JOB SEARCH PORTAL

by

SOWMYA MATHUKUMALLI

B. Tech., SASTRA University, India, 2014

A REPORT

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2016

Approved by:

Major Professor
Dr. Mitchell L. Neilsen

Abstract

JOB SEARCH PORTAL

Finding jobs that best suits the interests and skill set is quite a challenging task for the job seekers. The difficulties arise from not having proper knowledge on the organization's objective, their work culture and current job openings. In addition, finding the right candidate with desired qualifications to fill their current job openings is an important task for the recruiters of any organization. Online Job Search Portals have certainly made job seeking convenient on both sides. Job Portal is the solution where recruiter as well as the job seeker meet aiming at fulfilling their individual requirement. They are the cheapest as well as the fastest source of communication reaching wide range of audience on just a single click irrespective of their geographical distance.

The web application "Job Search Portal" provides an easy and convenient search application for the job seekers to find their desired jobs and for the recruiters to find the right candidate. Job seekers from any background can search for the current job openings. Job seekers can register with the application and update their details and skill set. They can search for available jobs and apply to their desired positions. Android, being open source has already made its mark in the mobile application development. To make things handy, the user functionalities are developed as an Android application. Employer can register with the application and posts their current openings. They can view the Job applicants and can screen them according to the best fit. Users can provide a review about an organization and share their interview experience, which can be viewed by the Employers.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to extend my sincere gratitude to my academic advisor, Dr. Mitchell L Neilsen for his motivation, support and constant guidance throughout the development of this project. I truly appreciate the trust he has placed in me. I take immense pleasure in expressing my sincere gratitude to my committee members, Dr. Daniel Andresen and Dr. Torben Amtoft for their encouragement and valuable feedback on the project. I thank them for serving on my committee.

I would like to acknowledge the support of the academic and technical staff of the Computer Science at K-State for their support throughout my graduate studies.

I would like to thank my parents and family for their unconditional love and support. Special thanks to my sister, Sravya Mathukumalli and my beloved friends for being there for me during my odds. Thank you for your strong belief in me.

Chapter 1 - Project Description

1.1 Introduction

Job Search Portal is a web application, which serves jobseekers to find available job vacancies and Employers to identify eligible job seekers with the prospect of selecting the most qualified candidates. The only way to select best-qualified candidate is to have a pool of eligible applicants, which is possible by drawing the interest of individuals in the market. Job search portals best serve this purpose. E-recruitment has become the standard means for employers and job seekers to meet their respective objectives. The traditional methods for recruitment includes Job fairs, University career employment services, Employee referrals, advertising in the newspapers, televisions etc. With the advancement in technology and growth of internet usage, the e-recruitment has revolutionized the way organizations hire and candidates search for jobs. With the Online Job search portals, the recruitment process is speeded up at every stage from job postings, to receiving applications from candidates, interviewing process. The cost of searching/posting jobs will be much less compared to the traditional way of advertising. Job search portal stands as an effective means for Employers to outline the job vacancies, responsibilities and qualifications to attract jobseekers. Using the portal jobseekers can extensively search for jobs in companies, organizations and regions they may otherwise have not learnt. In addition, candidates/Employers can write a review about an organization, which might help them to change the way things are done.

1.2 Motivation

The purpose of developing an Online Job Search Portal comes from my idea to make the job search efficient and handy. It helps the recruiters as a primary source of talent search. It also helps the job seekers to search for current vacancies at a single point. Therefore, we can say that Online Job Search Portal act as a bridge of communication between organizations and applicants. With the evolution of technology and internet being the main source of information for the applicants, these job portals and have become an excellent method to reach wide range of audience. Initially, when I am unaware of these portals, I used to do research about companies and their technology stack through their respective websites and apply if the job responsibilities matches my interests. This requires lots of effort and time. However, later when I realized the importance of job search portals, I am able to access jobs in companies, locations that I might not otherwise have learned.

Chapter 2 - Background

Based on the Job Search Portal application's requirements, I have made the choice of frameworks to be used. This application primarily consists of 2 main components; a web application that allows recruiters to outline job vacancies with required information for which jobseekers can view and apply according to their interests and an android application that displays the jobs that are available and applied by the applicants. This android application is developed by using the RESTful web services provided by the web application. The Java based web application is developed using Spring MVC and Hibernate to connect to the MySQL database. The front end is developed using HTML, CSS, JavaScript and jQuery. Web services are deployed in the Apache Tomcat server. In addition, the android application is developed using Java, SQLite, and Android studio.

2.1 Spring Framework

The Spring Framework is an application framework and inversion of control container for the Java platform [1]. The Spring Framework stands as a configuration model for developing Java-based enterprise applications.

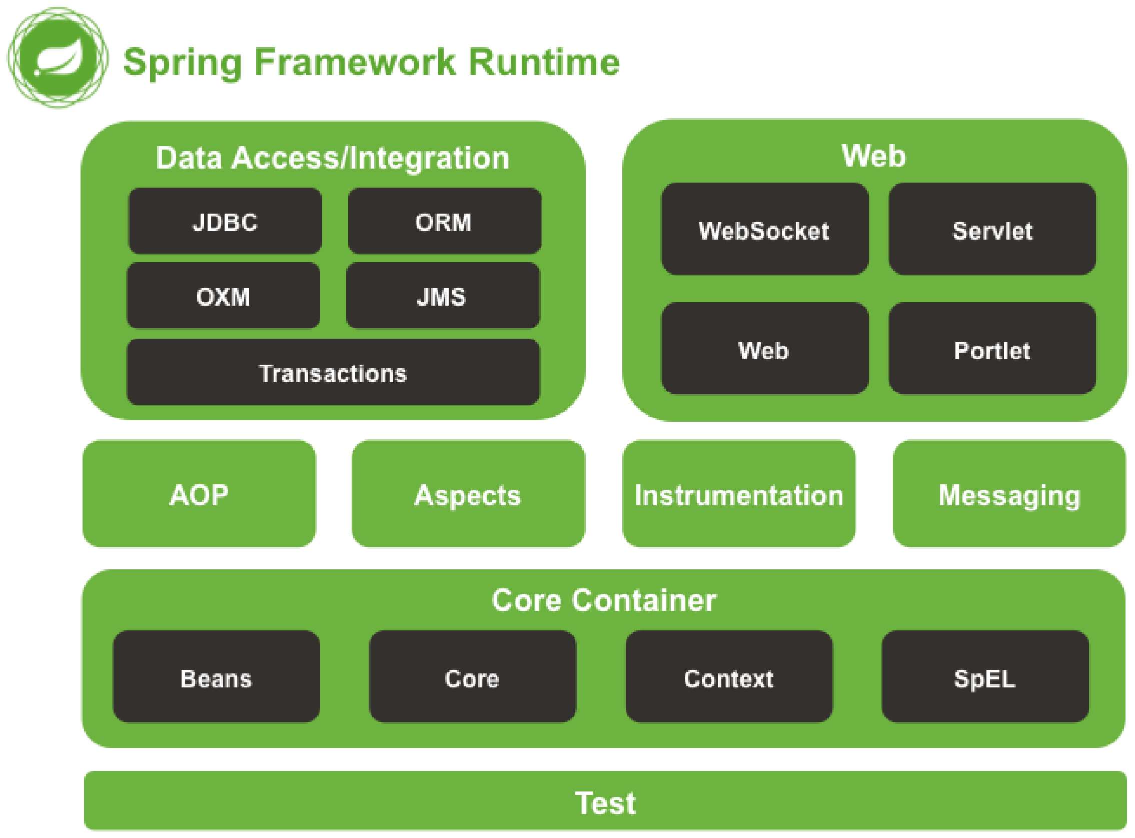


Figure 1: Spring Architecture **[1]**

It provides about 20 modules that can be used based on an application requirement.
In detail,

2.1.1 Core Container

The Core Container consists of Core module that provides important features like IoC and Dependency Injection, Beans that provides BeanFactory, Context that provides ApplicationContext interface and SpEL that provides expression language for querying and manipulating an object graph at runtime.

2.1.2 Data Access/Integration

The Data Access/Integration module consists of JDBC that provides a JDBC-abstraction layer, ORM module that provides integration layers for popular object-relational mapping APIs, including JPA, JDO, Hibernate etc. OXM module that provides an abstraction layer which supports Object/XML mapping implementations for JAXB, Castor, XMLBeans, JiBX, XStream, Java Messaging Service module that contains features for producing and consuming messages and Transactions that supports programmatic and declarative transaction management for classes that implement special interfaces and for all your POJOs.

2.1.3 WEB

The Web layer consists of Web module that provides basic web-oriented integration features and the initialization of the IoC container using servlet listeners and a web-oriented application context, the Web-MVC module that contains Spring's model-view-controller (MVC) implementation, the Web-Socket module provides support for two-way communication between client and server in web applications and Web-Portlet module which provides the MVC implementation to be used in a portlet environment and mirrors the functionality of Web-Servlet module.

2.2. Hibernate ORM

ORM (Object Relational Model) is a technique mapping data between an object-oriented model to a relational data model. Hibernate is an ORM framework for Java language. Its primary feature is to map from Java classes to database tables and from Java data types to SQL data types [2]. It also provides data query and retrieval facilities. This reduces the development time spent with writing the native SQL queries and lets developers develop persistent classes using object-oriented principles.

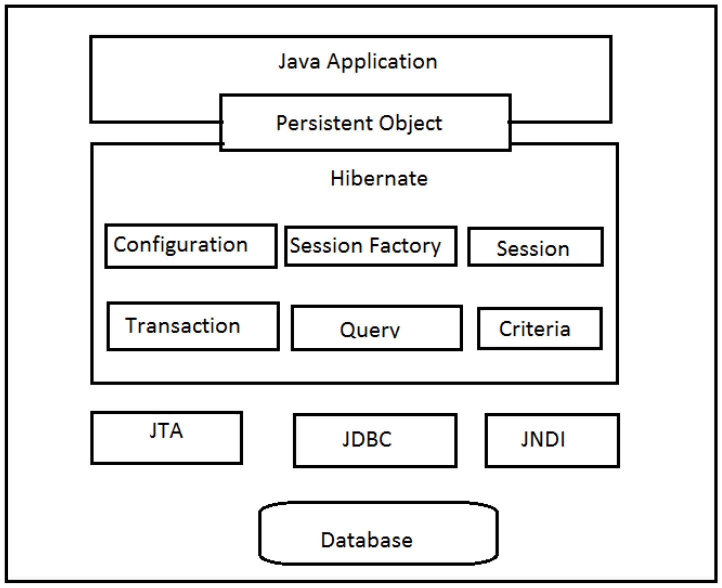


Figure 2: Hibernate ORM Architecture **[3]**

Hibernate uses Java APIs like JDBC, Java Transaction API (JTA), and Java Naming and Directory Interface (JNDI).

