

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF COMPUTER SCINECE

TITLE: ONLINE JOB PORTAL APPLICATION FOR GURAGHE ZONE

Prepared by:

Name ID

1. KEFIYALEW KUNTA NSR/0841/12

2. BIRUK ZERIHUN NSR/0375/12

3. AMIR JEMAL NSR/1787/12

Advisor Mr. Berihun

Wolkite University,Wolkite,Ethiopia

February 2023

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF COMPUTER SCINECE

**ONLINE JOB PORTAL APPLICATION FOR GURAGHE ZONE**

SUBMITTED TO DEPARTMENT OF COMPUTER SCIENCE

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR

THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Prepared by:

Name ID

1. KEFIYALEW KUNTA NSR/0841/12

2. BIRUK ZERIHUN NSR/0375/12

3. AMIR JEMAL NSR/1787/12

ADVISOR : MR. BERIHUN

Wolkite University, Wolkite, Ethiopia

February 2023

# DECLARATION

This is to declare that this project work which is done under the supervision of Mr Berihun and having the title developing Online Job Portal Application For Gurage Zone is the sole contribution of: Kefyalew Kunta, Amir Jemal and Biruk Zerihun

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 Members:**

Full Name Signature

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# APPROVAL FORM

This is to confirm that the project report entitled Online Job Portal Application For Gurage Zonesubmitted to Wolkite University, College of Computing and Informatics at the Department of Computer science by : Kefyalew Kunta, Amir Jemal and Biruk Zerihun . So that the advisor, department head who advises us and the examiner who evaluate us should be approved this proposed document by putting the following approval forms.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(Advisor Name) (Signature) (Date)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(Department Head Name) (Signature) (Date)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(Examiner-1) (Signature) (Date)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(Examiner-2) (Signature) (Date)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(Examiner-3) (Signature) (Date)

# ACKNOWLEDGEMENT

In successfully completing this project, many people have helped us. We would like to thank all those who are participate to this project. Primarily, we would thank you God for being able to complete this project with success. Secondly, we would like to express our gratitude to our advisor Mr Berihun for his guidance, support and his continuous assistance and encouragement throughout the preparation of this document as well as for the demonstration. He was both our advisor and source of information. We are also very grateful and extend our sincere thanks to some of the staff members of the Wolkite university society for their cooperation by sharing idea and giving primary and secondary sources of information what we need for this project. Finally, thanks to our friends, who have helped and given us suggestions and idea supports throughout the project.

Table of Contents

[DECLARATION i](#_Toc136215895)

[APPROVAL FORM ii](#_Toc136215896)

[ACKNOWLEDGEMENT iii](#_Toc136215897)

[LIST OF TABLES 1](#_Toc136215898)

[LIST OF FIGURES 2](#_Toc136215899)

[LIST OF ABBREVATIONS 4](#_Toc136215900)

[ABSTRACTS 5](#_Toc136215901)

[Chapter One 1](#_Toc136215902)

[1. Introduction 1](#_Toc136215903)

[1.1. Background of organization 1](#_Toc136215904)

[1.2. Statement of problem 2](#_Toc136215905)

[1.3. Objective of the Project 2](#_Toc136215906)

[1.3.1. General Objective 2](#_Toc136215907)

[1.3.2. Specific Objective 2](#_Toc136215908)

[1.4. Scope and Limitations 3](#_Toc136215909)

[1.4.1. Scope 3](#_Toc136215910)

[1.4.2. Limitations 3](#_Toc136215911)

[1.5. Feasibility Study 4](#_Toc136215912)

[1.6. Significance of the Project 4](#_Toc136215913)

[1.7. Beneficiary of the Project 5](#_Toc136215914)

[1.8. Methodology 5](#_Toc136215915)

[1.9. Project Plan 9](#_Toc136215916)

[Chapter Two 10](#_Toc136215917)

[2. DESCRIPTION OF THE EXISTING SYSTEM 10](#_Toc136215918)

[2.1 Introduction of Existing System 10](#_Toc136215919)

[2.2 Users of Existing System 11](#_Toc136215920)

[2.3 Major functions of the Existing System 11](#_Toc136215921)

[2.4 forms and other documentations of existing system 12](#_Toc136215922)

[2.5 Drawbacks of the Existing System 17](#_Toc136215923)

[2.6 Business rules 17](#_Toc136215924)

[Chapter three 19](#_Toc136215925)

[3. Proposed system 19](#_Toc136215926)

[3.1. Functional requirement 20](#_Toc136215927)

[3.2. Non-functional Requirements 21](#_Toc136215928)

[CHAPTER FOUR 24](#_Toc136215929)

[SYSTEM ANALYSIS 24](#_Toc136215930)

[4.1 System Model 24](#_Toc136215931)

[4.1.1 Use Case Model 24](#_Toc136215932)

[4.1.2 Use Case Diagram 26](#_Toc136215933)

[4.1.3 Use Case description 27](#_Toc136215934)

[4.2 Object Model 37](#_Toc136215935)

[4.2.1. Class Diagram 37](#_Toc136215936)

[4.2.2. Data Dictionary 39](#_Toc136215937)

[4.3 Dynamic Model 41](#_Toc136215938)

[4.3.1 Sequence Diagram 41](#_Toc136215939)

[4.3.2 Activity Diagram 50](#_Toc136215940)

[4.3.3 State Chart Diagram 55](#_Toc136215941)

[CHAPTER FIVE 62](#_Toc136215942)

[5. SYSTEM DESIGN 62](#_Toc136215943)

[5.1. Design Goals 62](#_Toc136215944)

[5.2. Proposed System Architecture 64](#_Toc136215945)

[Application tier 64](#_Toc136215946)

[Data-tier 64](#_Toc136215947)

[5.2.1. Subsystem Decomposition and Description 65](#_Toc136215948)

[5.2.2. Hardware/Software Mapping 67](#_Toc136215949)

[5.2.3. Detailed Class Diagram 68](#_Toc136215950)

[5.2.4. Persistent Data Management 70](#_Toc136215951)

[5.2.5. Access Control and Security 70](#_Toc136215952)

[5.3. Algorithm Design 71](#_Toc136215953)

[5.4. User interface design 73](#_Toc136215954)

[CHAPTER SIX 74](#_Toc136215955)

[6. IMPLEMENTATION AND TESTING 74](#_Toc136215956)

[6.1. Implementation 74](#_Toc136215957)

[6.2. Implementation of the Database 74](#_Toc136215958)

[6.3. Implementation of Class Diagram Methods implemented in our system: 76](#_Toc136215959)

[Administrator 76](#_Toc136215960)

[Recruiter 76](#_Toc136215961)

[User 76](#_Toc136215962)

[Variables and class implemented in our system: 76](#_Toc136215963)

[User class 76](#_Toc136215964)

[Recruiter class 77](#_Toc136215965)

[Job class 77](#_Toc136215966)

[Apply class 77](#_Toc136215967)

[6.4. Configuration of Application Server 78](#_Toc136215968)

[Flexibility: 78](#_Toc136215969)

[Scalability: 78](#_Toc136215970)

[Speed: 78](#_Toc136215971)

[Simple: 78](#_Toc136215972)

[Reusable middleware: 78](#_Toc136215973)

[6.5. Configuration of Application security 79](#_Toc136215974)

[Cross site request forgery (CSRF) protection 79](#_Toc136215975)

[SQL injection protection 79](#_Toc136215976)

[Host header validation 79](#_Toc136215977)

[6.6. Testing 80](#_Toc136215978)

[6.6.1.Testing Tools and Environment 80](#_Toc136215979)

[6.6.4. Unit Testing 81](#_Toc136215980)

[6.6.5. Integration Testing 82](#_Toc136215981)

[6.6.6. System Testing 82](#_Toc136215982)

[6.6.7. Acceptance Testing 82](#_Toc136215983)

[CHAPTER SEVEN 83](#_Toc136215984)

[7. CONCLUSION AND RECOMMENDATION 83](#_Toc136215985)

[7.1. Conclusion 83](#_Toc136215986)

[7.2. Recommendation 84](#_Toc136215987)

# LIST OF TABLES

[Table 4. 1 Use Case Identification 24](#_Toc128148203)

[Table 4. 2 usecase description of login 27](#_Toc128148204)

[Table 4. 3 usecase description of registration 28](#_Toc128148205)

[Table 4. 4 usecase description of update profile 28](#_Toc128148206)

[Table 4. 5 usecase description of change password 29](#_Toc128148207)

[Table 4. 6 usecase description of view jobs 30](#_Toc128148208)

[Table 4. 7 usecase description of search jobs by category 31](#_Toc128148209)

[Table 4. 8 usecase description of apply for job 31](#_Toc128148210)

[Table 4. 9 usecase description of post job 32](#_Toc128148211)

[Table 4. 10 usecase description of remove job 33](#_Toc128148212)

[Table 4. 11 usecase description of update job detail 34](#_Toc128148213)

[Table 4. 12 usecase description of view candidates applied 35](#_Toc128148214)

[Table 4. 13 usecase description of view users 36](#_Toc128148215)

[Table 4. 14 Data dictionary for user 39](#_Toc128148216)

[Table 4. 15 Data dictionary for recruiter 39](#_Toc128148217)

[Table 4. 16 Data dictionary for job 40](#_Toc128148218)

[Table 4. 17 Data dictionary for application 41](#_Toc128148219)

[Table 5. 1 Access Control and Security 71](#_Toc128149960)

# LIST OF FIGURES

[Figure 2. 1 posting vacancy 12](#_Toc128150876)

[Figure 2. 2 applicants registering form 13](#_Toc128150877)

[Figure 2. 3 biography of the applicant 14](#_Toc128150878)

[Figure 2. 4 family condition of the applicant. 15](#_Toc128150879)

[Figure 2. 5 contract agreement b/n applicant and recruiter 16](#_Toc128150880)

[Figure 4. 1 use case diagram 26](#_Toc128149516)

[Figure 4. 2 Class diagram 38](#_Toc128149517)

[Figure 4. 3 sequence diagram for user login 42](#_Toc128149518)

[Figure 4. 4 sequence diagram for user registration 43](#_Toc128149519)

[Figure 4. 5 sequence diagram for recruiter login 44](#_Toc128149520)

[Figure 4. 6 sequence diagram for recruiter registration 45](#_Toc128149521)

[Figure 4. 7 sequence diagram for admin login 46](#_Toc128149522)

[Figure 4. 8 sequence diagram for search job by category 47](#_Toc128149523)

[Figure 4. 9 sequence diagram for apply for job 48](#_Toc128149524)

[Figure 4. 10 sequence diagram for post job 49](#_Toc128149525)

[Figure 4. 11 sequence diagram for update job detail 50](#_Toc128149526)

[Figure 4. 12 activity diagram for login and registration 51](#_Toc128149527)

[Figure 4. 13 activity diagram for user 52](#_Toc128149528)

[Figure 4. 14 activity diagram for recruiter 53](#_Toc128149529)

[Figure 4. 15 activity diagram for admin 54](#_Toc128149530)

[Figure 4. 16 state chart diagram for view job 55](#_Toc128149531)

[Figure 4. 17 state chart diagram for search by category 55](#_Toc128149532)

[Figure 4. 18 state chart diagram for apply for job 56](#_Toc128149533)

[Figure 4. 19 state chart diagram for post job 57](#_Toc128149534)

[Figure 4. 20 state chart diagram for remove job 57](#_Toc128149535)

[Figure 4. 21 state chart diagram for update job detail 58](#_Toc128149536)

[Figure 4. 22 state chart diagram for view candidates applied 59](#_Toc128149537)

[Figure 4. 23 state chart diagram for view users 59](#_Toc128149538)

[Figure 4. 24 state chart diagram for view recruiters 60](#_Toc128149539)

[Figure 4. 25 state chart diagram for accept or reject recruiters 60](#_Toc128149540)

[Figure 4. 26 state chart diagram for remove users 61](#_Toc128149541)

[Figure 4. 27 state chart diagram for remove recruiters 61](#_Toc128149542)

[Figure 5. 1 Architecture of proposed system 65](#_Toc128149981)

[Figure 5. 2 Subsystem Decomposition 66](#_Toc128149982)

[Figure 5. 3 Hardware/Software Mapping 67](#_Toc128149983)

[Figure 5. 4 Detailed Class Diagram 69](#_Toc128149984)

[Figure 5. 5 persistent data management 70](#_Toc128149985)

[Figure 5. 6 user interface 73](#_Toc128149986)

# LIST OF ABBREVATIONS

|  |  |  |
| --- | --- | --- |
| No | Abbreviation | Description |
| 1 | CD-ROM | Compact Disk Read-Only Memory |
| 2 | CPU | Central Processing Unit |
| 3 | CSS | Cascading Style Sheet |
| 4 | CV I | Curriculum Vitae |
| 5 | MD5 | Message-Digest Algorithm 5 |
| 6 | RAM | Random Access Memory |
| 7 | SQLite3 | Standard Query Language |
| 8 | UI | User Interface |
| 9 | UML | Unified Modelling Language |

# 

# ABSTRACTS

The use of portal in information seeking is a large growing activity. Hence Job portal system  
deals with giving a common interface for the jobseeker and a company to connect to each other. Consequently most companies post jobs on their website and on papers. To oversee this problem we come up with a new system which is job portal system. The purpose of this project is, developing a web based Job Portal system for Gurage zone using django framework and to let the jobseeker create account and search for available jobs; after a jobseeker application the details including jobseeker’s degree paper are sent to the company then companies can send notification to the jobseeker. Different Methodology is used to solve the problem throughout the development of the project. Interview, direct observation for requirement gathering, Object-Oriented system analysis and design and agile model for development approach and Unified Modeling Language for designing the job portal system. The prototype is evaluated on it usability to ensure the successful implementation of the job portal system.

# Chapter One

# 1. Introduction

In the scenario of the assignment, we are required to develop a web-based application on Job Portal Management System. In this time of recession where everyone, is either experienced or fresher, is in search for a job. This job portal can prove to be very helpful since it allows users of different profile to upload their CVs, search job on the basis of their qualification. Every user can access through user id and apply for multiple jobs at a time. Currently, we are working on a manual system where data is stored in the form of registers. 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. Job seekers check the list of jobs available and apply the job. Then the agency will show available jobs for the job seeker for his qualifications and then updates the jobs database. The developed Job Portal management system is web-based which Requires Employee Registration & Profiles, Job Search, Employer Registration & Profiles, and Subscription Option for both Employee and Employer etc. Employer can add Own Profile and post jobs and Job Seeker can Search Jobs based on Geographical Area (Country, State, and City), Qualification, and Company wise or on the Basis of Experience and Expertise wise.

## 1.1. Background of organization

The Gurage Zone is one of the administrative zones in the southern Nations, Nationalities, and Peoples’ Region (SNNPR) of Ethiopia. The Gurage Zone Administrative Office is the administrative body responsible for overseeing the affair of the zone.

The Gurage Zone is located in the central part of the SNNPR and cover an area of approximately 6,500 square kilometer. The Zone is home to a diverse population of over one million people, with the majority of residents belonging to the Gurage ethnic group.

The Gurage Zone Administrative Office is responsible for managing various aspect of the zone’s governance, including the delivery of public services, infrastructure development, and local economic growth. The Office is headed by a Zonal Administrator, who is appointed by the regional government.

In recent years, the Gurage Zone Administrative Office has focused on promoting local economic development, with initiative aimed at boosting trade and investment, improving agricultural production and productivity, and developing the tourism sector. The Office has also been working to improve access to basic services, such as education and healthcare, for residents of the zone.

Overall, The Gurage Zone Administrative Office plays an important role in the governance and development of the Gurage Zone, working to improve the lives and livelihood of its residents, and driving local economic growth and development.

## 1.2. Statement of problem

Currently, in Gurage zone there is lack of communication between Companies or organizations and different employers who needs jobs, due to the lack of communication flow of information exchange. Let’s list some of the problems:-

* Big information gap between the communities of this zone of employers and Organizations.
* Problem with file sharing between employers who needs jobs and organizations.
* Problem in companies in hiring competent professionals, because lack of communication in applying Employers and companies.
* The manual system have many impact in time management.
* Problem for employers cost of transportation.
* Wastage of energy during applying jobs.

## 1.3. Objective of the Project

### 1.3.1. General Objective

The general objective of this project is to design and implement online job portal application for

Gurage zone.

