

JobH

Job Search Portal

Web Based Application



Thesis submitted to the Institute of Information Technology,
Quaid-i-Azam University, Islamabad, for the partial fulfillment

Master of Science
In
Information Technology

By
Muhammad Shahid

Supervised By
Mam Robina Rashid

INSTITUTE OF INFORMATION TECHNOLOGY
QUAID-I-AZAM UNIVERSITY ISLAMABAD

Session 2019-2021

Statement of Submission

This is certifying that “**Muhammad Shahid**” Registration No. “01161911027” has successfully completed the final project as “**web base Job Search Portal**” Quaid-I-Azam University, Islamabad to fulfill the partial requirement of the degree “Master of Science in Information Technology”.

External Examiner

Mam. Robina Rashid

ACKNOWLEDGEMENT

Firstly, I would like to express my gratefulness to Allah Almighty, whose blessings I have seen during my time at Quaid-I-Azam University in the form of achievements, success, and learning.

Also, I am obligated to several people for their support and generous contribution. I am thankful to my parents for their prayers, whose love and affection has been inspirational throughout my life.

I especially express my sincere gratitude to my advisor Mam Robina Rashid for guiding me throughout the process. I could not have imagined having a better advisor and mentor for my project.

I am also thankful to my friends (Shahid Iqbal, Waqar, Kashif, Farooq) for their continuous support during this journey.

- **Muhammad Shahid**

ABSTRACT

Finding jobs that best suits the interests and skill set is quite a challenging task for job seekers. The difficulties arise from not having proper knowledge of the organization's objective, their work culture and current job openings.

Job Portal is the solution where job seekers meet aiming at fulfilling their requirements. This is the cheapest as well as the fastest source of communication reaching a wide range of audiences on just a single click irrespective of their geographical distance.

The web application “JobH” (Job Search Portal) provides an easy and convenient search application for the job seekers to find their desired jobs. Job seekers from any background can search for the current job openings. Job seekers can register with the application and can get notification about coming jobs. They can search for available jobs and apply to their desired positions.

Functionalities are developed as a web application. The admin can add new jobs in the database. Also, he should update Jobs in the database. The user should also contact the admin through the “contact us” page and inform him about his reviews about the website. He should also give us advantageous advice through this “contact us” page. This system is implemented in PHP language.

Table of Contents

Chapter 1: Software Project Management Plan	1
1.1 Introduction	1
1.2 Problem Definition	1
1.3 Existing Systems	1
1.5 Proposed Solution	2
1.6 Motivation	2
1.7 Scope	3
1.8 Objectives	4
1.9 Project Deliverables	4
1.10 Project Organization	4
1.10.1 Software Process Model	4
1.10.2 Roles and Responsibilities	4
1.10.3 Tools and Techniques	4
1.11 Summary	4
Chapter 2: Requirements Gathering and Analysis	5
2.1 Product Overview	6
2.2 Definitions, Acronyms, and Abbreviations	6
2.3 Overview	6
2.4 Product Perspective	6
2.5 Application interface	6
2.6 User interface	6
2.7 Software Interface	7
2.8 Hardware Interface	7
2.9 User Characteristics	7
2.10 Assumptions and Dependencies	7
2.11 Software System Attributes	8
2.11.1 Reliability	8
2.11.2 Availability	8
2.11.3 Maintainability	8
2.11.4 Security	8
2.11.5 Portability	8

2.11.6 Performance	8
2.12 Use Case Diagram	9
2.13 Use Case Description	10
2.13.1 Use Case 1: Register	10
2.13.2 Use Case 2: Login	11
2.13.3 Use Case 3: View Job Seekers Details	12
2.13.4 Use Case 4: Suspend Job Seeker	13
2.13.5 Use Case 5: Search Jobs	14
2.13.6 Use Case 6: Filter By Location	15
2.13.7 Use Case 7: Filter By Category Stack	16
2.13.8 Use Case 8: Filter By Position	17
2.13.9 Use Case 9: Filter By Company	18
2.13.10 Use Case 10: Apply Jobs	19
2.13.11 Use Case 11: View Applied Jobs	20
2.13.12 Use Case 12: View Profile	21
2.13.13 Use Case 13: Post Job Vacancy	22
2.13.14 Use Case 14: View Posted Jobs	23
2.13.15 Use Case 15: Pause Job Post	24
2.13.16 Use Case 16: Resume Job Post	25
2.13.17 Use Case 17: Deleted Job Post	26
2.13.18 Use Case 18: Logout	27
2.14 System Sequence Diagrams	28
2.14.1 SSD for Register	28
2.14.2 SSD for Login	29
2.14.3 SSD for View Job Seeker Detail	30
2.14.4 SSD for Suspend Job Seeker	31
2.14.5 SSD for Search Job	32
2.14.6 SSD for Filter By Location	32
2.14.7 SSD for Filter By Position	33
2.14.8 SSD for Filter By Category Stack	33
2.14.9 SSD for Filter By Organization	34
2.14.10 SSD for Apply Job	34
2.14.11 SSD for View Applied Jobs	35
2.14.12 SSD for View Profile	35
2.14.13 SSD for Post Job	36
2.14.14 SSD for View Posted Jobs	36

2.14.15 SSD for Pause Job Post	37
2.14.16 SSD for Resume Job Post	37
2.14.17 SSD for Delete Job Post	38
2.14.28 SSD for Logout	39
Chapter 3: Software Design Description	40
3.1 Introduction	44
3.2 Interface Design	41
3.2.1 Simple and Appealing	41
3.2.2 Responsive	41
3.3 Interfaces	42
3.4.1 Login	42
3.4.2 Register	43
3.4.2 Applied Jobs	44
3.4.3 Manage Jobs	45
3.4.4 Post Jobs	46
3.4.3 Job Details	47
3.5 Sequence Diagram	48
3.5.1 Login	48
3.5.2 Suspend Job Seekers	48
3.5.3 Apply Jobs	49
3.5.4 Post Jobs	49
3.5.5 Register	50
3.6 Class Diagram	51
3.7 Summary	52
Chapter 4: Implementation	53
4.1 System Implementation	54
4.2 Detail of Implementation and Tools	54
4.2.1 Bootstrap	54
4.2.2 HTML5	54
4.2.3 CSS.....	55
4.2.4 JavaScript	55
4.2.5 PHP	55
4.2.6 MY SQL.....	56
4.2.7 Deployment Diagram.....	57
Chapter 5: Testing	58
5.1 Introduction	59
5.2 Testing	59

5.2.1 Software Testing	59
5.2.1.1 Module Testing.....	59
5.2.1.2 Unit Testing.....	60
5.2.1.3 Subsystem Testing.....	60
5.2.1.4 System Testing.....	60
5.2.1.5 Acceptance Testing.....	60
5.3 Evaluation	60
5.3.1 Purpose	61
5.3.2 Black Box Testing	61
5.4 Test Cases	61
5.5 Problems During Coding.....	69
5.6 Enhancement	70

Chapter 1

Software Project Management Plan

1.1 Introduction

This chapter first introduces the JobH. It also elaborates project organization and project planning. Finally, this chapter explains the scope and objectives of this project.

1.2 Problem Definition

No one would disagree with the fact that how much people have gone busier these days, no matter where but such is the life. They do not have enough time to get themselves notified of each job opening in the world or more specifically their city/region. But they wish to have some mean where they can find the job from good companies at one place or get jobs related to their city/province, certain technology stack or certain position.

1.3 Existing Systems

Indeed.com(<https://www.indeed.com>): With millions of job listings from thousands of websites, including company career sites, job boards, newspaper classifieds, associations, and other online sources of job postings. Users can also upload a resume and get a personalized resume link to share with employers. You can also apply for jobs, research salaries and job trends, set up job alerts, and use Indeed's job search app for mobile job hunting [1].

Rozee.pk(<https://www.rozee.pk/>): Jobseekers first answer a series of questions and select whether they are students, fresh graduates, or professionals. Upon uploading your cover letter and answering several in-depth questions about your job experience, education, and industry, you will be granted access to the site. (These answers are required and cannot be skipped.) Upon signing up, you will receive an activation code via email and be granted access to the site. You will be invited to update your profile, search for jobs, create job alerts, or connect your Facebook and LinkedIn accounts to Rozee [2].

Dice.com(<https://www.dice.com/>): Dice is the leading site for tech job seekers. You can search by company, job title, keyword, employment type, and location. Registered users can upload a resume, get salary information, store resumes and cover letters, and track jobs. You will also find career advice and tech news for job seekers [3].

GlassDoor.com(<https://www.glassdoor.com/>): Glassdoor is a career community that helps people find jobs and companies recruit top talent. Glassdoor members can see the latest job listings and get access to user-generated content, including company-specific salary reports, ratings and reviews, interview questions, and more[4].

1.4 Proposed Solution

The purpose of developing an Online Job Search Portal comes from my idea to make the job search efficient and handy. It helps the tech job seekers to search for current vacancies at a single point. Major difference in JobH relative to other existing systems are:

- JobH should be able to show jobs based on recent search history.
- JobH will archive all outdated jobs.
- JobH will display analytics based on all recent job posting.

This web application allows job seekers can view and search according to their interests. The front end is developed using HTML, CSS, JavaScript, and bootstrap.

1.5 Motivation

Nowadays in a fast and busy world, the job seeker does not have enough time to go through each company's website and scroll it all the way to find a job of their choice. The choice could be to get a job of a specific category. They would like to see a job related to their skills. Some people would want to get a job only in their city.

With the evolution of technology and the internet being the main source of information. For the applicants, these job portals and have become an excellent method to reach a wide range of audiences. Initially, when I am unaware of these portals, I used to do research about companies through their respective websites and apply if the job responsibilities match my interests. This requires lots of effort and time. However, later when I realized the importance of job search portals, I can access jobs in companies that I might not have heard.

1.6 Scope

The scope of the project is to extract and display new job openings that link to the source. JobH is also an effective means jobseeker to find out the information regarding the current openings in the organization.

Major Inputs

Major inputs are given below:

- Keywords to search for related job openings.
- Selection of company/location/category or position.
- Job posting details.

Major Outputs

Major outputs of the system are:

- General job openings from all job sources.
- Job openings based on the location selected.
- Job openings based on keywords typed.
- Job openings based on date.
- Job openings of the company's name selected.

Major functions

The major functions of the JOBH are:

- Scraping the whole job posting from different sources and display those with clickable links back to those sources.
- Displaying jobs based on typed keywords and location.
- Filtering jobs by category, position, particular date or based on company name.
- Displaying jobs based on recent search history.
- System should be able to send notifications on the job opening of a particular company.

Constraints

Users must have a working internet connection, some basic knowledge of computer and some means i-e computer/mobile to access the internet.

1.7 Objectives

The objective of the web application is to provide flexibility to the job seekers by providing the functionalities of both job search and job application in a single application.

1.8 Project Deliverables

Project deliverables are:

- Software Project Management Plan.
- Software Requirements Specification.
- Software Design Description.
- Software Test Document.
- Implementation.

1.9 Project Organization

1.9.1 Software Process Model

Agile Process Model is being used for the development of JobH. Agility is flexibility, it is a state of dynamic, adapted to the specific circumstances. The reason it is chosen is that it embraces changes, can expect the system requirements to change and so design the system to accommodate these changes. An agile approach combines the incremental and iterative approach by building a small portion of each feature, one by one, and then both gradually adding features and increasing their completeness.

1.9.2 Roles and Responsibilities

As this project is individually assigned to me, I am the only one responsible for all the roles to be performed in the accomplishment of it.

1.9.3 Tools and Techniques

The tools that are used for the implementation of this system are Draw.io for UML diagrams such as use case diagrams. Microsoft Word is used for documentation write-up. This system is implemented in PHP. MySQL is used as data base. XAMP Local Server and Sublime Text 3 Editor are also used in development of project.

1.10 Summary

This chapter has briefly introduced the system including problem description, proposed solution, scope, objective, and described the organization of the project and project management plan.

Chapter 2

Requirements Gathering and Analysis

2.1 Product Overview

This web-based application is going to be implemented in PHP programming language. This product performs crawling of jobs of different category from different local and international job sites

2.2 Definitions, Acronyms, and Abbreviations

Table 2.1 Abbreviation and Definition

Terms	Description
JOBH	Job's scraper/hunter.
UML	Unified Modeling Language
SSD	System Sequence Diagram
SD	Sequence Diagram
UC	Use Case
Users	Job Seekers/Admin

2.3 Overview

The rest of the chapter focuses on functional, non-functional and performance requirements. The overall functionality of the system, use cases, their description, and domain model have also been elucidated.

2.4 Product Perspective

Product is independent and self-contained.

2.5 Application interface

JobH is web application.

2.6 User interface

- Register: Job Seekers can create new account by providing valid identity information (email, number, and password) as a registered user.
- Login: Admin/Job Seekers will login to the application as a registered user.

- View Users Details: Admin can see the details of Job Seekers.
- Suspend Users: Admin can suspend the account of Job Seekers
- Search Jobs: Job Seekers can search jobs on a specific keyword.
- Filter Jobs: Job Seekers can filter jobs based on location, different category etc.
- Apply on Jobs: Job Seeker can apply on multiple jobs.
- View Profile: Job Seeker can see their profiles.
- Post Job: Admin can be able to post jobs.
- View Job Posts: Admin can see the job postings where they can pause, resume, and delete jobs.
- Log Out: Admin, Job Seekers can safely log out of the application.

2.7 Software Interface

Software uses for JobH is:

- Operating application: Windows/ iOS/ Linux.
- Database: MySQL.

2.8 Hardware Interface

As this software is web application so we will need internet supporting computer application (desktop/ smart phone/ tablet).

- Keyboard and Mouse (for desktop).
- Screen.

2.9 User Characteristics

It is expected that the users have computers or mobiles with an internet connection, and they also know how to use them to access websites.