Elements of Hibernate Architecture includes,

Configuration Object - creates the connection between the Java classes and database tables. *SessionFactory Object* - configures Hibernate for the application and is created at the start of the application that is shared among multiple threads. It is the factory for session objects.

Session Object - used to get a physical connection with a database. The Session object allows the persistent objects to be saved and retrieved.

Transaction Object - allows the application to define a transaction that is units of work, while maintaining abstraction from the underlying transaction implementation.

Query Object - Query Object uses object-oriented query language i.e., HQL (Hibernate Query Language) for creating objects and retrieving data from the database.

Criteria Object – Criteria is used to create object- oriented criteria queries for retrieving objects and controls the flow of execution.

2.3 Spring MVC

Spring MVC Framework is a model-view-controller framework for developing flexible and loosely coupled Java based web applications. It provides input logic, business logic and UI logic with a little dependency between them.

- The Model encapsulates the application data and in general, they will consist of POJO.
- The View renders the model data and generates HTML output to be viewed by the client. This application uses JavaScript, HTML, CSS and jQuery as front-end development technologies.
- The Controller processes the user requests and controls the flow of application logic between view and model.

2.4 MySQL

This application uses MySQL as a back end database. MySQL is a fast, open source Relational database management system for developing web-based applications. Since it a relational based database, data is stored in the form of tables and relations are established between tables using primary keys, foreign keys.

2.5 Tomcat Web Server

Tomcat is an open source Java Servlet Container by Apache Software Foundation for implementing Java Servlet, Java Servlet Pages (JSP). The web application is deployed into the Tomcat server. Using any web browser, we can run the web application that is deployed into the Tomcat Server.

2.6 Android

Android is an open source Linux based operating system. It is used to create various mobile applications on various smart phones or tablets available and to customize the device in many ways. It provides various components for an optimal application development and execution environment for mobile devices.

Each layer in the Android OS shown above provides various functionalities to the layers above it.

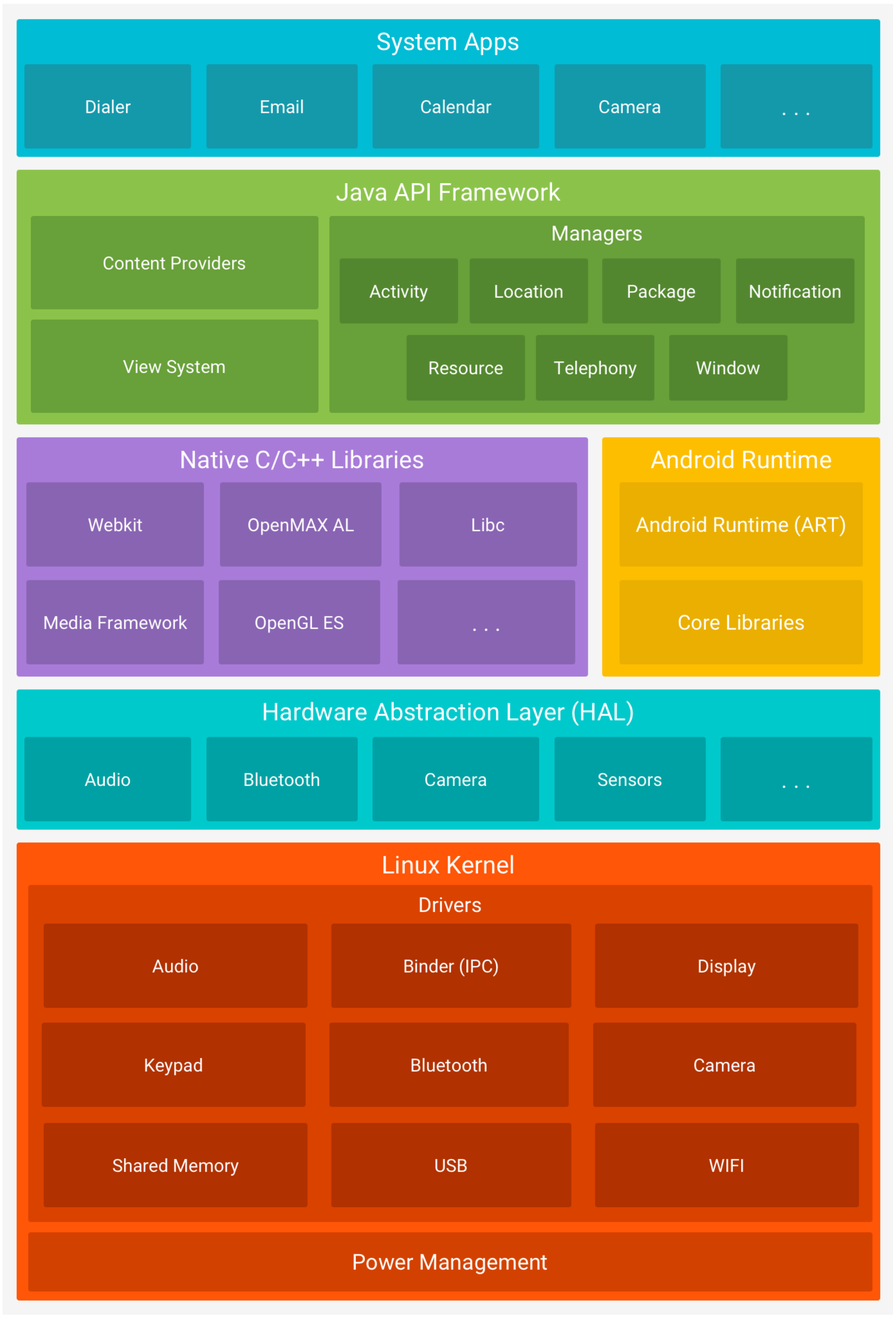


Figure 3: Android Architecture [9]

https://developer.android.com/guide/platform/index.html

In detail about each layer,

2.6.1 Linux Kernel

The Linux Kernel is the bottom layer of the Android software stack that provides a level of abstraction between the device hardware and the upper layers of the software stack [4]. The functionalities of the kernel include memory, process and power management. It also provides various device drivers for display, Wi-Fi and audio.

2.6.2 Android Libraries

The Libraries layer provides a set of Java-based libraries that facilitates interface design, graphics drawing and database access to list a few and C/C++ libraries for 2D and 3D graphics drawing, SSL communication, graphic layer and SQLite database management etc.,

2.6.3 Android Runtime

Replacing Dalvik, which is a runtime, Virtual Machine used by the Android Operating System for running Android application, the Android Runtime (ART). ART performs the translation of the application's bytecode into native instructions that are executed by the device's runtime environment [5]. The ART has two main features, Ahead-of-Time (AOT) compilation and improved Garbage Collection (GC).

2.6.4 Application Framework

The Application Framework layer provides various services to the applications. The services include, Activity Manager that controls the activity stack, Content Providers that allows sharing of data with other applications, Resource Manager that provides access to color settings and user interface layouts, Notifications Manager that takes care of the notification settings and View System to create user interfaces for the applications.

2.6.5 Applications

Applications form the top layer of the software stack in the Android Architecture. As a developer, you will write and install your application in this layer. This layer contains native as well as the third party applications installed by the user.

Chapter 3 - Related Work

3.1 Existing System

The existing system for job recruitment includes traditional methods like Employment agencies, advertising through newspapers, televisions and radios, college fairs etc., which are too slow and stressful. With the advancement of internet, jobseekers rely on the online job portals, which makes the job search efficient. Again, most of these are limited to the web/desktop applications, which requires jobseekers to have a laptop or desktop connected to internet and is not handy.

3.1.1 Disadvantages

- Time Consuming
- Stressful
- Challenging

3.2 Proposed System

Job Search Portal is a Java-based web application as well as Android application that provides functionalities of e-recruitment on desktop and on portable devices like Android based smart phones/tablets. Both applications do not require internet to perform the desired functionalities.

3.2.1 Advantages

- Cost and Time efficient
- Portable

3.2.2 Purpose of the System

Job Search Portal is developed to provide an effective means for the employers to post job openings with required qualification to have a better penetration into the job market and jobseekers to find out the information regarding the current openings in the organization. In addition, Employers can view the reviews provided by the applicants to make necessary improvements in their system if needed. Job search portal is both web based as well as an android application providing flexibility for the users.

3.2.3 Objective

The objective of the web as well as android application is to provide flexibility to the jobseekers by providing the functionalities of both job search and job application in a single application. In addition, this application provides an effective means for the employers to post job vacancies and view the job applications by the interested applicants in a single application. Employers can also view the reviews provided by the jobseekers.

Chapter 4 - Requirement Analysis

4.1 Requirement Gathering

Requirement Analysis is the first and important step in the Software development activity for building robust and user-friendly applications. I have started working on determining the functionalities that the application should provide. I have done a good amount of research on existing systems and the disadvantages of those. Once the functional requirements are finalized, I did research on the current technologies that are widely used in the industry and decided to use Spring MVC, hibernate replacing the traditional way of developing web applications using Struts, Servlets and JDBC. After a meeting with my Professor Dr. Mitchell L Neilsen, we decided to develop an android application as well to provide more flexibility to the users.

4.2 Requirement Specifications

Below are the technical requirements to develop Job Search Portal application

4.2.1 Software Requirements

- Operating System: Windows 10
- IDE: Eclipse, Android Studio IDE
- Application Server: Apache Tomcat 8.5.6
- Frameworks and APIs: Spring MVC, Hibernate, Android SDK Framework 10 or higher
- Database: MySQL, SQLite Database
- Front End: HTML5, CSS3, JavaScript, jQuery
- Web Service: RESTful web services
- Browser: Chrome or Firefox or Internet Explorer
- Emulator: SDK version 3.0 or higher

4.2.2 Hardware Requirements

- Processor: Intel core i7
- Processor speed: 3.40 GHz
- RAM: 8 GB

Chapter 5 - System Design

5.1 System Design

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements [6]. System designs are modeled using UML which is a standard object-oriented analysis and design language. The UML is a collection of diagrams and standard set of notations for specifying and visualizing various aspects such as requirements and design of software systems.

5.1.1 Use Case Diagram

A diagram is a visualization of set of elements and the relationships between them. Use case is a set of scenarios, which defines functionalities of the system from a user's perspective. The main components of a use case diagram include actors, use cases and their relationships. They depict the interaction between actors and system to achieve certain goal. This, a use case diagram is important in modelling the behavior of a system.

