**UNIVERSITY OF DAR ES SALAAM**



**COLLEGE OF INFORMATION AND COMMUNICATION**

**TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

##### PROGRESS REPORT CS 499

**PROJECT TITLE:** DESIGN AND IMPLEMENTATION OF JOB PORTAL WITH AUTOMATED INTERVIEW.

**STUDENT NAME:** MAHUNDI, YONA

**REGISTRATION NO:** 2013-04-02422

**ACADEMIC SUPERVISOR**: Dr. KALINGA

**SIGNATURE: ………………………………**

**DATE: 14th June 2017**

# DECLARATION

I MAHUNDI, YONA declare that this report and the work described in it is my own work, with any contributions from others expressly acknowledged and cited.

I declare the work in this report was carried out in accordance with regulations of University of Dar-es-salaam and has not been presented to any other University for examination in either Tanzania or Overseas. Any views expressed in this report are those of the author and in no way represent those of the University of Dar-es-salaam.

SIGNED: …………………………………………….

DATE: ……………………………………………….

# ACKNOWLEDGEMENT

First and foremost, I would like to thank GOD for giving me health and protectionthroughout the entire semester.

Secondly I would also like to thank my supervisor Dr. KALINDA who has invested her full effort in guiding me in achieving the goal. I have to appreciate the guidance given by other supervisor as well as the panels especially thanks to their comment and advices.

Thirdly, I would like to recognize contributions from staff and students of Department of Computer Science and Engineering of the University of Dar es Salaam for their support.

Furthermore, I would also like to acknowledge with much appreciation the crucial role of the staff of PSRS, who gave the permission to use all required equipment and the necessary materials to complete this task.

# LIST OF ACRONYMS AND ABBREVIATIONS

|  |  |
| --- | --- |
| API | Application program interface |
| COICT | College of Information and Communications Technology |
| MVC | Model–view–controller |
| HTML | Hypertext Markup Language |
| PHP | Hypertext Preprocessor |
| SQL | Structured Query Language |

# ABSTRACT

Technology is dynamic; it evolves to adapt to the needs of the world. With the advent of the

Internet, it is possible to perform online job search with automated interview. The World Wide Web has made it possible to integrate services that were done by different organizations on different platforms to single solution platforms.

This project aims to develop a system that will integrate the job portal system with interview so as to overcome difficulties that pertain the overall process of job recruitment.The methodology that has been used include the use of waterfall as a software development model.

Primary data was collected from PSRS system through open ended interview.

Secondary data was obtained from the internet, newspaper and so on.

Scheduling job applicants for their interviews is a crucial component of the hiring process.

Whether done by a company’s in house human resources department, through an outsourced service or recruiting company, or assigned to hiring coordinators, the task of scheduling and managing interview dates and times—as well as candidate information—can be tedious and timeconsuming, especially if these interviews are scheduled over the phone, in person. And based on recent survey results from the National Association for Business Economics that projects payrolls will rise an average of 170,000 employees per month in 2012, the need for efficient interview scheduling methods will become increasingly vital as the economy continues to improve.

# TABLE OF CONTENTS

[DECLARATION ii](#_Toc485090812)

[ACKNOWLEDGEMENT iii](#_Toc485090813)

[LIST OF ACRONYMS AND ABBREVIATIONS iv](#_Toc485090814)

[ABSTRACT v](#_Toc485090815)

[TABLE OF CONTENTS vi](#_Toc485090816)

[List of Figures ix](#_Toc485090817)

[List of Tables x](#_Toc485090818)

[CHAPTER ONE: INTRODUCTION 1](#_Toc485090819)

[1.1 Background 1](#_Toc485090820)

[1.2 Problem Statement 2](#_Toc485090821)

[1.3 Project Objectives 2](#_Toc485090822)

[1.3.1 General Objective 2](#_Toc485090823)

[1.3.2 Specific Objective 2](#_Toc485090824)

[1.4 Significance of the project 2](#_Toc485090825)

[1.5 Project Scope 3](#_Toc485090826)

[CHAPTER TWO: LITERATURE REVIEW 4](#_Toc485090827)

[2.1 Over View of Related System 4](#_Toc485090828)

[2.1.1 PSRS web system 4](#_Toc485090829)

[2.1.2 SONRU 4](#_Toc485090830)

[2.2 Overview of proposed system 4](#_Toc485090831)

[a) MYSQL database 4](#_Toc485090832)

[b) PHP using Code Igniter Framework 5](#_Toc485090833)

[c) Java script using JQuery Framework 5](#_Toc485090834)

[CHAPTER THREE: METHODOLODY 6](#_Toc485090835)

[3.1 Literature Review 6](#_Toc485090836)

[3.2 Design Approach 6](#_Toc485090837)

[3.3 Data collection 7](#_Toc485090838)

[CHAPTER FOUR: REQUIREMENT CAPTURE AND ANALYSIS 8](#_Toc485090839)

[4.1 Requirements Capture 8](#_Toc485090840)

[4.1.1 Functional Requirements 8](#_Toc485090841)

[4.1.2 Non-Functional Requirements 9](#_Toc485090842)

[4.2 Requirement Analysis 10](#_Toc485090843)

[4.2.1 Use Cases Identification 10](#_Toc485090844)

[4.2.2 Use case Description 11](#_Toc485090845)

[4.2.3 Use case Diagram of the system 18](#_Toc485090846)

[4.2.4 Conceptual Diagram 19](#_Toc485090847)

[4.2.5 System Sequence Diagram 20](#_Toc485090848)

[CHAPTER FIVE: SYSTEM DESIGN 23](#_Toc485090849)

[5.1 Design Class Diagram 23](#_Toc485090850)

[5.2 Database Design 24](#_Toc485090851)

[CHAPTER SIX: IMPLEMENTATION AND UNIT TESTING 25](#_Toc485090852)

[6.1 Overview 25](#_Toc485090853)

[6.2 Implementation stages 25](#_Toc485090854)

[6.2.1 Database Configurations 25](#_Toc485090855)

[6.2.2 System home page 26](#_Toc485090856)

[6.2.3 The Job Post section 27](#_Toc485090857)

[6.2.4 The Registration page 27](#_Toc485090858)

[6.2.5 The login page 30](#_Toc485090859)

[6.2.6 The home page for the job seeker 31](#_Toc485090860)

[6.2.7 The Employer home page 32](#_Toc485090861)

[6.2.8 Administrator home page 37](#_Toc485090862)

[CHAPTER EIGHT CONCLUSION AND RECOMMENDATION 39](#_Toc485090863)

[8.1 Summary 39](#_Toc485090864)

[8.2 Challenges Faced 39](#_Toc485090865)

[8.3 Recommendation 40](#_Toc485090866)

[8.4 Conclusion 40](#_Toc485090867)

[REFERENCES 41](#_Toc485090868)

[APPENDICES 42](#_Toc485090869)

[APPENDIX A 42](#_Toc485090870)

[Project Time Schedule and Budget 42](#_Toc485090871)

[APPENDIX B 45](#_Toc485090872)

[Use Case Description Template 45](#_Toc485090873)

# List of Figures

[Figure 2.1 MVC Diagram 5](#_Toc485091164)

[Figure 3.1: Waterfall model 7](#_Toc485091165)

[Figure 4.1: Use Case Diagram for the system 18](#_Toc485091166)

[Figure 4.2: Conceptual Diagram 19](#_Toc485091167)

[Figure 4.3: Sequence diagram for Job application 20](#_Toc485091168)

[Figure 4.4: Sequence diagram for registration 21](#_Toc485091169)

[Figure 4.5: Sequence diagram for Job Interview 22](#_Toc485091170)

[Figure 5.1: Design Class Diagram for System 23](#_Toc485091171)

[Figure 5.2: Database Design Diagram of System 24](#_Toc485091172)

[Figure 6.1 Tables of the system 26](#_Toc485091173)

[Figure 6.2: Home page 27](#_Toc485091174)

[Figure 6.3: User input validation. 28](#_Toc485091175)

[Figure 6.4: Employer input validation. 29](#_Toc485091176)

[Figure 6.5: Login Form 30](#_Toc485091177)

[Figure 6.6: Job seeker Home page. 31](#_Toc485091178)

[Figure 6.7: Job seeker Home functionalities. 31](#_Toc485091179)

[Figure 6.8: Employer Home page 33](#_Toc485091180)

[Figure 6.9: Employer Home page functionalities. 33](#_Toc485091181)

[Figure 6.10: Employer Post form. 35](#_Toc485091182)

[Figure 6.11: Employer Post form. 36](#_Toc485091183)

[Figure 6.12: Administrator homepage. 37](#_Toc485091184)

[Figure 6.13: Administrator homepage functionalites. 37](#_Toc485091185)

# List of Tables

[Table 4.1: Functional Requirements for the system 8](#_Toc485091106)

[Table 4.2: System Attributes 9](#_Toc485091107)

[Table 4.3: Identification of use cases 10](#_Toc485091108)

[Table 4.4: Register description 11](#_Toc485091109)

[Table 4.5: Update Profile description 12](#_Toc485091110)

[Table 4.6: View Profile description 13](#_Toc485091111)

[Table 4.7: Post Job description 14](#_Toc485091112)

[Table 4.8: Perform Interview description 15](#_Toc485091113)

[Table 4.9: Apply Job description 16](#_Toc485091114)

[Table 4.10: Set interview description 17](#_Toc485091115)

[Table A1: Budget Proposal 42](#_Toc485091116)

[Table A2: Time Schedule for Semester One 43](#_Toc485091117)

[Table A3: Time Schedule for Semester Two 44](#_Toc485091118)

[Table B1: Use Case Description Template 45](#_Toc485091119)

# CHAPTER ONE: INTRODUCTION

## 1.1 Background

Job seeking is an important step towards employment process. With developing county such as Tanzania the importance of employment is of great magnitude especially considering its economy. Job seekers tend to use various ways so as to obtain the information about a particular job post. Such ways include media such as Television, newspaper, radio and so on. But with growing use of information communication technology more methods of recruiting individual have appeared. Such method includes the use of web technology to advertise jobs.

They are known as job portal.

Job portal is a web based site that brings together diverse source of employment information so as individual seeking job can apply for one. Job portal are important as they centralize all job posts on a single system for easy use. Also through technology we can push much further and integrate automated interview system so as interviews can be performed seamlessly.

Automated interview is an online recruitment method conducted using computer-mediated communication (CMC) such as instant messaging, email, or video. Online interviews are separated into synchronous online interviews, for example via online chat which happen in real time online and asynchronous online interviews.

Recently in Tanzania there have been application of job portal technology but most systems have not integrated the interviewing process in this system. It is because our interviewers still use the face to face method of interviewing individuals. Face to face interview is common in Tanzania as most people have not yet been exposed to the internet or do not have technology required to access effectively this growing community of the internet.

With this system developed the process cost of the interview process will have decreased exponentially and individual shall be exposed to the online community so as to ease their entire life.