2.10 Assumptions and Dependencies

It is assumed that the users have computers or mobiles with an internet connection, and they also know how to use them to access websites.

2.11 Software System Attributes

2.11.1 Reliability

JobH should be reliable that is it should have almost no zero occurrences of failure. It should be able to work properly all the time. The system should give a proper response to every user action.

2.11.2 Availability

JobH should be available to users all the time. It should crawl and display jobs whenever it is updated on the source job site.

2.11.3 Maintainability

Usually, maintainability involves continuous improvement in the system or improve the reliability of the system based on maintenance experience. The application should be easy to extend. The code should be written in a way that favors the implementation of new functions.

2.11.4 Security

Since this system will be hosted on the internet, anyone should be able to access its services through an appropriate digital device with an internet connection. The system has its login credentials to use it. Users must enter their login credentials before using the system. The system maintains the login sessions of the user.

2.11.5 Portability

It is a web-based application and can run on any digital system having internet access and a browser like computers, laptops, and mobiles.

2.11.6 Performance

The system must have strong computing capabilities for data scrapping. The login information shall be verified within a second. System should be able to deal with numerous users at a time. Queries shall return results within milliseconds.

2.12 Use Case Diagram

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how).

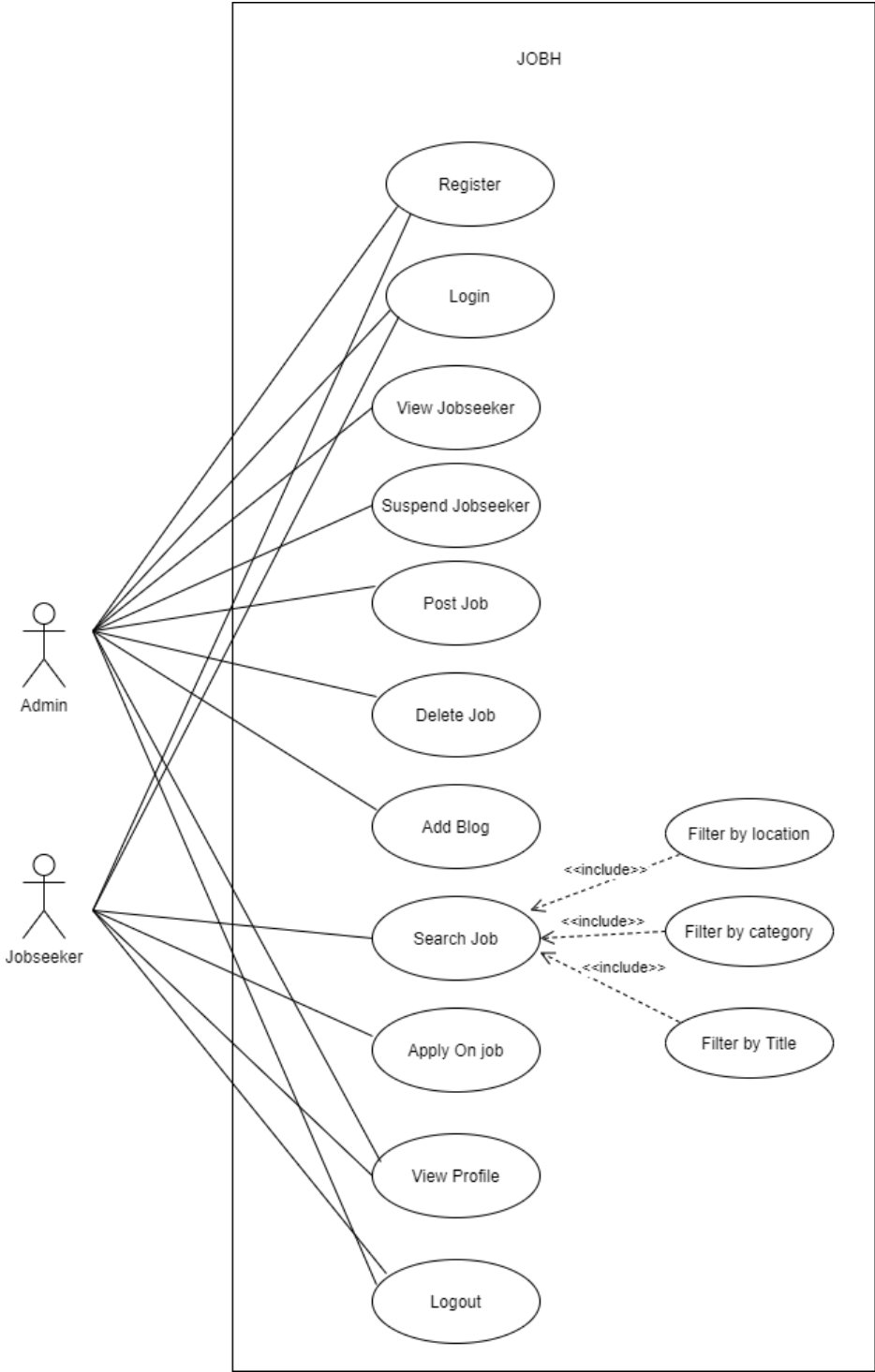


Figure 2.1 Use Case Diagram

2.13 Use Case Description

The use case diagram is a graphical representation of a user's interaction with the system. Use case diagrams can portray the different types of users of a system and ways that they interact with the system. A list of use-cases mentioned in the use-case diagram is described in detail so that we can look more precisely how the user can interact with the system to perform tasks. As this application is multiple users based and there are two main types of users.

2.13.1 Use Case 1: Register

Table 2.2 UC 1: Register

ID	UC1
Name	Register
Primary Actor	Job Seeker
Pre-Conditions	Job Seeker has opened the application.
Post-Conditions	Job Seeker has registered in the application
Main Success Scenario	<ol style="list-style-type: none"> 1. User opens the System. 2. System displays the register screen. 3. Users enter their information (name, email, password) 4. System validates credentials 5. System displays the home screen
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1. The server is down. <ol style="list-style-type: none"> i. System displays nothing. 2. Job Seeker presses the Register option without providing the required information. <ol style="list-style-type: none"> ii. System does not submit anything.
Frequency	Could be nearly continuous

2.13.2 Use Case 2: Login

Table 2.3 UC 2: Login

ID	UC2
Name	Login
Primary Actor	Admin, Job Seeker
Pre-Conditions	System displays the login screen.
Post-Conditions	Admin/ Job Seeker login to the system.
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin/ Job Seeker opens the JOBH application 2. System display log in screen 3. Admin/Jobseeker enter username and password 4. System display home page
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1. Admin enters incorrect credentials. <ol style="list-style-type: none"> i. System prompts to enter correct username 2. Admin submits information without filling all required fields. <ol style="list-style-type: none"> ii. System asks the user to fill all required fields.
Frequency	Could be nearly continuous

2.13.3 Use Case 3: View Job Seekers Details

Table 2.4 UC 3: View Job Seekers Details

ID	UC3
Name	View Job Seekers Details
Primary Actor	Admin
Pre-Conditions	Admin has logged into the application.
Post-Conditions	Admin has viewed the selected employer.
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin has logged into the application 2. System display home screen 3. Admin search job seeker from list of job seeker 4. System returns searched job seeker. 5. Admin view job seeker detail.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1. Admin related to that searched keyword is not available. <ol style="list-style-type: none"> i. System displays an empty result 2. Admin presses the Search key without entering anything. <ol style="list-style-type: none"> ii. System does not submit anything.
Frequency	Low

2.13.4 Use Case 4: Suspend Job Seeker

Table 2.5 UC 4: Suspend Job Seeker

ID	UC4
Name	Suspend Job Seeker
Primary Actor	Admin
Pre-Conditions	Admin has logged into the application.
Post-Conditions	Admin has suspended the selected Job Seeker.
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin log in into system 2. System display home screen 3. Admin searches job seeker from list of job seeker 4. System returns search job seeker 5. Admin suspended searched job seeker.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1. Admin related to that searched keyword is not available. <ol style="list-style-type: none"> i. System displays an empty result 2. Admin presses the Search key without entering anything. <ol style="list-style-type: none"> ii. System does not submit anything.
Frequency	Low

2.13.5 Use Case 5: Search Jobs

Table 2.6 UC 5: Search Jobs

ID	UC5
Name	Search Jobs
Primary Actor	Job Seeker
Pre-Conditions	Job Seeker has logged into the application.
Post-Conditions	Jobs appeared based on the keyword searched.
Main Success Scenario	<ol style="list-style-type: none"> 1. Job Seeker has logged into the application. 2. System displays the Home Screen. 3. Job Seeker search job based on a keyword. 4. System returns relevant jobs.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. Job Seeker waits until server response and internet connection are recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. Job Seeker restarts the system. 3) Jobs related to that searched keyword is not available. 4) Job Seeker clicks the Search key without entering anything.
Frequency	Could be nearly continuous

2.13.6 Use Case 6: Filter By Location

Table 2.7 UC 6: Filter By Location

ID	UC6
Name	Filter By Location
Primary Actor	Job Seeker
Pre-Conditions	Job Seeker has logged into the system.
Post-Conditions	System displays jobs based on location.
Main Success Scenario	<ol style="list-style-type: none"> 1. Job Seeker clicks the filter button. 2. Job Seeker select by location. 3. Job Seeker submits a query. 4. System displays relative jobs.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. Job Seeker waits until server response and internet connection is recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. Job Seeker restarts the system. 3) Jobs related to that searched keyword is not available. 4) Job Seeker clicks the Search key without entering anything.
Frequency	Could be nearly continuous

2.13.7 Use Case 7: Filter By Category Stack

Table 2.8 UC 7: Filter By Category Stack

ID	UC7
Name	Filter By Category Stack
Primary Actor	Job Seeker
Pre-Conditions	Job Seeker has logged into the system.
Post-Conditions	System displays jobs based on Category Stack.
Main Success Scenario	<ol style="list-style-type: none"> 1. Job Seeker clicks the filter button. 2. Job Seeker selects by Category Stack. 3. Job Seeker submits a query. 4. System displays relative jobs.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. User waits until server response and internet connection is recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. User restarts the system. 3) Jobs related to that searched keyword is not available. 4) Job Seeker clicks the Search key without entering anything.
Frequency	Could be nearly continuous

2.13.8 Use Case 8: Filter By Position

Table 2.9 UC 8: Filter By Position

ID	UC8
Name	Filter By Position
Primary Actor	Job Seeker
Pre-Conditions	Job Seeker has logged into the system.
Post-Conditions	System displays jobs based on Position.
Main Success Scenario	<ol style="list-style-type: none"> 1. Job Seeker clicks the filter button. 2. Job Seeker select by Position. 3. Job Seeker submits the query. 4. System displays relative jobs.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. User waits until server response and internet connection is recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. User restarts system. 3) Jobs related to that searched keyword is not available. 4) Job Seeker clicks Search key without entering anything.
Frequency	Could be nearly continuous

2.13.9 Use Case 9: Filter By Organization

Table 2.10 UC 9: Filter By Company

ID	UC9
Name	Filter By Organization
Primary Actor	Job Seeker
Pre-Conditions	Job Seeker has logged into the system.
Post-Conditions	System displays jobs based on Organization.
Main Success Scenario	<ol style="list-style-type: none"> 1. Job Seeker clicks the filter button. 2. Job Seeker selected by Organization. 3. Job Seeker submits the query. 4. System displays relative jobs.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. User waits until server response and internet connection is recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. User restarts the system. 3) Jobs related to that searched keyword is not available. 4) Job Seeker clicks the Search key without entering anything.
Frequency	Could be nearly continuous

2.13.10 Use Case 10: Apply Jobs

Table 2.11 UC 10: Apply Jobs

ID	UC10
Name	Apply Jobs
Primary Actor	Job Seeker
Pre-Conditions	Job Seeker has searched for the required job.
Post-Conditions	Job Seeker has applied for the searched job.
Main Success Scenario	<ol style="list-style-type: none"> 1. System displays the searched jobs. 2. Job Seeker clicks the apply button to apply on the job. 3. System reruns the form to fill out. 4. Job Seeker fills the required information.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. User waits until server response and internet connection is recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. User restarts the system. 3) Jobs related to that searched keyword is not available. 4) Job Seeker clicks the Search key without entering anything.
Frequency	Low

2.13.11 Use Case 11: View Applied Jobs

Table 2.12 UC 11: View Applied Jobs

ID	UC11
Name	View Applied Jobs
Primary Actor	Job Seeker
Pre-Conditions	Job Seeker has applied to one or multiple jobs.
Post-Conditions	Job Seeker has viewed the applied job.
Main Success Scenario	<ol style="list-style-type: none"> 1. System displays the Home Screen. 2. Job Seeker clicks the Profile button. 3. System displays the User Profile. 4. Job Seeker clicks the view applied jobs 5. System displays the list of applied jobs.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. User waits until server response and internet connection are recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. User restarts the system. 3) Jobs related to that searched keyword is not available. 4) Job Seeker clicks the Search key without entering anything.
Frequency	Low

2.13.12 Use Case 12: View Profile

Table 2.13 UC 12: View Profile

ID	UC12
Name	View Profile
Primary Actor	Job Seeker
Pre-Conditions	Job Seeker has logged into the system.
Post-Conditions	System displays Profile.
Main Success Scenario	<ol style="list-style-type: none"> 1. Job Seeker clicks View Profile button. 2. System displays Profile.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. Job Seeker waits until server response and internet connection is recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. Job Seeker restarts system.
Frequency	Low

2.13.13 Use Case 13: Post Job Vacancy

Table 2.14 UC 13: Post Job Vacancy

ID	UC13
Name	Post Job Vacancy
Primary Actor	Admin
Pre-Conditions	Admin has logged into the application.
Post-Conditions	Admin has posted the job.
Main Success Scenario	<ol style="list-style-type: none"> 1. System display home screen 2. Admin click on post job from option 3. System display form to fill out information 4. Admin fill information
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. User waits until server response and internet connection are recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. User restarts system.
Frequency	Could be nearly continuous

