSION['APPLICANTID'])) {

$sql = "INSERT INTO `tblattachmentfile` (FILEID,`USERATTACHMENTID`, `FILE_NAME`, `FILE_LOCATION`, `JOBID`) VALUES ('". date('Y').$fileid->AUTO."','{$_SESSION['APPLICANTID']}','Resume','{$location}','{$jobid}')";

$mydb->setQuery($sql);

$res = $mydb->executeQuery();

doUpdate($jobid,$fileid->AUTO);


    }else{


        $sql = "INSERT INTO `tblattachmentfile` (FILEID,`USERATTACHMENTID`, `FILE_NAME`, `FILE_LOCATION`, `JOBID`)

        VALUES ('". date('Y').$fileid->AUTO."','". date('Y').$applicantid->AUTO."','Resume','{$location}','{$jobid}')";

        // echo $sql;exit;
```

```
                    $mydb->setQuery($sql);
                    $res = $mydb->executeQuery();

                    doInsert($jobid,$fileid->AUTO);

                    $autonum = New Autonumber();
                $autonum->auto_update('APPLICANT');
                    }
            }
            $autonum = New Autonumber();
        $autonum->auto_update('FILEID');
}
function doInsert($jobid=0,$fileid=0) {
    if (isset($_POST['submit'])) {
    global $mydb;

                    $birthdate =  $_POST['year'].'-'.$_POST['month'].'-'.$_POST['day'];
                    $age = date_diff(date_create($birthdate),date_create('today'))->y;
                    if ($age < 20){
                    message("Invalid age. 18 years old and above is allowed.", "error");
                    redirect("index.php?q=apply&view=personalinfo&job=".$jobid);
                    }else{
                    $autonum = New Autonumber();
```

```
$auto = $autonum->set_autonumber('APPLICANT');

$applicant =New Applicants();

$applicant->APPLICANTID = date('Y').$auto->AUTO;

$applicant->FNAME = $_POST['FNAME'];

$applicant->LNAME = $_POST['LNAME'];

$applicant->MNAME = $_POST['MNAME'];

$applicant->ADDRESS = $_POST['ADDRESS'];

$applicant->SEX = $_POST['optionsRadios'];

$applicant->CIVILSTATUS = $_POST['CIVILSTATUS'];

$applicant->BIRTHDATE = $birthdate;

$applicant->BIRTHPLACE = $_POST['BIRTHPLACE'];

$applicant->AGE = $age;

$applicant->USERNAME = $_POST['USERNAME'];

$applicant->PASS = sha1($_POST['PASS']);

$applicant->EMAILADDRESS = $_POST['EMAILADDRESS'];

$applicant->CONTACTNO = $_POST['TELNO'];

$applicant->DEGREE = $_POST['DEGREE'];

$applicant->create();

$sql = "SELECT * FROM `tblcompany` c,`tbljob` j WHERE c.`COMPANYID`=j.`COMPANYID` AND JOBID = '{$jobid}'" ;

$mydb->setQuery($sql);

$result = $mydb->loadSingleResult();
```

```
                $jobreg = New JobRegistration();

                $jobreg->COMPANYID = $result->COMPANYID;

                $jobreg->JOBID     = $result->JOBID;

                $jobreg->APPLICANTID = date('Y').$auto->AUTO;

                $jobreg->APPLICANT          =    $_POST['FNAME']    .    '    '    .
$_POST['LNAME'];

                $jobreg->REGISTRATIONDATE = date('Y-m-d');

                $jobreg->FILEID = date('Y').$fileid;

                $jobreg->REMARKS = 'Pending';

                $jobreg->DATETIMEAPPROVED = date('Y-m-d H:i');

                $jobreg->create();

                message("Your application already submitted. Please wait for the
company confirmation if your are qualified to this job.","success");

                redirect("index.php?q=success&job=".$result->JOBID);

        }

}}

function doUpdate($jobid=0,$fileid=0) {

        if (isset($_POST['submit'])) {

        global $mydb;

                $applicant =New Applicants();

                $appl  = $applicant->single_applicant($_SESSION['APPLICANTID']);

                $sql = "SELECT * FROM `tblcompany` c,`tbljob` j WHERE
c.`COMPANYID`=j.`COMPANYID` AND JOBID = '{$jobid}'" ;

                $mydb->setQuery($sql);
```

```
                $result = $mydb->loadSingleResult();

                $jobreg = New JobRegistration();

                $jobreg->COMPANYID = $result->COMPANYID;

                $jobreg->JOBID     = $result->JOBID;

                $jobreg->APPLICANTID = $appl->APPLICANTID;

                $jobreg->APPLICANT   = $appl->FNAME . ' ' . $appl->LNAME;

                $jobreg->REGISTRATIONDATE = date('Y-m-d');

                $jobreg->FILEID = date('Y').$fileid;

                $jobreg->REMARKS = 'Pending';

                $jobreg->DATETIMEAPPROVED = date('Y-m-d H:i');