## 1.2 Problem Statement

With manual methods of searching for a job and interviewing an individual our community is faced with inefficient ways of recruiting individuals with variety of needed skills and yet maintain equality through the whole process. The inefficiency include employers are faced with additional cost of advertising job post in different media, payment for venues so as interview can be performed. By the other side of the coin job seekers also need additional cost for transport so as they can reach venue where they can perform the interview, they need to have access to communication media such as television or newspaper so they can be aware of the job post. Media such as television do not provide detail information about job post.

But with Information and communication technology we can be able to automate the process and save all cost as described above.

## 1.3 Project Objectives

#### 1.3.1 General Objective

The Objective of this project is to design and develop job search portal integrated with automated interview system that will allow individual who search for a job with required specification to perform a preset interview by an employer.

#### 1.3.2 Specific Objective

The following are specific objective of the project.

1. To capture and analyze system requirement.
2. To design and implement a module that will allow to post and view jobs.
3. To design and implement interviewing module.
4. To integrate module and perform system testing of all system modules.

## 1.4 Significance of the project

1. Job seekers can access system at any time for reference.
2. Employers will be able to target a specific people in the community. iii.

Carry out interviews with a very geographically dispersed population. iv.

Provide savings in costs to the job seekers and employers. v.

Record data quickly and accurately.

## 1.5 Project Scope

The system to be developed will focus on developing a web based system with modules that will allow posting and viewing jobs, interviewing an individual and alerting an individual via email. The system will use video interview using a computer web come and provide an alternative method of interview if needed by an interviewer.

# CHAPTER TWO: LITERATURE REVIEW

## 2.1 Over View of Related System

### 2.1.1 PSRS web system

Public Service Recruitment Secretariat (PSRS) is a Tanzanian government system that post various job posts on the Internet. The system provides various functionality such as job searching through various filters, registration, and SMS subscription. It provides users with interview tips but the system does not interview individuals online. The applicants that meet a certain job requirement have to go to a specific location so as they can be interviewed.

### 2.1.2 SONRU

SONRU is trusted global leader and pioneer of automated video interviewing for screening candidates, simplifies the recruiting process for recruiters and their candidates. Is system allows company to set their interview in terms of closed ended form and the candidate who answers the question is video interviewed via a web cam.

Every question when appear the web cam starts recoding after a predefined wait time. When then time expires the window automatically closes and it saves the interview session to the data base. The interview then can view each question with respective answer in terms of video.

## 2.2 Overview of proposed system

Job portal with automated interview is an integrated system that allows employers so post jobs and set predefined interviews so as job seekers can search and perform interviews on the web. The overall system will be implemented based on the following tools:

#### a) MYSQL database

MySQL databases are relational. A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

#### b) PHP using Code Igniter Framework

Code Igniter is a toolkit for people who build web applications using PHP. Its goal is to enable you to develop projects much faster than you could if you were writing code from scratch, by providing a rich set of libraries for commonly needed tasks, as well as a simple interface and logical structure to access these libraries. Code Igniter lets you creatively focus on your project by minimizing the amount of code needed for a given task. Code Igniter is based on the Model-View-Controller development pattern. Figure 2.1 show how the MVC works

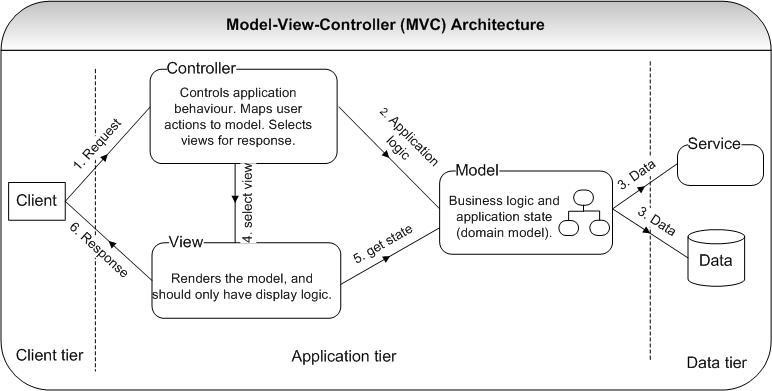


Figure 2.1 MVC Diagram

#### c) Java script using JQuery Framework

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility

# CHAPTER THREE: METHODOLODY

## Literature Review

For Successful completion of this project literature review on several subjects is required as pointed out below for this project.

a) Object Oriented (OO) System Design with UML

###### b) Database Design and Implementation with MYSQL, to store all useful data

1. Backend system development with PHP using Code Igniter Framework
2. Front end system development with Java script using JQuery Framework

## Design Approach

This project will use Waterfall software development model. The Waterfall model provides a framework for planning top – down systems development. The development flows down a number of successive activity stages the stages in the waterfall model overlap and feed information to each other. During design, problems with requirements are identified; during coding, design problems are found and so on. The development process is not a simple linear model but involves a sequence of iterations of the development activities.

With waterfall model the following are the task that are going to be done for successful completion of the project: -

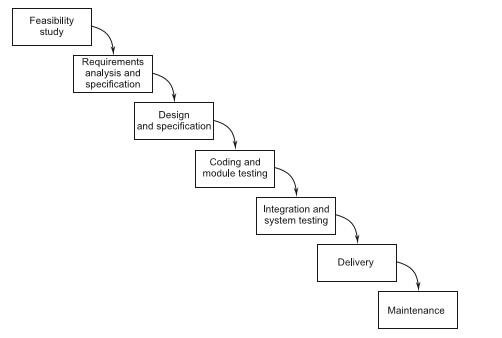


Figure 3.1: Waterfall model

## Data collection

Data collection was done through two ways which include primary data collection and secondary data collection. Primary data was obtained by visiting and interviewing stake holders of the PSRD system. Also primary data was obtained through observation of the whole job recruitment and interview process. Secondary data was obtained through books, internet and newspapers.

# CHAPTER FOUR: REQUIREMENT CAPTURE AND ANALYSIS

## 4.1 Requirements Capture

System requirements are the tasks which the proposed system is expected to perform; they can also be termed as System functions. System requirements can further be categorized into two namely functional requirements and non-functional requirements.

### 4.1.1 Functional Requirements

Functional requirements are tasks or processes which a system must perform for the user or for the system itself so as to attain specific desired system functionality. The function requirement are described in the Table 4.1.

Table 4.1: Functional Requirements for the system

|  |  |  |
| --- | --- | --- |
| **Ref#** | **Function** | **Category** |
| R1.1 | The system should allow Management of Job Seeker and Employer  Profiles | Evident |
| R1.2 | The system should allow admin to Add New Services, Related to  Employers or Job Seekers | Evident |
| R1.3 | The system should allow message interaction between actors | Evident |
| R1.4 | The system should allow search job seeker information | Evident |
| R1.5 | The System should allow posting of jobs | Evident |
| R1.6 | System should allow editing of user profile | Evident |
| R1.7 | The system should allow passwords reset by automated email. | Evident |
| R1.8 | The system should provide information on how much time left on each job posting and how long left on CV access | Evident |
| R1.9 | The system should allow employer to set up interview | Evident |

|  |  |  |
| --- | --- | --- |
| **Ref#** | **Function** | **Category** |
| R1.10 | The system should allow user registration. | Evident |
| R1.11 | The system should allow employers to view submitted interview |  |
| R1.12 | The system should allow uploading and downloading of CV | Evident |
| R1.13 | The system should allow search and application for a job | Evident |
| R1.14 | The system should allow job seekers to view Application History from profile. | Evident |
| R1.15 | The system should automatically calculate the remaining time for job post and interviews | Hidden |
| R1.16 | The system should allow interviewing of job seekers | Evident |
| R1.17 | The system should store its data into a database | Hidden |

### 4.1.2 Non-Functional Requirements

Non-functional requirements are the system constraints from which the operating of the system can be analyzed. The non-functional requirements are show in Table 4.2

Table 4.2: System Attributes

|  |  |  |
| --- | --- | --- |
| **Ref#** | **Attribute** | **Constraint** |
| RN1 | Fault-tolerance | The system should be able to recover from failure |
| RN2 | User friendliness | The system should meet user needs and should be easy to learn and use. |
| RN3 | Reliability | The system should perform the desired functions in a required period of time under stated condition. |
| RN4 | Security | The system should have high security by creating of access level. |
| **Ref#** | **Attribute** | **Constraint** |
| RN5 | Extensibility | The system can be upgraded by addition of functionalities and capabilities |
| RN6 | Maintainability | The system should be such that future maintenance and enhancements times and efforts are reduced. |

## 4.2 Requirement Analysis

Requirement analysis is the detailed study of various operations performed by a system and their relationships within and the outside the system. Modeling of the system was done using UML. Requirement analysis phase consists of identification of use cases, use case description, use case diagrams and a system conceptual diagram.(must reference table)

### 4.2.1 Use Cases Identification

The use cases are identified in Table 4.3

Table 4.3: Identification of use cases

|  |  |  |
| --- | --- | --- |
| **Actor** | **Actor Description** | **Use cases** |
| Job Seeker | A person who is actively looking for employment. Job seeker interacts with system by logging in and search for job post. They also perform an online interview. | Post cv, apply job, search job, perform interview, login, view profile info, update profile, register, send mail |
| Employer | A person or business that employs one or more people, especially for wages or salary.  They interact with the system by posting jobs and interviewing job seekers. | Update profile, register, search job seeker info, post job, login ,view profile, set interview, view interview, download cv, send mail, |
| Administrator | A system administrator (sysadmin) is a  Person who supports a multiuser computing environment and ensures continuous, optimal performance of IT services and support systems. | Manage accounts, add  services, provide roles, login, view profile, send mail, edit job info |

### 4.2.2 Use case Description

Table 4.4 through Table 4.10 Show the use cases description of few selected use cases of the system.