### 1.3.2. Specific Objective

* To identify the clients need and collect relevant data
* To study the existing manual system
* To provide the information about jobs
* To reduce the time of searching by using the web page that is developed to reduce the problem of job searching
* To post new job which is store in the job database
* To reduce the loss of applicant’s info, CV, tempo etc…by storing the jobseekers information on our database and retrieving as needed
* To minimize the cost of newspapers by announcing vacancy announcement using our web pages and attaching the announcement as the organization requirements
* To design an on-line recruitment system for that allows employers to post their job advertisements, which job seeker can refer to, when looking for jobs.

## 1.4. Scope and Limitations

### 1.4.1. Scope

The scope of the project that we are going to develop focused on Gurage zone especially for employers, recruiters who are worked on Gurage zone.

So the project focus on the following activities.

* users can search jobs based on
  + Location
  + Organization name
  + Job title
  + Start date and end date
* users can apply their cv
* The system suggest the recent search and delete old posts.
* maintain job seeker and employer records
* Maintain job posting details

### 1.4.2. Limitations

The Interface is provided only in English due to several constraints like shortage of time, linguists and budgets we are not going to implement the system other than English. So, the user should know English.

## 1.5. Feasibility Study

To bring the successful completion of this project goals and objectives the feasibility issues listed below will determine the project viability or the discipline of planning, organizing, and managing resources.

**1.5.1.** Economic Feasibility

It is the measure to determine the cost and benefit of the proposed system. A project is economical feasible which is under the estimated cost for its development. These benefits and costs may be tangible or intangible. Job Portal is the cost-effective project in which there is less possibility of intangible cost so there is no difficulty to determine the cost of the project.

**1.5.2. Technical Feasibility**

The system to be developed by using technologically system development techniques such as python, and SQLite3 database without any problems and the group members have enough capability to develop the project. So the system will be technically feasible.

**1.5.3. Operational Feasibility**

The system to be developed will provide accurate, active, secured service and decreases labor of workers and also it is not limited to particular groups or body. And also it is platform independent i.e. it run’s in all operating system.

## 1.6. Significance of the Project

Generally, the system will facilitate the existing system by providing the following:

* To get well organized information in short period of time
* Increase efficient use of properties, money and time.
* Minimize wastage of time, cost, and energy while hiring employers and properties and schedule of the organization.
* Minimize employees work load.
* **It promotes a neat image of the employer's company.**
* Reduce the organization cost that spent during the hiring process.
* **Reduces the time and energy required for the hiring process.**
* **Consolidation of candidate profile becomes easy.**

## 1.7. Beneficiary of the Project

**1.7.1. Wolkite zone and community:**

* Employers are benefit since it can save their time, energy and can simply search and apply to the companies by just visiting the site and creating an account.
* Save time and money.
* Reduce over load.
* To solve problems those are associated with the manual system.
* To use the document as a reference material for related projects.

**1.7.2. Companies**

* Companies can easily find competent employer.

**1.7.3. Team members**

* The project has been initiating our team to get knowledge of how to develop the required system.
* The team members are getting a lot of experience of solving problem while they are facing with some difficulties.
* It helps to create team Spirit among the development team.

## 1.8. Methodology

**1.8.1. Requirement Gathering**

The data collection instruments will use to gather accurate information about the existing system and the requirements for the new system. Interviews and questionnaires will administer to stakeholders like seller, cashier and business owner to collect user requirements. To get a precise data, the team member will use the following data collection techniques:

* **Interview:** To get the basic information and background information about the existing system, the team members will interview the marketer and some customer about the services that have given to them, and the problems associated with that environment.
* **Direct observation**:Even though interview is very important to gather information, direct observation is simple and we will physically observe information that cannot maintain from the interview or others and also it is important if they are unable to communicate with others because of the difficulties they have to the professional terms (language).
* **Questionnaire:** Since employee as well as manager of employee have work load they cannot able to answer or give information what we ask. So we will prepare some sample questions to get precise information.
* **Existing document**: To get more information about the project we will use earlier documents that help us to develop the project. During the analysis of documents, we will give a special consideration to those documents which can bring more features to the project.

**1.8.2. System Analysis and Design**

In the system analysis and design phase of a project we will use the object oriented approach that examines requirements from the perspective of the class and objects found in the problem domain. The reasons that we will use the object oriented approaches are:

* To simplify the design and implementation of complex program
* We can inherit properties of the class that are defined in the super class.
* We can reuse methods for avoiding redundancy.
* To make it easier for teams of designers and programmers to work in a single software project
* The data and functions are encapsulated in the objects that help us for easily debugging purpose.
* It will increase consistency among analysis, design and programming activities.
* It will improve communication among users, analysis, design and programming
* It enables us to comprehensively model a system before we develop it.
* Modification of the object implementation is easy because objects are loosely coupled.
* Understanding of the structure is easy because object oriented modelling represents real world entities.
* Direct manipulation of architectural components is possible because several object oriented programming languages exist.

**1.8.3. System Development Model**

In the system development model to develop good software we will use agile model because we have different reason such:

* We will iterate and increment with in each phase.
* Direct collaboration with the customers
* The benefit of the Agile Development Model can be conserving of our time as well as money.
* We can easily control it, and it is flexible for developers/us.
* Working software will be delivered constantly, i.e. in Weeks or Months.
* Regularly or weekly interaction among entrepreneurs and developers promotes software development speed.
* It primarily concentrates on the deliverable and fewer about paperwork.
* Customer, developers, and tester continuously interact with each other.

**1.8.4. Development Tools**

* Modelling
* Enterprise Architect
* Visual paradigm
* UML modelling tools
* Programming language
* Python
* Database design
* SQLite3
* Framework
* Django(used to develop web side) admin side
* Bootstrap(for front-end CSS decorator)

**1.8.5. Testing Procedure**

Developing software is a complex process. No matter how hard we try to eliminate all faults simply by going through the phases of requirements, analysis, design, specification, and implementation, however through good practice we can make sure that the most series fault does not occur in the first place. In addition we need a separate testing phase, with the goal of elimination all remaining faults before release. To simplify the testing process the project team will followed different types of test mechanism that break the testing process up into the distinct levels. These types testing are unit testing, integration testing and system testing.

**Unit Testing**

In this level of testing process, we will developers will test different sub procedures, functions and will test by applying the black and white box testing.

Sample Tests will be:

* Check whether the return type of the functions is correct.
* Check how the sub procedures or functions are called correctly.
* Check if the correct output is produced for different inputs.
* Check the efficiency of the code with respect to the memory and CPU time.

**Integration Testing**

In this level of testing we will examine how the different procedures work together to achieve the goal of the sub system.

**System Testing**

In this level of testing process we will examine how nicely the subsystems of the whole online job portal work together to achieve the desired goal.

**1.8.6.** **Operating Environment**

* Hardware environment
* Processing power
* 64-bit operating system
* Intel(R) core™i3 CPU 550@3.20GHz
* Memory & Secondary storage
* RAM: 4GB and above
* 500GB Hard-disk: and swap space (if RAM is insufficient).
* Peripherals
* CD-ROM drives,
* Network devices, etc.

## 1.9. Project Plan

Table 1. 1 Project Plan

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activities** | Duration | Time | | | | | | |
| 23/03/15  -  09/04/15 | | 10/04/15  -  27/04/15 | 28/04/15  -  23/05/15 | 24/05/15  -  20/09/15 | 17/09/15  -  22/09/15 | 23/09/15 |
| **Requirement Elicitation** | 14 |  | |  |  |  |  |  |
| Test | 2 |  | |  |  |  |  |  |
| **Requirement Analysis** | 15 |  | |  |  |  |  |  |
| test | 2 |  | |  |  |  |  |  |
| **Proposed system design** | 25 |  | |  |  |  |  |  |
| test | 2 |  | |  |  |  |  |  |
| **Implementation & Coding** | 80 | |  |  |  |  |  |  |
| **test** | 2 | |  |  |  |  |  |  |
| Testing | 5 | |  |  |  |  |  |  |
| **project Defence** | 1 | |  |  |  |  |  |  |

# Chapter Two

# 2. DESCRIPTION OF THE EXISTING SYSTEM

## Introduction of Existing System

When the project team was analysing the existing system of the job searching and posting process, the team has tried to study the detailed nature and procedure of the tasks and operations performed by the employers and the job seekers in the job requirement process. Most people either recruit or recruiter use the traditional way of finding job because they are more familiar with the old process or because of the gaps with the existing job portals.

We are looking at the current system in two ways. This is because of some agencies announce job vacancies through their web sites and also some websites are available to meet recruiters and job seekers, and almost all job seekers, recruiters and agencies are accomplishing job portal manually or through new peppers. For the reason, no popular websites are available to meet their requirements.

Almost all agencies hire employees and applicants or job seekers seek for job manually. This system requires applicants to search through print and visual media for job opportunities. Applicants need to apply for jobs using conventional methods and appear for interview on a specified date at a specified location. In this manual system jobseekers send their resumes or information through postal mails or they use person to person contacts with each other. It will take a long time to send their requirement through this type of communication, and then wait for an interview request.

Employers need to advertise the vacancies advertising through newspapers, televisions, and radios, and sort all applicant details, conduct selection procedures and complete the formalities. The job application process took quite some time. This approach is tedious and requires much effort and resources. This process was equally frustrating for recruiters. As it often took some time to fill positions with qualified candidates, online recruitments speed up and make the process much more efficient.

## Users of Existing System

The major players of this existing system are the Applicants and the recruiters. Job seeker is a person that wants to find job at Gurage zone. if the job seeker wants to work at the Gurage zone community, he has to attend the vacancies of the companies or governmental office from newspaper if they are popular and by searching the agencies clipboard. After the registration of every applicant the applicants wait for a week to know the final result of the computation.

## 2.3 Major functions of the Existing System

* Posting new vacancy announcements to the applicants as needed by the company.
* Evaluating the applicants by giving oral and paper exams and evaluating their technical skills.
* Screening the applicants by their GPA, experience etc.
* Giving clearance for employer that leave the agency.
* Working with the annual plan of the employers and technical workers (technical assistance) of the agency reporting the budget needed for the monthly salary of the

employees on the coming year.

* Reporting the number and quality of professionals to the agency at the end of the year.