2.13.14 Use Case 14: View Posted Jobs

Table 2.15 UC 14: View Posted Jobs

ID	UC14
Name	View Posted Jobs
Primary Actor	Admin
Pre-Conditions	Admin has logged into the application.
Post-Conditions	Admin has viewed jobs.
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin clicks on the View Jobs from options 2. System displays all jobs.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. Admin waits until server response and internet connection are recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. Admin restarts the system.
Frequency	Low

2.13.15 Use Case 15: Pause Job Post

Table 2.16 UC 15: Pause Job Post

ID	UC15
Name	Pause Job Post
Primary Actor	Admin
Pre-Conditions	Admin has logged into the application.
Post-Conditions	Admin has paused the job.
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin clicks on the Pause Job from options 2. System pauses the jobs.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. Admin waits until server response and internet connection are recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. Admin restarts the system.
Frequency	Low

2.13.16 Use Case 16: Resume Job Post

Table 2.17 UC 16: Resume Job Post

ID	UC16
Name	Resume Job Post
Primary Actor	Admin
Pre-Conditions	Admin has logged into the application.
Post-Conditions	Admin has Resume the job.
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin clicks on the Resume Job from options 2. System resumes the jobs.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. Admin waits until server response and internet connection are recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. Admin restarts the system.
Frequency	Low

2.13.17 Use Case 17: Deleted Job Post

Table 2.18 UC 17: Deleted Job Post

ID	UC17
Name	Delete Job Post
Primary Actor	Admin
Pre-Conditions	Admin has logged into the application.
Post-Conditions	Admin has paused the job.
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin clicks on the Delete Job from options 2. System deletes the job.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. Admin waits until server response and internet connection are recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. Admin restarts system.
Frequency	Low

2.13.18 Use Case 18: Logout

Table 2.19 UC 18: Logout

ID	UC18
Name	Logout
Primary Actor	Admin, Job Seeker
Pre-Conditions	Admin/ Job Seeker has logged into the application.
Post-Conditions	Admin/ Job Seeker logout from the system.
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin/ Job Seeker clicks the Logout button. 2. System logout the Admin/ Job Seeker.
Alternative Flows or Extensions	<ol style="list-style-type: none"> 1) Server down or internet connection not working. <ol style="list-style-type: none"> i. Admin/ Job Seeker waits until the server response and internet connection are recovered. 2) At any time, system crashes. <ol style="list-style-type: none"> ii. Admin/ Job Seeker restarts system.
Frequency	Could be nearly continuous

2.14 System Sequence Diagrams

The purpose of the **System Sequence Diagram (SSD)** is to illustrate the successful scenario of a use case in a visual format.