Table 4.4: Register description

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case | Register |
| Actors | Employee, Job Seekers |
| Short description | A use case allows users to register to the system |
| Pre-condition |  |
| Post-condition | A user is ready to access the system online services. |
| Main Flow | 1. The user select Registration service center 2. The user select register an account 3. The system provide the member with the “Online register form”. 4. User fills the form with the appropriate details. 5. User submits the duly filled form. 6. System validates the form and successively updates database fields to allow access to online service. |
| Alternative Flow(s) | Unfilled field/invalid data   1. System validation of the submitted form fails due to unfilled form field(s) or invalid data. 2. System ask the member to re-fill the form (the unfilled fields or fields with invalid data) 3. Go back to Main Flow 5. |
| Exception Flow(s) | If a member cancels the registration process, a member does not get |
|  | registered for online services and use case ends. |

Table 4.5: Update Profile description

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case | Update Profile |
| Actors | Employee, Job Seekers |
| Short description | A use case allows users to change various profile information including user name, password etc. |
| Pre-condition | The user must be registered  The user must be logged in |
| Post-condition | User profile information is updated |
| Main Flow | 1. The system present the user with an editable view of present details (personal info, password, username) 2. A user then edits the editable fields 3. A member submits the updated details to the system. 4. The system validates the details provided 5. The system saves the new details and present the updated details to the member in a read-only view |
| Alternative Flow(s) | Unfilled field/invalid data   1. System validation of the submitted form fails due to unfilled form field(s) or invalid data. 2. System ask the member to re-fill the form (the unfilled fields or fields with invalid data) 3. Go back to Main Flow 3. |
| Exception Flow(s) | If a member cancels the update process, a members details does not get updated and use case ends. |

Table 4.6: View Profile description

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case | View Profile |
| Actors | Employer, Job Seekers |
| Short description | A use case allow registered user to view their personal information |
| Pre-condition | The user must be registered  The user must be logged in. |
| Post-condition | A user is presented with his/her information. |
| Main Flow | 1. System fetch information depending on the logged in user id. 2. System displays profile info to the user. |
| Alternative Flow(s) |  |
| Exception Flow(s) |  |

Table 4.7: Post Job description

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case | Post Job |
| Actors | Employer |
| Short description | A use case allows the employers to post job |
| Pre-condition | The user must be registered  The user must be logged in. |
| Post-condition | A user is presented with his/her information. |
| Main Flow | 1. The system present the employer with a form with all necessary job detail required. 2. An employer then insert all the information as required. 3. An employer submits the form to the system. 4. The system validates the details provided 5. The system saves the new job post the publish it. |
| Alternative  Flow(s) | 1. If the employer selects the interview button 2. The employer sets interview question |
|  | 1. The employer sets the interview time limit in seconds 2. Go back to Main Flow 3 |
| Exception  Flow(s) | If an employer cancels the posting process, a job post does not get published and use case ends. |

Table 4.8: Perform Interview description

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case | Perform Interview |
| Actors | Job Seeker |
| Short description | A use case allow the job seeker to perform inline interview. |
| Pre-condition | The user must be registered  The user must be logged in.  Must have requirement as specified by the Employer.  Must have working video camera |
| Post-condition | The interview is automatically uploaded to the employer account |
| Main Flow | 1. The system presents the user with series of questions while recording 2. The user answers the questions 3. The system analyses the videos recorded |
| Alternative  Flow(s) |  |
| Exception  Flow(s) |  |

Table 4.9: Apply Job description

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case | Apply Job |
| Actors | Job Seeker |
| Short description | A use case allow registered user to apply for job depending on their |
|  | qualification |
| Pre-condition | The user must be registered  The user must be logged in. |
| Post-condition | An application form is submitted to the employer’s database. |
| Main Flow | 1. Employee is presented with different job on screen. 2. The employee will be presented with a text area to type his application latter 3. The employee will also require to upload all necessary documentation. 4. The employee submits the form. |
| Alternative  Flow(s) |  |
| Exception  Flow(s) |  |

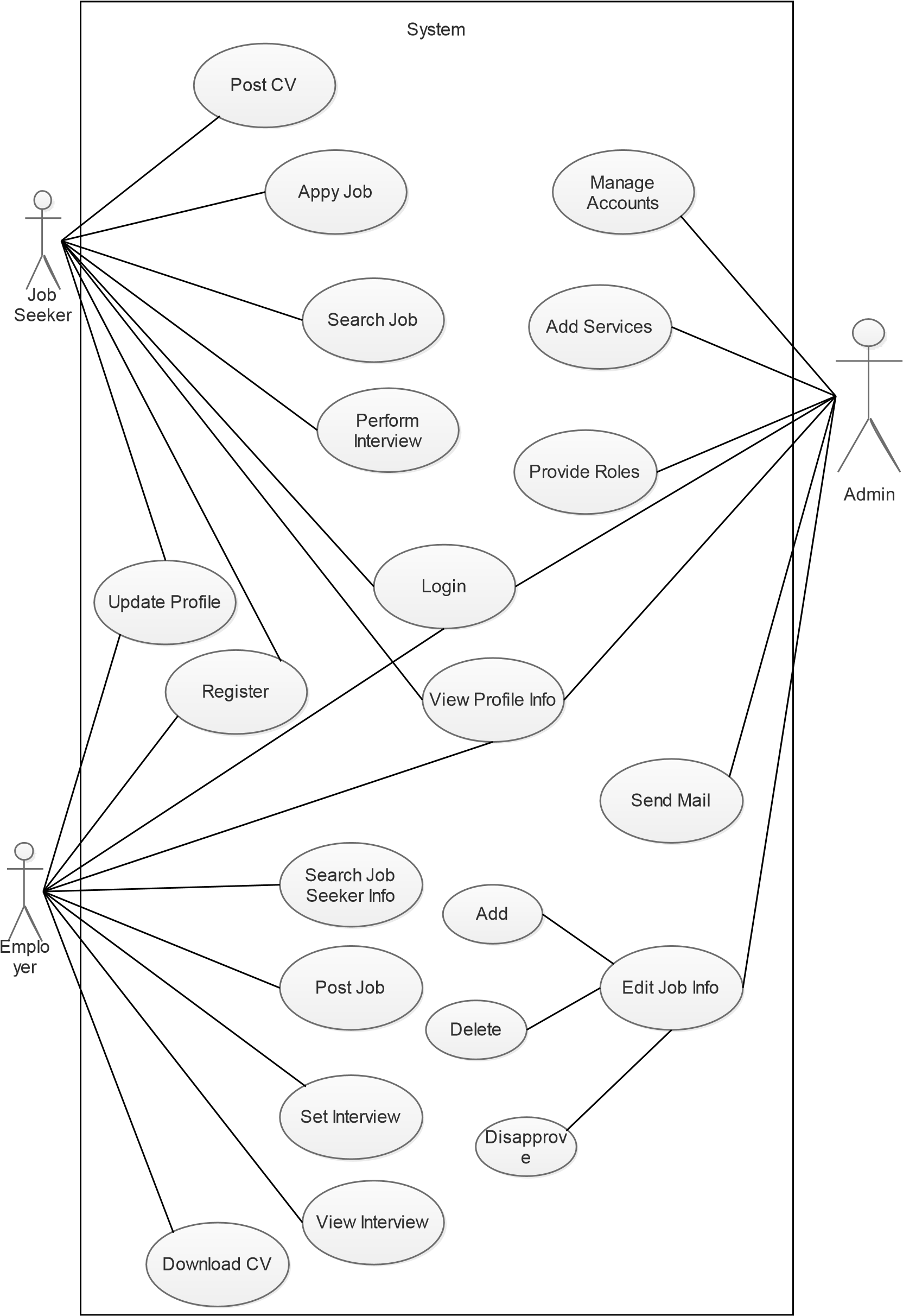
Table 4.10: Set interview description

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case | Set interview |
| Actors | Employer |
| Short description | A use case allow an employer to set interview question. |
| Pre-condition | * The user must be registered * The user must be logged in. |
| Post-condition |  |
| Main Flow | 1. An employer is presented with a form that requires to set question and maximum amount of time used by the employee to answer the question. 2. The employer fills the form. 3. The employer submits the form. |
| Alternative Flow(s) |  |
| Exception Flow(s) |  |

### 4.2.3 Use case Diagram of the system

Figure 4.1 represents the interaction between selected actor and the system use cases

Figure 4.1: Use Case Diagram for the system



### 4.2.4 Conceptual Diagram

Figure 4.2 show various concepts and their relationships.

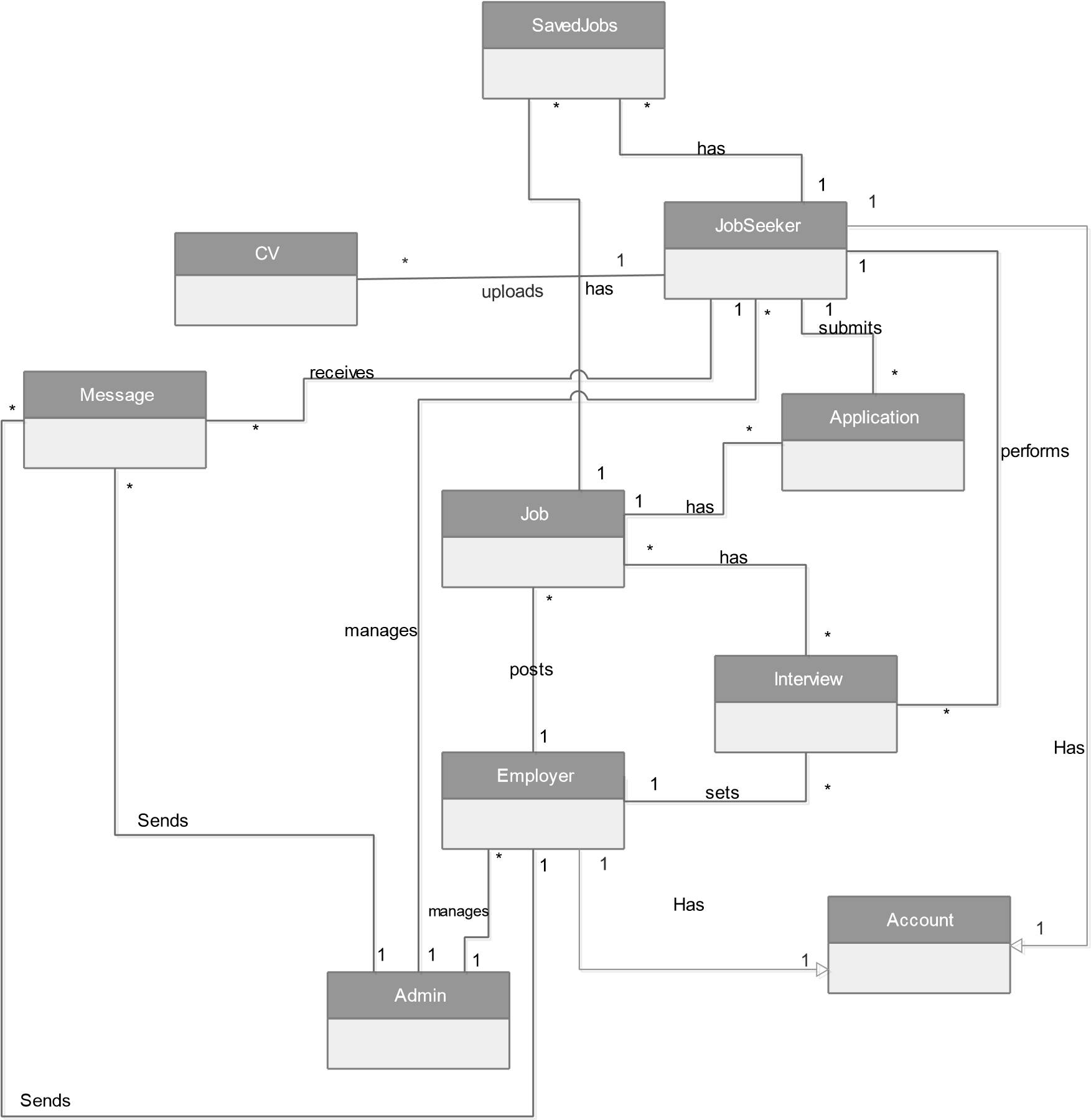


Figure 4.2: Conceptual Diagram

### 4.2.5 System Sequence Diagram

System sequence diagrams depict the interaction between the system and any user involved with the system. They show different events initiated by user and their corresponding system response. Figure 4.3 through 4.5 show the system sequence diagram of most important use cases