## 2.4 forms and other documentations of existing system

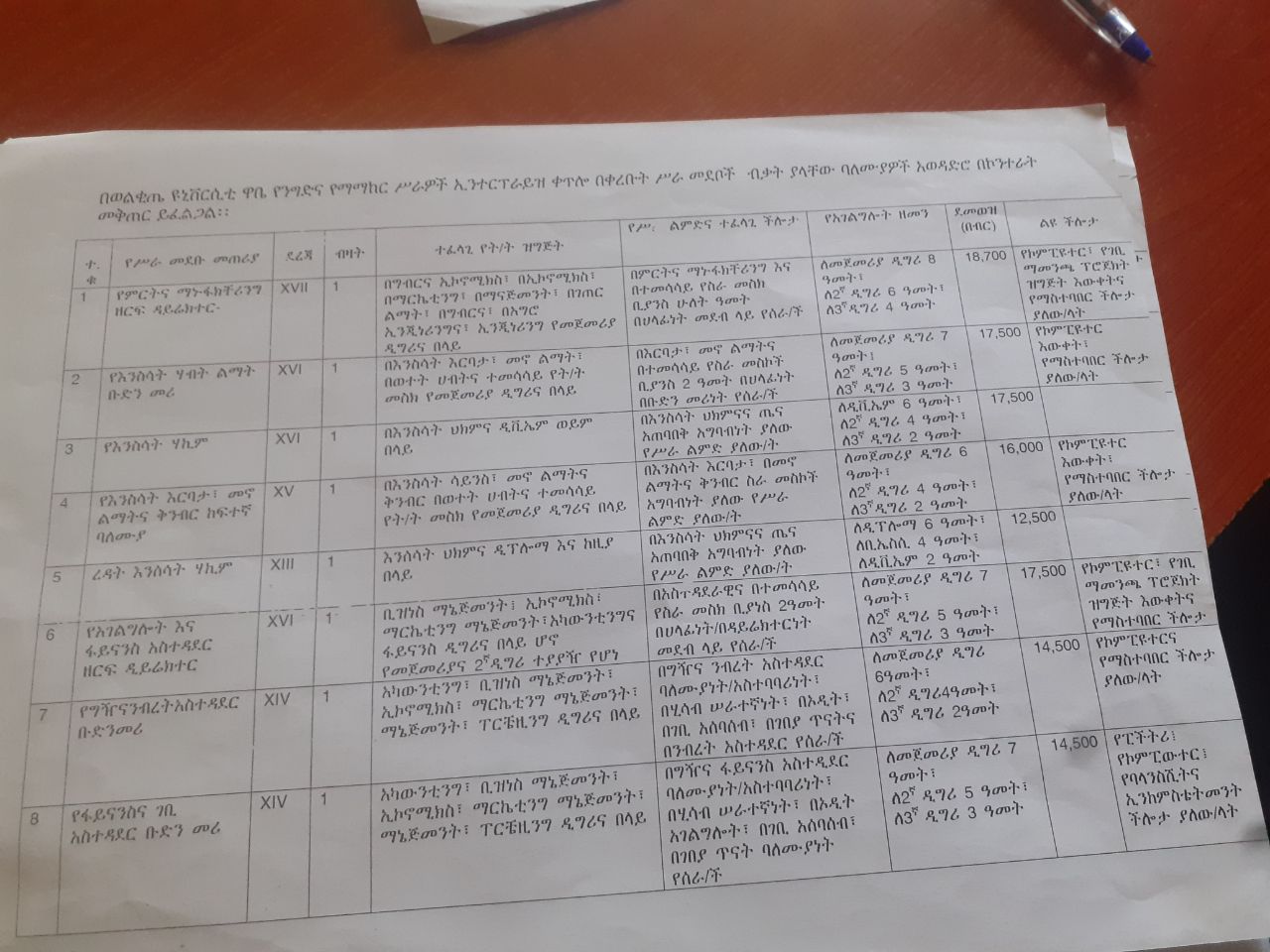


Figure 2. 1 posting vacancy

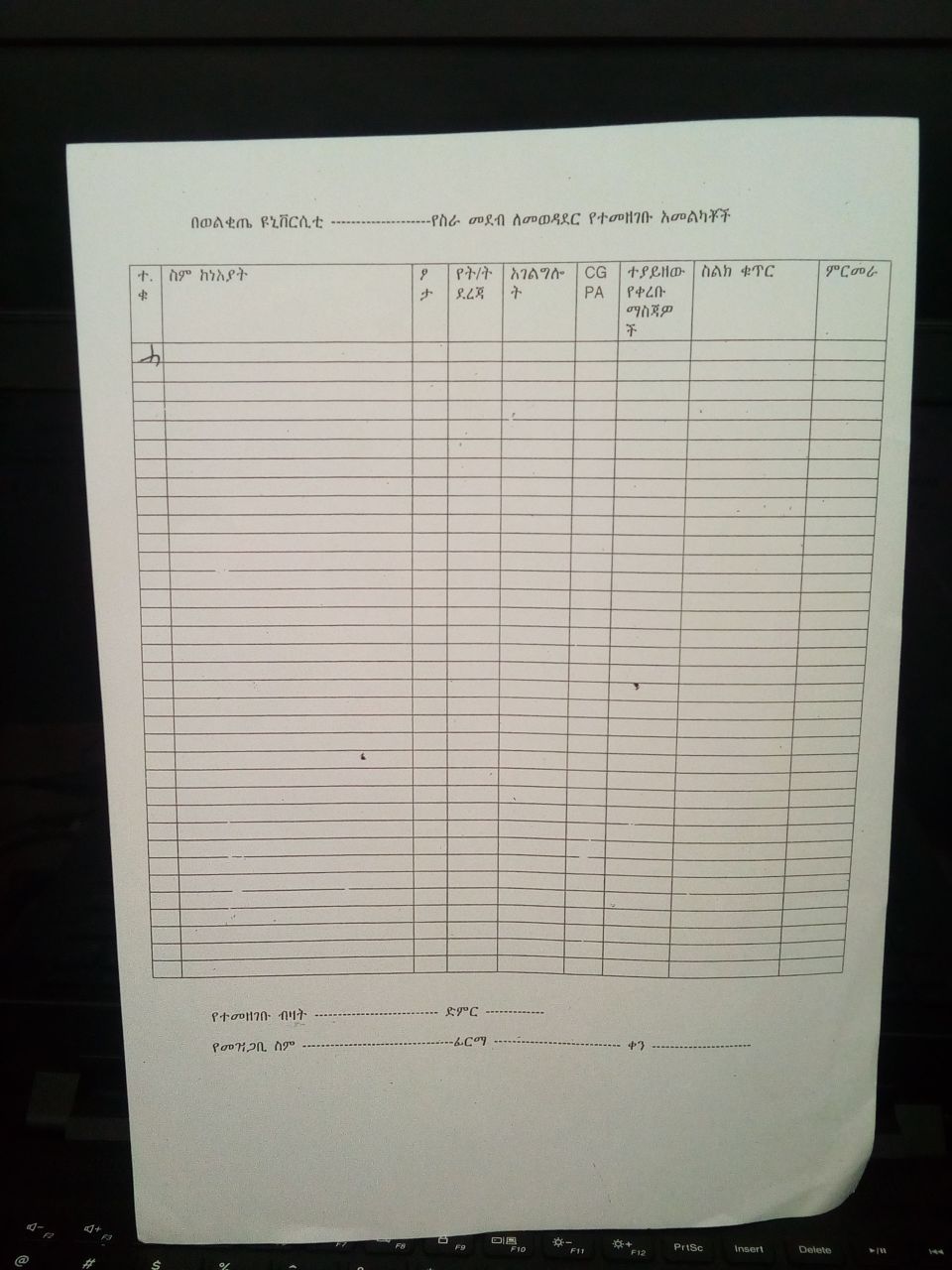


Figure 2. 2 applicants registering form

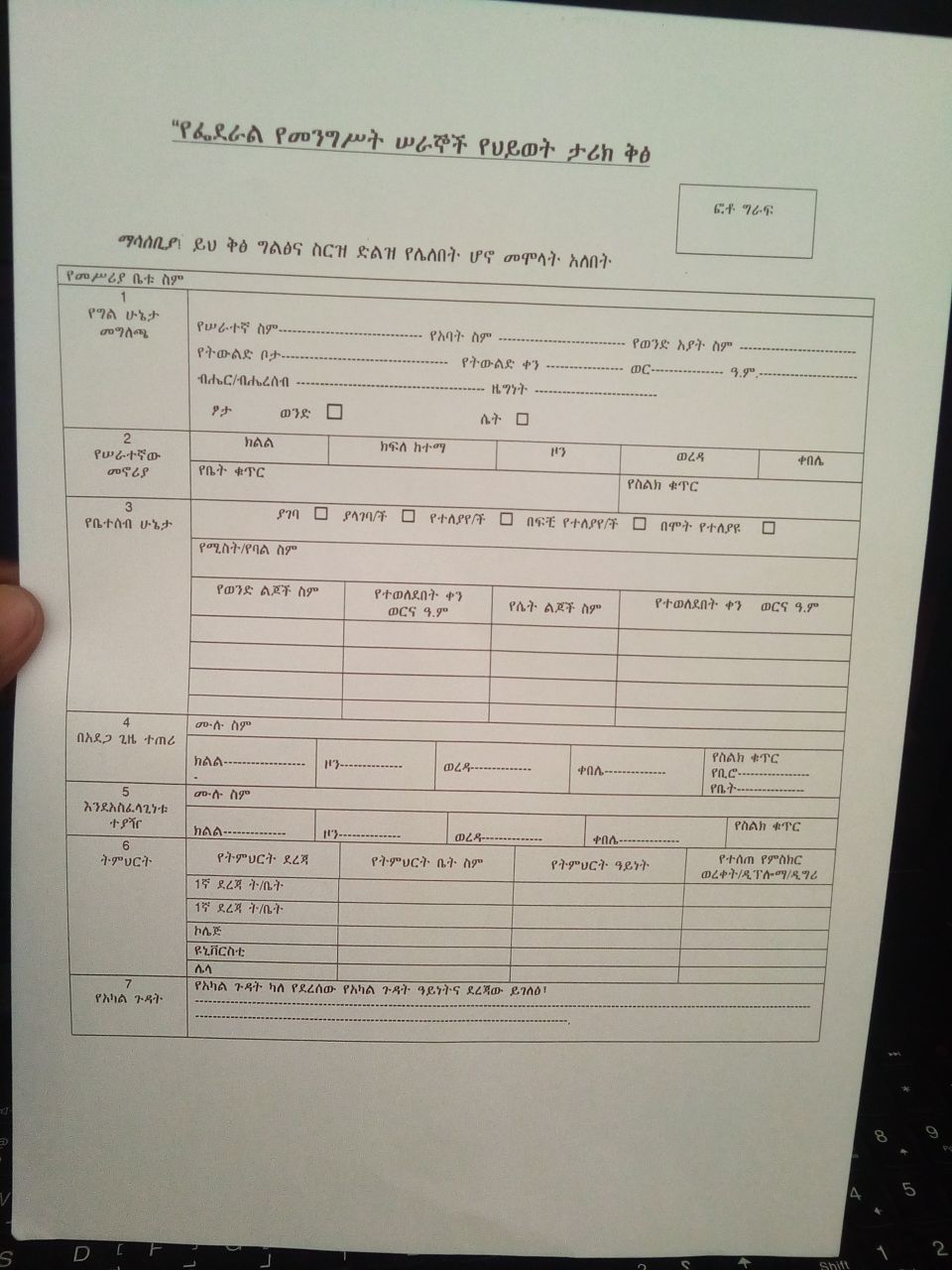


Figure 2. 3 biography of the applicant

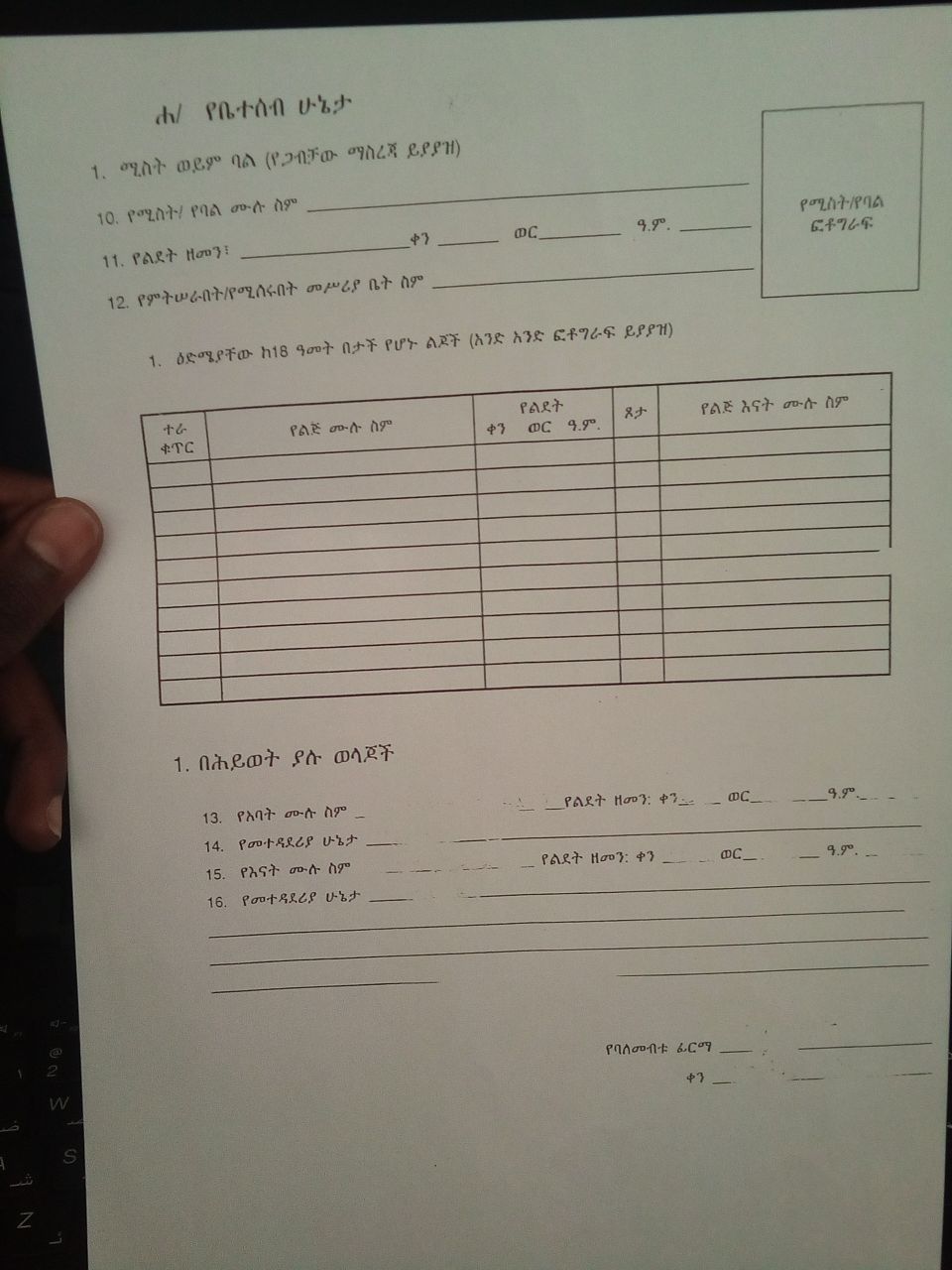


Figure 2. 4 family condition of the applicant.

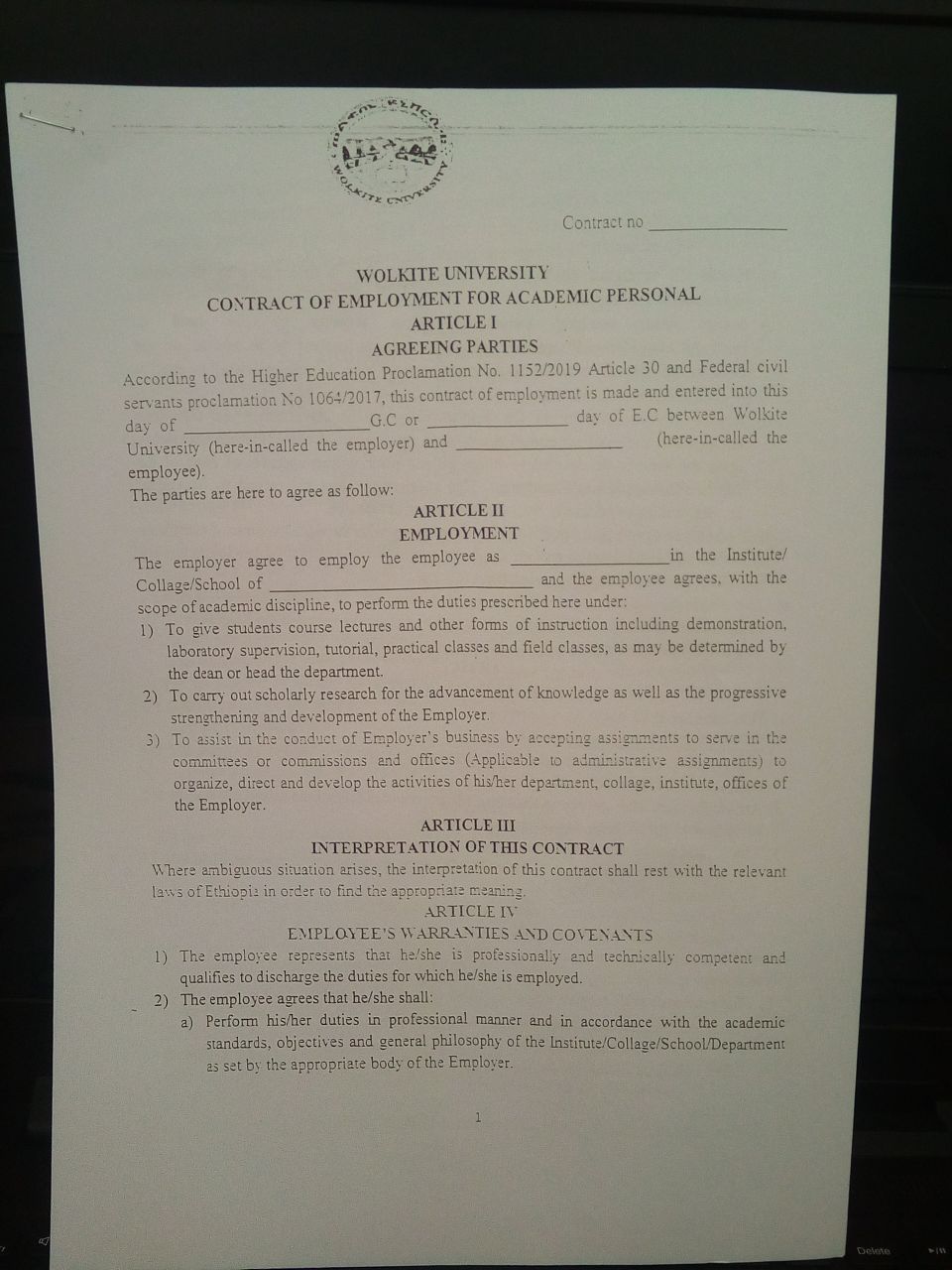


Figure 2. 5 contract agreement b/n applicant and recruiter

2.5 Drawbacks of the Existing System

Currently there are main problems between the job seeker and the employers of any agency to find right employee. The existing problems includes: -

* A job seeker cannot submit his/her CV because of transport, distance to the agency

bureau, family cases etc...

* Existing job portals are not reaching woreda and kebele level of Gurage zone.
* The Job Seekers are unable to get the opportunity for job according to qualification.
* Employer of the company limits the job description when he/she spreads the paper in

order to minimize the number of papers sticks at the board.

* Companies do not get employees which are qualified and experienced because only

few individuals who are close to them can know the vacancy and those are not as the

company need in their qualifications.

* Companies need more time starting from job vacancy is announced until they hire the

right employee.

## 2.6 Business rules

In every organization or institution there are rules and policies, which is used to govern all activities it also functions for our project as well it’s identified below.

1. The employee represents that she/he professionally and technically competent and qualified to discharge the duties for which he/she is employed.

2. The employee agrees that she/he shall.

A) Performs her/his duties in professional manner and in accordance with the academic standards, objective and general philosophy of the institution as set forth by the appropriate body.

B). The employ agrees to submit to the institute along with the signed contract a medical certificate of her/his physical fitness to perform the duties.

C). the employee agrees to refrain from using her/his official position of institute facilities for the dissemination of her/his official position of institute facilities for the dissemination of her/his personal actions which incite or cause disruption of the institute.

D). the employee agrees to abide by the regulations of the company and laws of Ethiopia.

# Chapter three

# 3. Proposed system

The web-based job portal system allows applicants and employers to register their details. Applicants can browse through the vacancy posted, can apply for jobs online. Employers can view the applicant detail. The system provides an effective means for the employers to post job openings with required qualification to have a better penetration into the job market and job seekers to find out the information regarding the current openings in the organization. Job seekers can view all the vacancies posted in a particular city and a particular job title. Most of the requirement process is covered in this system including the job searching and finding process up to assigning interview date. Data storage and retrieval is faster and easier to maintain because data is stored in a systematic manner and in a single database. Easier and faster data transfer through the latest technology associated with computer and communication. These features increase the efficiency, accuracy, and transparency of the job portal system. The project is identified by the merits of the system offered to the user. The merits of this project are as follows:

* Free service, no cost
* 24 X 7 Online support system for an employee as well as an employer.
* No duplication of Jobs.
* Filter, search facilities for job seekers according to their required vacancy.
* Sending resume saves effort, time and cost of job seeker.
* All vacancies are available on a single interface.

## 3.1. Functional requirement

**Registration**

The user and recruiter have to fill the fields like first name, last name, username, password and so on to create an account and can’t login without creating account.

**Login**

The admin, user and recruiter have their own login page and fill username and password field and login to the system.

**Manage job**

The recruiter manages job,

* **Post new job:** this can help the applicants to search jobs they want by their educational qualifications. The recruiter (upload) vacancy Announcement. The posted new job is displayed in the jobseekers (users) page this can help the system to make reviews every time when a new announcement is posted.
* **Remove job:** the recruiter can remove unwanted job from his/her posted joblist.
* **Update job detail**

The recruiter can edit the description of the job from his/her post.

**Manage User**

The administrator can create admin users with giving them the privilege to manage jobseeker, recruiter. The admin can change the status of recruiter from pending to accepted or rejected.