2.14.1 SSD for Register

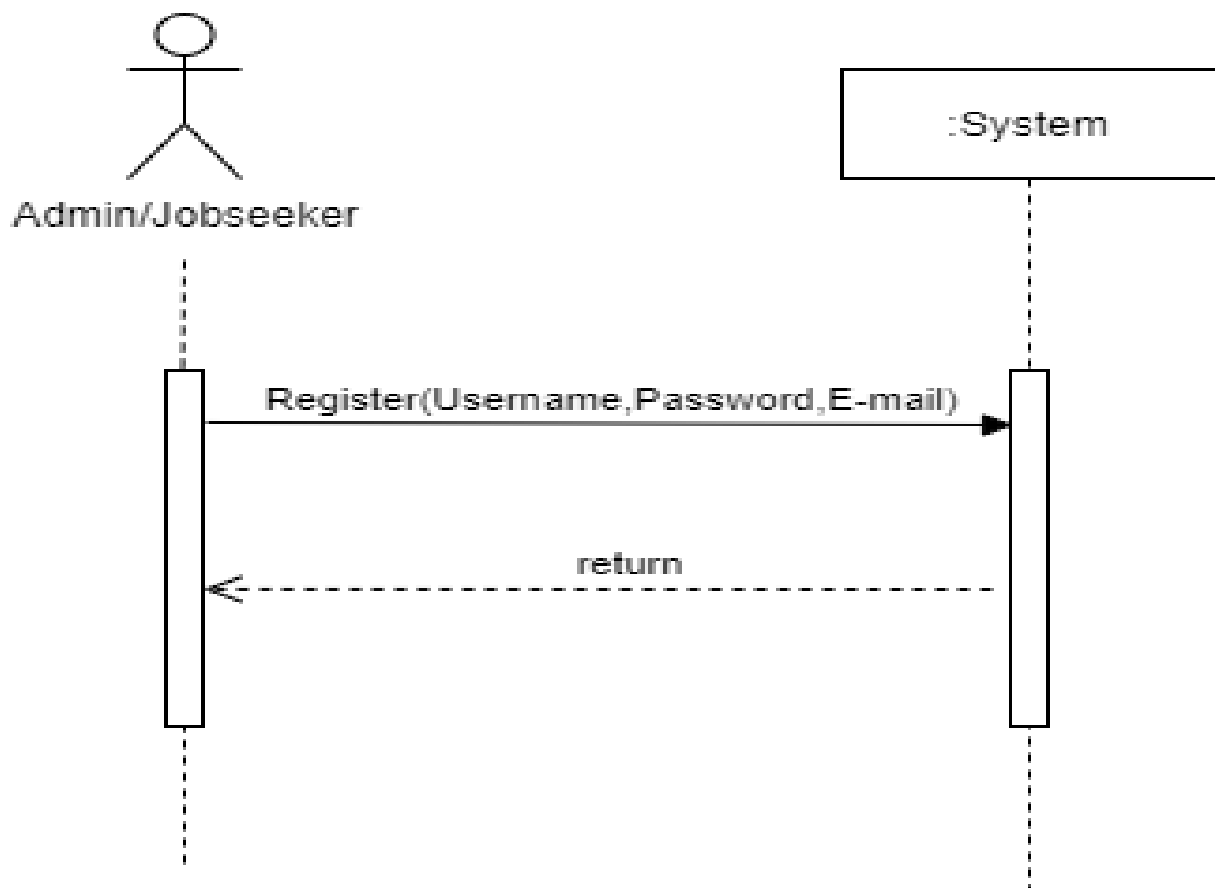


Figure 2.1 SSD for Register

2.14.2 SSD for Login

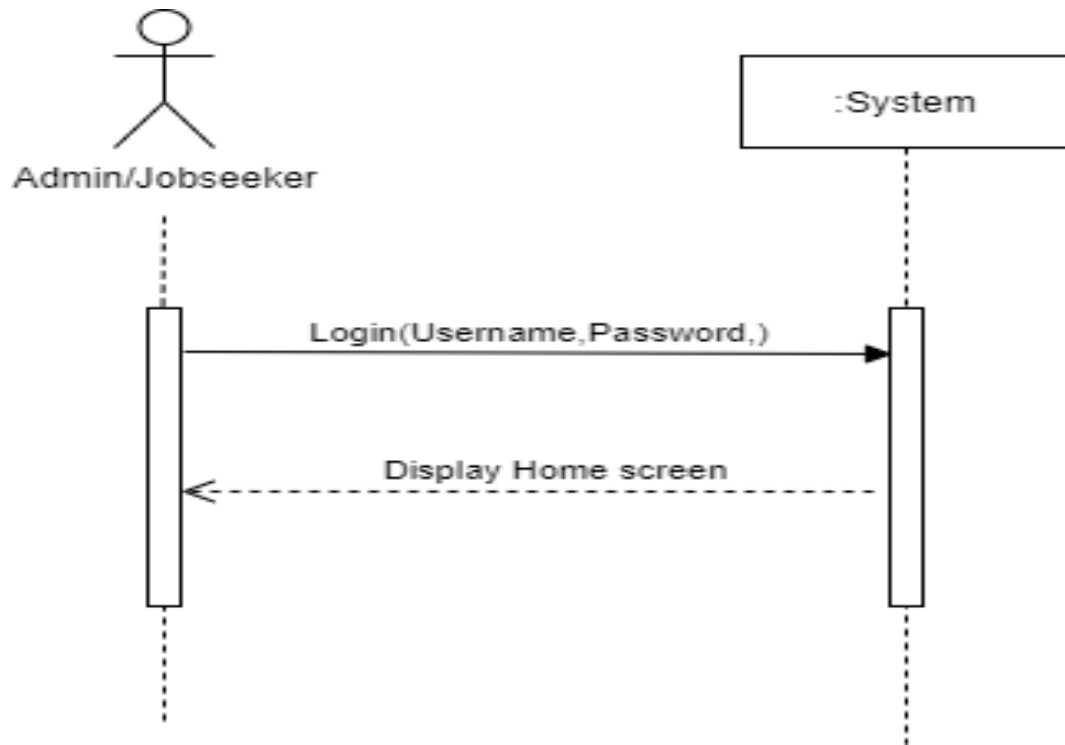


Figure 2.2 SSD for Login

2.14.3 SSD for View Job Seeker Details

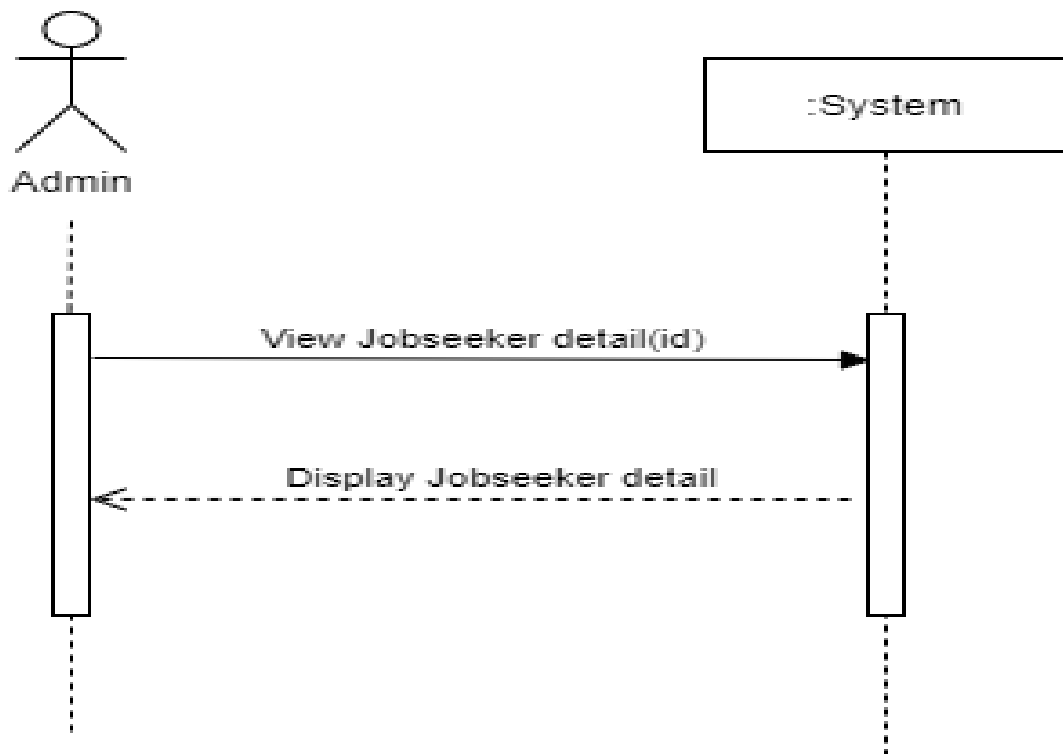


Figure 2.3 SSD for View Job Seeker Detail

2.14.4 SSD for Suspend Job Seeker

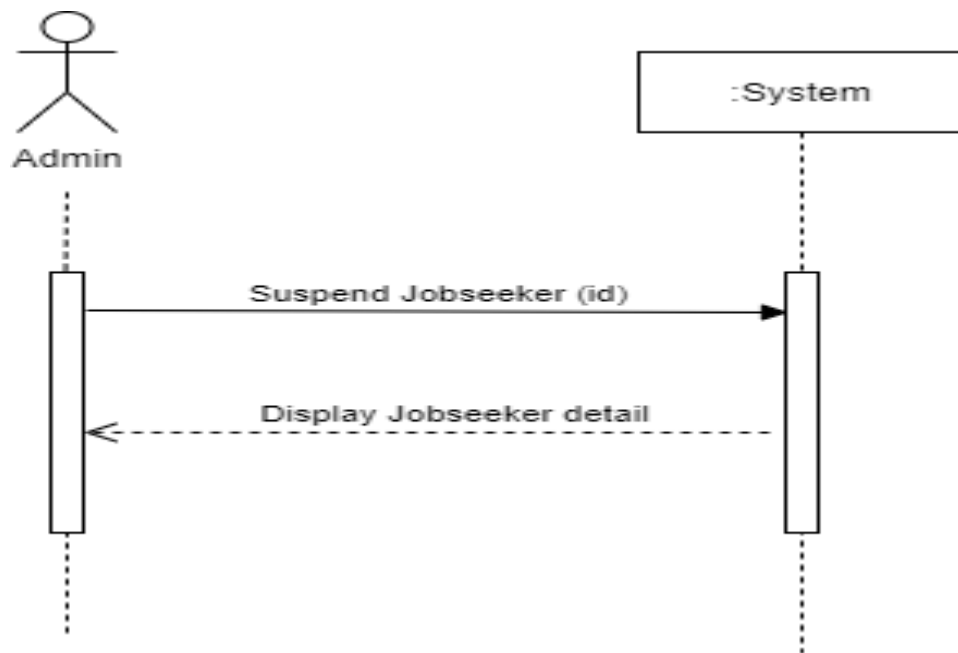


Figure 2.4 SSD for Suspend Job Seeker

2.14.5 SSD for Search Job

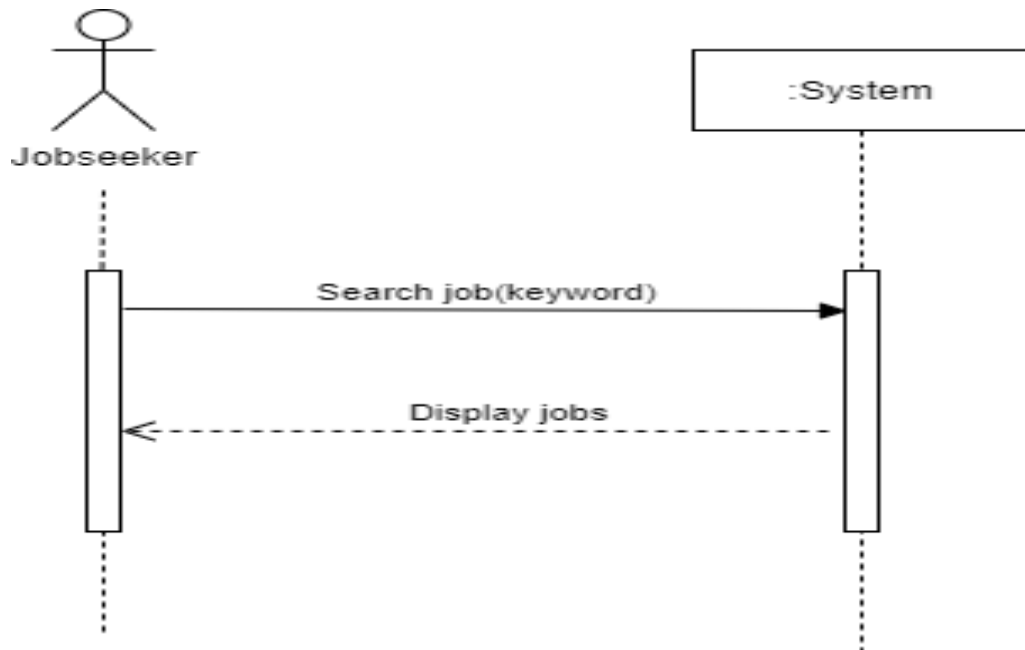


Figure 2.5 SSD for Search Job

2.14.6 SSD for Filter By Location

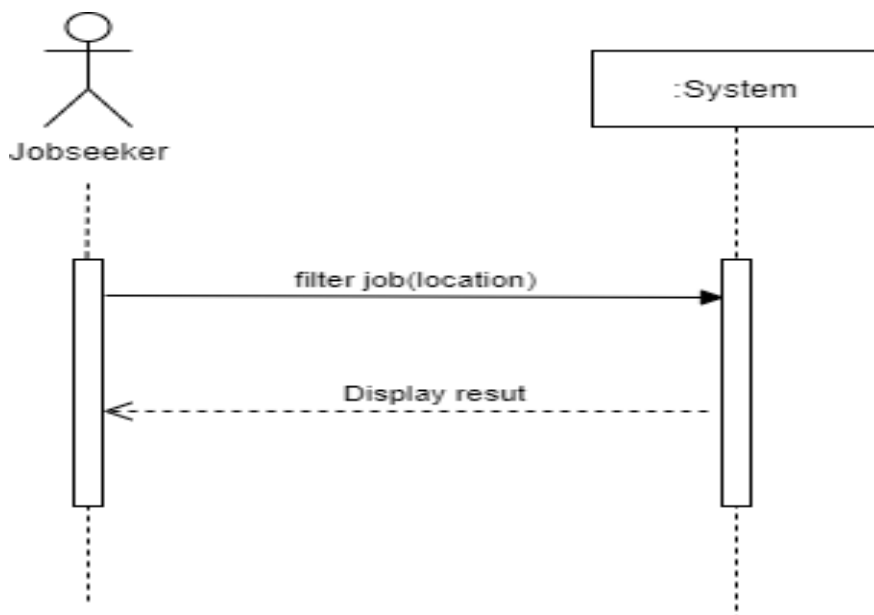


Figure 2.6 SSD for Filter By Location

2.14.7 SSD for Filter By Position

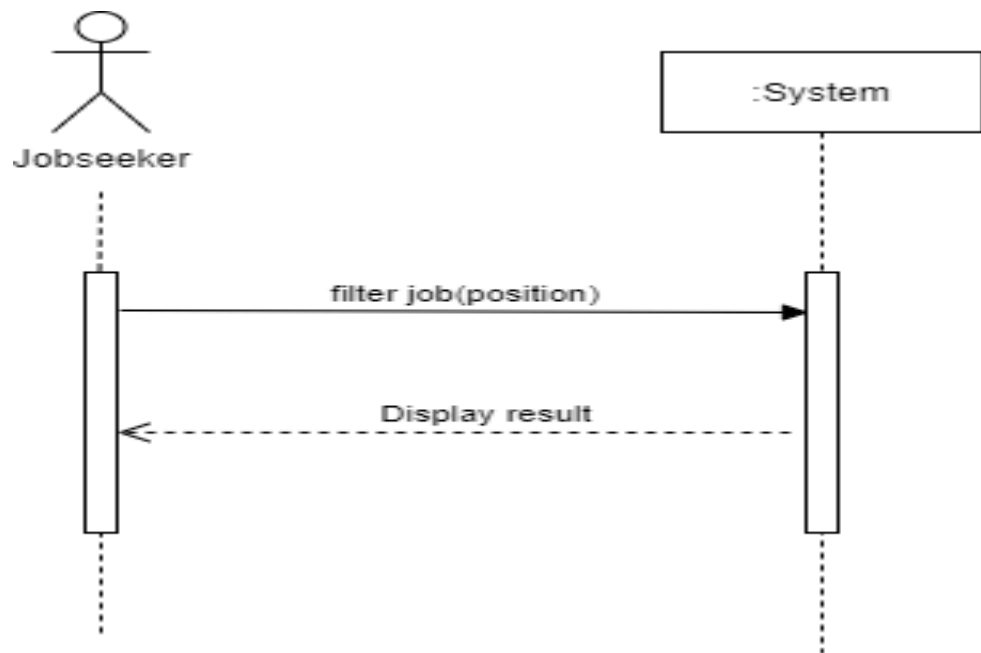


Figure 2.7 SSD for Filter By Position

2.14.8 SSD for Filter By Different Category Stack

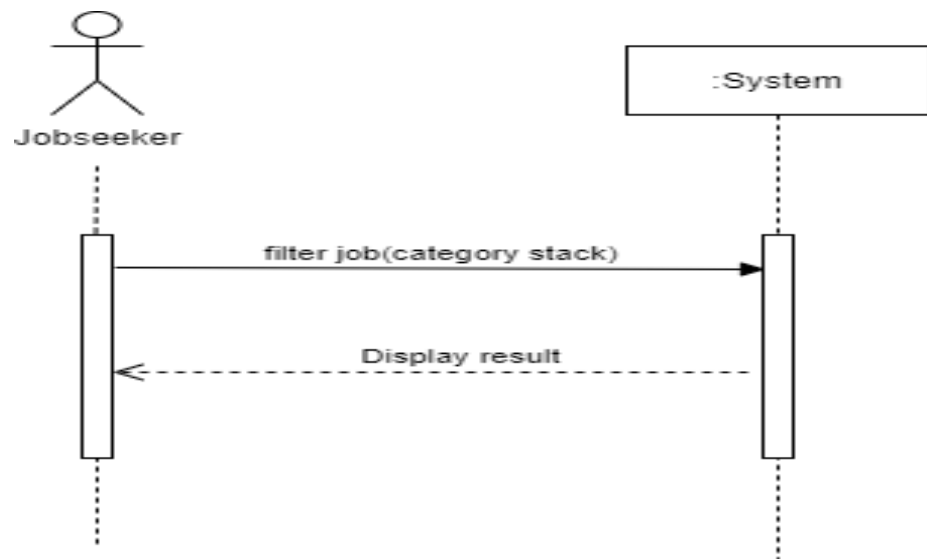


Figure 2.8 SSD for Filter By Category Stack

2.14.9 SSD for Filter By Organization

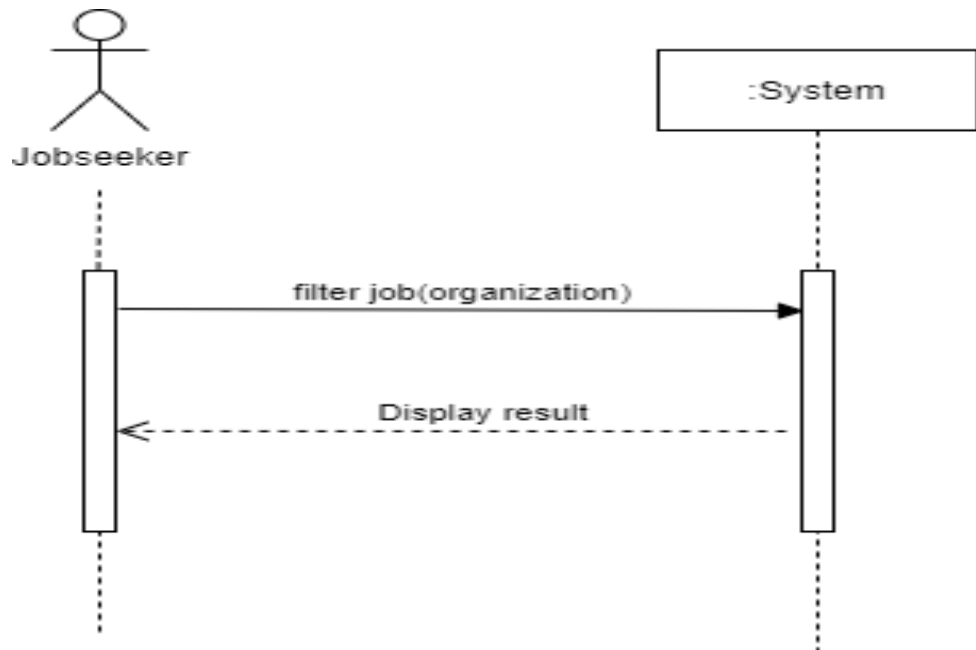


Figure 2.9 SSD for Filter By Organization

2.14.10 SSD for Apply Job



Figure 2.10 SSD for Apply Job

2.14.11 SSD for View Applied Jobs

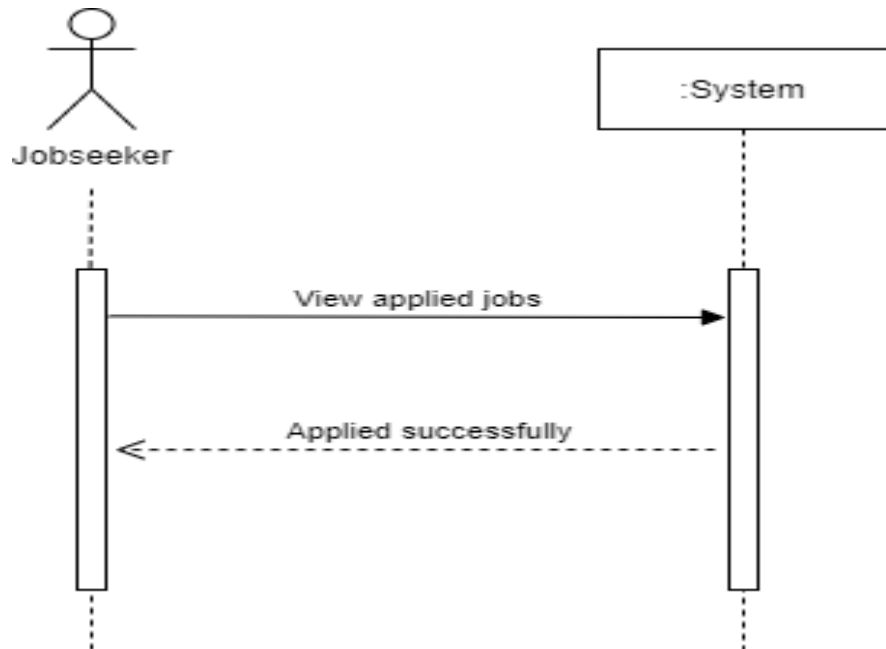


Figure 2.11 SSD for View Applied Jobs

2.14.12 SSD for View Profile

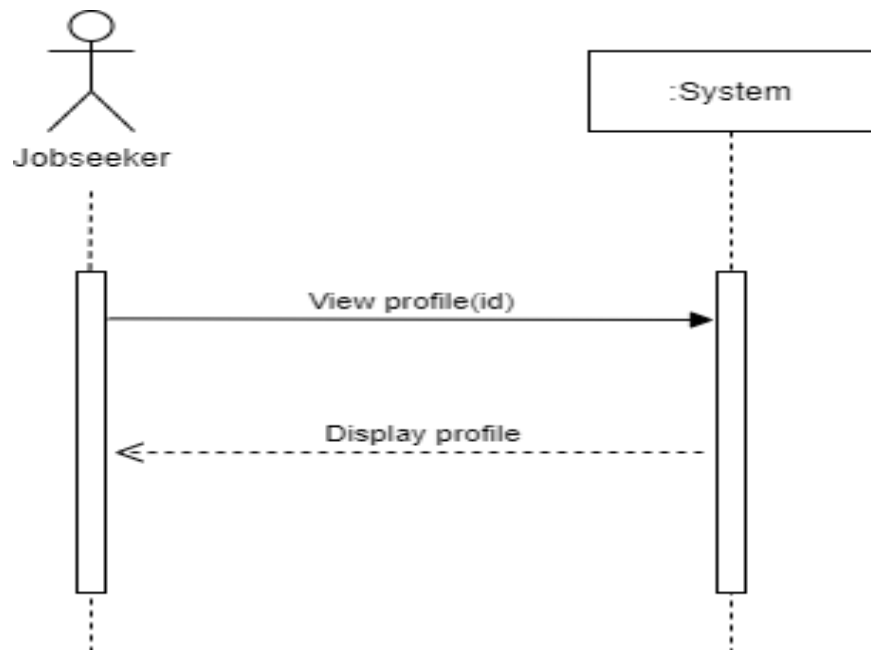


Figure 2.12 SSD for Filter By View Profile

2.14.13 SSD for Post Job

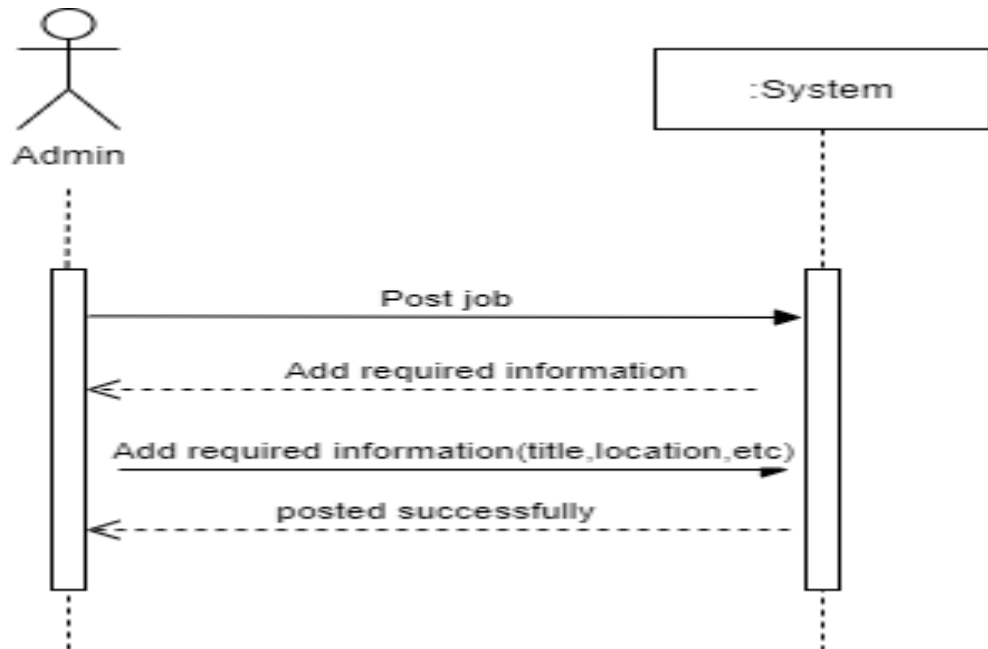


Figure 2.13 SSD for Post Job

2.14.14 SSD for View Posted Jobs

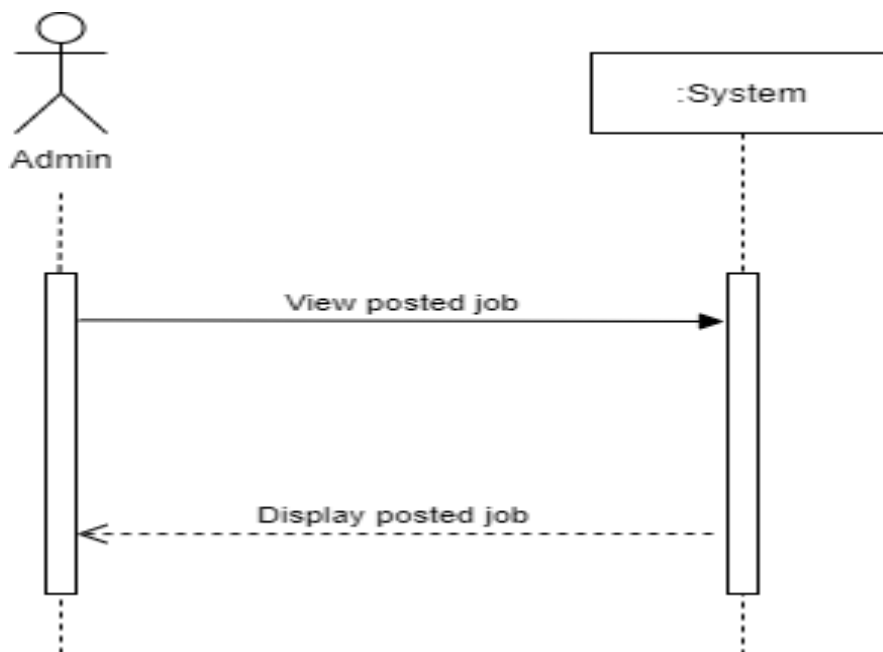


Figure 2.14 SSD for View Posted Jobs