Use Case Diagram for Job Search Portal

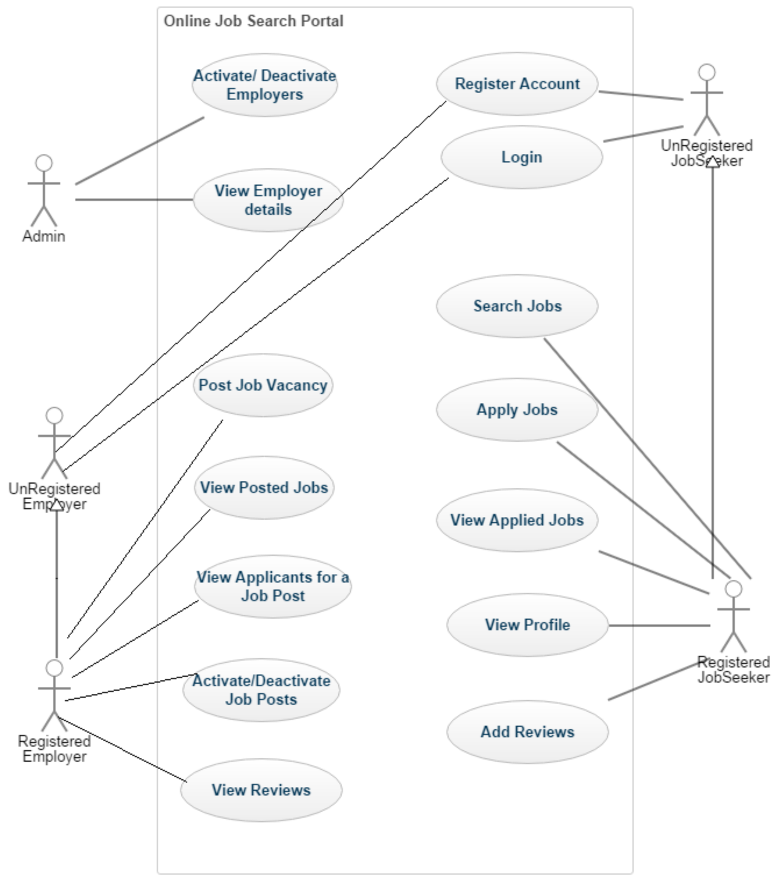


Figure 4: Use Case Diagram

Actors:

The Actors of the system are Admin, Unregistered Employer, Register Employer, Unregistered Jobseeker and Registered Jobseeker.

Use cases:

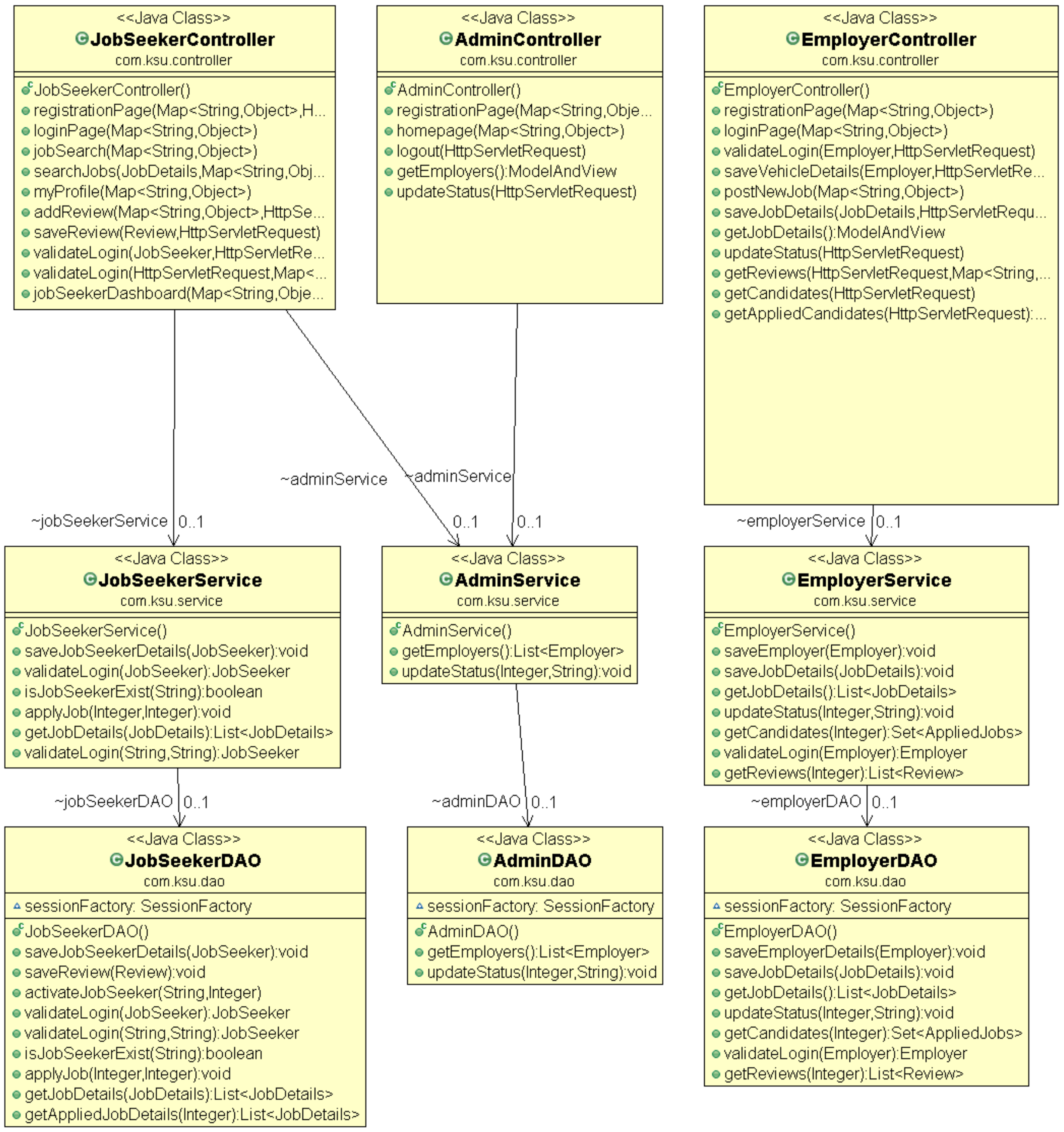
I have identified a set of use cases based on the functionalities and goals of the application.

- **Register Account-** This use case denotes a set of actions required for Employer and Job seeker to register with the application.
- **Login-** This use case denotes a set of actions required for Employer and Job seeker to login into the application.

- **Activate/De-activate Account-** This use case denotes a set of actions required for admin to activate or de-activate the Employers.
- **View Employers-** This use case denotes a set of actions required for admin to view the Employers that are registered with the application.
- **Add Job Vacancy-** This use case denotes a set of actions required for Employer to post a job vacancy.
- **Activate/De-activate Job Post-** This use case denotes a set of actions required for Employer to change the status of the Job Post.
- **View Applicants for a Job Post-** This use case denotes a set of actions required for Employer to view the list of applicants for a particular job post.
- **View Reviews-** This use case denotes a set of actions required for Employer to view Reviews provided by the applicants.
- **View Job Posts-** This use case denotes a set of actions required for Employer to view all the jobs posted by the Employer.
- **Search Job Posts-** This use case denotes a set of actions required for Job Seeker to search available and active jobs.
- **Apply for Job-** This use case denotes a set of actions required for Job Seeker to apply for an available job vacancy.
- **Add Reviews-** This use case denotes a set of actions required for Job Seeker to add Reviews for an organization that can be viewed by the Employer.

5.1.2 Class Diagram

Class diagram is a graphical representation of the static view of the system. It describes the design and structure of the system by displaying the system’s classes, their attributes, methods and relationships among objects.



<<Java Class>>
JobSeekerController
com.ksu.controller
JobSeekerController()
registrationPage(Map<String,Object>,H...
loginPage(Map<String,Object>)
jobSearch(Map<String,Object>)
searchJobs(JobDetails,Map<String,Obj...
myProfile(Map<String,Object>)
addReview(Map<String,Object>,HttpSe...
saveReview(Review,HttpServletRequest)
validateLogin(JobSeeker,HttpServletRe...
validateLogin(HttpServletRequest,Map<...
jobSeekerDashboard(Map<String,Obje...
<<Java Class>>
AdminController
com.ksu.controller
AdminController()
registrationPage(Map<String,Obje...
homepage(Map<String,Object>)
logout(HttpServletRequest)
getEmployers():ModelAndView
updateStatus(HttpServletRequest)
<<Java Class>>
EmployerController
com.ksu.controller
EmployerController()
registrationPage(Map<String,Object>)
loginPage(Map<String,Object>)
validateLogin(Employer,HttpServletRequest)
saveVehicleDetails(Employer,HttpServletRe...
postNewJob(Map<String,Object>)
saveJobDetails(JobDetails,HttpServletRequ...
getJobDetails():ModelAndView
updateStatus(HttpServletRequest)
getReviews(HttpServletRequest,Map<String,...
getCandidates(HttpServletRequest)
getAppliedCandidates(HttpServletRequest):...
~adminService
~adminService
~jobSeekerService 0..1
0..1
0..1
~employerService 0..1
<<Java Class>>
JobSeekerService
com.ksu.service
JobSeekerService()
saveJobSeekerDetails(JobSeeker):void
validateLogin(JobSeeker):JobSeeker
isJobSeekerExist(String):boolean
applyJob(Integer,Integer):void
getJobDetails(JobDetails):List<JobDetails>
validateLogin(String,String):JobSeeker
<<Java Class>>
AdminService
com.ksu.service
AdminService()
getEmployers():List<Employer>
updateStatus(Integer,String):void
<<Java Class>>
EmployerService
com.ksu.service
EmployerService()
saveEmployer(Employer):void
saveJobDetails(JobDetails):void
getJobDetails():List<JobDetails>
updateStatus(Integer,String):void
getCandidates(Integer):Set<AppliedJobs>
validateLogin(Employer):Employer
getReviews(Integer):List<Review>
~jobSeekerDAO 0..1
~adminDAO 0..1
~employerDAO 0..1
<<Java Class>>
JobSeekerDAO
com.ksu.dao
sessionFactory: SessionFactory
JobSeekerDAO()
saveJobSeekerDetails(JobSeeker):void
saveReview(Review):void
activateJobSeeker(String,Integer)
validateLogin(JobSeeker):JobSeeker
validateLogin(String,String):JobSeeker
isJobSeekerExist(String):boolean
applyJob(Integer,Integer):void
getJobDetails(JobDetails):List<JobDetails>
getAppliedJobDetails(Integer):List<JobDetails>
<<Java Class>>
AdminDAO
com.ksu.dao
sessionFactory: SessionFactory
AdminDAO()
getEmployers():List<Employer>
updateStatus(Integer,String):void
<<Java Class>>
EmployerDAO
com.ksu.dao
sessionFactory: SessionFactory
EmployerDAO()
saveEmployerDetails(Employer):void
saveJobDetails(JobDetails):void
getJobDetails():List<JobDetails>
updateStatus(Integer,String):void
getCandidates(Integer):Set<AppliedJobs>
validateLogin(Employer):Employer
getReviews(Integer):List<Review>

Figure 5: Class Diagram

5.1.2.1 Controller classes: The Project contains three controller classes i.e. AdminController, JobSeekerController and EmployerController. These classes are responsible for handling HTTP requests and returns HTTP response.