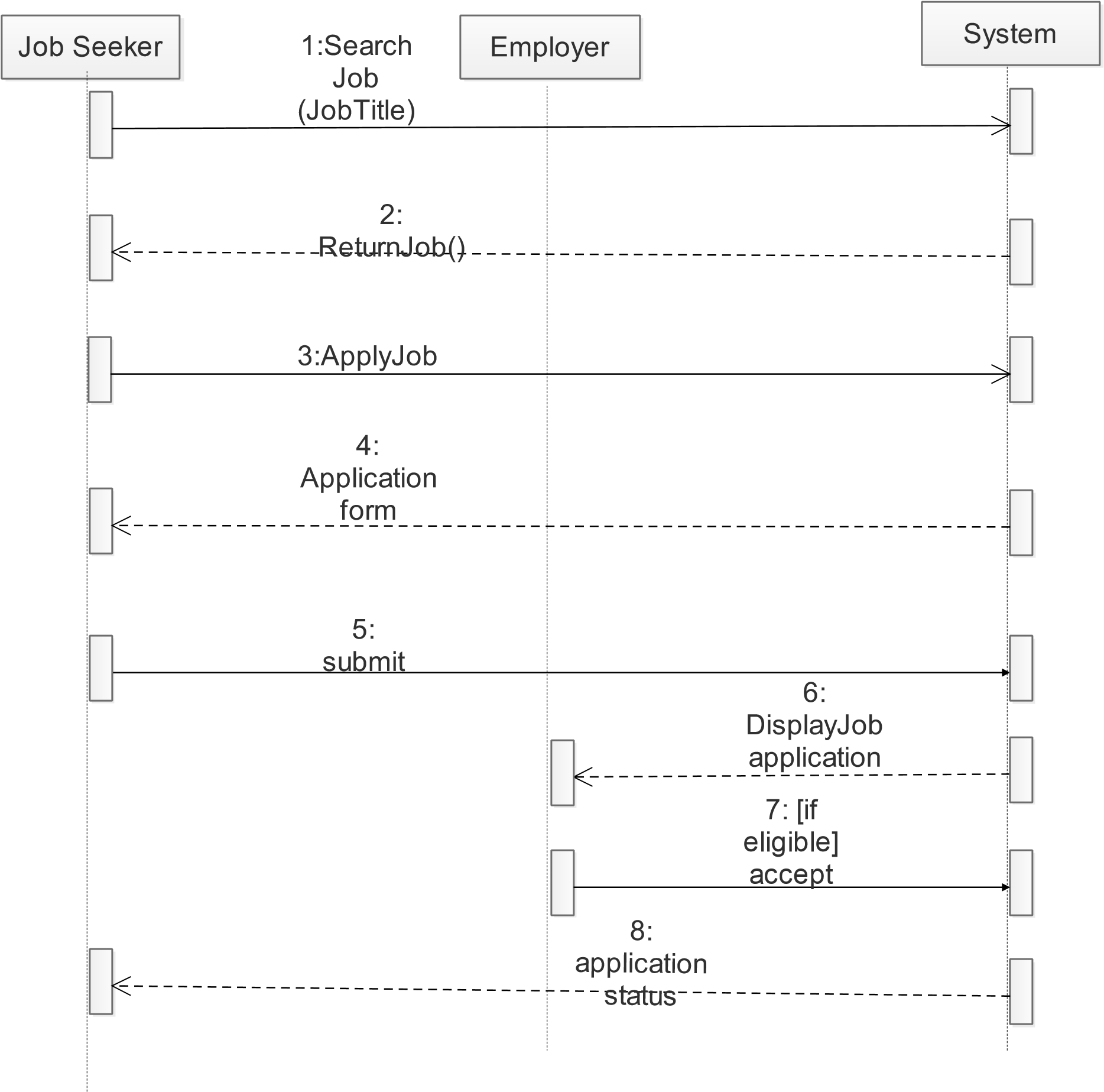


Figure 4.3: Sequence diagram for Job application

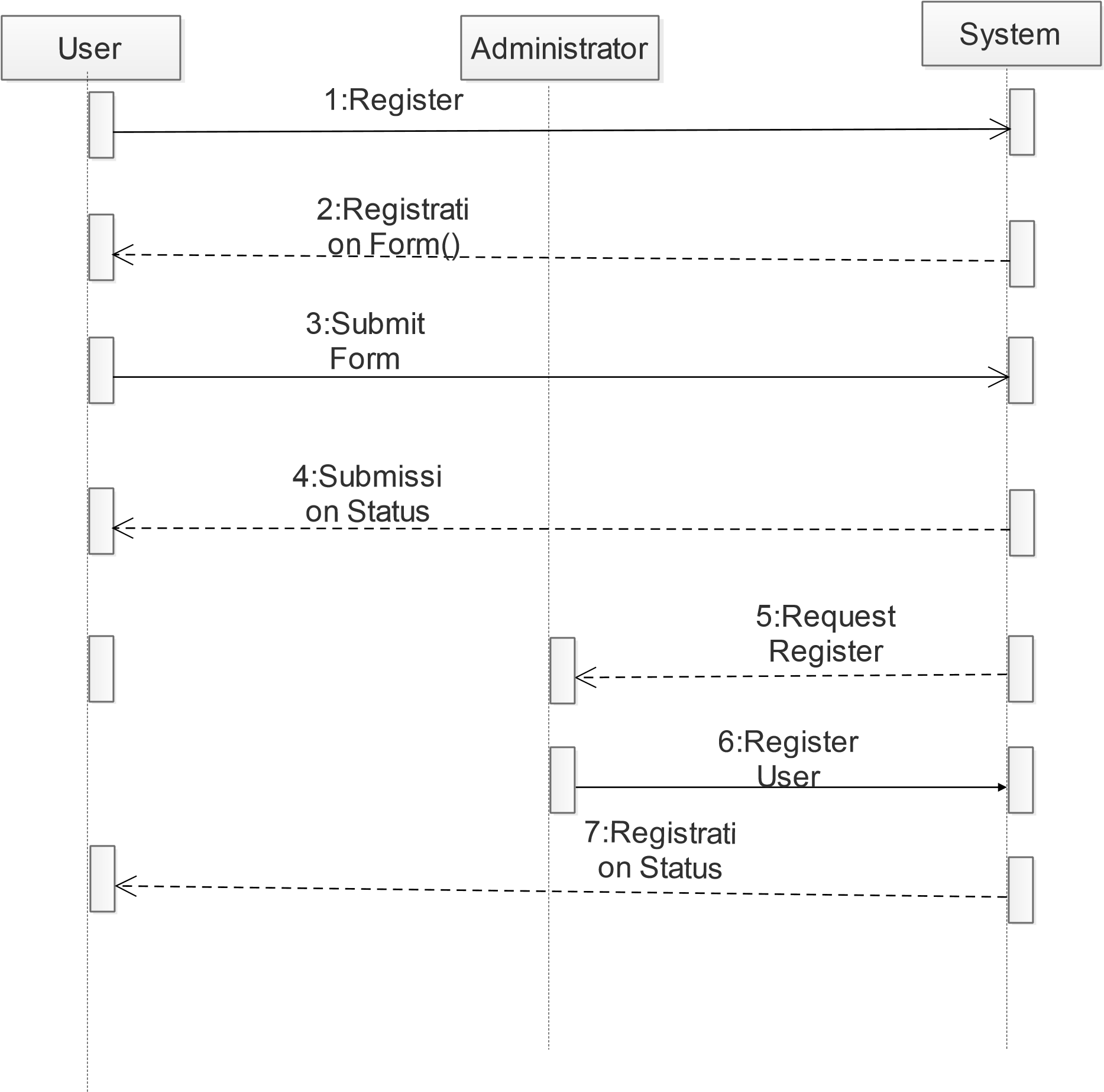


Figure 4.4: Sequence diagram for registration

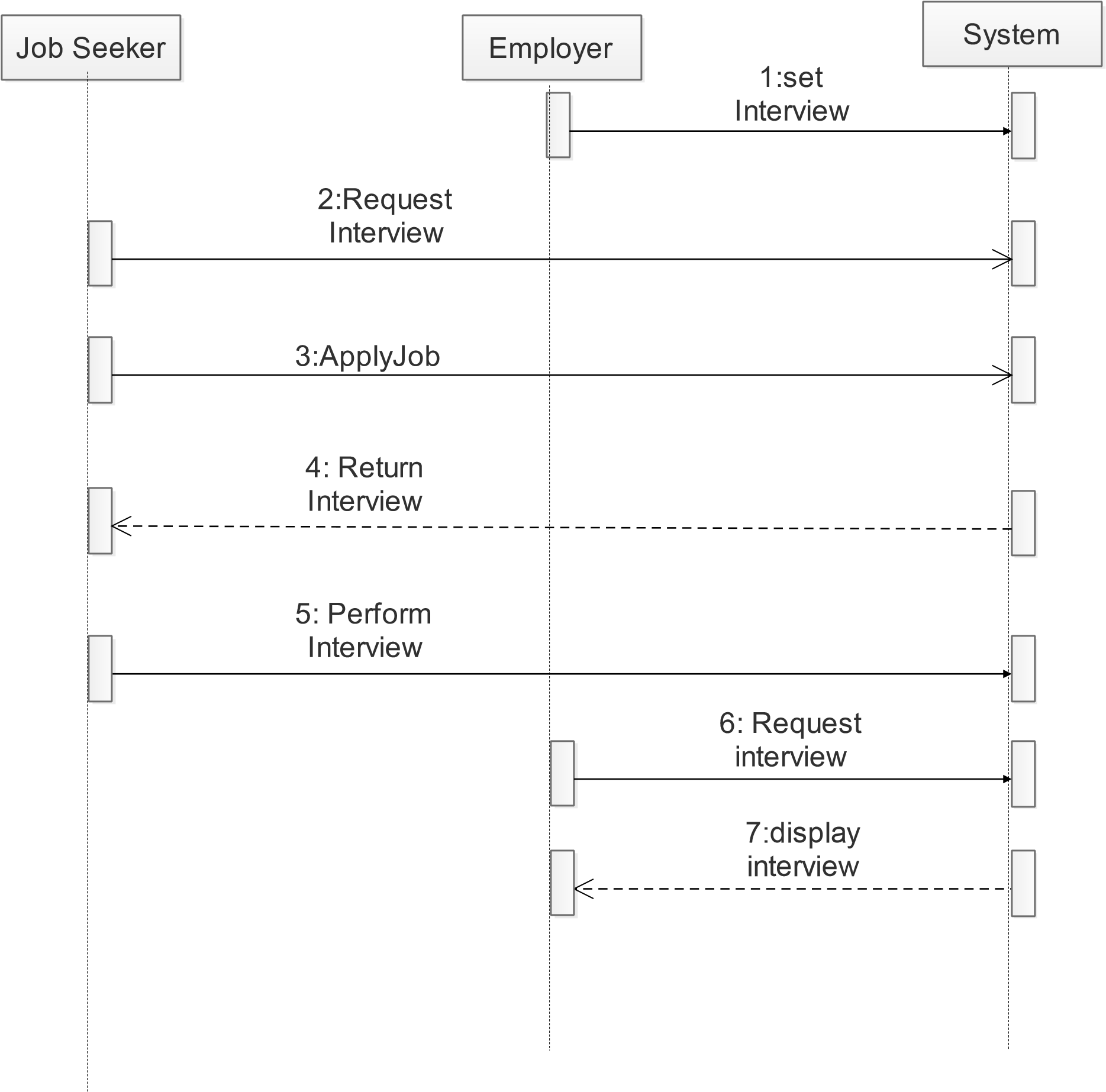


Figure 4.5: Sequence diagram for Job Interview

# CHAPTER FIVE: SYSTEM DESIGN

## 5.1 Design Class Diagram

Design Class Diagram is the description of the classes on an Object Oriented System, their fields, methods and relationship between the classes that interact or inherit from each other. Figure 5.1 shows the Design Class diagram of the system.