2.14.15 SSD for Pause Job Post

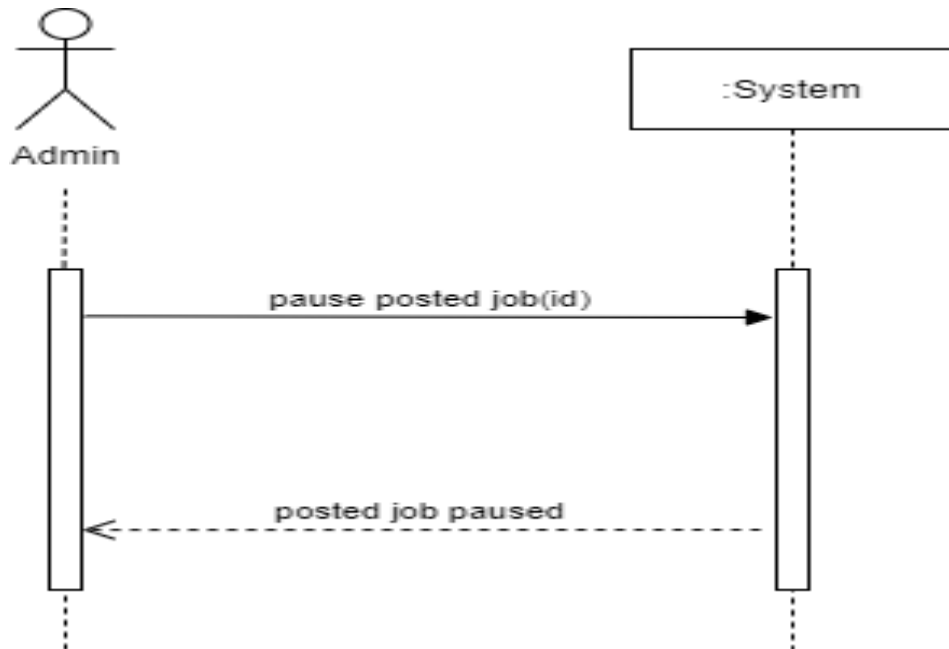


Figure 2.15 SSD for Pause Job Post

2.14.16 SSD for Resume Job Post

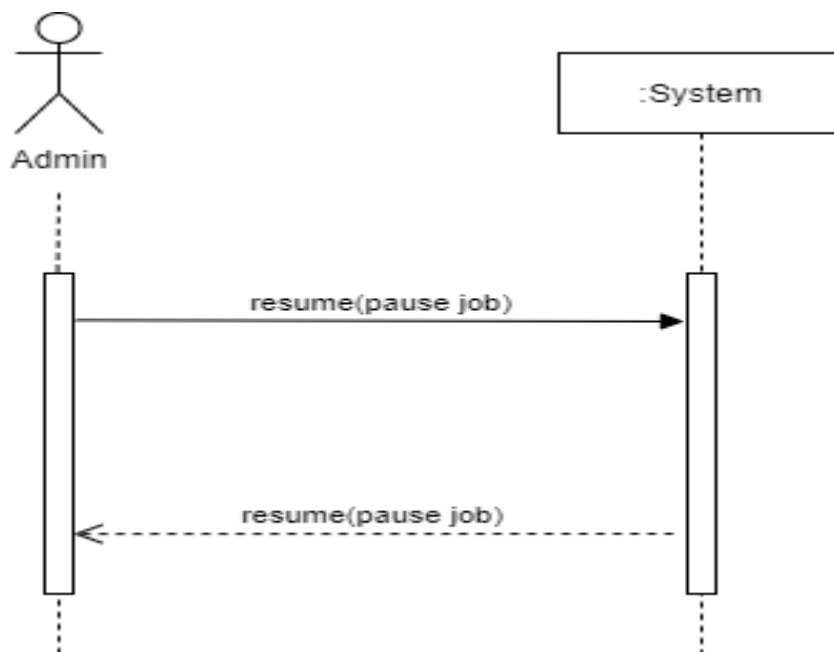


Figure 2.16 SSD for Resume Job Post

2.14.17 SSD for Delete Job Post

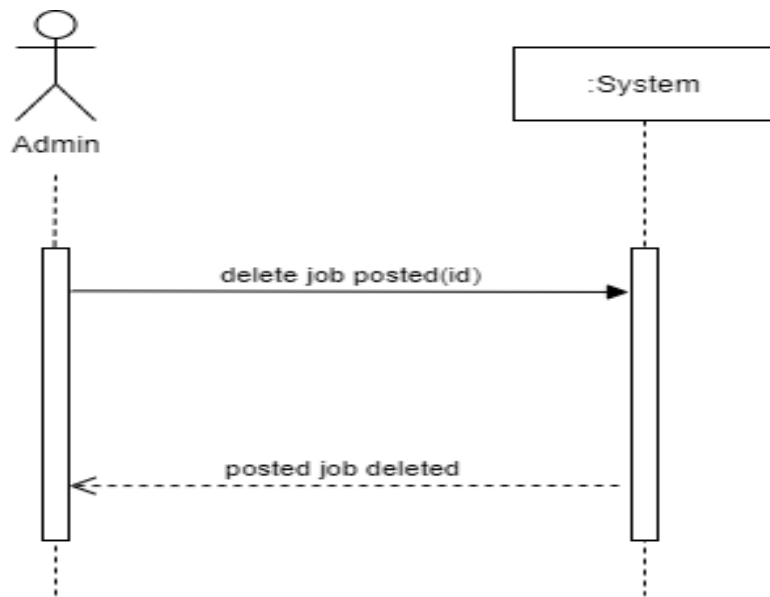


Figure 2.17 SSD for Delete Job Post

2.14.18 SSD for Logout

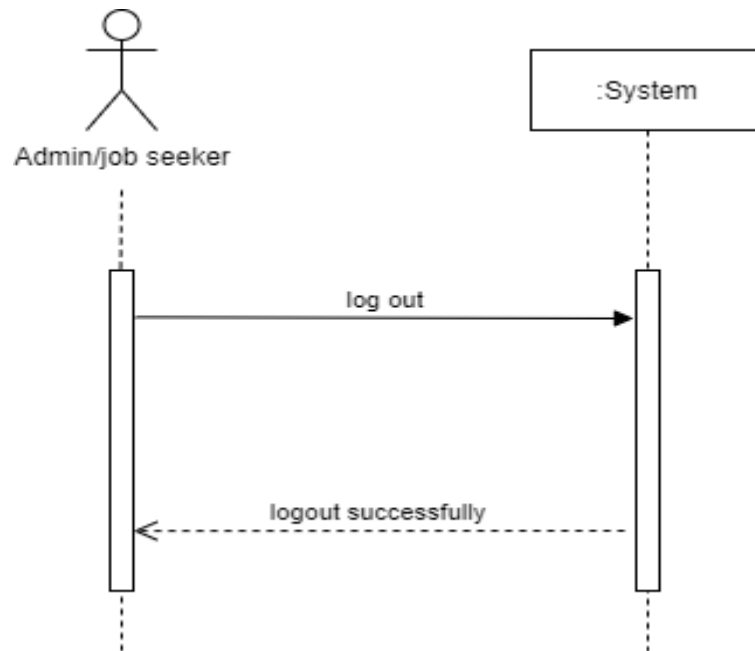


Figure 2.18 SSD for Logout

Chapter 3

Software Design Description

3.1 Introduction

Software Design Description (SDD) is a document that provides a complete description of the design of the software to be developed before the actual development. The SDD document describes the system architecture design in detail and provides a complete description of the different components. It also describes how the different components will communicate with each other. The SDD document also contains the interface design, architecture diagram, sequence diagrams, and class diagram.

3.2 Interface Design

3.2.1 Simple and Appealing

The interface is simple to use, a naïve user can also use it very easily and efficiently to get their required information. The image buttons are used that make the website appealing.

3.2.2 Responsive

The interface is responsive this means it can adjust its size according to the device's display Like (small screens, large screens), etc.

3.3 Interfaces

3.3.1 Login

This picture will show the interface of the login screen. Which is used in the project. It can contain two fields of email and password.