5.1.2.2 Service classes: The Project contains three Service classes i.e. AdminService, JobSeekerService and EmployerService. The Controller classes pulls data from the request and passes it to the appropriate service class. The Service classes are responsible for called or more DAO class.

5.1.2.3 DAO classes: The Project contains three Service classes i.e. AdminDAO, JobSeekerDAO and EmployerDAO. The DAO classes contains the query code and directly interacts with the model classes. The DAO classes send back model classes to the Controller class in order to be sent to the view layer.

Chapter 6 - Database Design

The database that is used to design the web application is MySQL. MySQL workbench is used to create tables and run queries. In this application development, we have used MySQL to store employer details, jobseeker details, applied jobs by the applicants, jobs posted by the employer. Hence, we have identified five tables to achieve desired functionality.

i. Employer table: holds details of Employer
ii. Jobseeker table: holds details of applicant
iii. Applied_Jobs table: holds details of jobs applied by the job seeker
iv. Posted_Jobs table: holds details of jobs posted by the Employer
v. Reviews table: holds the reviews for interview, salary, work life provided by the jobseeker

When employer registers with the application, the application inserts the details of the employer into the Employer table. Similarly, when a jobseeker creates an account, his/her details will be inserted into the Jobseeker table. When jobseeker searches for the available job vacancies, the application queries the database to retrieve the job vacancies that are posted by the employer from the Posted_Jobs table. Similarly, when an employer wishes to view the applicants for a particular job posting, the application queries the database to retrieve the details of the job and job seeker from the Applied_Jobs table. In addition, the employer can activate or deactivate the job status thus updating the database. The jobseeker can provide reviews about an organization and will be saved in the Reviews table.

Chapter 7 - Implementation

The Online Job Search Portal is a web-based and android application, which revolutionizes the way companies hire the candidates and jobseekers search for job vacancies. The employers can view reviews given by the jobseekers and make improvements in their system accordingly. The application provides a flexible and easy to use environment on desktops as well as portable devices like smart phones/tablets for the users to achieve their respective objective

The modules that I have implemented in the Job Search Portal are as listed.

- Admin
- Employer
- Jobseeker

7.1 Admin

Spring Security provides the Admin login. The Admin module provides various functionalities. The Admin users are responsible for activating and deactivating the employer accounts. In addition, Admin users can view the list of employers registered with the application.

7.2 Employer

Employer users will be able to perform functions such as registering with the application and creating an account by providing the details of Employer Name, Employer Code, Address, Company E-mail, Mobile Number, Login Name, and Password that are stored in the Employer table of MySQL database. Once the account is activated, this module allows employers to post jobs summarizing responsibilities and expected skills that will be saved in the Posted_Jobs table of MySQL database. The employer will also be given privilege to activate or deactivate jobs. He/she can view the list of job postings that are active. He/she can also view the applicant details that have applied for a particular job posting. The employer will be able to view reviews provided by the jobseeker.

7.3 Jobseeker

The Jobseeker users will be able to perform functions such as registering with the application and creating an account by providing the details of First Name, Last Name, E-mail, Password, Mobile Number, Primary Skill and Experience that are stored in the Jobseeker table of MySQL database.

Once the account is activated, jobseekers can search, view and apply for active job openings. All the applied jobs details are stored in the Applied_Jobs table of MySQL database. The applicants can also write/update reviews for the companies.

The Online Job Search web application is developed on Eclipse IDE using Java 1.8 and Spring MVC framework. I have used annotation configuration for spring replacing XML configurations. The application uses Hibernate framework for mapping object-oriented domain model to a relational database. The backend database for the web application is MySQL and I have used MySQL workbench and MySQL query browser for managing the database. The web application is deployed in the tomcat server. An android application is developed for the jobseeker module using Android Studio IDE, Android SDK framework and Java using the RESTful web services provided by the web application.

Chapter 8 - Graphical User Interface

The Graphical User Interface are developed using HTML5, CSS, JavaScript, jQuery and Bootstrap to provide an easy to understand interactive screens. Various screens and the navigation between screens are discussed below.

Web application screens:

8.1 Home Page

The home page appears on the start of the application. The screen will provide various functionalities like Home, Admin, Employer Registration, Employer Login, Jobseeker Registration and Jobseeker Login. When clicked on the button, it navigates to the respective screens.

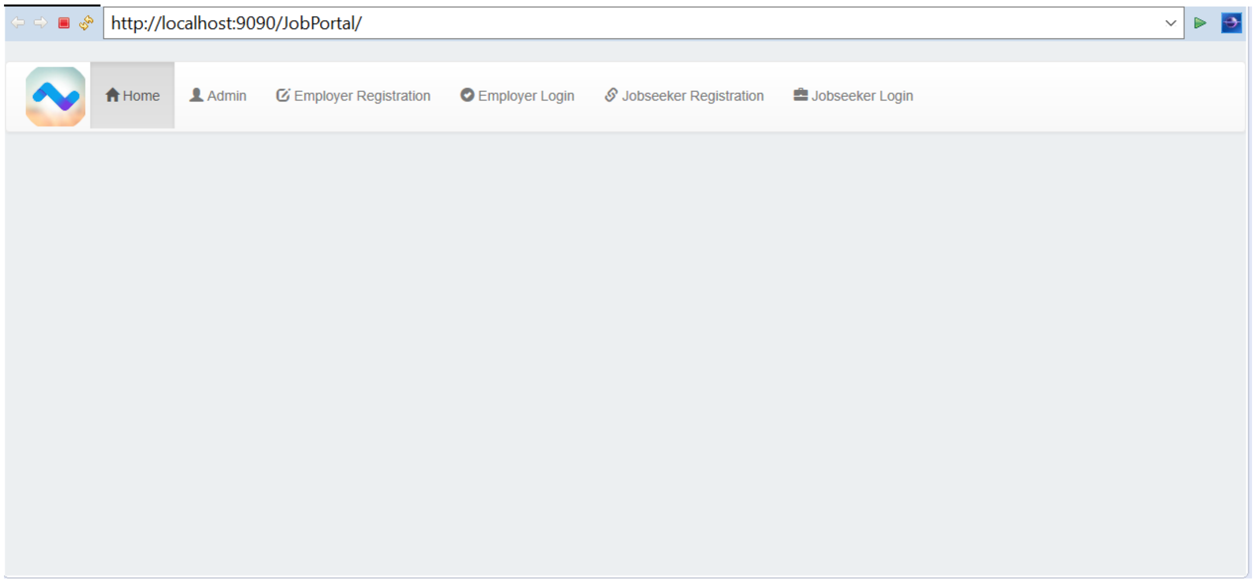
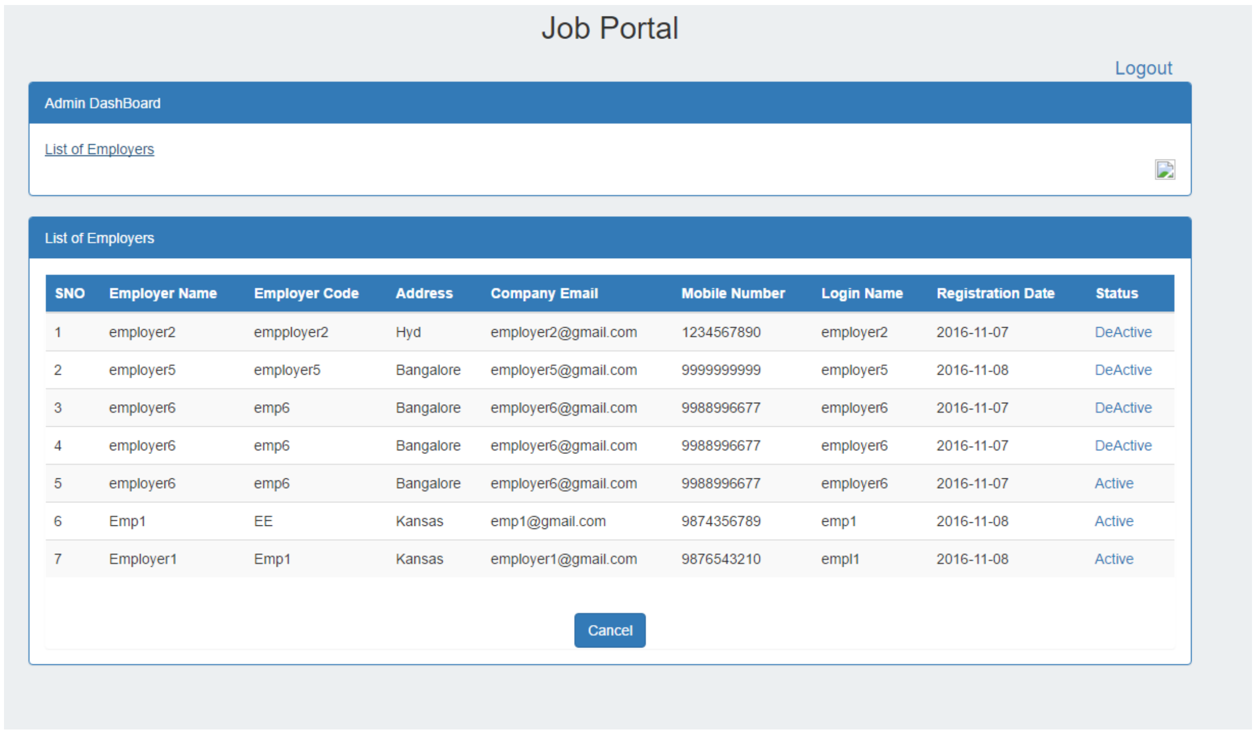


Figure 6: Home Screen

8.2 Admin

After successful login, the admin navigates to the Admin dashboard page where he/she can access the details of the List of Employers registered with the application. He can also activate or deactivate Employers.