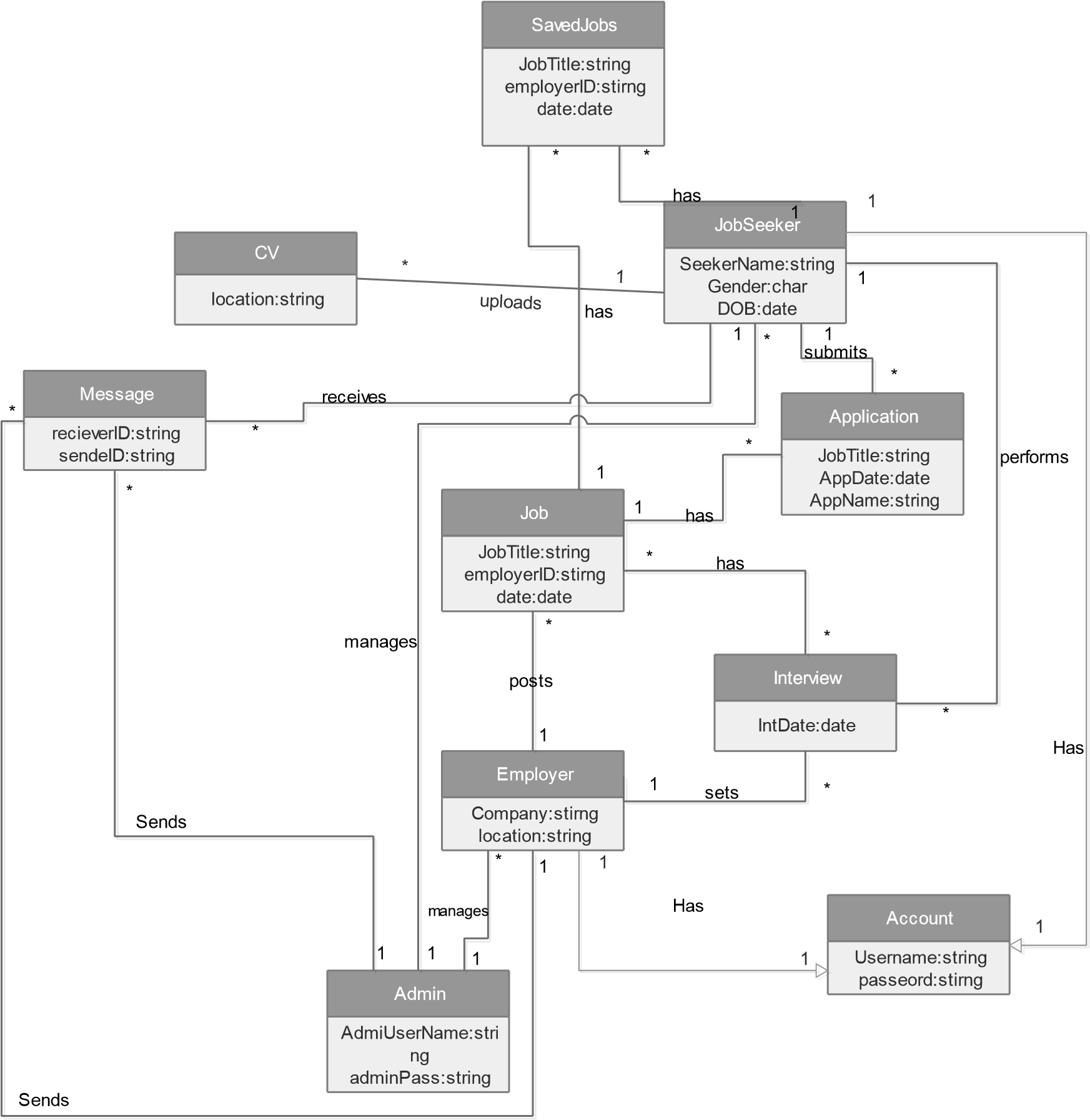


Figure 5.1: Design Class Diagram for System

## 5.2 Database Design

Database design is the description of various parts of the design of the database system. Figure 5.2 shows the data base table relation of the system.

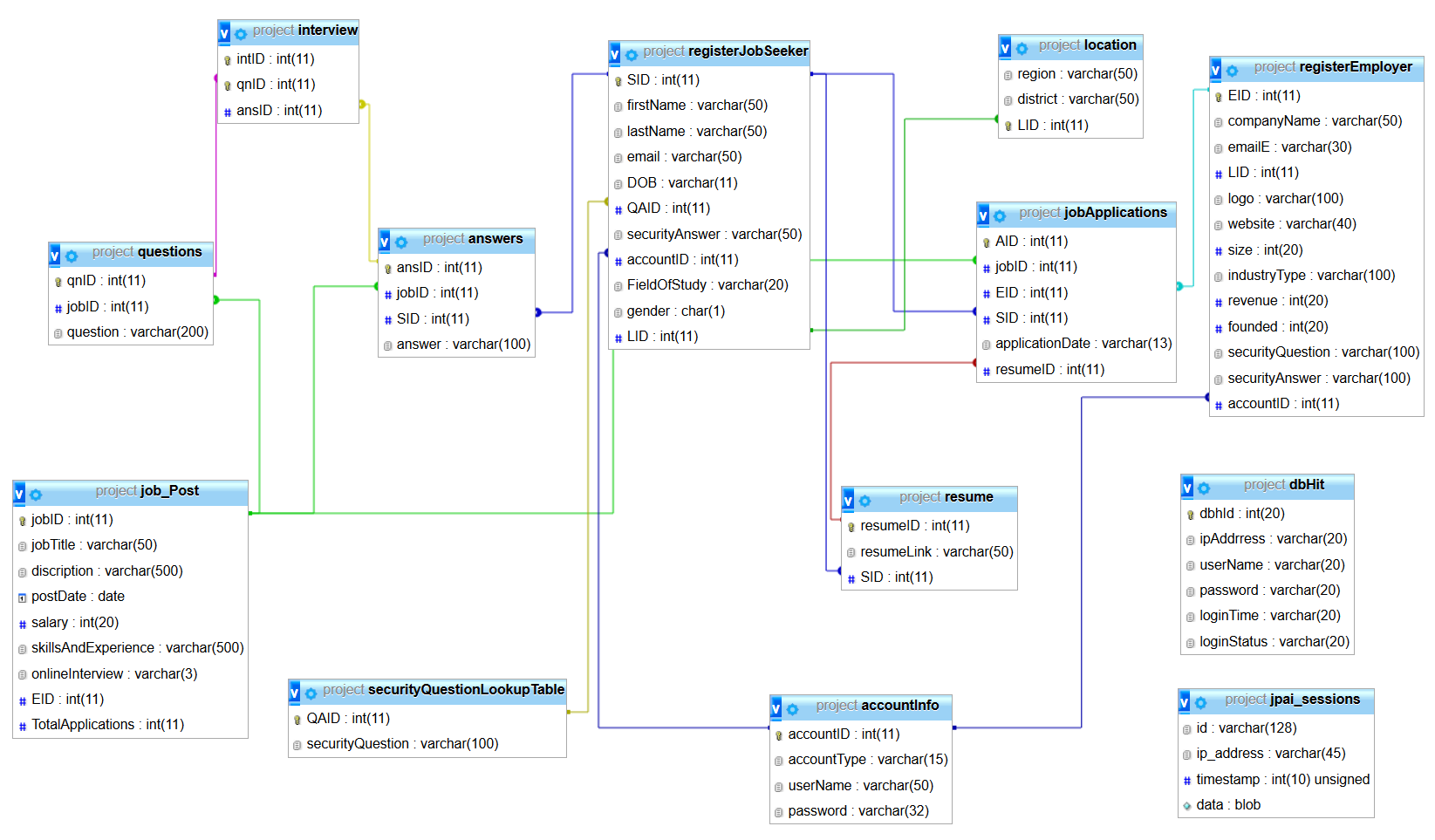


Figure 5.2: Database Design Diagram of System

# CHAPTER SIX: IMPLEMENTATION AND UNIT TESTING

## Overview

Implementation Phase involves writing actual code to implement system requirements according to the Design.

It has two distinct parts: -

1. Writing programs that implement the design.
2. Conducting unit tests.

Generally, these activities include.

1. Coding computer instructions and data definitions,
2. Building databases, populating databases,
3. And other activities needed to implement the design.

## Implementation stages

### Database Configurations

1. Installation of Xampp

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. Xampp provided the necessary tools for hosting the database and act as php serve nessesary to run the system.