**Search and Apply for Job**

Job seekers can search job as their requirement and after looking the vacancy detail they can apply for job.

**View candidates applied:** The recruiter can view jobseekers applied to his/her post.

# 3.2. Non-functional Requirements

Non-functional requirements describe aspects of the system that are not directly related to the functional behavior of the system, only related to how the system will do the functional requirement. Non-functional requirements include a broad variety of requirements that apply to many different aspects of the system, from usability to performance.

**User Interface and Human Factors**

The system we are going to develop follows a good interface principle and it will provide easy, attractive, simple, and interactive interfaces for the user of our system. We will design a beautiful user interface using the Django framework which is a high-level Python Web framework that encourages rapid development and clean, pragmatic design and this framework enables us to design a responsive user interface. The user of the system should have a mobile phone or computer to connect to the internet and to access the system. The proposed system will be intuitive enough that only needs a few hours of training the user. No need for computer professions. Only the person who knows basic computer knowledge and knows how to browse the internet can use our system. Administrators should be skillful and professional to use and manage the proposed system.

**Hardware Consideration**

The system will be developed by considering hardware requirements. The system will support computers, smartphones and digital cameras positioned in the classroom to obtain snapshots and secondary memory to store all the images and database to provide the service for the user. Basic hardware device which our proposed system want is:

⮚ Computers

⮚ Deployment server

**Security Issues**

Protecting user data is an essential part of any system, the proposed will allow only authorized users to enter using their username and password that will be encrypted using MD5 encryption algorithm. Session and cookie will be used in our proposed system. Users will access only their privilege concerning areas of the system and data modification should be done only by privileged users. Message Digest Algorithm 5(MD5) is a one-way cryptographic function that accepts a message of any length as input and returns as output a fixed-length digest value to be used for authenticating the original message.MD5 maps a set of data to a bit string of a fixed size called the hash value. Hash functions are used for crypto currency, password security, and message security. MD5 algorithms are widely used to check the integrity of the files. The big use of MD5 is the storage of 30 passwords. Due to the insecurity of databases and the need to store the passwords somewhere, many passwords were stored in this manner. It is very easy to generate a message digest of the original message using this algorithm. Generally, MD5 processes messages quickly. Because it’s fast.

**Performance Consideration**

The proposed system will have easy and efficient code manipulation and a clear database.

**Response Time:** Upon the request for user query the system under normal conditions will display results as quickly as possible.

**Processing Time:** Since the system will be developed with efficient programming language and database upon request for users’ activity. The system under normal conditions will process the request as quickly as possible.

**Concurrent - Processing:** since the proposed system is web-based it concurrently deals with the requests rather than waiting for a previous one to be completed, the system will support multiple users at a time.

**Error Handling and Validation**

Error handling is an important skill. The system will check user inputs to the system to handle errors. Django provides full support for cookie- and session-based messaging, for both anonymous and authenticated users. The messages framework allows you to temporarily store messages in one request and retrieve them for display in a subsequent request (usually the next one). Every message is tagged with a specific level that determines its priority (e.g., info, warning, or error). It handles and shows errors in a user-friendly manner, without exaggerating the user. For server-side error, it is only visible on the server log for the developer. All users will be notified that there was a server error.

**Documentation**

Our system will have well-defined documents which help to easily maintain the system. We will also prepare a short and understandable file for users on how to use the system. And the development process will be provided for the user to read to know about the process and what type of model used to develop the system.

# 

# CHAPTER FOUR

## SYSTEM ANALYSIS

## 4.1 System Model

System modeling is the process of developing abstract models of a system, so with each model we will present (show) a different view or perspective of our system using some kind of graphical notation, which is now almost always based on notations in the Unified Modeling Language (UML). Models help to understand the functionality of a system.

### 4.1.1 Use Case Model

Use case model is a model of how different types of users interact with the system to solve a problem. And it describes the goals of the users, the interactions between the users and the system, and the required behavior of the system in satisfying these goals. And it consists of a number of model elements such as use cases, actors and the relationship between them.

Table 4. 1 Use Case Identification

|  |  |  |
| --- | --- | --- |
| Actors | Use case Under Identified Actor | Use Case |
| User | Login   * Logout | 01 |
| Register | 02 |
| Update profile | 03 |
| Change password | 04 |
| View jobs | 05 |
| Search job by category | 06 |
| Apply for job   * Upload resume | 07 |
| Recruiter | Login   * logout | 01 |
| Register | 02 |
| Update profile | 03 |
| Change password | 04 |
| Post job | 08 |
| Remove job | 09 |
| Update job detail | 10 |
| View candidates applied   * View resume | 11 |
| Login   * Logout | 01 |
| Administrator | Update profile | 03 |
| Change password | 04 |
| View users | 12 |
| Remove users | 13 |
| View recruiters | 14 |
| Accept or reject recruiters | 15 |
| Remove recruiters | 16 |

### 4.1.2 Use Case Diagram

A use case diagram below created to visualize interactions between an actor and use cases in a proposed system. Users are specified by the actor and their functionalities are specified by the use case.

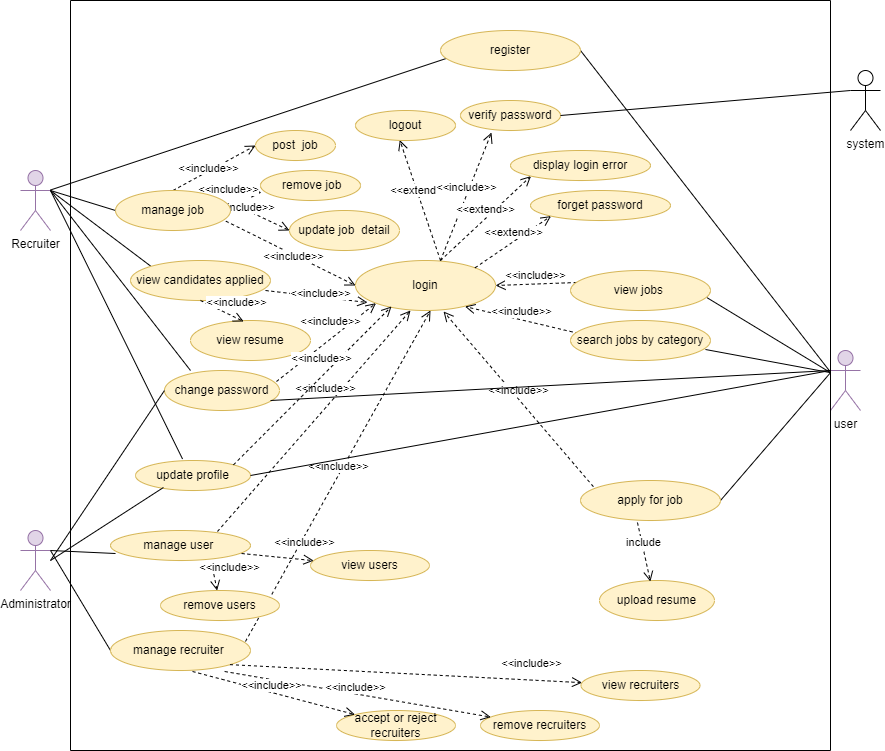


Figure 4. 1 use case diagram

## 

### 4.1.3 Use Case description

The following tables show Use case description based on the system business or context of our system.

Table 4. 2 usecase description of login

|  |  |  |
| --- | --- | --- |
| Usecase Name | Login | |
| Usecase Identifier | UC-01 | |
| Actor | User, Recruiter, Administrator | |
| Description | This usecase describes how users, recruiters and administrator login to the system | |
| Precondition | The Actor must be registered before | |
| Basic Course of Action | Actor action | System response |
| **Step 1**. open the system webpage  **Step 3**. click the actor name login link  **Step 5.**enter username and password and then click login  link | **Step 2**. the system displays homepage  **Step 4.** the system displays login page  **Step 6.**validate username and password  **Step 7.** The displays the homepage with privilege of actor |
| Alternative Course of action | **Alt. Course A:** if the input data(username and password) not valid  **Step A.1:** the system display error message and returns to step 4 | |
| Post Condition | If The user logged into system wants to terminate the session can logout | |

Table 4. 3 usecase description of registration

|  |  |  |
| --- | --- | --- |
| Usecase Name | registration | |
| Usecase Identifier | UC-02 | |
| Actor | User, Recruiter | |
| Description | Allows users and recruiters to create account | |
| Precondition | Not registered before | |
| Basic Course of Action | Actor action | System response |
| **Step 1**.open the system webpage  **Step 3**. click the actor name login link  **Step 5.** click create account  **Step 7.**fill the form and click submit | **Step 2**. the system displays homepage  **Step 4.**the system displays the login page  **Step 6.** The system displays signup page  **Step 8.** validate the required fields  **Step 9.** save to database and display login page |
| Alternative Course of action | **Alt.1:** if the input data are not valid  the system displays error message and returns to step 6 | |

Table 4. 4 usecase description of update profile

|  |  |  |
| --- | --- | --- |
| Usecase Name | Update profile | |
| Usecase Identifier | UC-03 | |
| Actor | User, Recruiter and Administrator | |
| Description | Here the actors can update their existing profile | |
| Precondition | User must login and must go to it’s intended page | |
| Basic Course of Action | Actor action | System response |
| **Step 1**. open the system webpage  **Step 3**.login to the system  **Step 5.** Change the permitted fields and click submit | **Step 2**. the system displays homepage  **Step 4.** The system displays actor home page  **Step 6.** validate the required fields  **Step 7.**save the changes to database |
| Alternative Course of action | **Alt.1:** if the input data are not valid  the system display error message and returns to step 4 | |

Table 4. 5 usecase description of change password

|  |  |  |
| --- | --- | --- |
| Usecase Name | Change password | |
| Usecase Identifier | UC-04 | |
| Actor | User, Recruiter and Administrator | |
| Description | Here the actors can change their password | |
| Precondition | User must login and must go to it’s intended page | |
| Basic Course of Action | Actor action | System response |
| **Step 1**. open the system webpage  **Step 3**.login to the system  **Step 5.** Click change password on the navigation bar  **Step 7.**fill the current, new and confirmation password field and click submit | **Step 2**. the system displays homepage  **Step 4.** The system displays actor home page  **Step 6.**the system displays change password page  **Step 8.**validate the required fields  **Step 9.**save changes to database and display homepage |
| Alternative Course of action | **Alt.1:** if the input data are not valid  the system display error message and returns to step 6 | |

Table 4. 6 usecase description of view jobs

|  |  |  |
| --- | --- | --- |
| Usecase Name | View jobs | |
| Usecase Identifier | UC-05 | |
| Actor | Users, recruiters, administrator | |
| Description | Here the actors can view the job which is posted from recruiter side | |
| Precondition | Here the user must the system home page and click latest jobs link | |
| Basic Course of Action | Actor action | System response |
| **Step 1**.open the system webpage  **Step 3.** Click latest jobs link on navigation bar | **Step 2**. the system displays homepage  **Step 4.**the system displays latest jobs page |

Table 4. 7 usecase description of search jobs by category

|  |  |  |
| --- | --- | --- |
| Usecase Name | Search jobs by category | |
| Usecase Identifier | UC-06 | |
| Actor | Users, recruiters, administrator | |
| Description | Here the actors can search the job by location, company name, end date etc. which is posted from recruiter side | |
| Precondition | Here the user must the system home page and click latest jobs link | |
| Basic Course of Action | Actor action | System response |
| **Step 1**. open the system webpage  **Step 3.** Click latest jobs link on navigation bar  **Step 5.** Enter the search text on search field | **Step 2**. the system displays homepage  **Step 4.**the system displays latest jobs page  **Step 6.** The system filters the jobs based on the text and displays to the users |

Table 4. 8 usecase description of apply for job

|  |  |  |
| --- | --- | --- |
| Usecase Name | Apply for job | |
| Usecase Identifier | UC-07 | |
| Actor | User | |
| Description | Allows users to apply for job | |
| Pre-condition | The user must login to the system and go to intended page | |
| Post condition | If the use case is successful user apply for a job. | |
| Basic Course of Action | Actor action | System response |
| **Step 1**. open the system webpage  **Step 3**.login to the system  **Step 5.** click joblist link  **Step 7.** click apply  **Step 9.** click apply for job  **Step 11.** Select the file and click upload resume | **Step 2**. the system displays homepage  **Step 4.** the user displays home page  **Step 6.** The system displays joblist page and apply button.  **Step 8.** The system displays detail of the job vacancy and apply for job button.  **Step 10.** check for date and displays upload resume page  **Step 12.** The system displays successfully applied message and goes to joblist page |
| Alternative Course of action | **Alt.1:** if the date for application is not started or ended the system display the application date is not valid and return to step 8  **Alt.2:** if the resume file is not selected the system displays error message | |

Table 4. 9 usecase description of post job

|  |  |  |
| --- | --- | --- |
| Usecase Name | Post job | |
| Usecase Identifier | UC-08 | |
| Actor | Recruiter | |
| Description | Allows recruiters to post | |
| Pre-condition | The recruiter must login to the system and go to intended page | |
| Post condition | If the use case is successful the job will be posted and jobseekers will view the vacancy. | |
| Basic Course of Action | Actor action | System response |
| **Step 1**. open the system webpage  **Step 3**.login to the system  **Step 5.** Click add job link  **Step 7.** Fill the required job description fields and click submit button | **Step 2**. the system displays homepage  **Step 4.** the system displays the recruiter home page  **Step 6.** The system displays add\_job page  **Step 8.** validate the required fields and save to database  **Step 9.** The system displays joblist page |
| Alternative Course of action | **Alt.1:** if the input data are not valid the system display error message and returns to step 6 | |

Table 4. 10 usecase description of remove job

|  |  |  |
| --- | --- | --- |
| Usecase Name | Remove job | |
| Usecase Identifier | UC-09 | |
| Actor | Recruiter | |
| Description | Allows recruiter to remove job from his/her joblist | |
| Pre-condition | The recruiter must login to the system and go to intended page | |
| Post condition | If the use case is successful the job will be removed from joblist of the user and recruiter page. | |
| Basic Course of Action | Actor action | System response |
| **Step 1**. open the system webpage  **Step 3**.login to the system  **Step 5.** Click joblist link  **Step 7.** click delete button corresponding to job to be removed | **Step 2**. The system displays home page  **Step 4.** The system displays the recruiter home page  **Step 6.** The system displays the recruiters joblist based on the id of recruiter  **Step 8.** The system displays confirmation message and the job is removed. |
| Alternative Course of action | **Alt.1:** if the cancel button is clicked on confirmation message the system returns to step 6 | |

Table 4. 11 usecase description of update job detail

|  |  |  |
| --- | --- | --- |
| Usecase Name | Update job detail | |
| Usecase Identifier | UC-10 | |
| Actor | Recruiter | |
| Description | Allows recruiters to update job detail over which he/she posted | |
| Pre-condition | The recruiter must login to the system and go to intended page | |
| Post condition | If the use case is successful the job detail will be updated and the changes are saved to database. | |
| Basic Course of Action | Actor action | System response |
| **Step 1**. open the system webpage  **Step 3**.login to the system  **Step 5.** Click joblist link  **Step 7.** click edit button corresponding to job to be updated  **Step 9.**change the required fields and click submit button | **Step 2**. The system displays home page  **Step 4.** The system displays the recruiter home page  **Step 6.** The system displays the recruiters joblist based on the id of recruiter  **Step 8.** The system displays update job detail page.  **Step 10.** validate the required fields and save the changes to database  **Step 11.**the system displays joblist page |
| Alternative Course of action | **Alt.1:** if the input data are not valid the system display error message and returns to step 6 | |

Table 4. 12 usecase description of view candidates applied

|  |  |  |
| --- | --- | --- |
| Usecase Name | View candidates applied | |
| Usecase Identifier | UC-11 | |
| Actor | Recruiter | |
| Description | Allows recruiters to see users applied to job. | |
| Pre-condition | The recruiter must login to the system and go to intended page | |
| Post condition | If the use case is successful the applicant detail is displayed | |
| Basic Course of Action | Actor action | System response |
| **Step 1**. open the system webpage  **Step 3**.login to the system  **Step 5.** Click view candidates applied link | **Step 2**. The system displays home page  **Step 4.** The system displays the recruiter home page  **Step 6.** The system displays candidates applied based on the id of recruiter |

Table 4. 13 usecase description of view users

|  |  |  |
| --- | --- | --- |
| Usecase Name | View users | |
| Usecase Identifier | UC-11 | |
| Actor | Administrator | |
| Description | Allows Administrator to view users registered to the system | |
| Pre-condition | The administrator must login to the system and go to intended page | |
| Post condition | If the use case is successful the administrator can view users registered to the system | |
| Basic Course of Action | Actor action | System response |
| **Step 1**. open the system webpage  **Step 3**.login to the system  **Step 5.** Click view users | **Step 2**. The system displays home page  **Step 4.** The system displays the admin home page  **Step 6.** The system displays users detail |

## 4.2 Object Model

Object Modeling develops the constant formation of the system concerning the object. It recognizes the objects and the relationship between them. It identifies the attributes and functions of each class.