Job Portal

Logout

Admin DashBoard

List of Employers

List of Employers

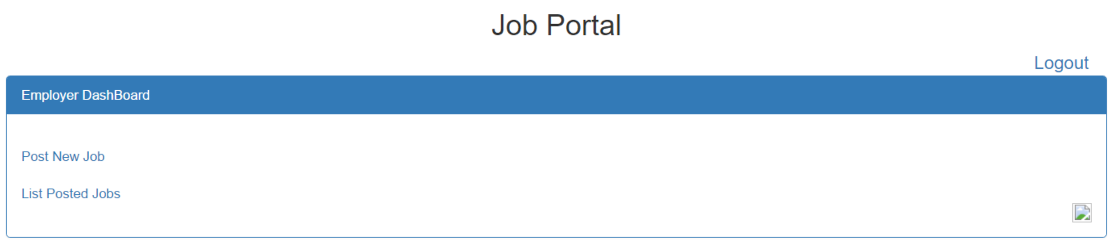
SNO	Employer Name	Employer Code	Address	Company Email	Mobile Number	Login Name	Registration Date	Status
1	employer2	empployer2	Hyd	employer2@gmail.com	1234567890	employer2	2016-11-07	DeActive
2	employer5	employer5	Bangalore	employer5@gmail.com	9999999999	employer5	2016-11-08	DeActive
3	employer6	emp6	Bangalore	employer6@gmail.com	9988996677	employer6	2016-11-07	DeActive
4	employer6	emp6	Bangalore	employer6@gmail.com	9988996677	employer6	2016-11-07	DeActive
5	employer6	emp6	Bangalore	employer6@gmail.com	9988996677	employer6	2016-11-07	Active
6	Emp1	EE	Kansas	emp1@gmail.com	9874356789	emp1	2016-11-08	Active
7	Employer1	Emp1	Kansas	employer1@gmail.com	9876543210	empl1	2016-11-08	Active

Cancel

Figure 7: Admin Dashboard

8.3 Employer Dashboard

On successful registration and login, employer navigates to the Employer dashboard page where he can post a job or view the list of jobs posted.


Job Portal

Logout

Employer DashBoard

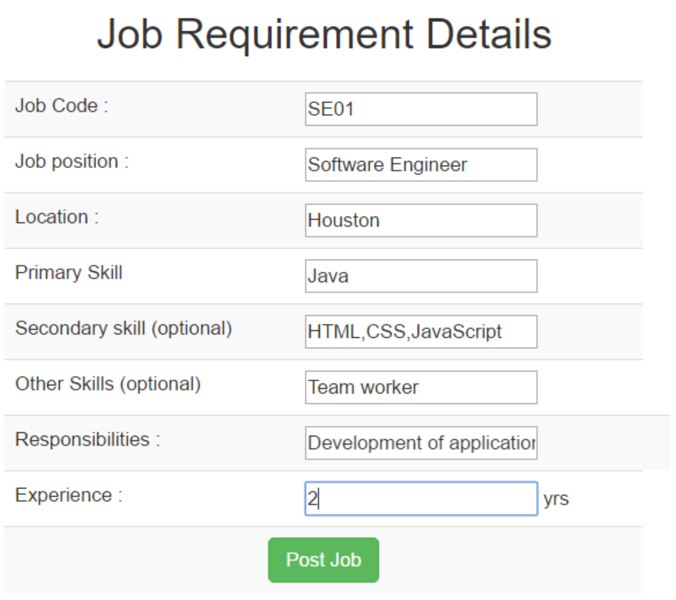
Post New Job

List Posted Jobs

Figure 8: Employer Dashboard

8.4 Post New Job

On clicking the Post New Job tab on the Employer page, the employer navigates to the Post New Job page where he posts current job opening with summary of qualifications and responsibilities.

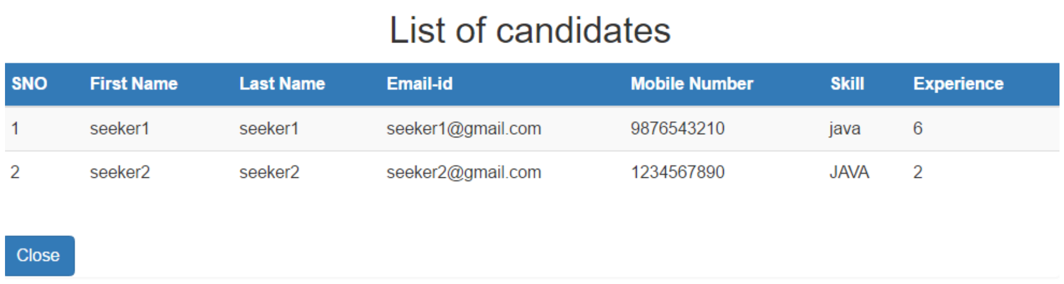

Job Requirement Details

Job Code :	SE01
Job position :	Software Engineer
Location :	Houston
Primary Skill	Java
Secondary skill (optional)	HTML,CSS,JavaScript
Other Skills (optional)	Team worker
Responsibilities :	Development of applicatior
Experience :	2 yrs

Figure 9: Post New Job

8.5 List Applied Candidates

The Employer might choose to list posted jobs and view the list of candidates applied for a particular job posting.

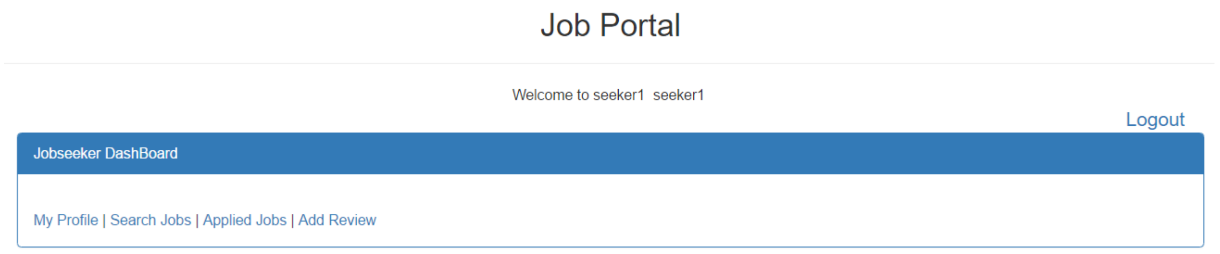

List of candidates

SNO	First Name	Last Name	Email-id	Mobile Number	Skill	Experience
1	seeker1	seeker1	seeker1@gmail.com	9876543210	java	6
2	seeker2	seeker2	seeker2@gmail.com	1234567890	JAVA	2

Figure 10: Applied Candidates

8.6 Jobseeker Dashboard

After successful registration and login, jobseeker navigates to Job Portal Page facilitating the functionalities of viewing his/her profile, Search Jobs, View applied jobs and add reviews.


Job Portal

Welcome to seeker1 seeker1

Logout

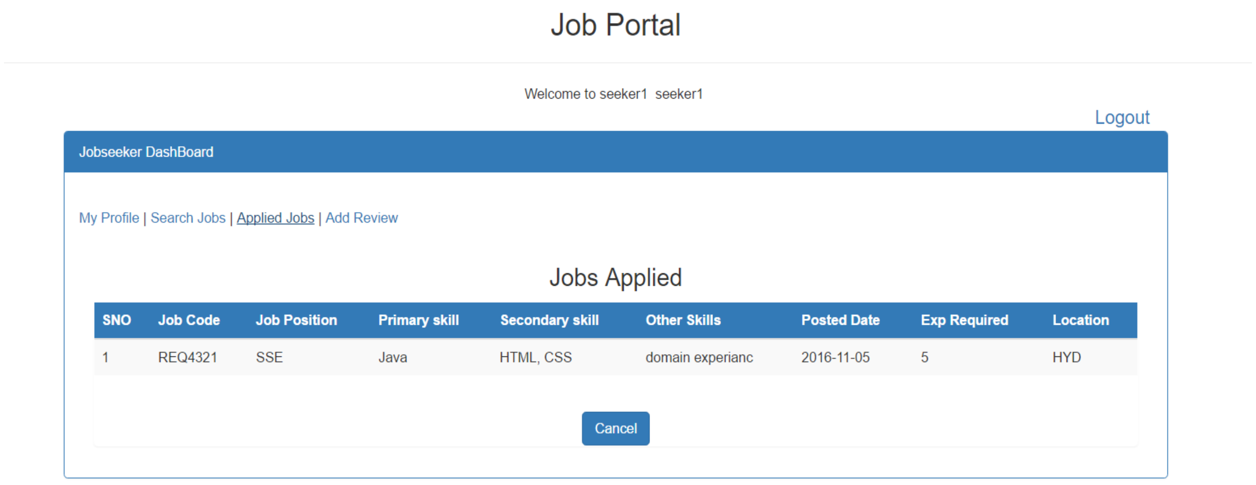
Jobseeker DashBoard

My Profile | Search Jobs | Applied Jobs | Add Review

Figure 11: Job Seeker dashboard

8.6.1 Applied Jobs

On clicking the applied jobs tab, the jobseeker will be navigated to the Applied Jobs page which displays the details of the jobs applied.


Job Portal

Welcome to seeker1 seeker1

Logout

Jobseeker DashBoard

My Profile | Search Jobs | Applied Jobs | Add Review

Jobs Applied

SNO	Job Code	Job Position	Primary skill	Secondary skill	Other Skills	Posted Date	Exp Required	Location
1	REQ4321	SSE	Java	HTML, CSS	domain experianc	2016-11-05	5	HYD

Cancel

Figure 12: Applied Jobs

8.6.2 Reviews

The job seeker can also provide reviews including the interview, salary, work balance ratings.