                $jobreg->create();

                message("Your application already submitted. Please wait for the company confirmation if your are qualified to this job.","success");

                redirect("index.php?q=success&job=".$result->JOBID)

        }}
function doRegister(){
        global $mydb;
        if (isset($_POST['btnRegister'])) {

                $birthdate =  $_POST['year'].'-'.$_POST['month'].'-'.$_POST['day'];

                $age = date_diff(date_create($birthdate),date_create('today'))->y;

                if ($age < 20){

                message("Invalid age. 20 years old and above is allowed.", "error");

                redirect("index.php?q=register");
```

```
}else{
$autonum = New Autonumber();
$auto = $autonum->set_autonumber('APPLICANT');
$applicant =New Applicants();
$applicant->APPLICANTID = date('Y').$auto->AUTO;
$applicant->FNAME = $_POST['FNAME'];
$applicant->LNAME = $_POST['LNAME'];
$applicant->MNAME = $_POST['MNAME'];
$applicant->ADDRESS = $_POST['ADDRESS'];
$applicant->SEX = $_POST['optionsRadios'];
$applicant->CIVILSTATUS = $_POST['CIVILSTATUS'];
$applicant->BIRTHDATE = $birthdate;
$applicant->BIRTHPLACE = $_POST['BIRTHPLACE'];
$applicant->AGE = $age;
$applicant->USERNAME = $_POST['USERNAME'];
$applicant->PASS = sha1($_POST['PASS']);
$applicant->EMAILADDRESS = $_POST['EMAILADDRESS'];
$applicant->CONTACTNO = $_POST['TELNO'];
$applicant->DEGREE = $_POST['DEGREE'];
$applicant->create();
$autonum = New Autonumber();
$autonum->auto_update('APPLICANT');
```

```
                    message("You are successfully registered to the site. You can login now!","success");
                    redirect("index.php?q=success");        }}}
function doLogin(){
        $email = trim($_POST['USERNAME']);
        $upass  = trim($_POST['PASS']);
        $h_upass = sha1($upass);
  //it creates a new objects of member
    $applicant = new Applicants();
    //make use of the static function, and we passed to parameters
    $res = $applicant->applicantAuthentication($email, $h_upass);
    if ($res==true) {
        message("You are now successfully login!","success");
      // $sql="INSERT INTO `tbllogs` (`USERID`,USERNAME, `LOGDATETIME`, `LOGROLE`, `LOGMODE`)
      //      VALUES ('".$_SESSION['USERID']."','".$_SESSION['FULLNAME']."','".date('Y-m-d H:i:s')."','".$_SESSION['UROLE']."','Logged in')";
      //    mysql_query($sql) or die(mysql_error());
        redirect(web_root."applicant/");
    }else{
         echo "Account does not exist! Please contact Administrator.";
    }
function UploadImage($jobid=0){
        $target_dir = "applicant/photos/";
```

```
$target_file = $target_dir . date("dmYhis") . basename($_FILES["picture"]["name"]);
$uploadOk = 1;
$imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);
if($imageFileType != "jpg" || $imageFileType != "png" || $imageFileType != "jpeg"
|| $imageFileType != "gif" ) {
 if (move_uploaded_file($_FILES["picture"]["tmp_name"], $target_file)) {
  return  date("dmYhis") . basename($_FILES["picture"]["name"]);
 }else{
  message("Error Uploading File","error");
redirect(web_root."index.php?q=apply&job=".$jobid."&view=personalinfo");
 }}
 else{
  message("File Not Supported","error");
 redirect(web_root."index.php?q=apply&job=".$jobid."&view=personalinfo");

}} ?>
```

CHAPTER SEVEN

7. Conclusions and Recommendations

7.1 Conclusion

As the scope of this project describes we developed the development of job search engine for Ethiopian job search to make it reliable. The system will help to search job, view new job, apply for organization that has been posted the , deliver message to job applier to be approved or not and perform additional task that will be performed by the system. So the system will Minimize the work load,Minimize the time required to perform task,Reduce the number of employee involved .This document use the object oriented technology called UML (unified modeling language) that enable the user to understand the software easily.

7.2 Recommendation

The system we have developed is Web based system it needs a skilled person to work with the system. So, we recommend the system should be entitled to the responsible and skilled person. We highly recommend the system should be kept in highly safe and favorable condition.

References

[1] ethiojob.net, mjobs.com. (2004, 05 14). Retrieved from employeeethiopia.com.

[2] https://www.nih.gov/research-training/science-education. (2021, Dec 20). Retrieved jun 1, 2022, from https://www.nih.gov/research-training/science-education: https://www.nih.gov/research-training/science-education

[3] JEFFERY A.HOFFER, J. (2004). Modern system analysis and design (third edition ed.). Retrieved 01 15, 2022

[3[research-training/science-education. (2021, Dec 20). Retrieved jun 1, 202, from https://www.nih.gov/research-training/science-education.

[4] ScottAmbler. (2003). The_Elements_of_UML_Style.

[5] supplements/nih4/technology/guide. (208, 05 16). Retrieved from http://www.science.education.nih.gov/.