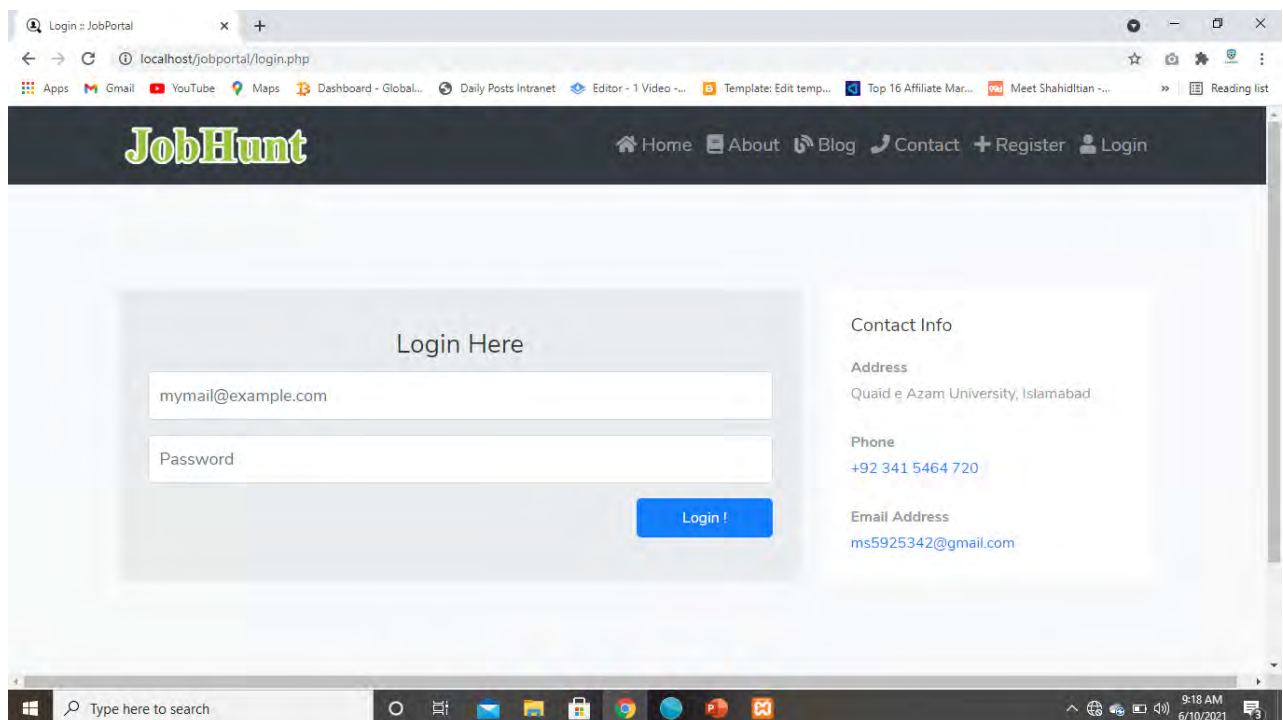


Figure 3.1 Login Interface

3.3.2 Register

This picture will show the interface of the register screen. Which is used in the project. It can contain a field of personal details and account details.

Register : JobPortal

localhost/jobportal/register.php

User Type

Job Seeker

Name *

* John Smith

Phone *

03001234567

Email *

mymail@example.com

Password *

City

Select City

Contact Info

Address

Quaid e Azam University, Islamabad

Phone

+92 341 5464 720

Email Address

ms5925342@gmail.com

More Info

If you want know to more about JobH.What is scope and purpose of this Job Hunter.To get all information.Click on the button

Learn More

Figure 3.2 Register Interface

3.3.3 Applied Jobs

This picture will show the interface of applied jobs screen. Which is used in the project. It shows the jobs on which the user applied.

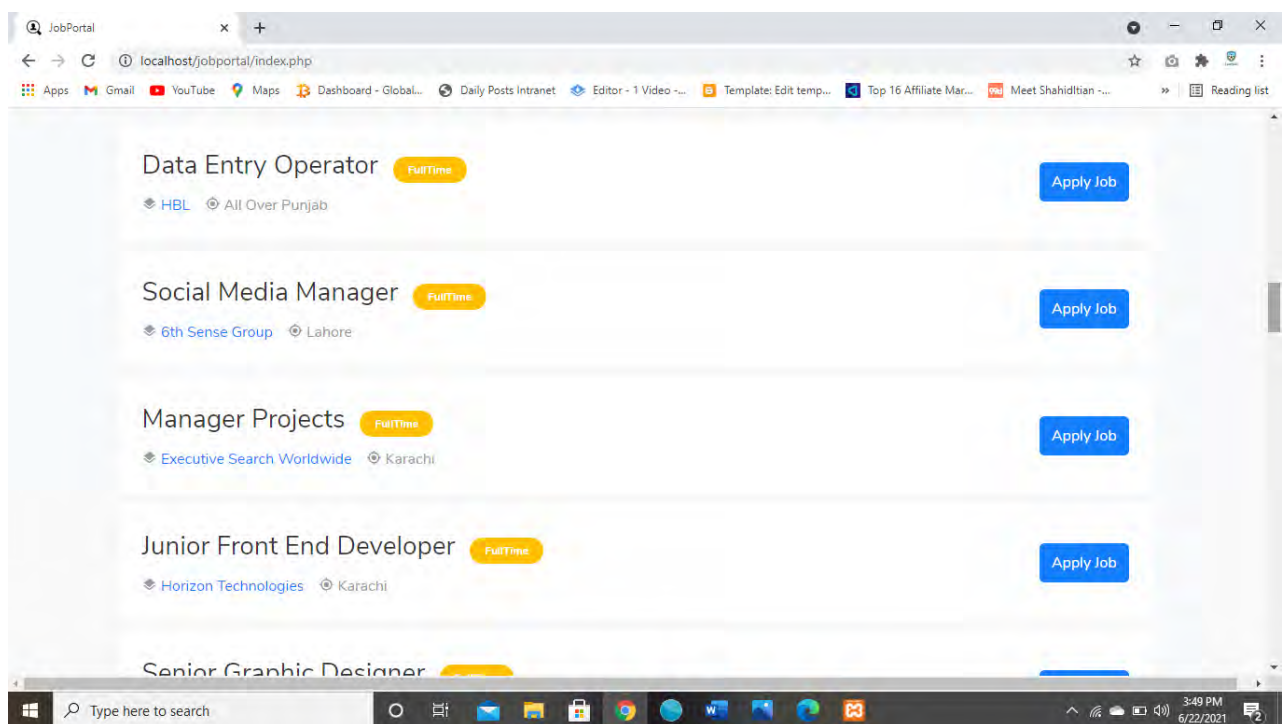


Figure 3.3 Applied Jobs Interface

3.3.3 Manage Jobs

This picture will show the interface of manage jobs screen. Which is used in the project. It shows the jobs that an employer/admin can manage.

The screenshot shows a web browser window with the URL `localhost/jobportal/manage_jobs.php`. The page title is "JobHunt" and the user is logged in as "Muhammad Shahid Admin". The sidebar on the left contains the following menu items: Dashboard, Add Category, Manage Category, Post Job, Manage Jobs, and CVs. The main content area is titled "Manage Jobs" and displays a table of jobs.

#	Job	Detail	Qualification	Sector	Action
40	Social Media Manager Designing and implementing social media strategy to align with business goals. www.paperads.com	Manager (FullTime) 31,000 Lahore	BBA 2021-06-23	6th Sense Group Social Media Advertising 2021-06-08	Delete
39	Manager Projects Achieves operational objectives by contributing information and recommendations to strategic plans and reviews. www.glassdoor.com	Manager (FullTime) 50,000 Karachi	MBA 2021-06-26	Executive Search Worldwide Project Management 2021-06-08	Delete

Figure 3.3 Manage Jobs Interface

3.3.4 Post Jobs

This picture will show the interface of the post jobs screen. Which is used in the project. It shows how an employer can post a job.

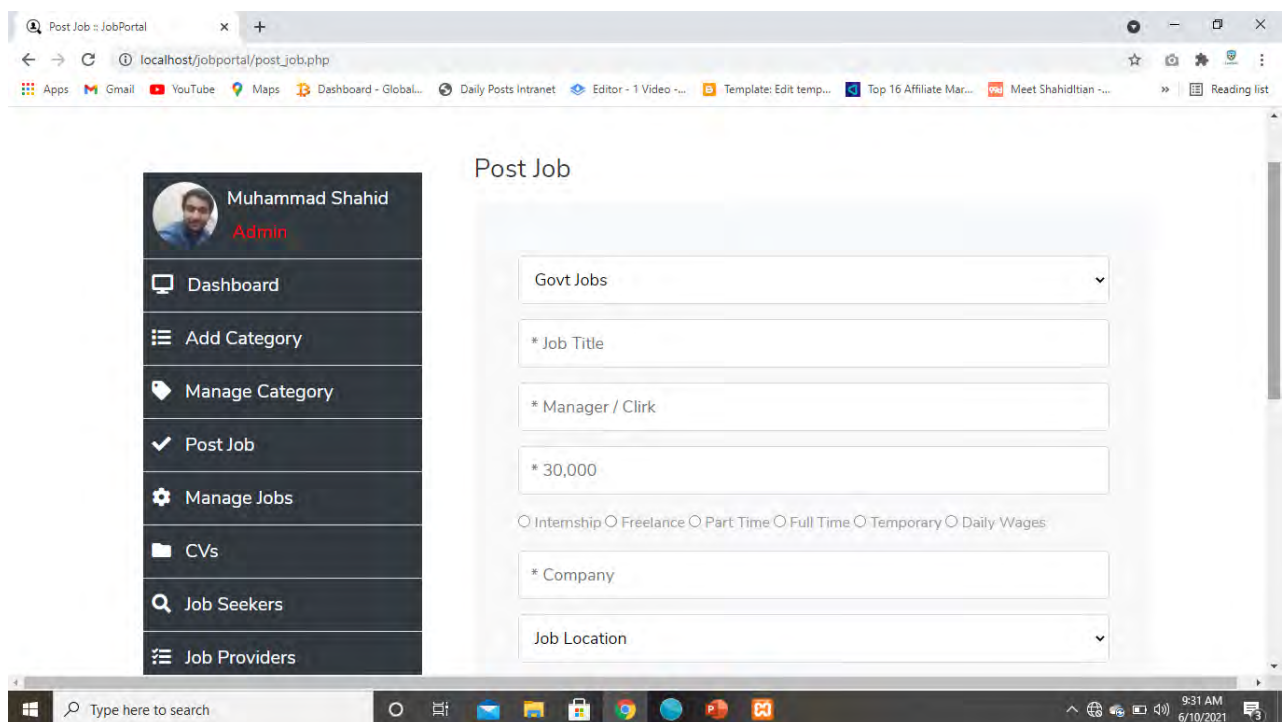


Figure 3.4 Post Jobs Interface

3.3.5 Job Details

This picture will show the interface of manage jobs screen. Which is used in the project. It shows the job details.