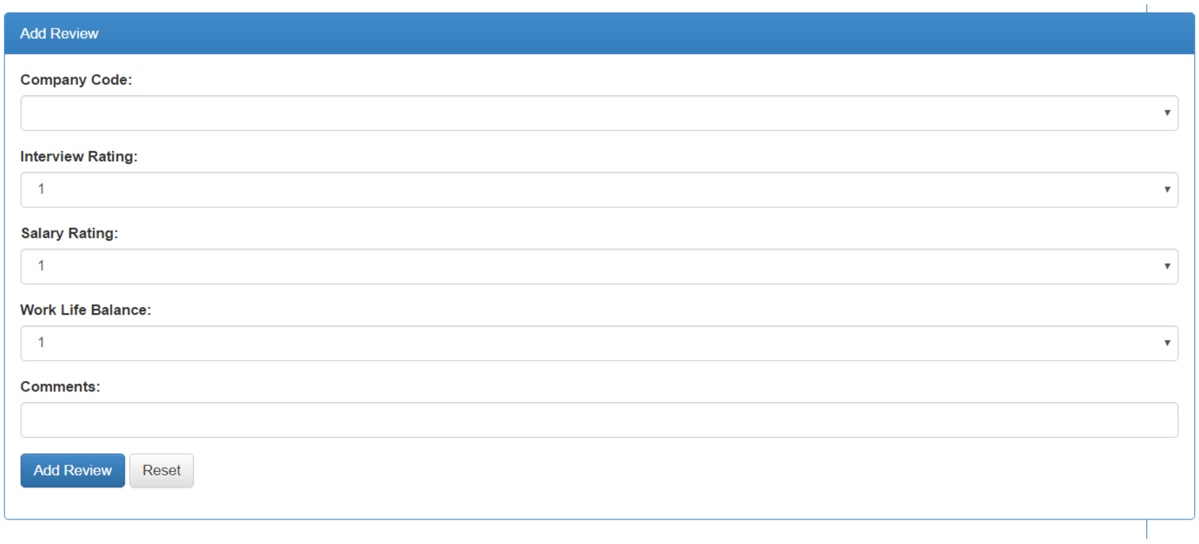
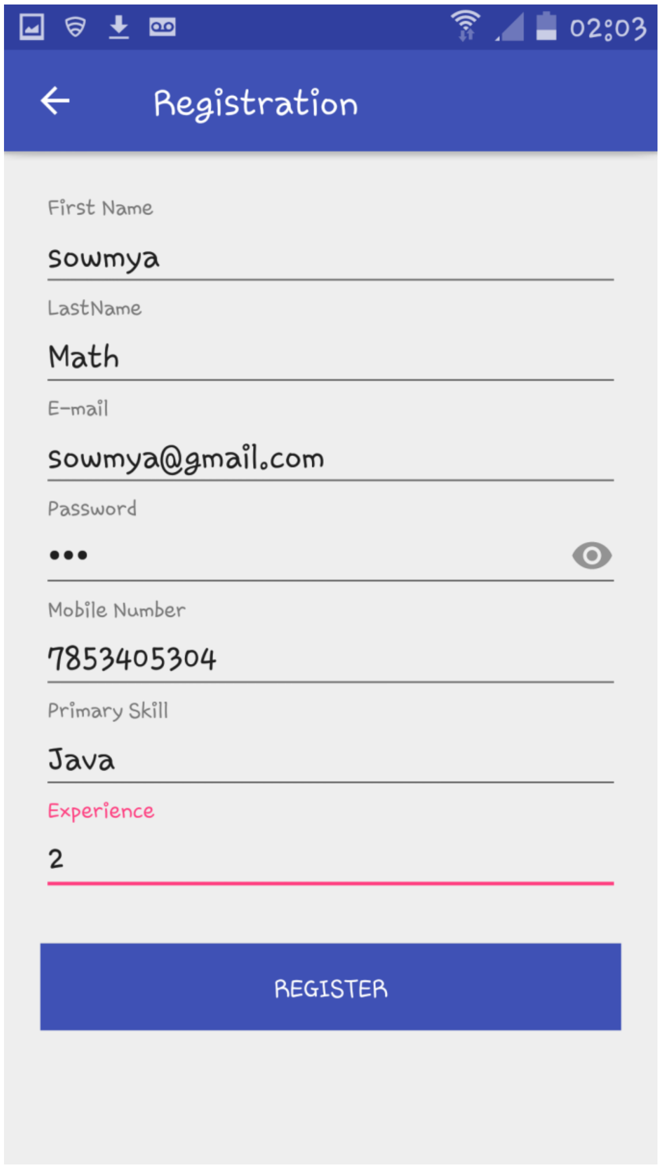


Figure 13: Add Review

Android application screens for jobseeker

8.7 Login/Registration Screen

The Job seeker can login or sign up with the application.



First Name
sowmya

LastName
Math

E-mail
sowmya@gmail.com

Password
•••

Mobile Number
7853405304

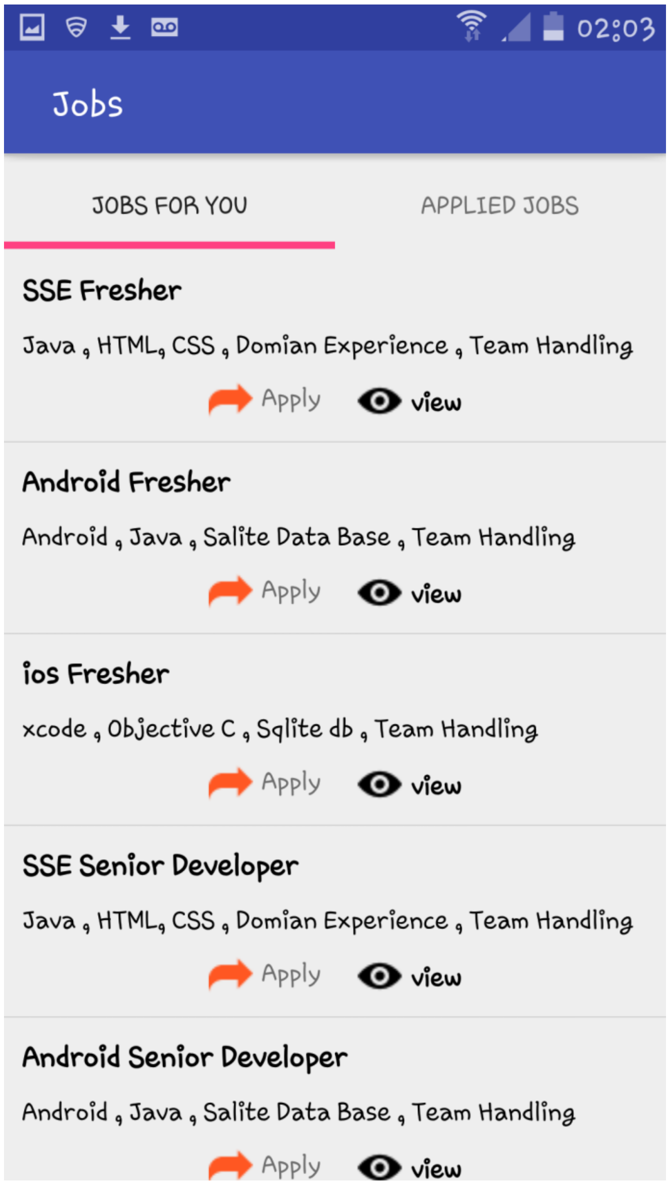
Primary Skill
Java

Experience
2

Figure 14: Login/Registration Page

8.8. Jobs Screen

The Jobs Screen provides the list of available jobs and also the list of jobs applied. One can view the listed jobs and apply if interested.


02:03

Jobs

JOBS FOR YOU | APPLIED JOBS

SSE Fresher

Java , HTML, CSS , Domian Experience , Team Handling

Apply view

Android Fresher

Android , Java , Salite Data Base , Team Handling

Apply view

ios Fresher

xcode , Objective C , Sqlite db , Team Handling

Apply view

SSE Senior Developer

Java , HTML, CSS , Domian Experience , Team Handling

Apply view

Android Senior Developer

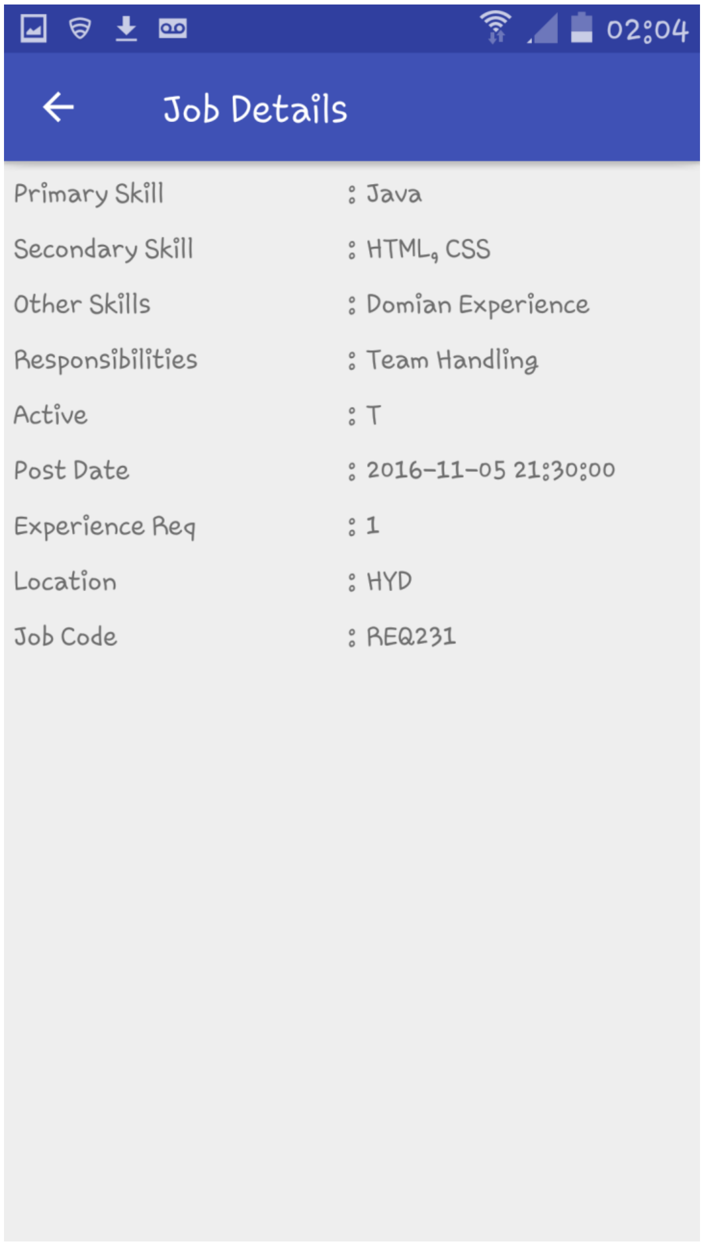
Android , Java , Salite Data Base , Team Handling

Apply view

Figure 15: Jobs Page

8.9 View Jobs Screen

After choosing to view the job posting in the Jobs screen, the details of the jobs will be displayed.