1. Installation of codegniter frame work CodeIgniter is installed in four steps:

* Unzip the package.
* Upload the CodeIgniter folders and files to server.
* setting base URL in application/config/config.php as shown in code snippet 6.1



###### Code Snippet 6.1: setting base url

c) Creating the database via sql query

Database tables that were created can be seen at figure 6.1

Figure 6.1 Tables of the system



### System home page

The general appearance of the homepage can be seen on figure 6.2

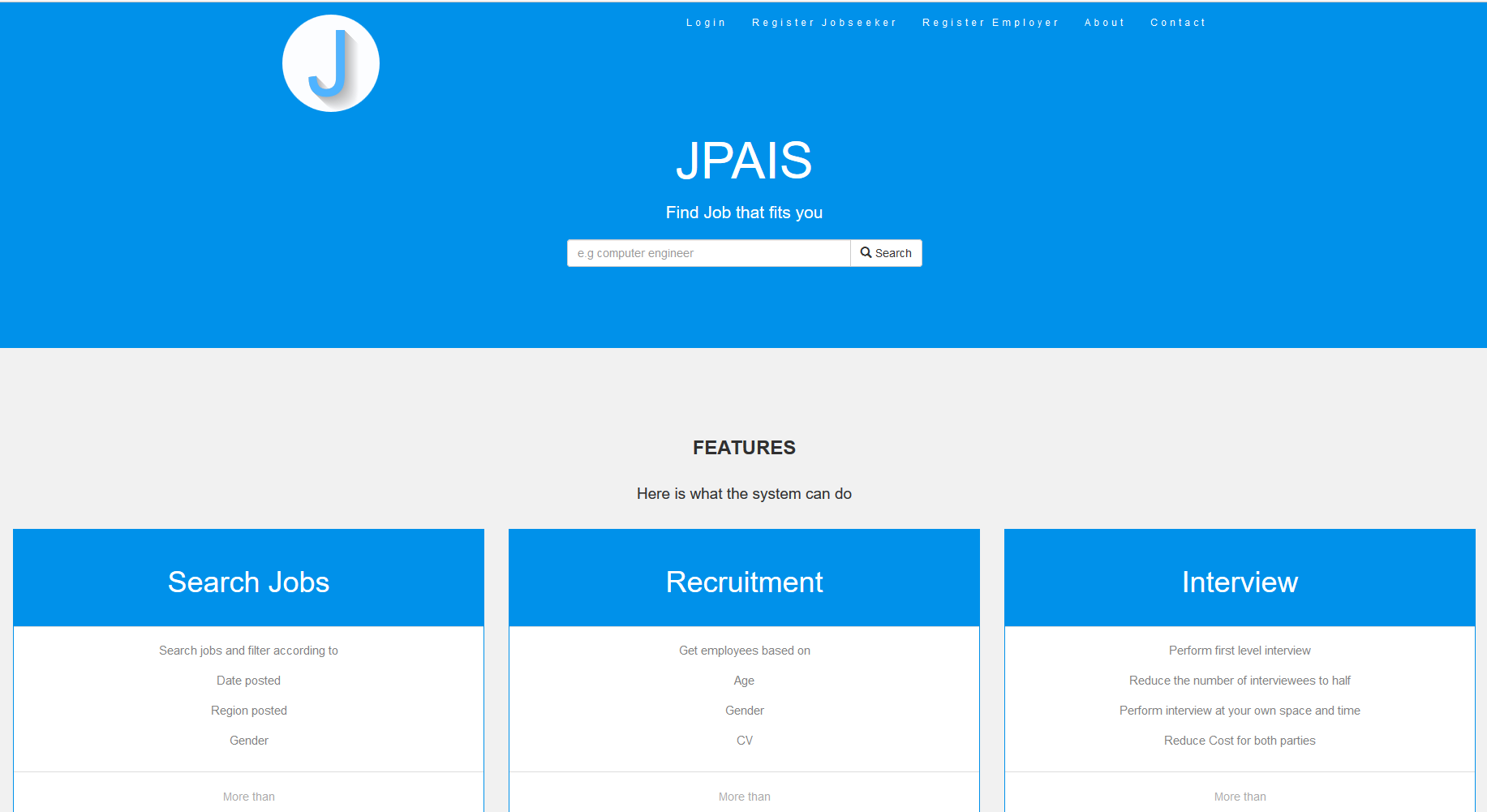
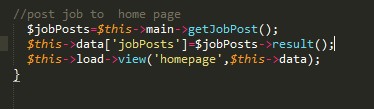


Figure 6.2: Home page

### The Job Post section

The homepage is made from the function index() which is in the home controller. So in order to load the page we need to pass database request so as to get the posted jobs so far. The index function calls a model main.php with function. Refer to the code snippet 4.7.



###### Code snippet 6.2: Post job

### The Registration page

The system registers so far two types of users as from the design: -

1. Job seeker
2. Employer

The regions and districts from the form are fetched from the database. The form can be seen in figure 6.3

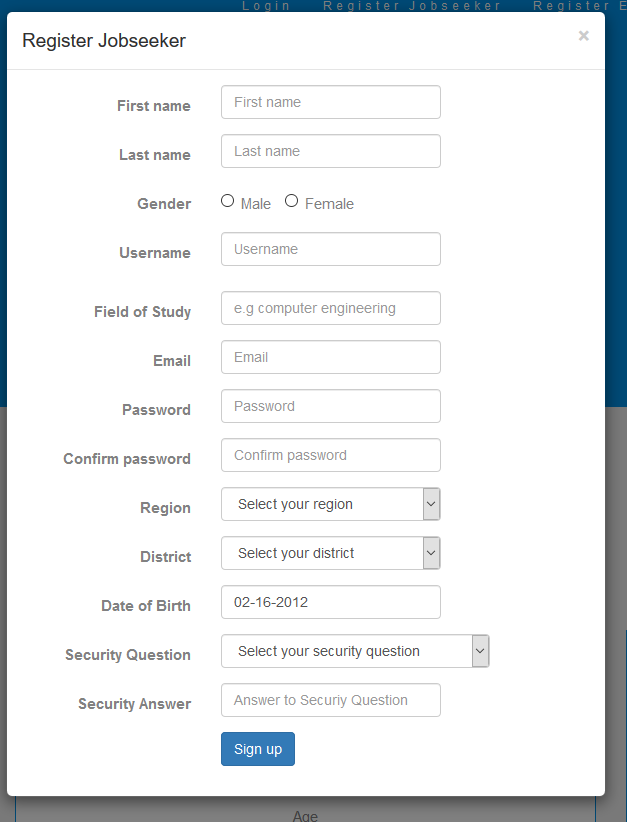


Figure 6.3: User input validation.

Employer registration page

The employer also has a unique registration page as seen in figure 6.4

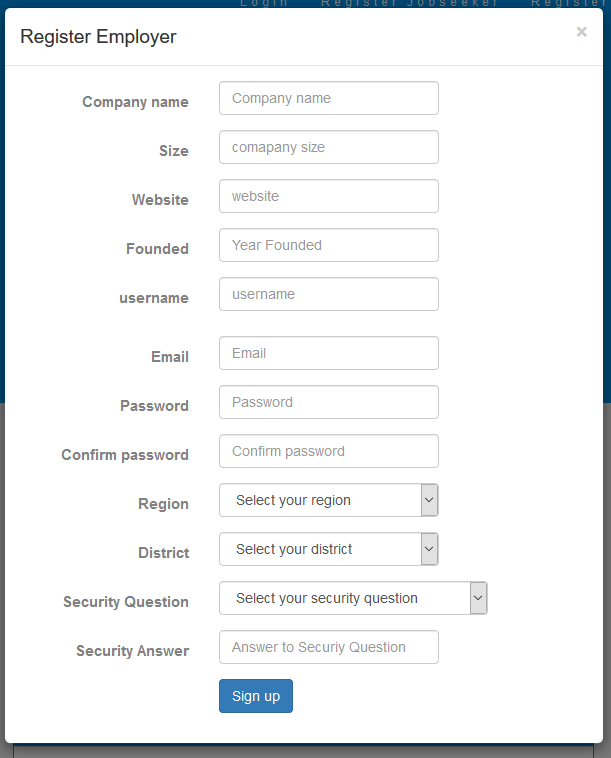


Figure 6.4: Employer input validation.

### The login page

The login form provides the options to sign in both the employer and the job seeker as seen in figure 6.5

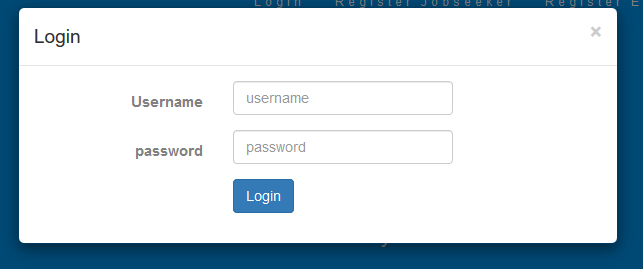
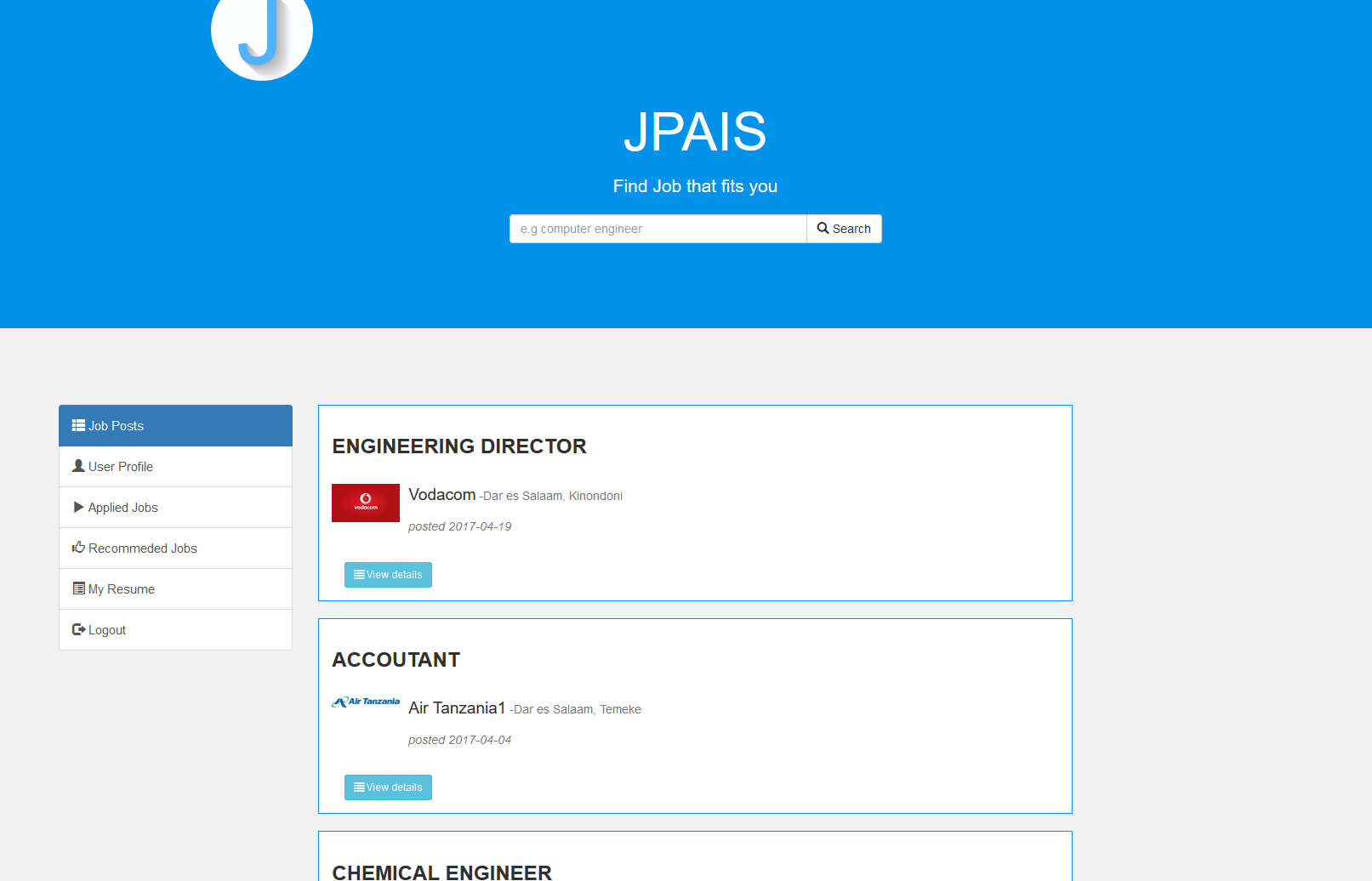


Figure 6.5: Login Form

### The home page for the job seeker

The job seeker home page can be seen in figure 6.6 along with the functionalities in figure 6.7.