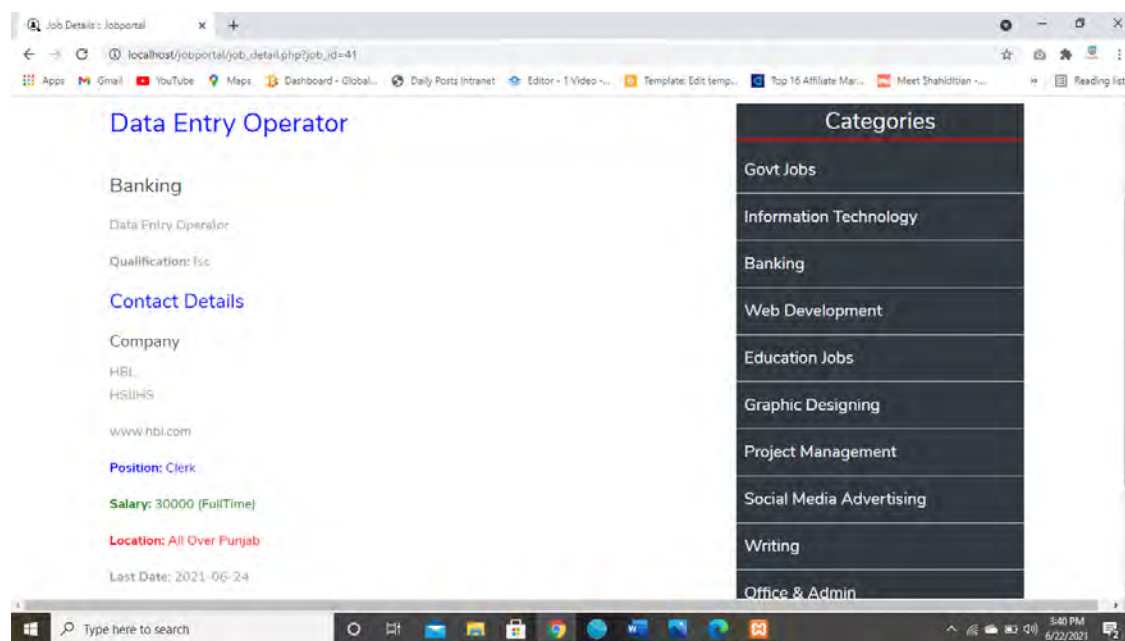


Figure 3.5 Job Details

3.4 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

3.4.1 Login

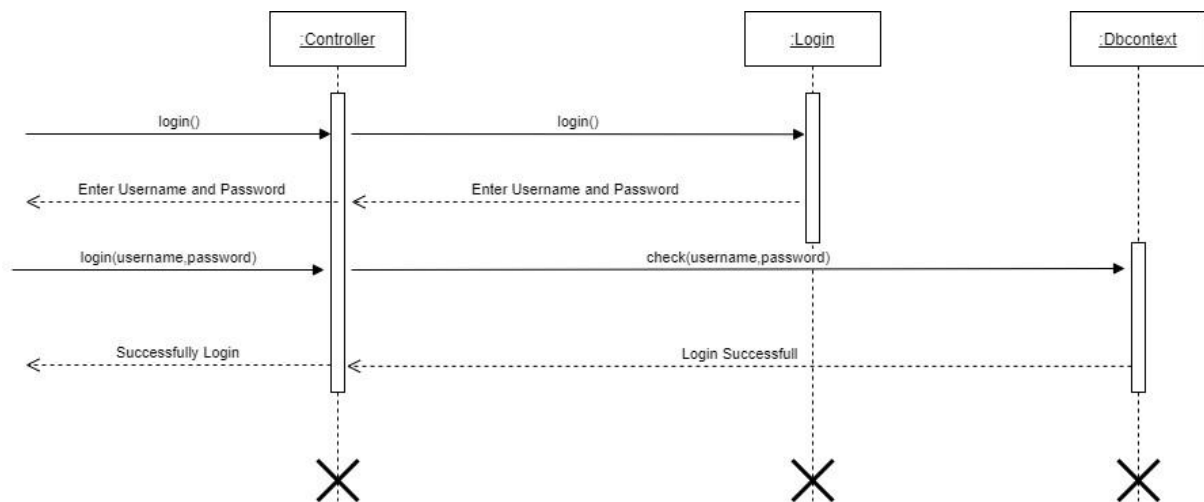


Figure 3.7 SD-1: Login

3.4.2 Suspend Job Seekers

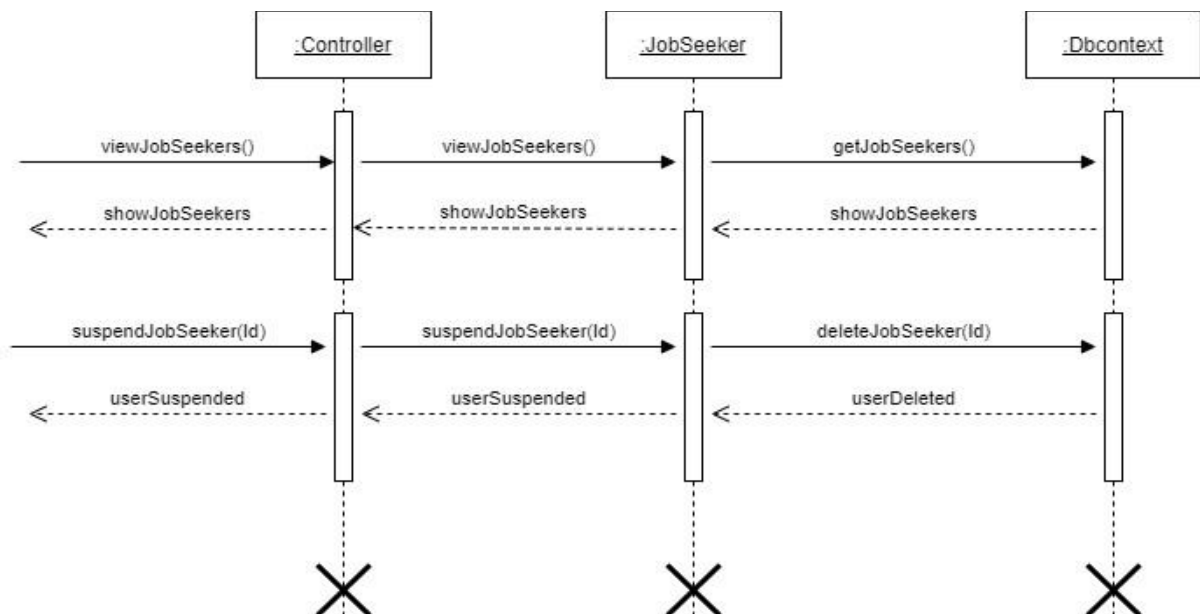


Figure 3.8 SD-2: Suspend Job Seekers

3.4.3 Apply Jobs

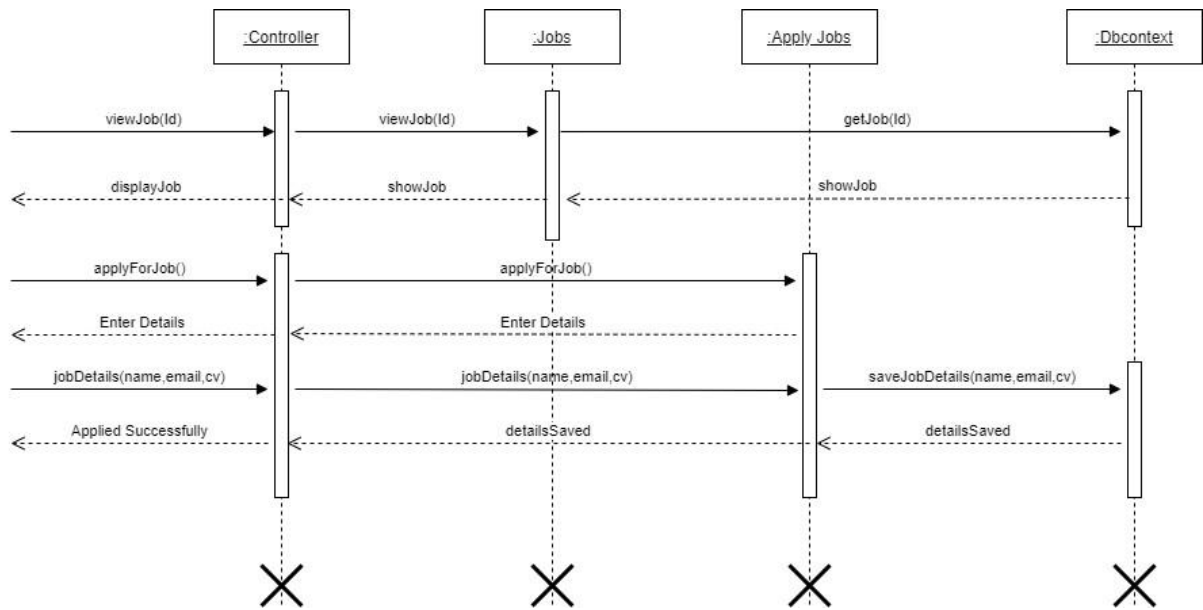


Figure 3.9 SD-3: Apply Jobs

3.4.4 Post Jobs

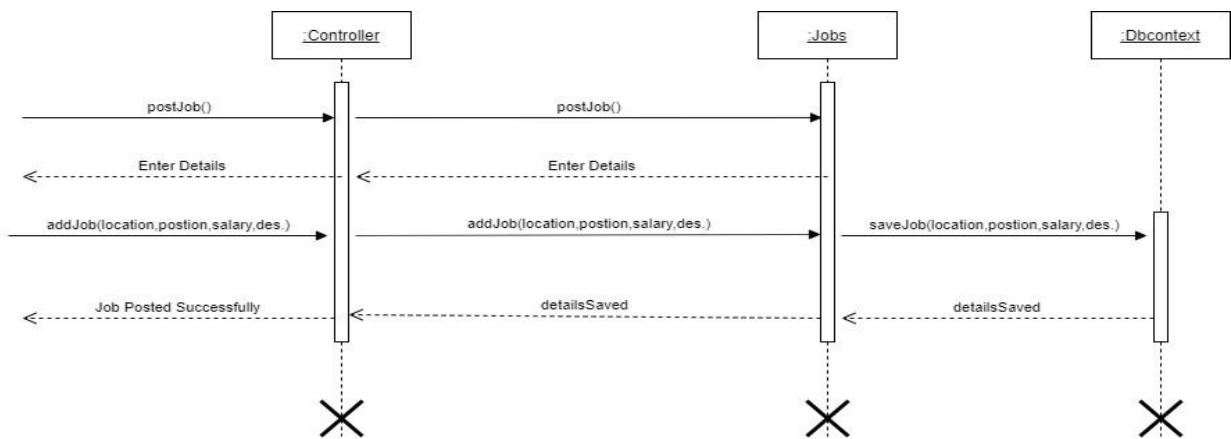


Figure 3.10 SD-4: Post Jobs

3.4.5 Register

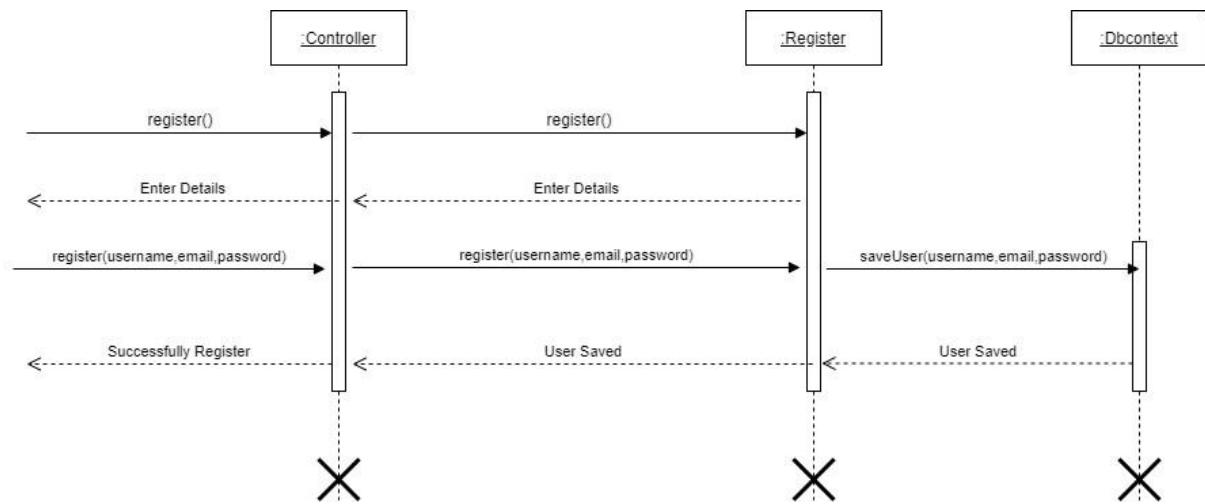


Figure 3.11 SD-5: Register

3.5 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects

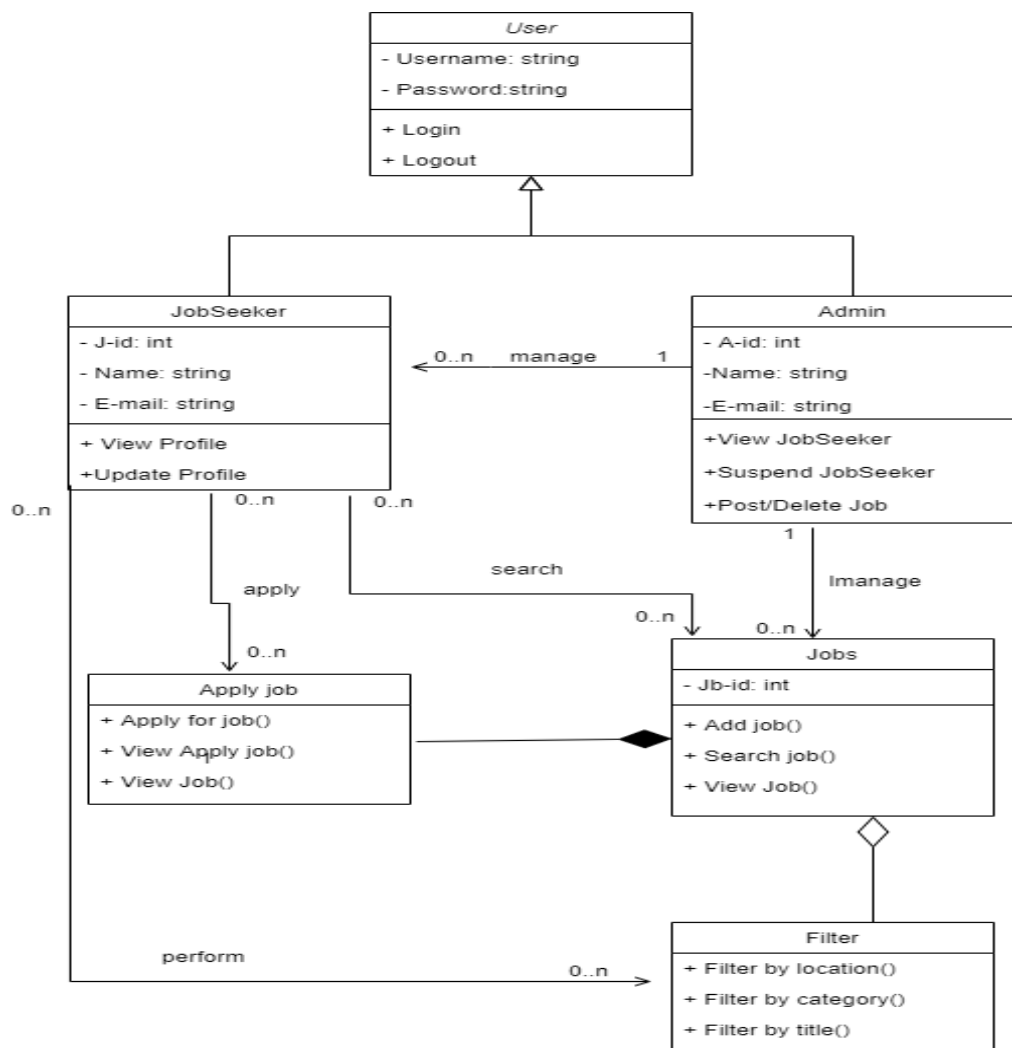


Figure 3.12 Class Diagram

3.7 Summary

This chapter gives the complete description of software design. It gives details description regarding architecture design, components of the system and user interface description. Finally, interaction between the object and human actor are shown by interaction diagram and relationship between the instances is shown by class diagram.

Chapter 4

Implementation

4 System Implementation

System implementation is the process of defining how the information system should be built ensuring that the information system is operational and used and meets quality standard. The choice of the technologies used was based on the principle of modularity and performance. I choose bootstrap 4 for designing the front end of the system such as client side. I choose PHP language for the purpose of back-end server-side implementation. After it, I use JavaScript language for handling the events in the front side theme. Furthermore, I choose jQuery language for performing the different types of functions in the project designing. In addition, the JSON is also used in the front side of project theme to perform the task without browsing. HTML (Hypertext Markup Language) is also used for the setting of text in the website front end. CSS (Cascading language) is also used for designing the project front end theme. In this project, the MySQL is used as database to store the data of the project.