02:04

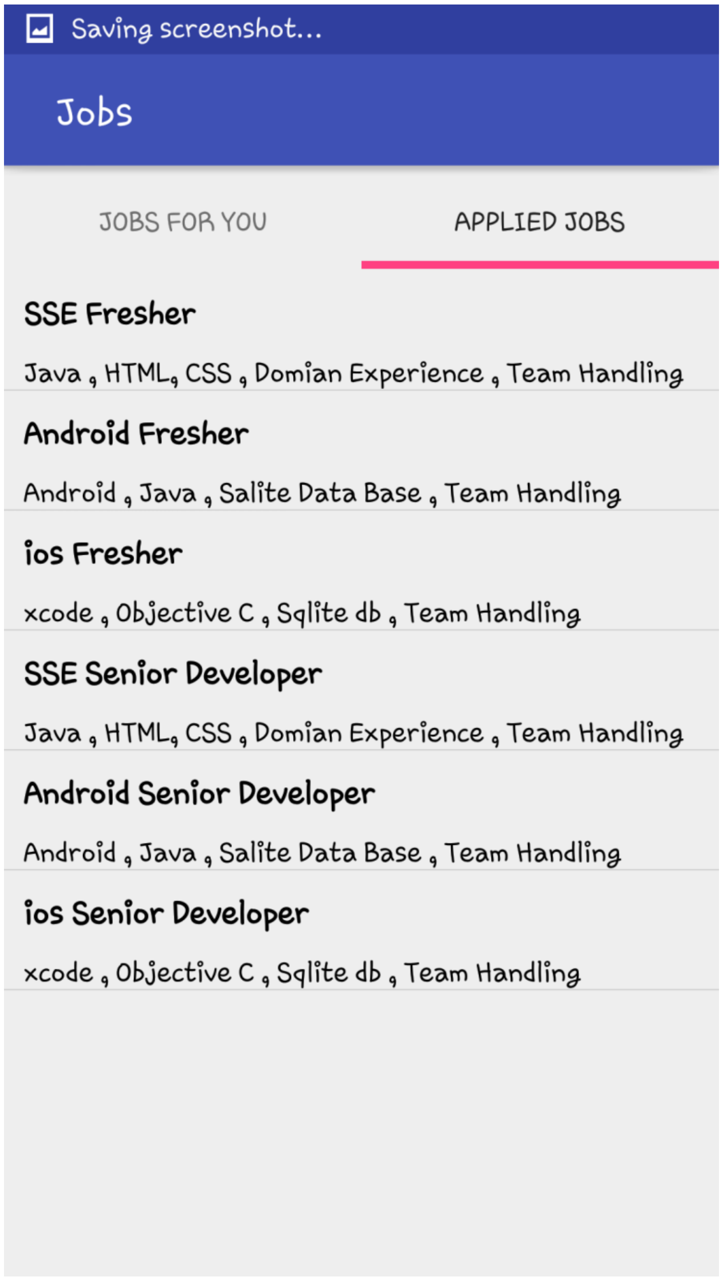
Job Details

Primary Skill	: Java
Secondary Skill	: HTML, CSS
Other Skills	: Domian Experience
Responsibilities	: Team Handling
Active	: T
Post Date	: 2016-11-05 21:30:00
Experience Req	: 1
Location	: HYD
Job Code	: REQ231

Figure 16: Job Details

8.10 Applied Jobs Screen

Once can view the list of applied jobs navigating from the Jobs screen.



SSE Fresher

Java , HTML, CSS , Domian Experience , Team Handling

Android Fresher

Android , Java , Salite Data Base , Team Handling

ios Fresher

xcode , Objective C , Sqlite db , Team Handling

SSE Senior Developer

Java , HTML, CSS , Domian Experience , Team Handling

Android Senior Developer

Android , Java , Salite Data Base , Team Handling

ios Senior Developer

xcode , Objective C , Sqlite db , Team Handling

Figure 17: Applied Jobs

Chapter 9 - Testing

Testing is a process of executing a program with the intent of finding bugs that makes the application fail to meet the expected behavior. Regardless of the development methodology, the ultimate goal of testing is to make sure that what is created does what it is supposed to do. Testing plays a critical role for assuring quality and reliability of the software. I have included testing as a part of development process. The test cases should be designed with maximum possibilities of finding the errors or bugs. Various level of testing are as follows.

9.1 Testing Levels

- ***Unit testing***: Unit testing tests the functionality of individual units of source code. It is the smallest component of a testable software that works in isolation with other parts of the code. I have done unit testing for various individual components of the source code to uncover errors within the boundary of the application.
- ***Integration testing***: Integration testing focuses on the design and construction of the software. Here the individual components that are tested using unit tests are combined and tested as a group. Its primary purpose is to expose the defects associated with the interfacing of modules. It checks if the modules perform the desired functionality when integrated together.
- ***System testing***: System testing is performed on a completely integrated system to see if it meets the requirements.
- ***Regression testing***: Regression testing aims at verifying the functionality of the software that is previously tested and to which changes are made. It is to ensure the old software still works with new changes.
- ***Acceptance testing***: Acceptance testing is conducted to verify if the system compliance the business requirements.

Adhering to the levels of testing, Unit testing is performed on individual components of the system ensuring the expected behavior. Later, I have integrated various components together and performed Integration testing. Once the integration testing is done, I have performed System testing and ensured the application works as per the requirements. Finally, acceptance testing is performed to check if the client accepts the system.

9.2 Test Cases

A test case is a set of rules or conditions to check if the system or one of its feature works in accordance to the requirement. It is a document with a set of details which includes, set of test data, expected results, actual results, environment information and soon.

I have designed and executed a few test cases to check if the application meets the functional requirements.

Below are the test cases for the Job Search Portal web application.

TEST MODULE	**TEST CASE**	**EXPECTED RESULT**	**TEST RESULT**
ADMIN	Provide valid login credentials	User successfully logged in and directed to the admin dashboard page	PASS
ADMIN	Enters invalid login credentials	Displays Error message	PASS
ADMIN	Upon successful login, click on the ‘List of Employers’ tab.	Displays the details of list of active employers registered with the application	PASS
ADMIN	Click on ‘Active/Deactivate’ tab under status of the employer	The status of the employer will be changed to active/deactivate.	PASS
EMPLOYER	Provide details for registration	Employer successfully registered with the application	PASS
EMPLOYER	Upon successful login, click on ‘Post New Job’ tab	Employer posts jobs with the required details	PASS
EMPLOYER	Employer trying to post job with insufficient details	Prompts to fill in all the necessary details of the job	PASS
EMPLOYER	Employer clicks on the ‘List Posted Jobs’ tab	All the jobs posted by the employer will be displayed.	PASS

EMPLOYER	Employer clicks on ‘Active/deactivate’ under Status	The status of the job posting will changed to active/deactivated.	PASS
EMPLOYER	Employer clicks on the ‘view’ tab under candidates column	The list of the details of applicants for a particular job posting are displayed.	PASS
JOBSEEKER	Provide details for registration	Jobseeker successfully registered with the application	PASS
JOBSEEKER	Enters invalid login credentials	Error message displayed	PASS
JOBSEEKER	Upon successful login, click on ‘My Profile’ tab	List details of jobseeker	PASS
JOBSEEKER	Upon successful login, click on ‘Search Jobs’ tab	Details of the active job postings are displayed.	PASS
JOBSEEKER	Upon successful login, click on ‘Applied Jobs’ tab	Details of the jobs that are applied by the jobseeker are displayed	PASS
JOBSEEKER	Click on ‘Add Review’ tab	Displays a form to fill in the review details of the company	PASS
JOBSEEKER	Logout	Redirects to the Home page of the application	PASS

Table 1-Unit Test Cases for web application

The test cases for the android application are as follows.

Test Objectives: Navigation from Splash screen to Jobs screen

TEST CONDITION	**INPUT SPECIFICATION**	**OUTPUT SPECIFICATION**	**PASS/FAIL**
The user is currently on the Splash screen	User enters credentials and clicks on login button	Directs to Jobs screen	PASS

Table 2-Test Case for navigation to Jobs screen

Test Objectives: Navigation from Jobs Screen to Job details Screen

TEST CONDITION	**INPUT SPECIFICATION**	**OUTPUT SPECIFICATION**	**PASS/FAIL**
The user is currently on the Jobs screen	User clicks on the view against a particular job	Directs to Job details screen	PASS

Table 3-Test Case for navigation to job details screen

Test Objectives: Successfully submits the Job application

TEST CONDITION	**INPUT SPECIFICATION**	**OUTPUT SPECIFICATION**	**PASS/FAIL**
The user is currently on the Jobs page	User clicks on the ‘Apply’ tab against a job post	Prompts a message as “Successfully applied to this job”	PASS

Table 4-Test Case for applying jobs

Test Objectives: User checks for applied jobs

TEST CONDITION	**INPUT SPECIFICATION**	**OUTPUT SPECIFICATION**	**PASS/FAIL**
The user is currently on the Jobs page	User clicks on ‘Applied Jobs’ tab	Lists all the jobs that are applied	PASS

Table 5-Test Case for listing all jobs

Chapter 10 - Performance Testing

Performance testing is performed to determine how well the system can perform in terms of responsiveness under all kinds of load. The web application is tested to see if it can sustain huge amount of requests providing higher throughput under different loads. I have simulated multiple hits on various pages of the application to evaluate the overall performance.

10.1 Performance Analysis Tools

I have used Apache JMeter to test functional and performance both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more) [7]. It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types.

10.1.1 System Configuration

The details of the configuration of the system that has been used for testing the application are listed below:

Operating System	Windows 10 (64 bit)
RAM	8 GB
Processor	Intel core i7
Processor Speed	3.40 GHz

Table 6- System Configuration for Testing

The below listed tests are conducted on the system with the above-mentioned configuration.

10.1.2 JMeter-Test Plan Results

10.1.2.1 Throughput Analysis

JMeter is configured to an increased user count while having the same Ramp-Up time and Loop Count. The results for various test cases while mare displayed in the table below.

Users Count	Ramp Up	Loop Count	Samples	Throughput
50	1	10	500	10,932/min
100	1	10	1000	13,324min
12	1	10	2000	23,132/min

Table 7- Performance Testing Analysis for 100 records

The analysis is done with initial 100 records for jobs posted.

Extending the testing to increased records of 300 for jobs posted, I have observed a decrease in throughput under the same Test Plans.

Users Count	Ramp Up	Loop Count	Samples	Throughput
50	1	10	500	9,190/min
100	1	10	1000	10,753min
12	1	10	2000	15,534/min

Table 8- Performance Testing Analysis for 300 records

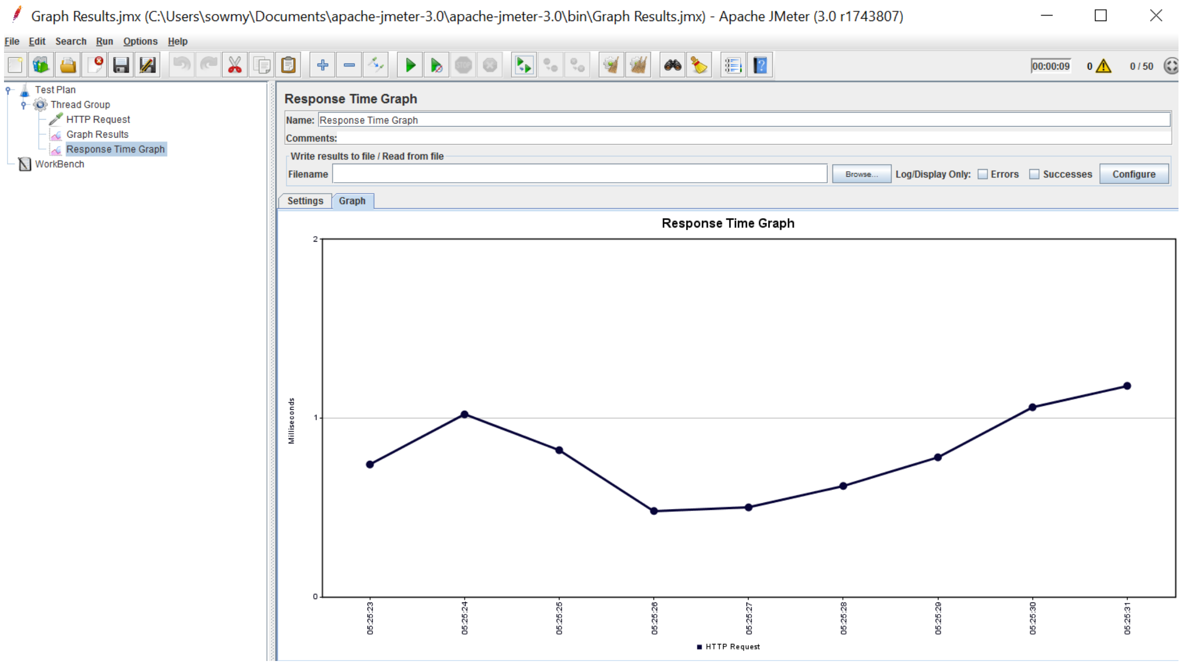
In both testing analysis, there is an increase in average throughput time with an increase in user's count. The first test case is executed for 500 samples i.e., simulating 500 requests from 50 users with a loop count of 10. Similarly, the second and third test cases are executed with 100 and 200 users simulating 1000 and 2000 samples respectively. From the above results, we can see that with the increase in the number of users the throughput increases thus ensuring good performance of the application.