### 4.2.1. Class Diagram

UML class diagrams is the main building block of object-oriented modeling. The class diagram below shows the attributes, operations, and association of the classes within our proposed system.

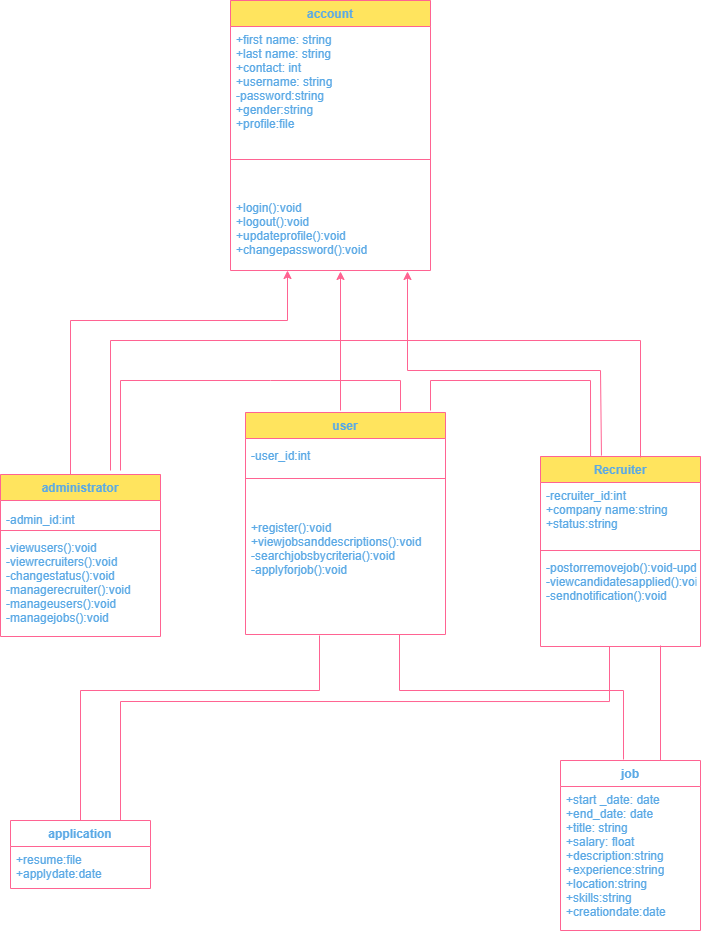


Figure 4. 2 Class diagram

### 4.2.2. Data Dictionary

The following table shows attributes, data type and key constraint of the classes in our proposed system.

Table 4. 14 Data dictionary for user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Caption | Example | Data type | Attribute size | Constraints |
| id | Id | 1 | int | 10 | Primary key |
| First\_name | First name | Tsadiku | varchar | 20 | Not null |
| Last\_name | Last name | Kibru | varchar | 20 | Not null |
| username | Username | tsadiku@gmail.com | varchar | 30 | Not null |
| password | Password | tsadiku@123456 | varchar | 15 | Not null |
| mobile | Mobile | 0923697891 | int | 10 | Not null |
| image | image | Profile.png | file | 1MB | Not null |
| gender | Gender | Male | varchar | 10 | Not null |
| type | type | User | varchar | 20 | Not null |

Table 4. 15 Data dictionary for recruiter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Caption | Example | Data type | Attribute size | Constraints |
| id | Id | 1 | int | 10 | Primary key |
| First\_name | First name | Tsadiku | varchar | 20 | Not null |
| Last\_name | Last name | Kibru | varchar | 20 | Not null |
| username | Username | tsadiku@gmail.com | varchar | 30 | Foreign key |
| password | Password | tsadiku@123456 | varchar | 15 | Not null |
| mobile | Mobile | 0923697891 | int | 10 | Not null |
| image | image | Profile.png | file | 1MB | Not null |
| gender | Gender | Male | varchar | 10 | Not null |
| type | type | Recruiter | varchar | 20 | Not null |
| company | Company | Commercial bank of Ethiopia | varchar | 40 | Not null |
| status | Status | Pending, accepted, rejected | varchar | 10 | Not null |

Table 4. 16 Data dictionary for job

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Caption | Example | Data type | Attribute size | Constraints |
| Id | Id | 1 | int | 10 | Primary key |
| Recruiter | Recruiter | Mezgebu mulat | varchar | 20 | Foreign key |
| start\_date | Start date | Jan. 24, 2023 | date | 20 | Not null |
| end\_date | End date | Jan. 25, 2023 | date | 30 | Not null |
| Title | Title | Python django developer | varchar | 15 | Not null |
| Salary | Salary | 9000.2 | float | 10 | Not null |
| Image | Image | Profile.png | file | 1MB | Not null |
| Description | Description | Basic python programming | varchar | 10 | Not null |
| Experience | Experience | 3 | int | 10 | Not null |
| Location | Location | wolkite | varchar | 40 | Not null |
| Skills | Skills | Basic python programming | varchar | 10 | Not null |
| Creationdate | Creation date | Jan. 8, 2023 | date | 10 | Not null |

Table 4. 17 Data dictionary for application

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Caption | Example | Data type | Attribute size | Constraints |
| id | Id | 1 | int | 10 | Primary key |
| job | Job title | python django developer | varchar | 20 | Foreign key |
| Job\_seeker | Job seeker | amir jemal | varchar | 20 | Not null |
| resume | Resume | Apply.pdf | file | 30 | Not null |
| applydate | Apply date | Jan. 8, 2023 | date | 10 | Not null |

## 4.3 Dynamic Model

Dynamic Model is used to express and model the behavior of the system over time. It concerned with the temporal changes in the states of the objects in a system. We use sequence diagram, activity diagram and state diagram in our system.

### 4.3.1 Sequence Diagram

Sequence diagram is used to describe the dynamic behavior of several objects over time. It is good for real-time specifications. It provides a sequential map of messages passing between objects over time. How users interact with the system and what happens internally to get the work done.