Figure 6.6: Job seeker Home page.



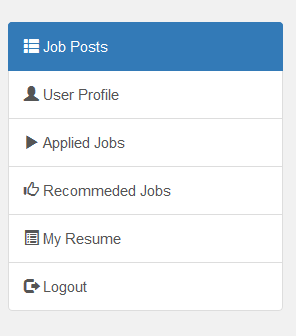


Figure 6.7: Job seeker Home functionalities.

Job seeker home page contains the following options

* Job Posts

This sections provide all the necessary information to the job seeker about the name of the company, date of post, job title, description and option to apply or save the job post for letter reference.

* User profile

Provided user personal information including DOB, name, username etc.

* Applied jobs
* Recommended Jobs

Here the system intelligently observe all posted jobs and Auto recommends depending on the field of study of the Jobseeker.

* My Resume (Upload CV)

This section contains information about the user CV. Here the user can upload the his/her CV which shall be sent automatically to the employer when they apply.  Account setting  Logout option.

### The Employer home page

The job seeker home page can be seen in figure 6.8 along with the functionalities in figure 6.9.

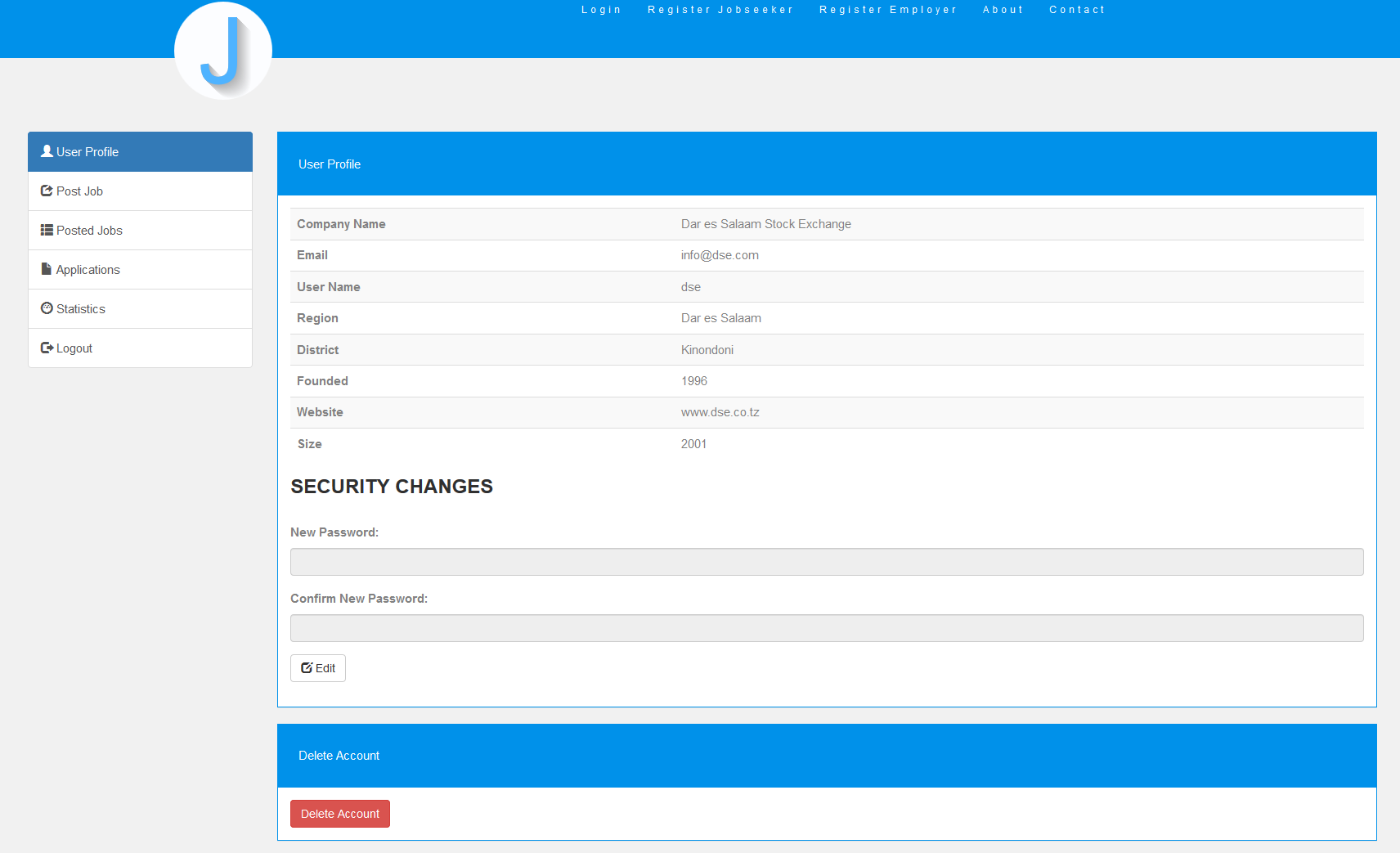


Figure 6.8: Employer Home page

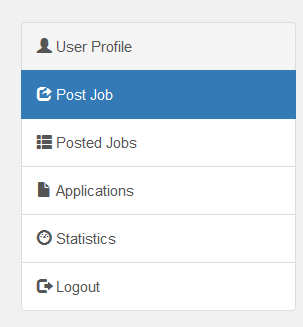


Figure 6.9: Employer Home page functionalities.

The employer home page contains the following options

* 1. User Profile

Provides the necessary information including company name, email, user name, location information and so on.

* 1. Post Job

The post form is seen in figure 6.10

This form provide a unique way of posting jobs because it’s a step form that include job descriptions and setting of interview questions.

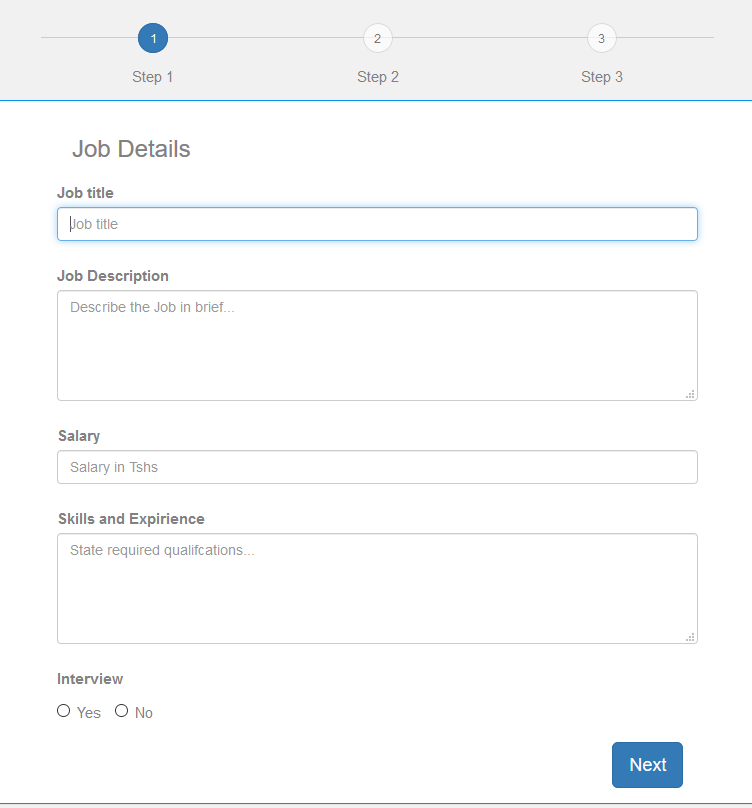


Figure 6.10: Employer Post form.

Here employer can post job by providing the job title, salary as the form require. As of pressing next the employer shall be prompt to 10 interview Question as shown in figure 6.11

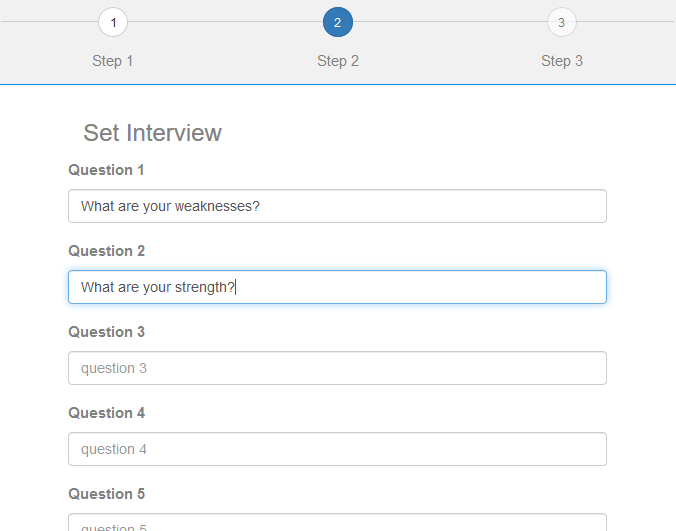


Figure 6.11: Employer Post form.