4.1 Details of implementation and tools

The details of the languages, tools, and technologies we used are following:

4.1.1 Bootstrap

Get started with Bootstrap, the world's most popular framework for building responsive, mobile-friendly sites, with Bootstrap CDN and a template starter page. Bootstrap is a free and open-source CSS framework directed at responsive, mobile- first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. On January 31, 2012, Bootstrap 2 was released, which added built-in support for Glyph icons, several new components, as well as changes to many of the existing components. This version supports responsive web design, meaning the layout of web pages adjusts dynamically, considering the characteristics of the device used (whether desktop, tablet, or mobile phone).

4.1.2 HTML5

HTML5 is the latest version of Hypertext Markup Language, the code that describes web pages. It's actually three kinds of code: HTML, which provides the structure.

Cascading Style Sheets (CSS), which take care of presentation; and JavaScript, which makes things happen. HTML5 is the latest evolution of the standard that defines HTML. The term represents two different concepts. It is a new version of the language HTML, with new elements, attributes, and behaviors, and a larger set of technologies that allows the building of more diverse and powerful Web sites and applications. This set is

sometimes called HTML5 & friends and often shortened to just HTML5. HTML5 is a programming language whose acronym stands for Hyper Text Markup Language. It is a system that allows the modification of the appearance of web pages, as well as adjusting their appearance. It also used to structure and present content for the web. With HTML5, browsers

like Firefox, Chrome, Explorer, Safari and more, can know how to display a particular web page, know where the elements are, where to put the images and where to place the text. Apart from HTML5, there are other languages that are necessary to give format and interactivity to a site, but the basic structure of any page is first defined in the HTML5 language.

4.1.3 CSS (Cascading style sheet)

CSS stands for Cascading Style Sheets with an emphasis placed on “Style.” While HTML is used to structure a web document (defining things like headlines and paragraphs, and allowing you to embed images, video, and other media), CSS comes through and specifies your document’s style— page layouts, colors, and fonts are all determined with CSS. Think of HTML as the foundation (every house has one), and CSS as the aesthetic choices (there is a big difference between a Victorian mansion and a mid-century modern home). You might be wondering how this CSS code is applied to HTML content, though. Much like HTML, CSS is written in simple, plain text through a text editor or word processor on your computer, and there are three main ways to add that CSS code to your HTML pages. CSS code (or Style Sheets) can be external, internal, or inline. External style sheets are saved as .CSS files and can be used to determine the appearance of an entire website through one file (rather than adding individual instances of CSS code to every HTML element you want to adjust).

4.1.4 JavaScript ES6

ECMAScript 6, also known as ECMAScript 2015, is a significant update to the language, and the first update to the language since ES5 was standardized in 2009.

Implementation of these features in major JavaScript engines is underway now [1]. There are many new and important features added to the language such as DE structuring, arrow functions, let and cons, modules, classes which are a syntactical sugar for prototypal inheritance in JavaScript and other features which we used extensively throughout our project. However, ES6 JavaScript code still cannot run on all browsers therefore it needs to be converted to ES5 and it’s done through a transpire. I used Next.js which uses Babel-core and Web pack to convert ES6 code to ES5 code. We do not need to do it ourselves Next.js or CRA (create-react-app, though I did not use it in my project) do it for us out of the box.

4.1.5 PHP

PHP is a server-side scripting language. That is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages. PHP scripts can only be interpreted on a server that has PHP installed. PHP is a general-purpose scripting language especially suited to web development. It was originally created by Danish-Canadian Programmer Ramus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initials PHP: Hypertext Preprocessor. There are

different versions of PHP are evolved with the passage of time. The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge. The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification. As of December 2020, 72 % of PHP websites use discontinued versions of PHP, i.e., PHP 7.2 or lower no longer supported by The PHP Development Team; and large additional fraction is on PHP 7.3 that is by now "supported for critical security issues only." Over 40% of all PHP websites use version 5.6 or older, that not even Debian supported (Debian 9 supported version 7.0 and 7.1).

4.1.6 MySQL

MySQL, the most popular Open-Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

The original website provides the latest information about MySQL software.

MySQL is a database management system:

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

MySQL database are relational:

Open-Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs.

The MySQL database server is very fast, reliable, scalable, and easy to use:

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

MySQL server works in client/ server or embedded systems:

The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

4.1.7 Deployment Diagram:

Figure 4.1 Deployment Diagram

Chapter 5

Testing and Evaluation

5.1 Introduction:

This chapter describes the testing and evaluation of the website developed. Section 6.2 reviews the testing that took place at three levels. Software testing refers to the testing of the produced courseware. It looks at questions such as: whether the web pages are generated correctly, whether all the links work, whether all the audio files exist. Section 6.3 covers the evaluation phase of the system. It reports on the implementation and whether it works properly or not. The system is then evaluated from a software point of view.

5.2 Testing:

This section covers the testing of the template. It deals with concept testing, software testing and the testing of the courseware itself. The concept testing asks the question as whether the developed website properly works.

5.2.1 Software testing:

Software testing is a critical element of software quality assurance and represents the ultimate reuse of specification.

Software testing involves both verification and validation. Verification involves checking that the program conforms to its specification, while validation involves checking that the program as implemented meets the expectations of the user. Static checking techniques include program inspections and analysis. Dynamic techniques (tests) involve exercising the system. Accordingly, the verification stages employ what in the technical lingo are known as the white box testing techniques whilst the validation stage uses black box testing techniques.

The testing process normally has five stages. Firstly, individual units are tested in unit testing. Module testing tests modules (usually a collection of dependent units). Sub-system testing tests collections of modules and often exposes sub-system interface mismatches. System testing tests the system as a whole and finally, there is (user) acceptance testing.

5.2.1.1 Module testing:

Module testing is defined as a software testing type, which checks individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommends testing the smaller building blocks of the program. Module testing is largely a white box oriented.

Module Testing is recommended because:

- Probability of identifying errors or bugs on smaller chunks of program becomes higher.

- Multiple modules can be tested simultaneously and hence supports parallel testing.
- Complexity of testing can be easily managed.

5.2.1.2 Unit Testing:

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

5.2.1.3 Subsystem testing:

Subsystem verification testing is performed as a prelude to system testing. It is performed in the operational environment using installed system hardware and software.

Testing at the subsystem level should be performed:

- When different developers, vendors, or contractors have been responsible for delivering stand-alone subsystems.
- When the complete functionality of a subsystem could not be tested at a lower level because it had not been fully integrated with the necessary communication infrastructure.
- When it was previously impossible to connect to the field devices for the testing phase.

5.2.1.4 System testing:

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications.

Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

5.2.1.5 Acceptance Testing:

Acceptance testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

5.3 Evaluation:

Evaluation is a systematic determination of a subject's merit, worth and significance, using criteria governed by a set of standards. It can assist an organization, program, design, project or any other intervention or initiative to assess any aim, realizable concept/proposal, or any alternative, to help in decision-making; or to ascertain the degree of achievement or value regarding the aim and objectives and results of any such action that has been completed.

Evaluation is the structured interpretation and giving of meaning to predicted or actual impacts of proposals or results. It looks at original objectives, and at what is either predicted or what was accomplished and how it was accomplished.

5.3.1 Purposes:

Purposes Evaluation can be conducted for the purposes of decision making, judgements, conclusion, findings, new knowledge, organizational development, and capacity building in response to the needs of identified stakeholders leading to improvement, decisions about future programming, and/or accountability ultimately informing social action ameliorating social problems and contributing to organizational or social value.

5.3.2 Black Box testing

Black box testing is also known as specification-based testing. Black box testing refers to test activities using specification-based testing methods and criteria to discover program errors based on program requirements and product specifications.

The major testing focuses:

- Specification-based function errors
- Specification-based component/system behavior errors
- Specification-based performance errors
- User-oriented usage errors
- Black box interface errors

5.4 Test Case: Following are the Test Cases for our project Online Tourism Development Management System:

TC1: Registration	
Test Case ID:	UC-001
Wrote By:	Muhammad Shahid
Test Type:	Black box testing
Product Name:	JOBH
Test Item:	Web App
Documented Date:	01/02/2021
Test Suite:	1a
Version Number:	1.0
Test case description:	This test case is designed to successfully sign up
Operation procedure:	<ol style="list-style-type: none"> 1. Go to homepage of project 2. Click on the sign up 3. Clicking on signup will open sign up page which allows user to sign up using email address. 4. Credentials are then sent to server which save them into MySQL 5. Web App redirects to email verification page. 6. After verification Web App redirects to login page.
Pre-conditions:	Internet is required and Web App must be running
Post-conditions:	The credentials are verified and stored in MySQL
Required test scripts:	No

TC2: Login	
Test Case ID:	UC-002
Wrote By:	Muhammad Shahid
Test Type:	Black box testing
Product Name:	JOBH
Test Item:	Web App, MySQL
Documented Date:	01/02/2021
Test Suite:	1a
Version Number:	1.0
Test case description:	This test case is designed to successfully sign in
Operation procedure:	<ol style="list-style-type: none"> 1. Go to homepage of project 2. Click on the login 3. Clicking on login will open login page which allows option to login with Email address. 4. The user is verified, and his credentials is sent to the Web App. 5. Web App redirects to home page which then redirects to user's panel / dashboard
Pre-conditions:	Internet is required and Web App must be running
Post-conditions:	Web App redirects to home page.
Required test scripts:	No

TC3: Post Job	
Test Case ID:	UC-003
Wrote By:	Muhammad Shahid
Test Type:	Black box testing
Product Name:	JOBH
Test Item:	Web App, MySQL
Documented Date:	01/02/2021
Test Suite:	1a
Version Number:	1.0
Test case description:	This test case is designed to successfully for adding rooms.
Operation procedure:	<ol style="list-style-type: none"> 1. Admin will go to his app page. 2. Admin can post job. 3. Admin click on post job from option 4. System display form to fill requirements 5. Admin Fill all requirements.
Pre-conditions:	Internet is required and Web App must be running
Post-conditions:	Job posted successfully
Required test scripts:	No

TC4: Search Job	
Test Case ID:	UC-004
Wrote By:	Muhammad Shahid
Test Type:	Black box testing
Product Name:	JOBH
Test Item:	Web App, MySQL
Documented Date:	01/02/2021
Test Suite:	1a
Version Number:	1.0
Test case description:	This test case is designed to successfully search job
Operation procedure:	<ol style="list-style-type: none">1. The job seeker goes to homepage of project after log in2. Job seeker search job based on position, location, different category, and company3. System shows relevant jobs.
Pre-conditions:	Internet is required and Web App must be running
Post-conditions:	Web App redirects to home page.
Required test scripts:	No

TC5: Apply job	
Test Case ID:	UC-005
Wrote By:	Muhammad Shahid
Test Type:	Black box testing
Product Name:	JOBH
Test Item:	Web App, MySQL
Documented Date:	01/02/2021
Test Suite:	1a
Version Number:	1.0
Test case description:	This test case is designed to apply on job
Operation procedure:	<ol style="list-style-type: none"> 1. System display search job 2. Job seeker clicks apply button to apply on job 3. System redirects to source and show form to fill out information 4. Job seeker fill form and submit 5. Web App redirects to home page which then redirects to user's panel / dashboard
Pre-conditions:	Internet is required and Web App must be running
Post-conditions:	Applied for search job
Required test scripts:	No

TC6: Delete job	
Test Case ID:	UC-006
Wrote By:	Muhammad Shahid
Test Type:	Black box testing
Product Name:	JOBH
Test Item:	Web App, MySQL
Documented Date:	01/02/2021
Test Suite:	1a
Version Number:	1.0
Test case description:	This test case is designed to delete job
Operation procedure:	<ol style="list-style-type: none">1. Admin clicks on delete job from option2. Job seeker clicks apply button to apply on job3. System deleted jobs
Pre-conditions:	Internet is required and Web App must be running
Post-conditions:	Deleted job successfully
Required test scripts:	No

TC7: Logout	
Test Case ID:	UC-007
Wrote By:	Muhammad Shahid
Test Type:	Black box testing
Product Name:	JOBH
Test Item:	Web App, MySQL
Documented Date:	01/02/2021
Test Suite:	1a
Version Number:	1.0
Test case description:	This test case describes the process by which the user can logout his account successfully.
Operation procedure:	<ol style="list-style-type: none">1. User will login his account.2. User can click on logout button for logout account.3. System will return on log in page.
Pre-conditions:	Internet is required and Web App must be running
Post-conditions:	Log out successfully.
Required test scripts:	No

5.5 Problems During Coding:

When I implemented the Web Site, many difficulties were met. I tried to find the problems and solve them through the study.

Firstly, Installation was the very tedious step as I used this for this first time. I installed three to four time to get exact libraries and modules.

Secondly, I have to learn and revised all the basic concepts of HTML, JavaScript, and SQL.

5.6 Enhancement:

In future, I have planned to include more facilities. I shall try to make this website more visible by changing some things and more interestingly by adding some more Things. Like by adding international jobs and scholarships.

References

- [1] Alex Hales, “Best Job Search Portals Review”, techwonderkinds.com <https://techwonderkinds.com/> [accessed Nov.15,2019]
- [2] ALISON DOYLE, “Top 10 Best Websites For Jobs”, thebalancecareers.com <https://www.thebalancecareers.com/top-best-job-websites-2064080> [accessed Nov.15,2019]
- [3] Brianna Johnson, “Rozee.pk: A Review of Pakistan’s Top Job Portal”, pakwired.com <https://pakwired.com/rozee-pk-a-review-of-pakistans-top-job-portal/> [accessed Nov.15,2019]
- [4] Ed Han , “What are the best job search engines in 2019?”, quora.com <https://www.quora.com/What-are-the-best-job-search-engines-in-2019> [accessed Nov.15,2019]
- [5] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE *Recommended Practice* for Software Requirements Specifications”, June 25, 1998.
- [6] Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative development, Third Edition by Craig Layman.
- [7] UML Distilled (Second Edition) A Brief Guide to the Standard Object Modeling Language– Martin Fowler with Kendall Scott.