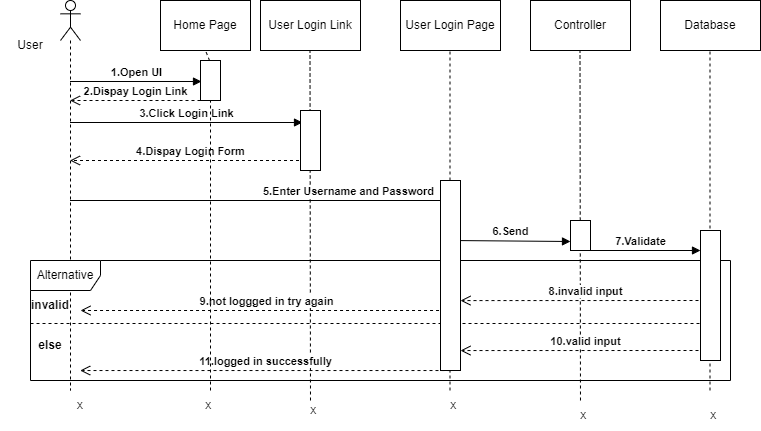


Figure 4. 3 sequence diagram for user login

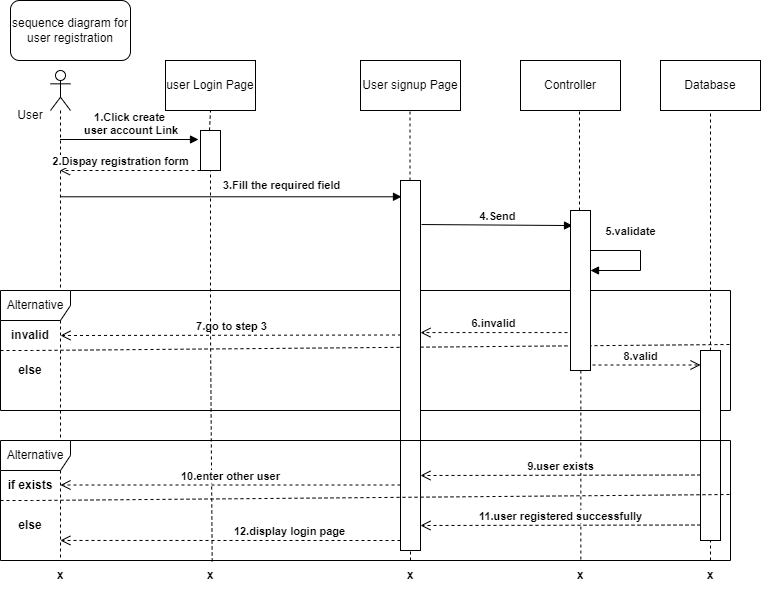
****

Figure 4. 4 sequence diagram for user registration

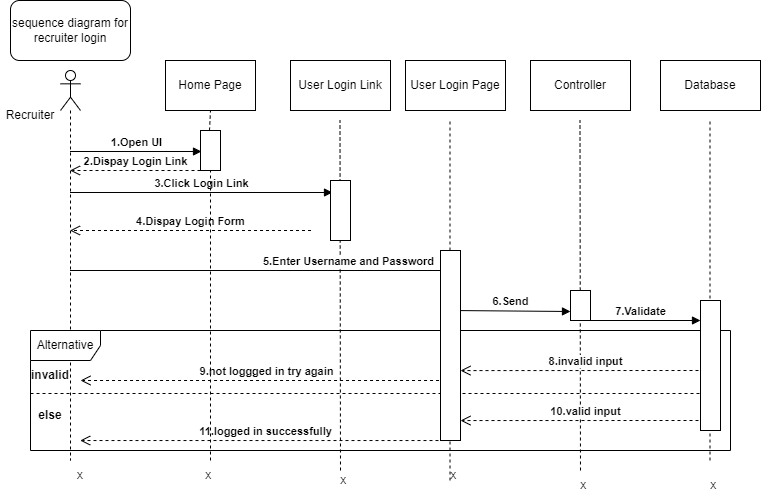


Figure 4. 5 sequence diagram for recruiter login

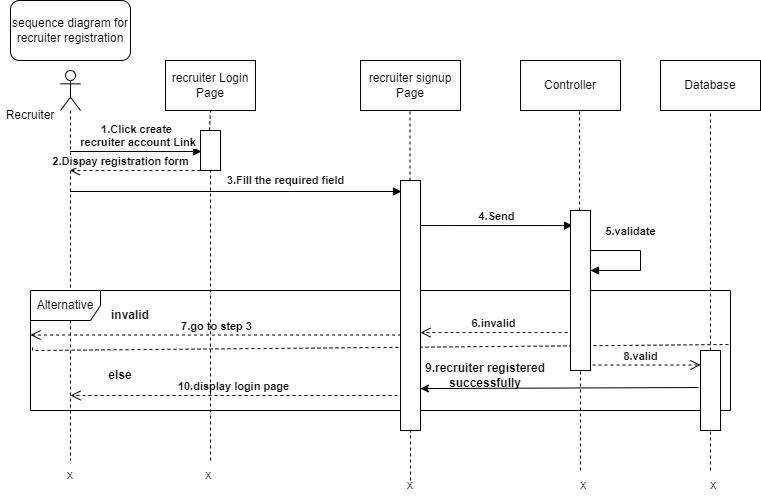


Figure 4. 6 sequence diagram for recruiter registration

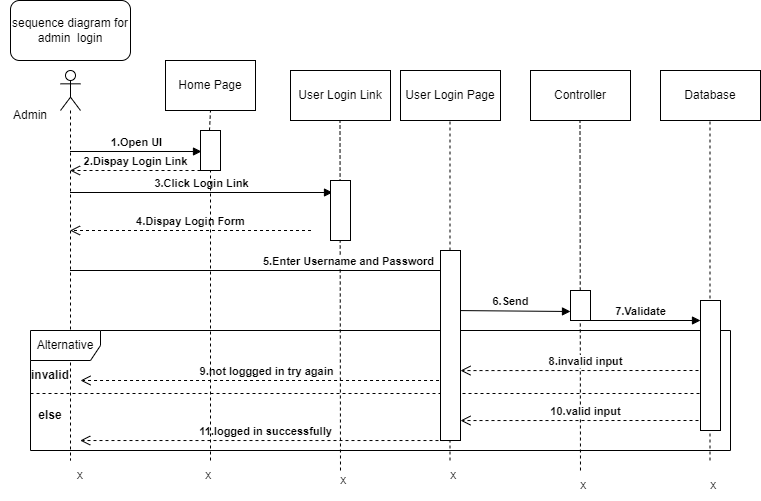
****

Figure 4. 7 sequence diagram for admin login

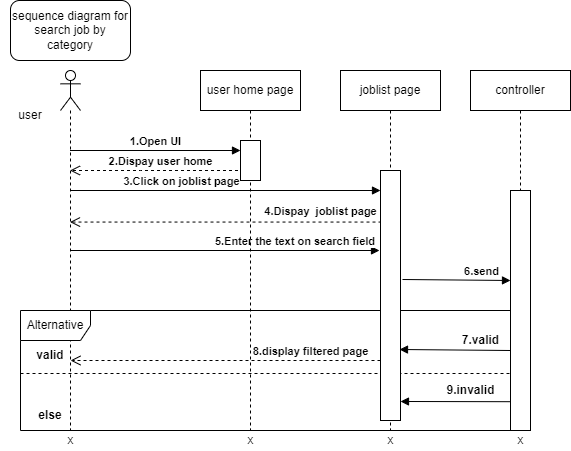


Figure 4. 8 sequence diagram for search job by category

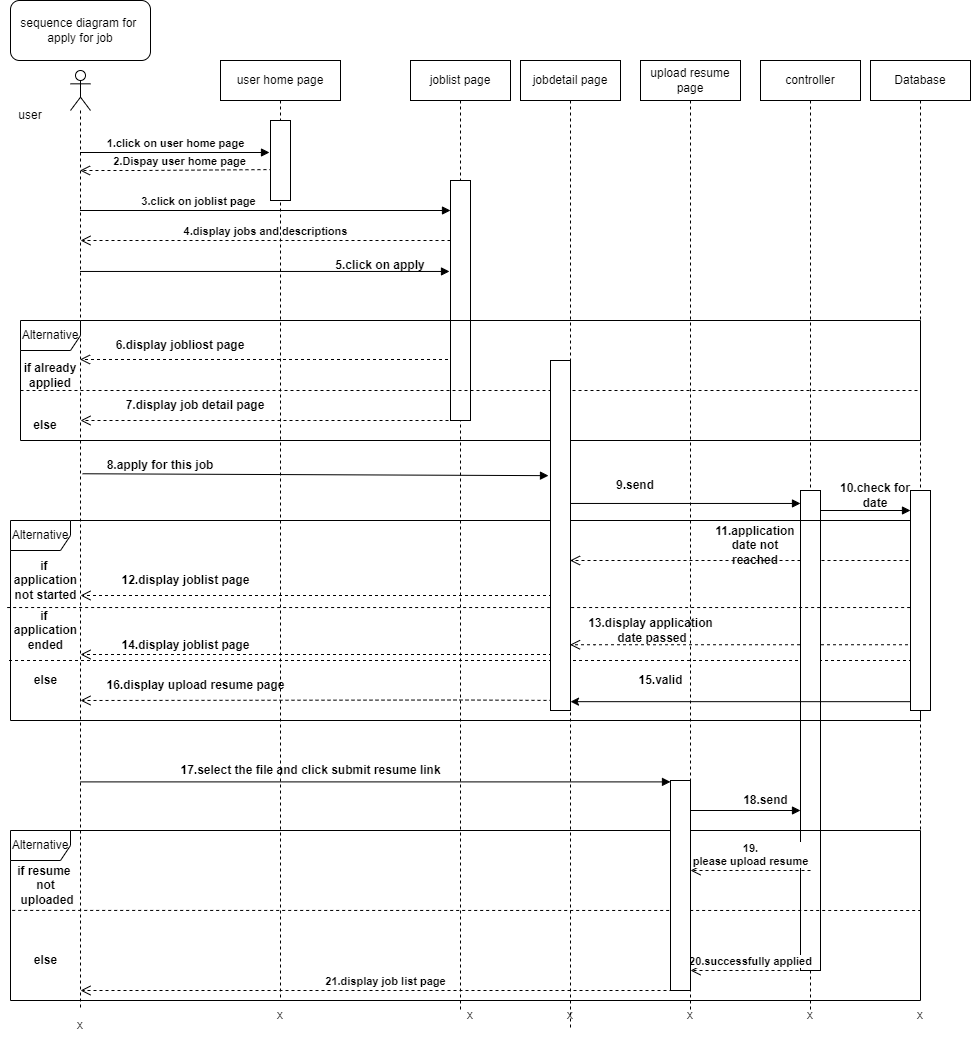


Figure 4. 9 sequence diagram for apply for job

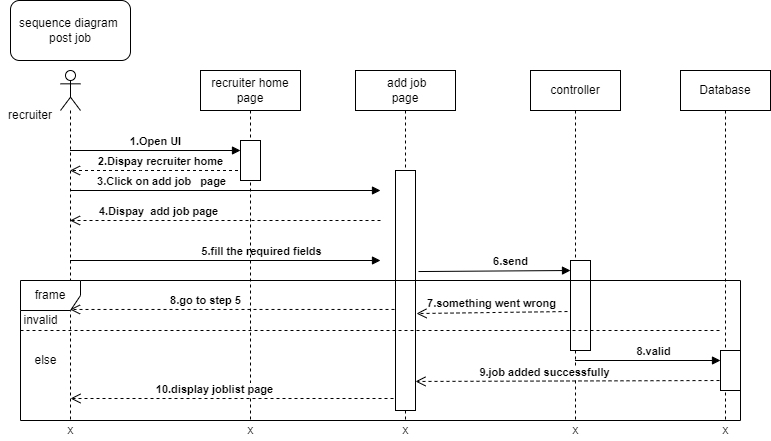
****

Figure 4. 10 sequence diagram for post job

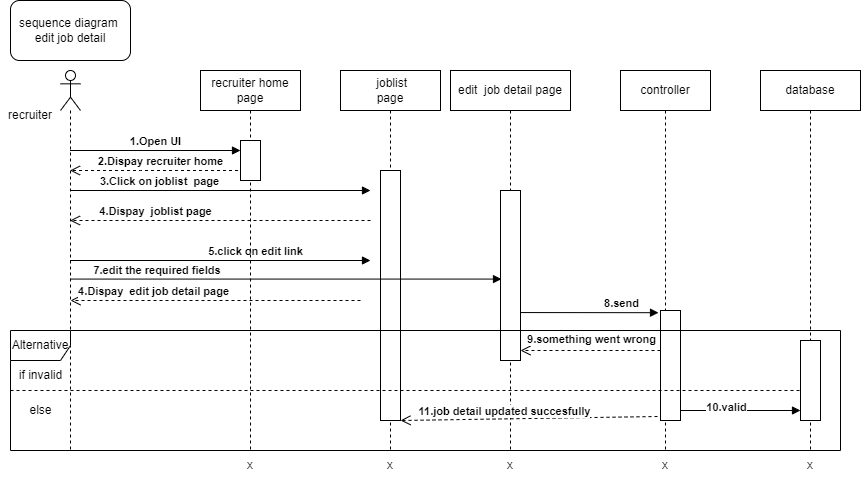


Figure 4. 11 sequence diagram for update job detail

### 4.3.2 Activity Diagram

Activity diagram used to emphasize the flow of control from activity to activity or to model the flow of an object as it moves from the state at different points in the flow of control.

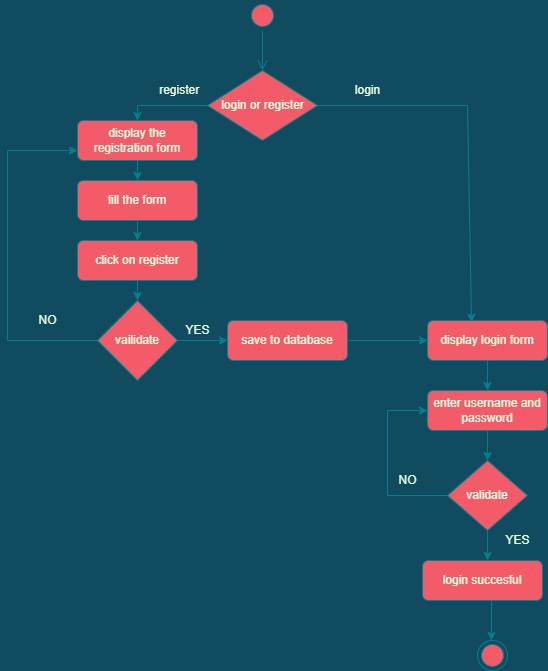


Figure 4. 12 activity diagram for login and registration

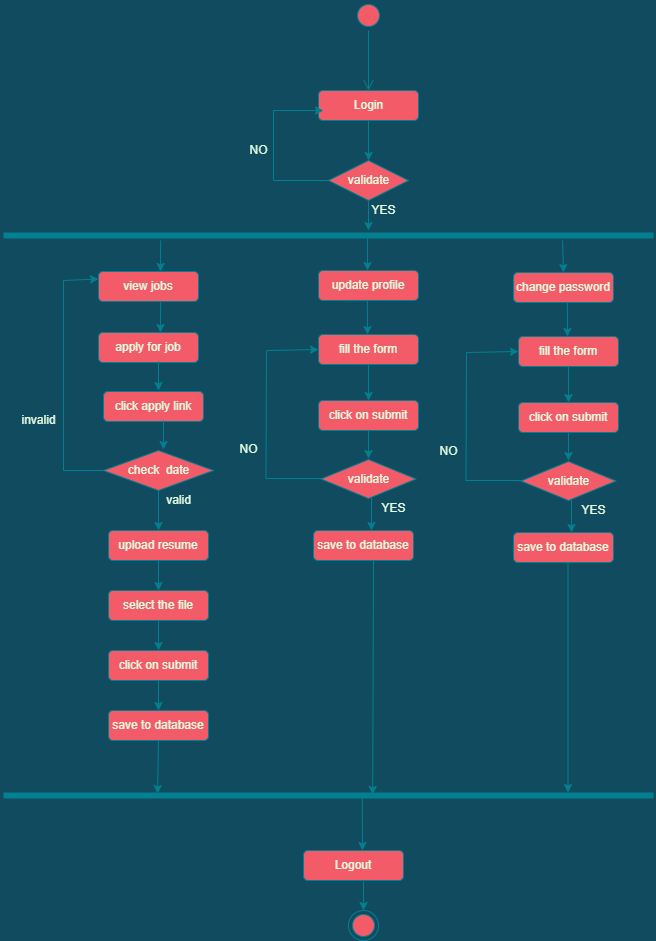


Figure 4. 13 activity diagram for user

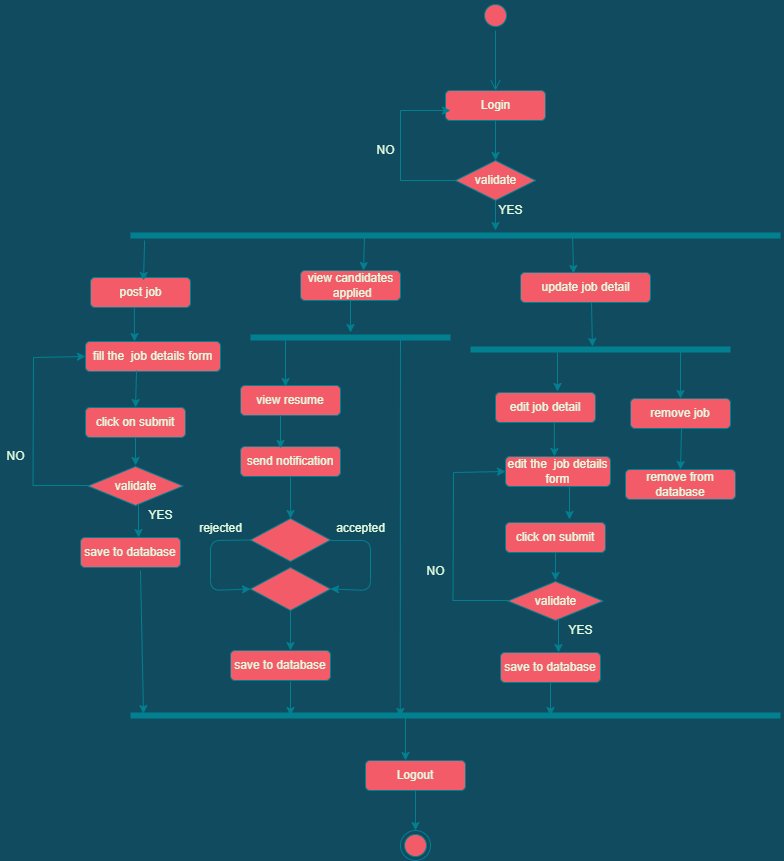
****

Figure 4. 14 activity diagram for recruiter

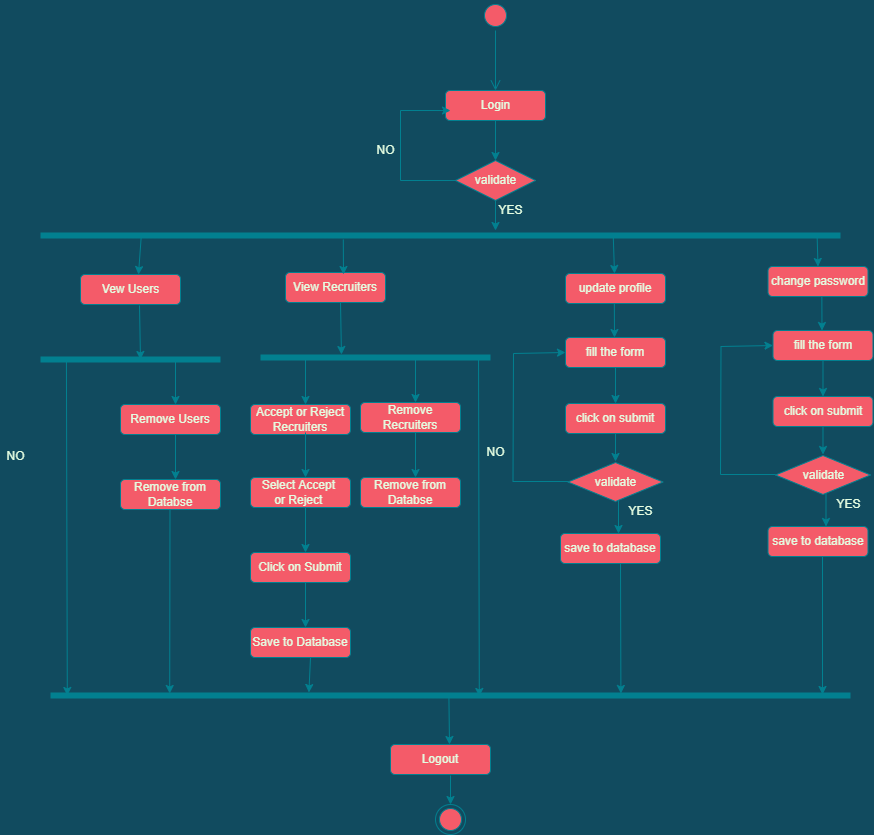
****

Figure 4. 15 activity diagram for admin

## 

### 4.3.3 State Chart Diagram

State chart diagram is UML diagrams used to model the dynamic nature of a system. It is simply a presentation of a state machine that shows the flow of control from state to state.