* 1. Posted Jobs

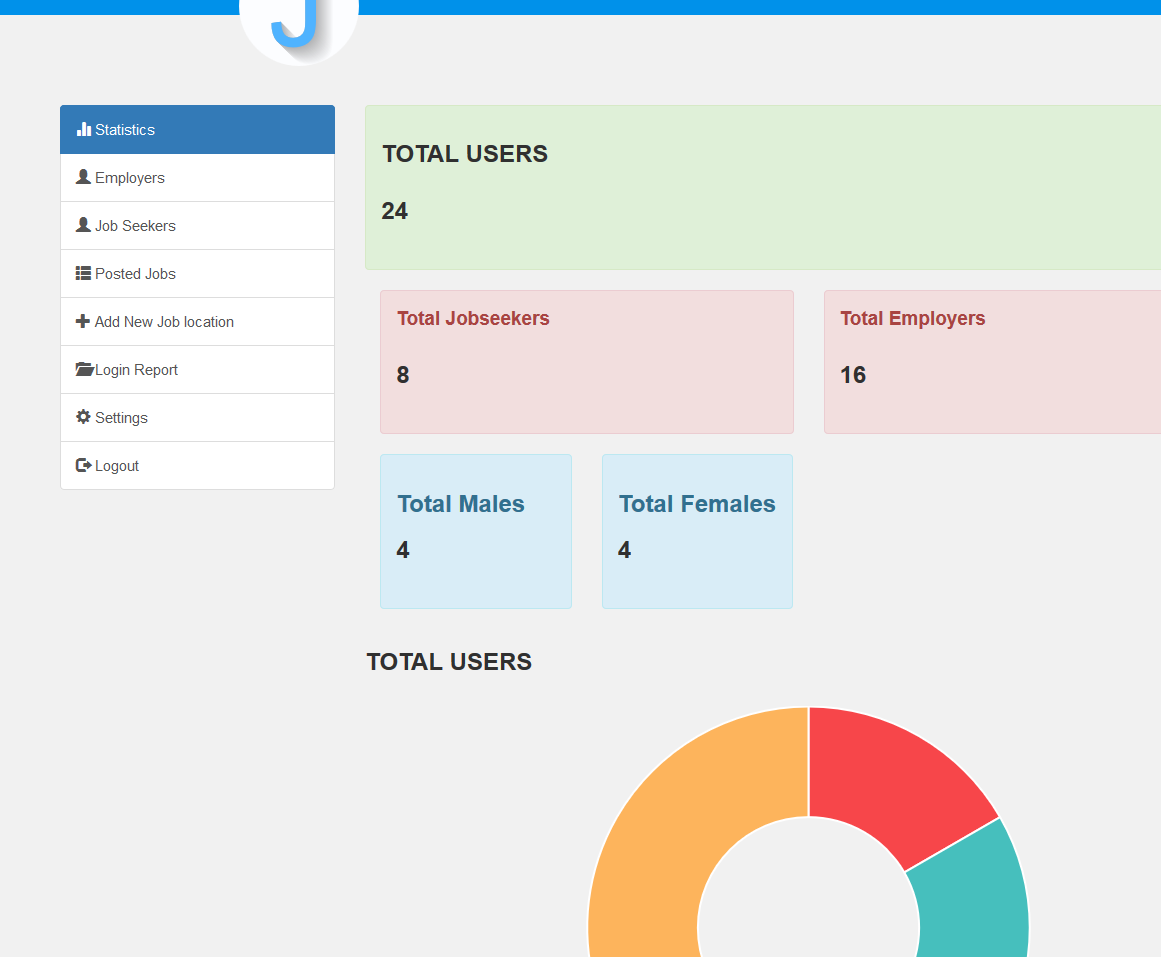
The section provides with the employer information about the posted jobs of which they can decide to delete or edit.

* 1. Applications
  2. Statistics
  3. Account Settings
  4. Logout

### Administrator home page

The system administrator home page be seen in figure 6.12 along with its various options in figure 6.13.

Figure 6.12: Administrator homepage.



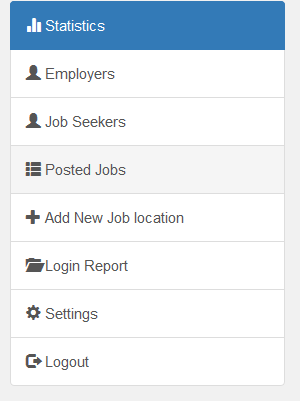


Figure 6.13: Administrator homepage functionalites.

The administrator of the system has the following functionalities

1. Statistics

Here the admin can view the overall users of the system and have a clear pie chat describing the total users

1. Employers and Jobseekers

Here the administrator can add or delete users of the system.

1. Posted jobs

The administrator can view and edit posted jobs whenever necessary

1. Add New Location

Add mini can add new region or district so as they can be selected by job seeker or employers during job seekers during registration.

# CHAPTER EIGHT CONCLUSION AND RECOMMENDATION

## Summary

JPAIS was developed following the waterfall model as the software development model. This system is expected to simplify the whole process finding a job of your field of study and also employers to reach even remote people, it also minimizes the total cost for both jobseekers and employer during the whole recruitment process.

## Challenges Faced

Any Engineering activity involves solving a particular existing problem. The problems in one way or another cannot be solved freely; there are challenges which are encountered when taking particular engineering work. Such challenges are as follows;

1. **Integrating different systems and technologies.** There are thousands of different technologies, systems and applications available for businesses. Integrating third-party or other custom applications, such websites, or an inventory management database, adds to the complexity of the project as the new software solution must conform to the external constraints of these existing systems. Even a simple requirement, such as supporting multiple browsers, significantly increases the complexity of building and testing software solutions.
2. **Change of requirements.** It’s common for projects to continuously build new ideas throughout the design process. It’s almost too easy to get caught up in all the things we want to do, that we forget what we originally set out to create. When the solution is live, it’s almost certain new feature needs or desired capabilities will reveal themselves.

## Recommendation

The following are my recommendations

1. **Use of modular design**

Modules act as building blocks that can be glued together to achieve overall system functionality. They hide the details of their implementation behind a public interface, which provides all the methods that should be used. Users should code and test to the interface rather than the implementation. Thus, concrete implementation details can change without impacting downstream users of the module. Application programming interfaces (APIs) can be shared between different implementation providers.

1. **Documenting everything**

Comprehensive documentation helps other developers who may take over your code, and will also help you in the future. Use code comments for in-line documentation, especially for any technically challenging blocks, and public interface methods. However, there is no need for comments that mirror the exact detail of code line by line.

1. **Performing of extensive test**

Unit tests are software tests which are executed automatically on a regular basis. In test driven development, the tests are written first, serving as a specification and checking every aspect of the intended functionality as it is developed. One must make sure that unit tests exhaustively simulate all possible not only that which seems reasonable inputs to each method.

1. **Keeping it simple**

Every software project starts somewhere. A rule of thumb is tostart as simply as you possibly can*.* Significantly more problems are created by over-engineering than under-engineering. Simplicity starts with design: a clean and elegant data model is a kind of simplicity that leads naturally to efficient algorithms.

## Conclusion

Effective implementing automated interview is a challenge in any country, but may be even more so in Tanzania. Among other reasons, the researching can encourage a rapid turnover of staff, with the result that knowledge about legacy systems is lost.

The recommendations above give a brief introduction to established best practices in JPAIS engineering that may serve as a useful reference. Some of these recommendations may be debated in some contexts, but nevertheless are important to understand and master.

# REFERENCES

Affiliates, E. C. (2013). DuckWeb. Retrieved November 30, 2013, from DuckWeb Information System: https://duckweb.uoregon.edu/pls/prod/twbkwbis.P\_WWWLogin

Bagile, D. (2013, November 22). CS 431, Database Systems. Dar es Salaam, Tanzania.

Kalinga, D. ( 2013, March). CS332 OBJECT-ORIENTED PROGRAM DESIGN AND ANALYSIS. Dar es Salaam, Tanzania: CSE, CoICT, UDSM,.

Kramer, J. (2014). Up and Running with Bootstrap 2. Retrieved April 17, 2014, from lynda.com: http://www.lynda.com/Bootstrap-tutorials/Up-Running-Bootstrap/110885-2.html

Pressman,R., Software Engineering – A Practitioner’s Approach (4th Edition), McGraw Hill, 1997.

Priestley ,M., Practical Object-Oriented Design, McGraw-Hill, 1997.

Runbaugh J., I. Jacobson and G. Booch,The Unified Modelling Language Reference Manual, Addison- Wesley, 1999.

Schach S.R., Classical and Object-Oriented Software Engineering (3rd Edition), IRWIN, 1996.

Sommerville I., Software Engineering (5th Edition), Addison-Wesley, 1995.

PSRS at htttp://www.ajira.go.tz

SONRU at [http://www.sonru.com](http://www.sonru.com/)

# APPENDICES

## APPENDIX A

# Project Time Schedule and Budget

To accomplish the process of developing this system hence making this project a success, Table A3 gives the project estimated costs.

Table A1: Budget Proposal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NO** | **MATERIALS** | **UNITS** | **UNIT COST** | **COST (T/SH).** |
| 1 | Internet Surfing | 42 days | 1,500 | 63,000/= |
| 2 | Report Printing | 200 pages | 100 | 20,000/= |
| 3 | Report Binding | 4 reports | 3500 | 14,000/= |
| 4 | Transport expenses | 1 person | 48,000 | 48,000/= |
| 5 | Communication | 1 person | 30,000 | 30,000/= |
| 6 | Overheads |  |  | 43,000/= |
|  | **TOTAL** |  |  | **218,000/=** |

Table A1 shows the time schedule for semester one

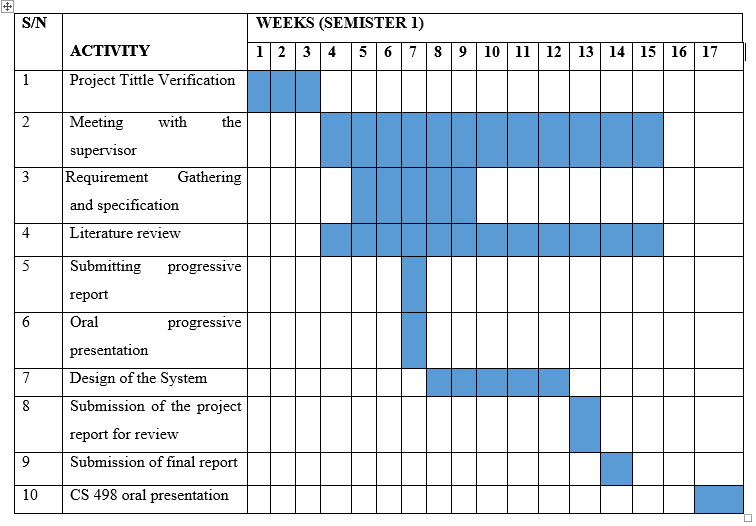
Table A2: Time Schedule for Semester One 

Table A2 shows the time schedule for semester two

Table A3: Time Schedule for Semester Two

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S/N** | **ACTIVITY** | **WEEKS (SEMISTER 2)** | | | | | | | | | | | | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** |
| 1 | Setting out development environment |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Meeting with  the supervisor |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Database implementation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Developing the System front  end |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Making the  users accounts |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Implementation of Job post module |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Progress report preparation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Submission  of the progress project report |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Presentation of the Progress  report |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Implementation of the  interviewing module |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | Making the  administrator account |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# APPENDIX B

## Use Case Description Template

Table B1 shows the template for use case description and Table B2, B3 shows the sample use case description of the system.

Table B1: Use Case Description Template

|  |  |
| --- | --- |
| **Use case:** | Use Case name |
| **Actors:** | Role names of people or external entities the use case |
| **Short Description:** | A brief description of the Use Case |
| **Cross Reference:** | Related use cases and system functions. |
| **Pre-Conditions:** | A description of the conditions that must be satisfied before the use case is invoked |
| **Post-Conditions:** | A description of what has happened at the end of the use case. |
| **Main Flow:** | A list of the system interactions that take place under the most common scenario. |
| **Alternate Flow(s):** | A description of possible alternative interactions |
| **Exception Flow(s):** | A description of possible scenarios where unexpected events have taken place |