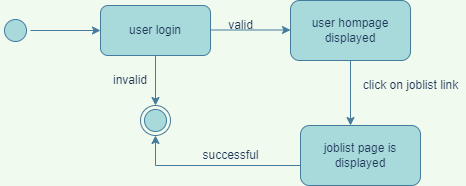


Figure 4. 16 state chart diagram for view job

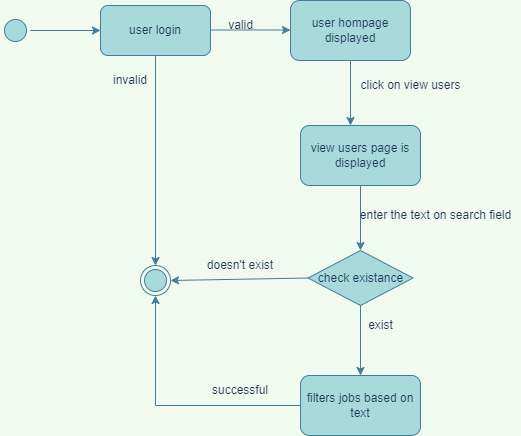
****

Figure 4. 17 state chart diagram for search by category

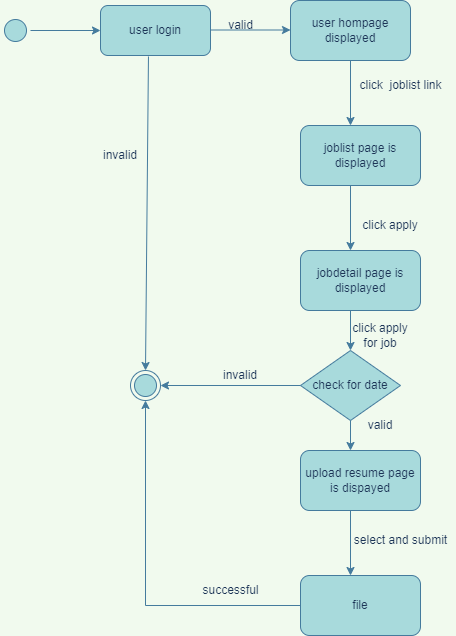


Figure 4. 18 state chart diagram for apply for job

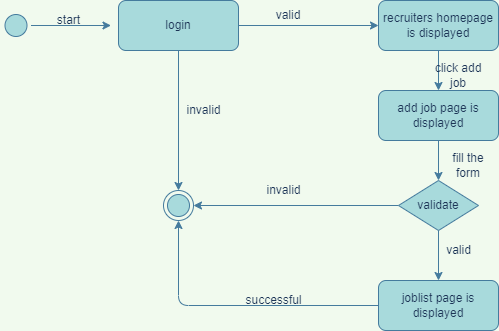
****

Figure 4. 19 state chart diagram for post job

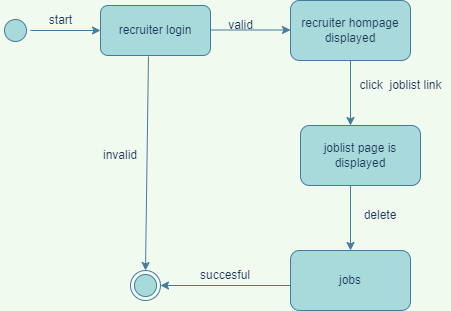


Figure 4. 20 state chart diagram for remove job

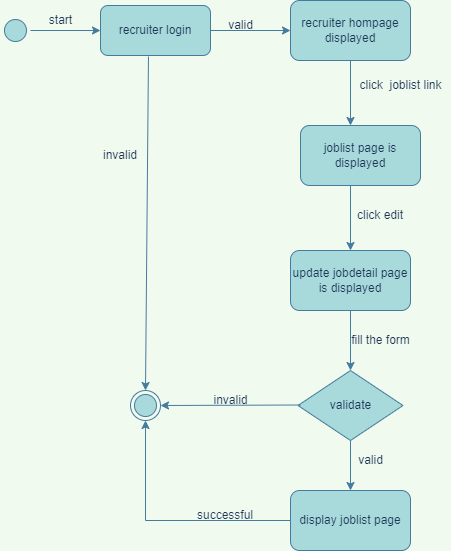


Figure 4. 21 state chart diagram for update job detail

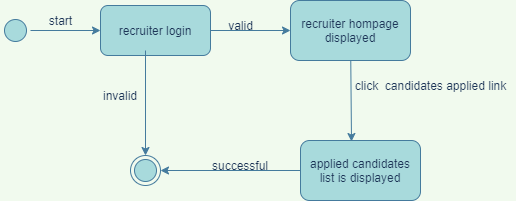


Figure 4. 22 state chart diagram for view candidates applied

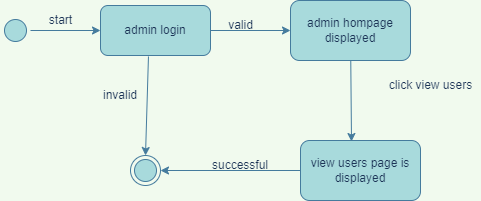


Figure 4. 23 state chart diagram for view users

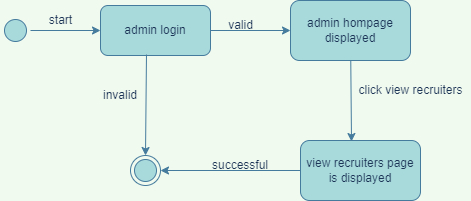


Figure 4. 24 state chart diagram for view recruiters

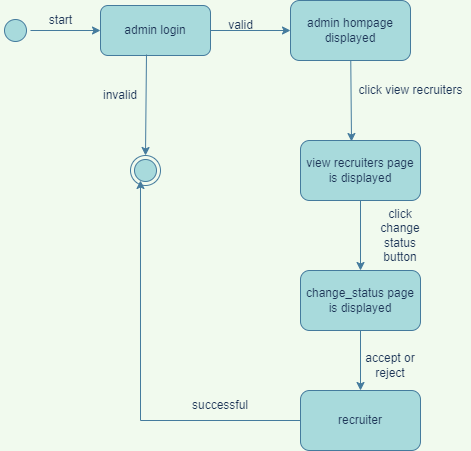


Figure 4. 25 state chart diagram for accept or reject recruiters

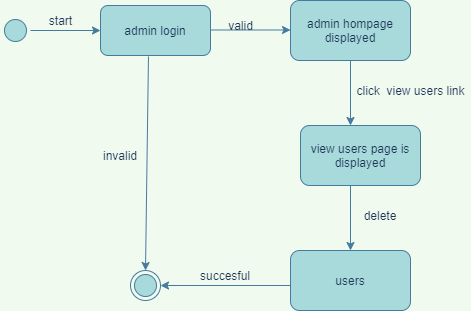
****

Figure 4. 26 state chart diagram for remove users

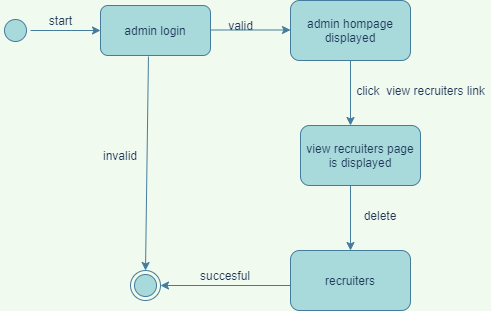
****

Figure 4. 27 state chart diagram for remove recruiters

# CHAPTER FIVE

## 5. SYSTEM DESIGN

## 5.1. Design Goals

The design goals are derived from non-functional requirements that means nonfunctional  
requirement is the description of the feature characteristics and attributes of our proposed system.  
The purpose of designing is to show the direction of how the system is built and to obtain clear  
and enough information needed to drive the actual implementation of the system. The objectives  
of design are to model the system with high quality. Goal of system design is to manage complexity  
by dividing the system into manageable pieces.

**Performance** is one key factor that influences system quality. Therefore, our proposed system is responsible for data interchange between client and server. In performance, we consider the  
following: -

**Response Time:** -As Django framework provides a strong caching framework, no need to  
recompute again and again the same value. So it saves value that is repetitively requested by the  
client on cache and it minimizes the time the request takes to reach the database every time the  
client requests.

**Availability:** as long as there is an internet connection the system will be available always.

**Security:** Django does a great job at user authentication and authorization. Passwords are hashed  
using a Password-Based Key Derivation Function (PBKDF2) by default user passwords are  
protected well.

**Maintenance**  
In times of failure or need modification, the system needs to be maintained. To be maintainable  
the system should meet the following maintenance criteria: -

**Extensibility:** if it is needed to add new functionality to the system, this must be achieved by only making a separate page and integrating this page with the existing system. The system we proposed is only for the classroom if it is needed to be used in a bureau for staff management; it can be simply added without destructing other functionalities of the system.

**Modifiability**: When functionality in a proposed system requires modification, it must be done  
specifically to that function or page without affecting the overall system organization.

**Portability**: The proposed system is developed to be viewed and retrieved from any web browser regardless of their version and platform it resides in.

**End-User:**  
From the user point of view the system should provide the following end user criteria:  
**Utility:** in order to help the user, to easily understand and interact with the system.  
**Usability:** the system is easy and user-friendly. It doesn`t require a high level of expertise in  
computers. Any user can access it simply using the user guide. The system should be designed  
incorporating the following usability concepts: -

**Consistent page pattern:** The proposed system should be designed to meet user expectations. To  
determine where the text content, navigation, images and other important elements are placed on  
the website. Users become aware of the standard location for elements. So elements in our system  
should be consistent on every page. Users of the system spend less time trying to use the website  
and more time engaging with the content.

**Less over-crowded interface:** Our system reduces user frustration that happens when it is not  
clear what to do next. The user interface must be simple to use and navigate from one page to  
another page. By displaying instructional text or line (arrow) that implies the next task. Displaying  
error messages that make the user know what happened and how to correct it.

## **5.2**. Proposed System Architecture

In our system we will use multi-tier architecture: presentation tier, application tier and database  
tier.  
**Presentation tier**

The presentation tier is the user interface and communication layer of the application, where the  
end-user interacts with the application. Its main purpose is to display information and collect  
information from the user.

Application tier

The application tier, also known as the logic tier or middle tier, is the heart of the application. In  
this tier, information collected in the presentation tier is processed - sometimes against other  
information in the data tier - using business logic, a specific set of business rules. The application  
tier can also add, delete or modify data in the data tier.

Data-tier

The data tier, sometimes called the database tier, data access tier, or back-end, is where the  
information processed by the application is stored and managed.

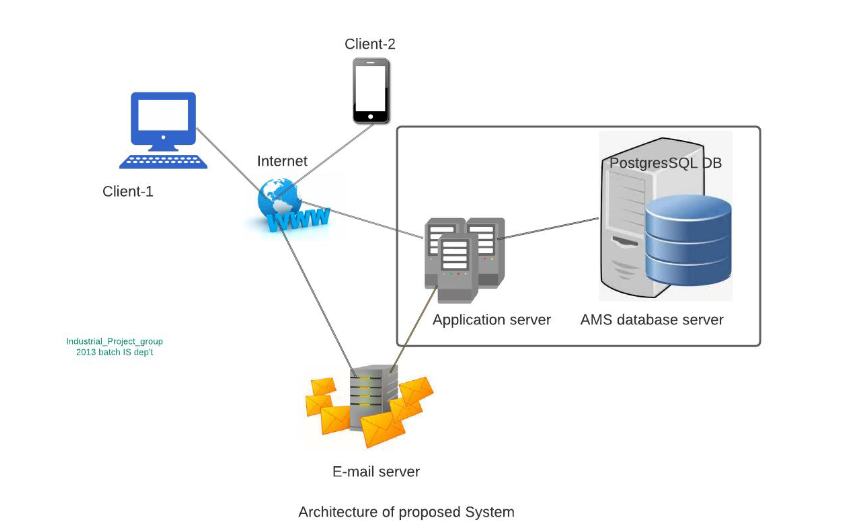


Figure 5. 1 Architecture of proposed system

### 5.2.1. **Subsystem Decomposition and Description**

Decomposing the system enables us to minimize the complexity of the system by decomposing it  
into subsystems. Subsystems identified from our system are: -  
**Manage User Subsystem:** for managing user information and performing those operations.

⮚ view users  
 ⮚ remove users  
**Manage Recruiter Subsystem:** for managing recruiter information and performing those operations.

⮚ view recruiters  
 ⮚ remove recruiters

* Accept or Reject recruiter

**Manage Job Subsystem: for managing job and perform the following activities**

* Post job
* Remove job
* Update job detail

**Application Management Subsystem:**

* View candidates applied
* View resume

**Application Process Subsystem:**

* Apply for job
* Upload resume

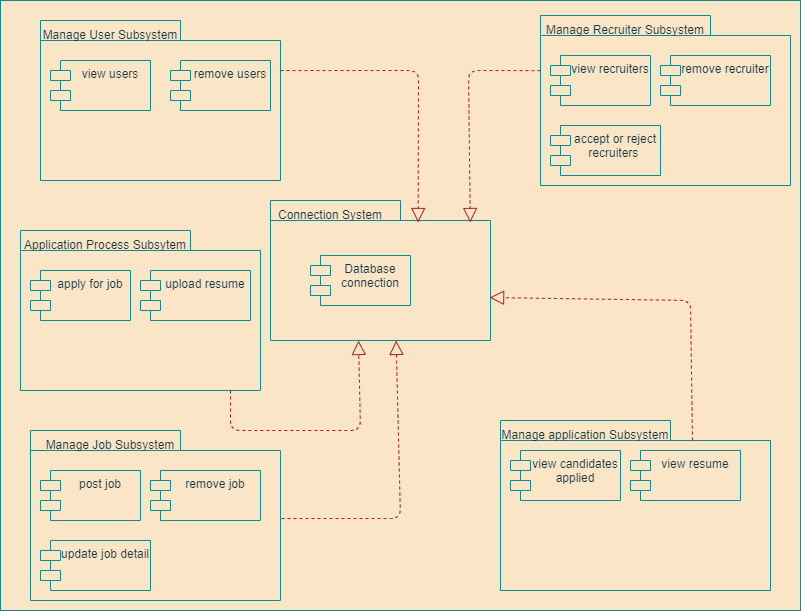


Figure 5. 2 Subsystem Decomposition

5.2.2. Hardware/Software Mapping  
Hardware or software mapping describes how subsystems are assigned to hardware and  
customized components. We, use a UML deployment diagram to diagrammatically illustrate the  
hardware/software mapping of our proposed system.

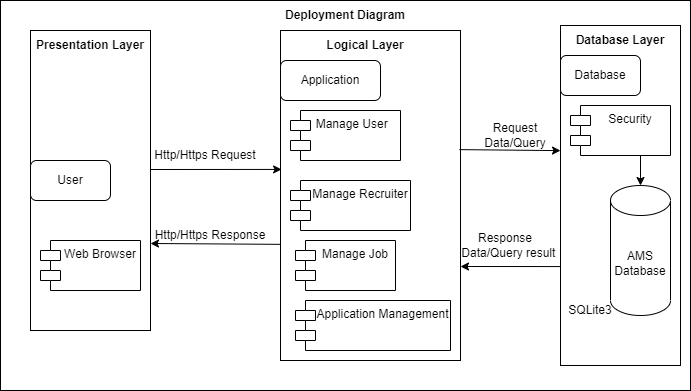


Figure 5. 3 Hardware/Software Mapping

5.2.3. Detailed Class Diagram

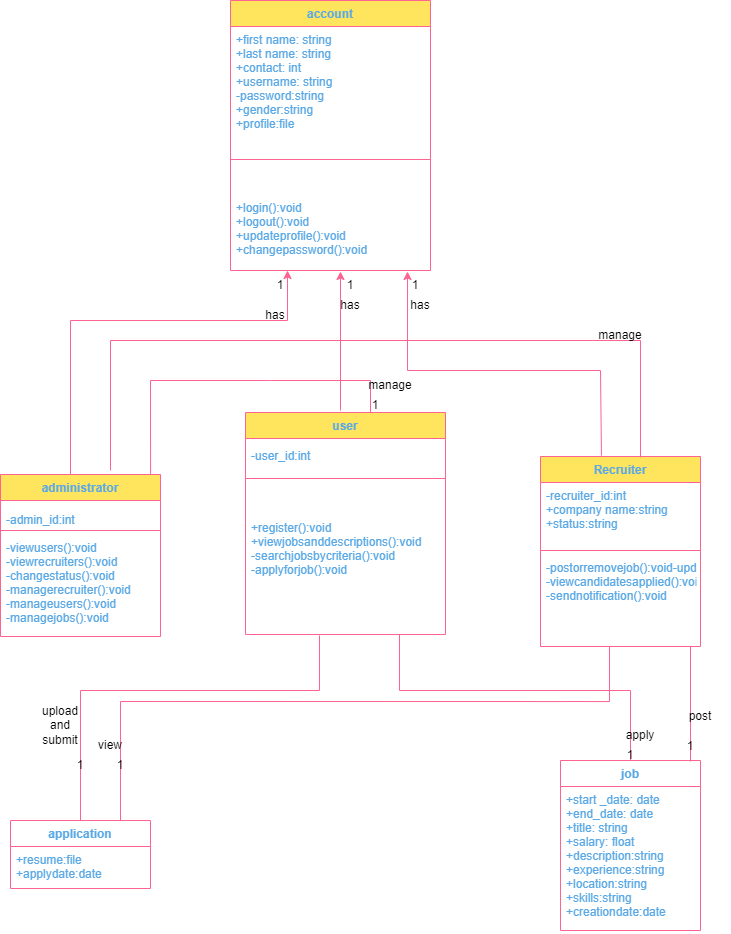


Figure 5. 4 Detailed Class Diagram

5.2.4. Persistent Data Management

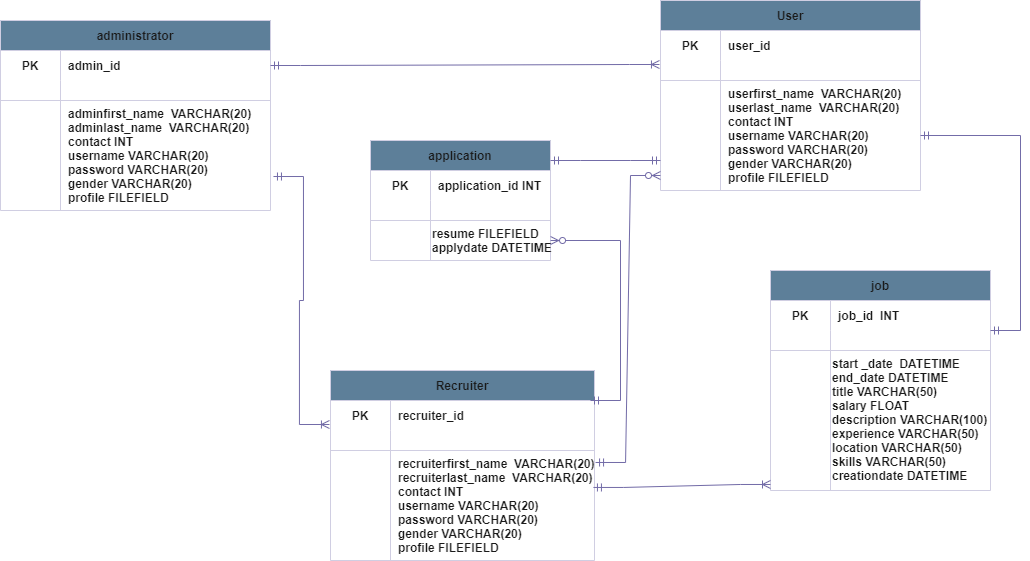


Figure 5. 5 persistent data management

5.2.5. Access Control and Security

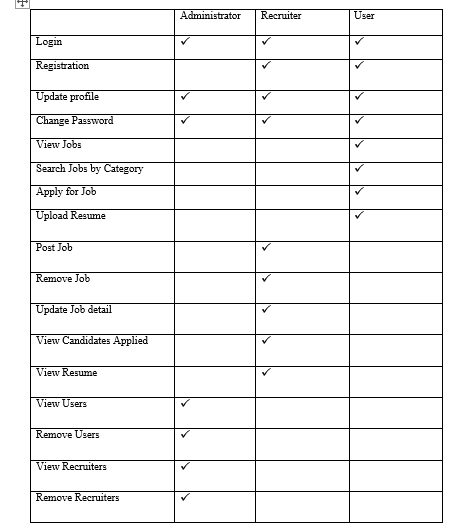
****

Table 5. 1 Access Control and Security

5.3. Algorithm Design  
Algorithms are designed to show the flow of programs in the system. Pseudocode is a method of  
planning which enables the programmer to plan without worrying about syntax. Algorithms show the flow and steps of logic in each function.  
Pseudocode for Login  
step1: BEGIN  
step2: enter username and password on the given form.  
Step3: check Database with entered input  
step3.1: IF valid, THEN logged in successfully  
step3.2: IF invalid, THEN not logged in successfully and try again with correct username and  
password.

Pseudocode for Registration  
step1: BEGIN  
step2: IF actor clicks registration link, THEN form displayed.  
Step3: Actor fill required data.  
Step4: If filled data valid and actor with this data is not exist, then add actor to Database  
Step5: Display actor registered successfully.  
Step6: IF filled data invalid or the actor with this information is preexist, THEN go to step3 and re-enter valid input, or  
step7: ENDIF

Pseudocode for apply for job

Step1: BEGIN

Step2: Login to the system as user

Step3: IF user click on joblist link then list of jobs are displayed

Step4: Click on apply on right of jobs

Step5: Display job detail page

Step6: Click on apply for job

Step7: Check start date and end date

Step7.1: IF valid then upload resume page is displayed

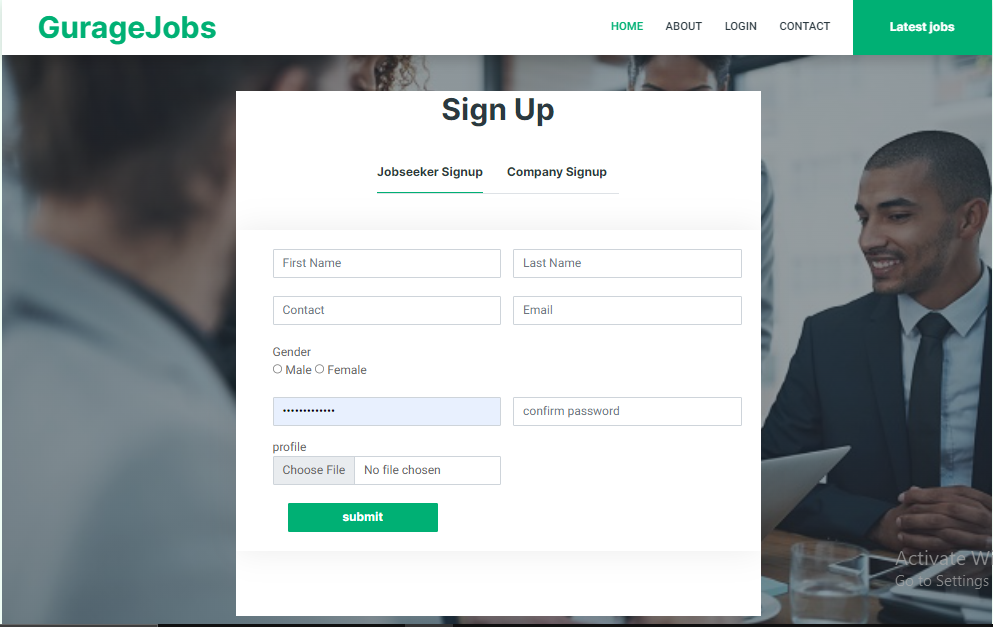
Step7.2: IF start date is not reached, display application date is not reached message

Step7.3 IF end date is passed, display application date is passed message

Step8: select file and click submit button

Step9: END

## 5.4. User interface design



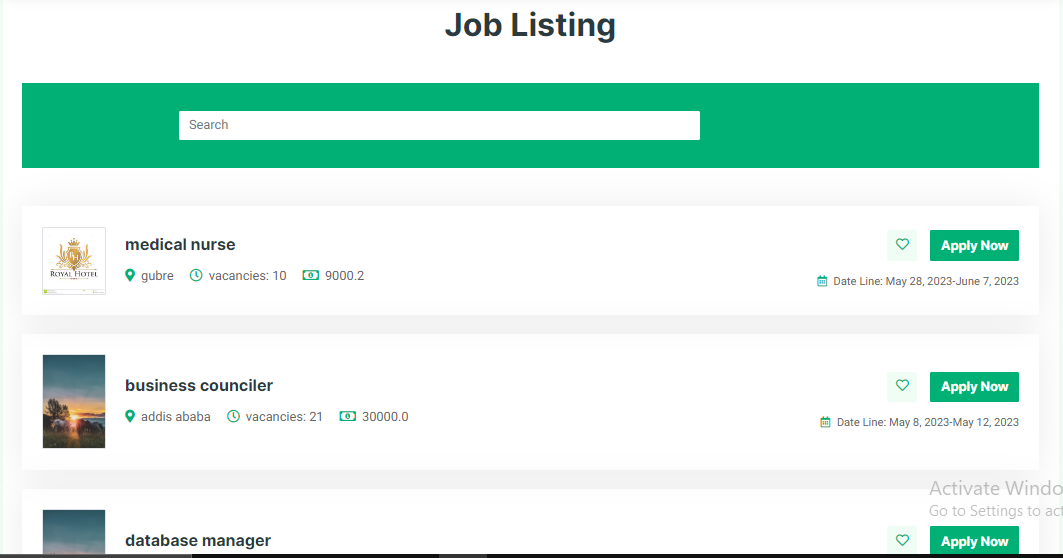


Figure 5. 6 user interface

# CHAPTER SIX

# 6. IMPLEMENTATION AND TESTING

6.1. Implementation  
Implementation is one of parts of the project development phase where project inputs are  
converted to project outputs. In implementation phase, the main task is coding. In implementation;  
we tried to put into practice what was proposed in the project document i.e., transforming the  
project proposal into the actual project.

6.2. Implementation of the Database  
We have chosen SQLite3 database because SQLite3 is a highly scalable open source,  
object-relational database management system (ORDBMS) that’s designed to handle a range of  
operations and perform well with complex datasets.

* It is an advanced open-source object-relational system which applies SQL language.  
  SQLite3 allows you to store large and sophisticated data safely. It helps developers to  
  build the most complex applications, run administrative tasks and create integral  
  environments.
* SQLite3 supports advanced data types and performance optimization features, which  
  are only available in expensive commercial databases, like Oracle and SQL Server.
* It is backed by an experienced community of developers who have made tremendous  
  contributions to make it a highly reliable DBMS.

from django.db import models

# Create your models here.

from django.contrib.auth.models import User

class StudentUser(models.Model):

user=models.ForeignKey(User,on\_delete=models.CASCADE)

mobile=models.CharField(max\_length=15,null=True)

image=models.FileField(null=True)

gender=models.CharField(max\_length=10,null=True)

type=models.CharField(max\_length=15,null=True)

def \_\_str\_\_(self):

return self.user.username

class Recruiter(models.Model):

user=models.ForeignKey(User,on\_delete=models.CASCADE)

mobile=models.CharField(max\_length=15,null=True)

image=models.FileField(null=True)

gender=models.CharField(max\_length=10,null=True)

company=models.CharField(max\_length=100,null=True)

company\_detail=models.CharField(max\_length=300,null=True)

type=models.CharField(max\_length=15,null=True)

status=models.CharField(max\_length=20,null=True)

def \_\_str\_\_(self):

return self.user.username

class Job(models.Model):

recruiter=models.ForeignKey(Recruiter,on\_delete=models.CASCADE)

start\_date=models.DateField()

end\_date=models.DateField()

title=models.CharField(max\_length=100)

salary=models.FloatField(max\_length=20)

vacancy=models.IntegerField()

nature=models.CharField(max\_length=50)

image= models.FileField()

description=models.CharField(max\_length=300)

experience=models.CharField(max\_length=50)

location=models.CharField(max\_length=100)

skills=models.CharField(max\_length=100)

category=models.CharField(max\_length=100,null=True)

creationdate=models.DateField()

def \_\_str\_\_(self):

return self.title

class Apply(models.Model):

job=models.ForeignKey(Job,on\_delete=models.CASCADE)

student=models.ForeignKey(StudentUser,on\_delete=models.CASCADE)

resume=models.FileField(null=True)

applydate=models.DateField()

portifolio\_website=models.CharField(max\_length=100)

coverletter=models.CharField(max\_length=300)

def \_\_str\_\_(self):

return self.id

class Profile(models.Model):

degree=models.CharField(max\_length=50)

university=models.CharField(max\_length=50)

skills=models.CharField(max\_length=300)

experience=models.CharField(max\_length=50)

about\_you=models.CharField(max\_length=50)

languages=models.CharField(max\_length=50)

def \_\_str\_\_(self):

return self.id

## 6.3. Implementation of Class Diagram Methods implemented in our system:

### Administrator

-manageUsers ()

-manageRecruiters ()

### Recruiter

-manageJob()

-viewCandidatesApplied()

### User

+register()

-viewjobs()

-searchJobbyCategory()

-applyforjob()

## Variables and class implemented in our system:

### User class

# ForeignKey first\_name

# ForeignKey last\_name

+CharField mobile

# ForeignKey username

# ForeignKey password

+ CharField gender

+ FileField image

+ CharField type

### Recruiter class

+ CharField company

+ CharField status

### Job class

+ DateField start\_date

+ DateField end\_date

+ CharField title

+ FloatField salary

+ FileField image

+ CharField description

+ CharField experience

+ CharField location

+ CharField skills

+ DateField creationdate

### Apply class

# ForeignKey job

# ForeignKey student

+ FileField resume

+ DateField applydate

## **6.4. Configuration of Application Server**

We used Web Server Gateway Interface (WSGI) server for running Python web applications.  
Django provides a default WSGI development server. In addition to that, WSGI has the following  
advantages like:

Flexibility: You can actually change the web stack components without changing the code at all,  
and without even changing the application that runs the WSGI servers.

Scalability: WSGI servers can handle thousands of requests concurrently and route them from the web server through the best possible means.

Speed: WSGI helps speed up Python web application development because you don’t need to  
know any complicated things, just a basic knowledge of how the interface works.

Simple: The learning curve of WSGI is simple, making it easier to pick up, and with no  
configuration or installation hassles. This is one of the biggest benefits of WSGI, and developers  
wholeheartedly embrace it.

Reusable middleware: You can enhance the functionality of WSGI through existing middleware components like authentication/authorization, caching, filtering, etc. The reusability feature saves time.

## 6.5. Configuration of Application security

Since we are using the Django framework in python language, we were able to achieve the  
following security issues.

Cross site request forgery (CSRF) protection

CSRF protection works by checking for a secret in each POST request. This ensures that a  
malicious user cannot “replay” a form POST to your website and have another logged in user  
unwittingly submit that form. The malicious user would have to know the secret, which is user  
specific (using a cookie).

When deployed with HTTPS. CsrfViewMiddleware will check that the HTTP referrer header is  
set to a URL on the same origin (including subdomain and port). Because HTTPS provides  
additional security, it is imperative to ensure connections use HTTPS where it is available by  
forwarding insecure connection requests and using HSTS for supported browsers.

SQL injection protection

SQL injection is a type of attack where a malicious user is able to execute arbitrary SQL code on  
a database. This can result in records being deleted or data leakage.

Django’s querysets are protected from SQL injection since their queries are constructed using  
query parameterization. A query’s SQL code is defined separately from the query’s parameters.  
Since parameters may be user-provided and therefore unsafe, they are escaped by the underlying  
database driver.

Host header validation

Django uses the Host header provided by the client to construct URLs in certain cases. While these values are sanitized to prevent Cross Site Scripting attacks, a fake Host value can be used for Cross-Site Request Forgery, cache poisoning attacks, and poisoning links in emails.

## 6.6. Testing

Software Testing is considered as one of the important and necessary phases in SDLC because of

the following reasons:

* Testing always helps to verify that complete software requirements are implemented

correctly or not, means it is implemented according to the defined requirements or not.

* Testing definitely helps in identifying defects/bugs and ensuring they are

recognized/addressed before software deployment stage.

* Testing also demonstrates that software/application appears to be working according to

specifications and the performance requirements that have been defined is also met.

### 6.6.1.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 VsCode, Google chrome and Django frame-work and SQLlite are  
used.

**VsCode**

VSCode (Visual Studio Code) is a versatile code editor widely used by programmers to edit a variety of programming languages, including Python, PHP, HTML, JavaScript, and CSS. Its powerful features and extensive ecosystem make it a popular choice among developers who value flexibility and efficiency.

VSCode offers a user-friendly code editor that enhances the programming experience. It provides a clean and customizable interface, allowing programmers to personalize their coding environment according to their preferences. With a vast library of themes and extensions, developers can easily tailor the editor's appearance and functionality to suit their needs.

**Django framework**

Django is an open-source framework for backend web applications based on Python one of the top web development languages. Its main goals are simplicity, flexibility, reliability, and scalability.

Django has its own naming system for all functions and components (e.g., HTTP responses are

called “views”). It also has an admin panel, which is deemed easier other technical features,

including:

* Simple syntax
* Its own web server for development
* MVT (Model-View-Template) core architecture
* “Batteries included” (comes with all the essentials needed to solve solving common cases)
* An ORM (Object Relational Mapper)
* HTTP libraries
* Middleware support and
* A Python unit test framework.

### 6.6.4. 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. It is a way of testing each of the system functionality independently.

### 6.6.5. Integration Testing

In this level of testing, we have examined how the different procedures work together to achieve

the goal of the subsystem. So, we integrate each component from single function to the main

function incrementally.

* Check the interaction between individual functionality which performs the specific tasks.
* Evaluate the functionality of the subsystem after combination all individual functionality.
* Identify the Independence of each subsystem with another subsystem

### 6.6.6. System Testing

In this level of testing process, we have examined how the whole subsystems work together to

achieve the desired goal. Generally, under this testing is mainly concerned with areas such as

performance, security and validation.

* We evaluated the functionality of the subsystem after a combination of individual

subsystem whether it works correctly or not.

* Check the coherence and coupling of each subsystem.
* Check the overall functionality that achieves the user’s requirement.

### 6.6.7. Acceptance Testing

Acceptance testing is a level of the software testing where a system is tested for acceptability. The processes whereby actual users are test a completed information system. We gave our system to Instructor which is one of our system users. While we made user acceptance testing our system user is satisfied (when we describe in percent the user gives us 75% out of 100). And the user gives us some feedback that should be improved. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

# CHAPTER SEVEN

# 7. CONCLUSION AND RECOMMENDATION

## 7.1. Conclusion

In conclusion, the Online Job Portal System presented in this documentation offers an efficient and user-friendly platform for job seekers to showcase their qualifications, search for relevant job listings, and apply to multiple positions simultaneously.

we conclude the project documentation for the Online Job Portal System. Throughout this documentation, we have presented a comprehensive overview of the system's design, functionality, and implementation. The Online Job Portal System provides a platform for users of different profiles to upload their CVs, search for jobs based on their qualifications, and conveniently apply for multiple positions simultaneously using their unique user IDs.

The system's success in achieving its goals and providing a seamless user experience makes it a valuable tool in today's competitive job market. By continuously updating and enhancing the system's features, it has the potential to revolutionize the way job seekers connect with employers and facilitate the hiring process in the future.

## 7.2. Recommendation

In this project, we provide recommendations for the future development and improvement of the Online Job Portal System. These suggestions aim to enhance the system's functionality, user experience, and overall effectiveness in connecting job seekers with potential employers. The following recommendations are based on emerging trends and potential areas for further innovation within the online job portal domain.

* **Implement AI-based Job Matching**:Consider integrating artificial intelligence (AI) algorithms into the system to provide personalized job recommendations to users. By analyzing user profiles, qualifications, and job preferences, the system can offer tailored suggestions that match their skills and interests. AI can also help in predicting the likelihood of success in specific job applications, increasing the efficiency of the job search process.
* **Introduce Skill Assessment Tests:**Integrate skill assessment tests within the system to help job seekers evaluate their strengths and weaknesses. By providing users with the option to take skill tests relevant to their desired positions, the system can offer personalized recommendations for skill development and training opportunities, enhancing their employability.
* **Mobile Application Development**:Consider developing a mobile application for the Online Job Portal System to provide users with a more convenient and accessible experience. A mobile app would allow job seekers to search for jobs, apply for positions, and receive notifications on-the-go, increasing engagement and responsiveness.

By implementing these recommendations, the Online Job Portal System can evolve into a more comprehensive and dynamic platform, providing enhanced value to both job seekers and employers. Continual innovation and adaptation to industry trends will ensure the system remains at the forefront of the online job portal landscape, effectively addressing the evolving needs of users in the competitive job market.