In addition, the average throughput of application for 2000 samples accessing jobs page is around 11000/minute for 300 records but when samples are above 2000 then throughput is decreasing that means the application works good for up to 2000 requests.

10.1.2.2 Response Time Analysis

The graphs below show the results of response time analysis for under various test cases.

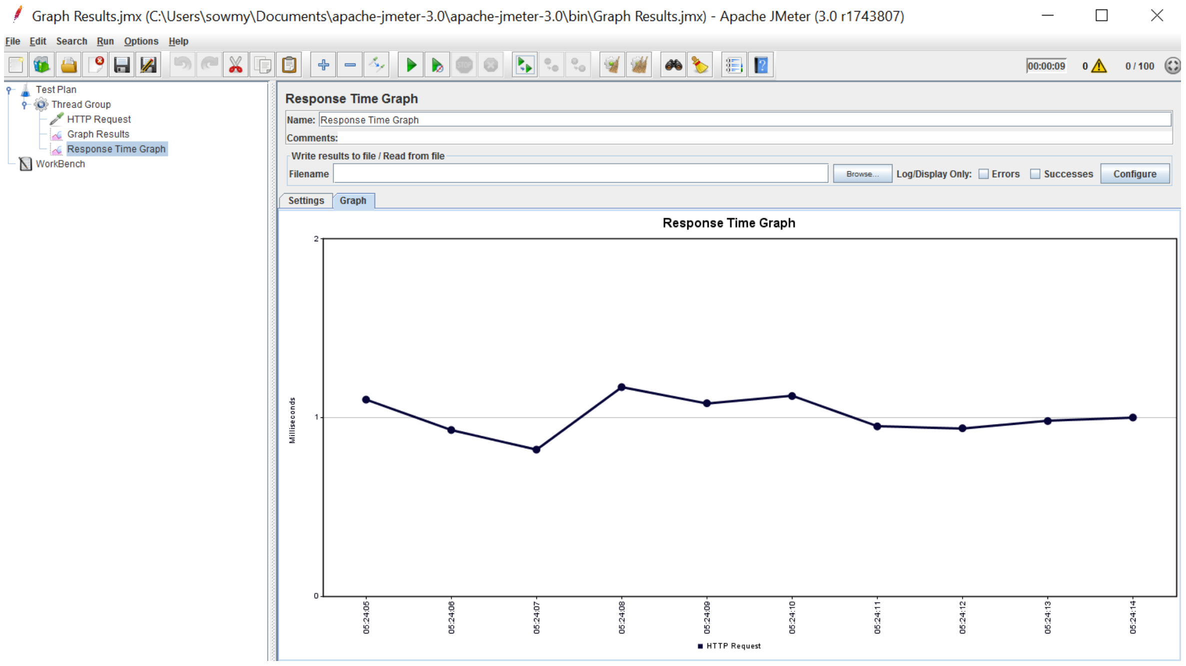
Test Case 1: User Count = 50


Graph Results.jmx (C:\Users\sowmy\Documents\apache-jmeter-3.0\apache-jmeter-3.0\bin\Graph Results.jmx) - Apache JMeter (3.0 r1743807)

Name: Response Time Graph

Figure 18: Response Time Analysis Graph for Test Plan 1

Test Case 2: User Count = 100


Graph Results.jmx (C:\Users\sowmy\Documents\apache-jmeter-3.0\apache-jmeter-3.0\bin\Graph Results.jmx) - Apache JMeter (3.0 r1743807)

Name: Response Time Graph

Figure 19: Response Time Analysis for Test Plan 2

Test Case 3: User Count= 200

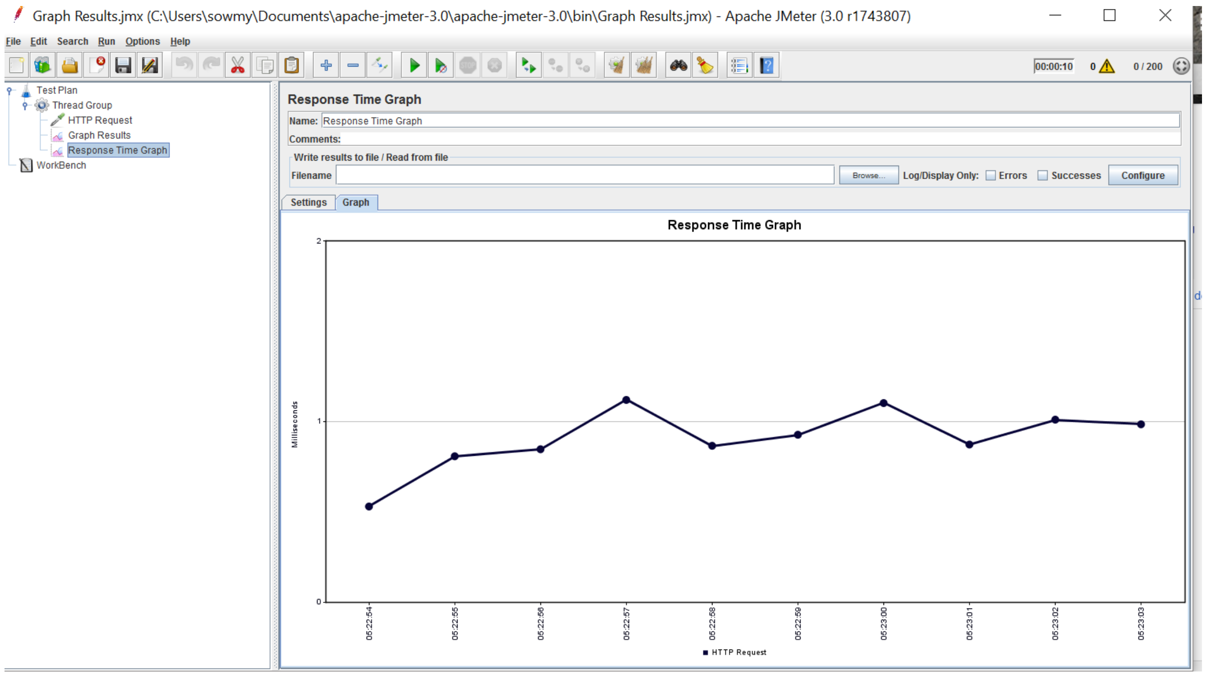


Figure 20: Response Time Analysis for Test Plan 3

The graphs above show Response Times for different User counts. The vertical axis is the Response Time in milliseconds and the horizontal axis is Elapsed Time i.e. granularity/1000 milliseconds. In all the three test cases, we can see that the Response Times are almost the same varying from 0.5 milliseconds to 1.25 milliseconds.

10.2 Rendering Analysis Tools

I have used profiling tools available in the developer options to record and visualize the rendering of the application. The application is tested on Samsung Galaxy S5 with the following specifications.

10.2.1 System Configuration

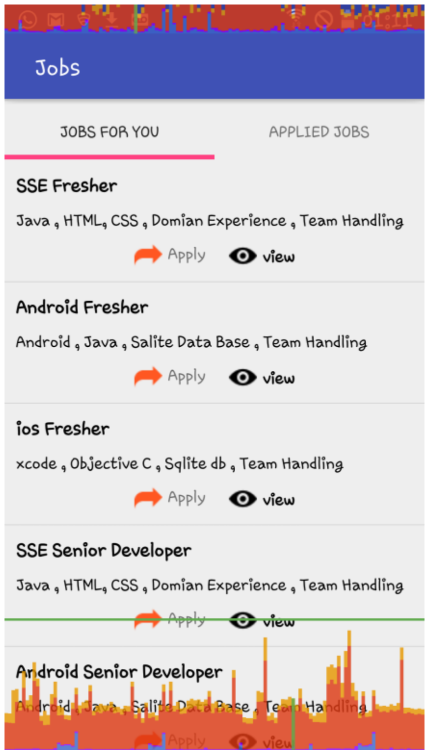
Operating System	Android OS, v6.0 (Marshmallow)
CPU	Quad-core 2.5 GHz Krait 400
GPU	Adreno 330
Internal	32 GB, 2 GB RAM

10.2.2 Profiling GPU Rendering

Profile GPU Rendering gives you a quick visual representation of how much time it takes to render the frames of a UI window relative to the 16-ms-per-frame benchmark [8]. It helps you to see how a UI window performs against the 16-ms-per-frame target. It helps in identifying if any part of the rendering pipeline stands out in using processing time.

The tool displays a graph for each application. The tool shows up vertical bars on the screen, graphing performance of the application. The taller the bar, the longer it takes to render. The graph has colored sections representing the phase of the rendering pipeline.

- The **green** line is the 16 milliseconds Reference Bar. Any time a bar pushes above this line, there may be pauses in the animations.
- The **blue** section of the bar represents the time used to create and update the View's display lists.
- The **purple** section of the bar represents the time spent transferring resources to the render thread.
- The **red** section of the bar represents the time spent by Android's 2D renderer issuing commands to OpenGL to draw and redraw display lists.
- The **orange** section of the bar represents the time the CPU is waiting for the GPU to finish its work.



Jobs

JOBS FOR YOU

APPLIED JOBS

SSE Fresher

Java , HTML, CSS , Domian Experience , Team Handling

Apply view

Android Fresher

Android , Java , Salite Data Base , Team Handling

Apply view

ios Fresher

xcode , Objective C , Sqlite db , Team Handling

Apply view

SSE Senior Developer

Java , HTML, CSS , Domian Experience , Team Handling

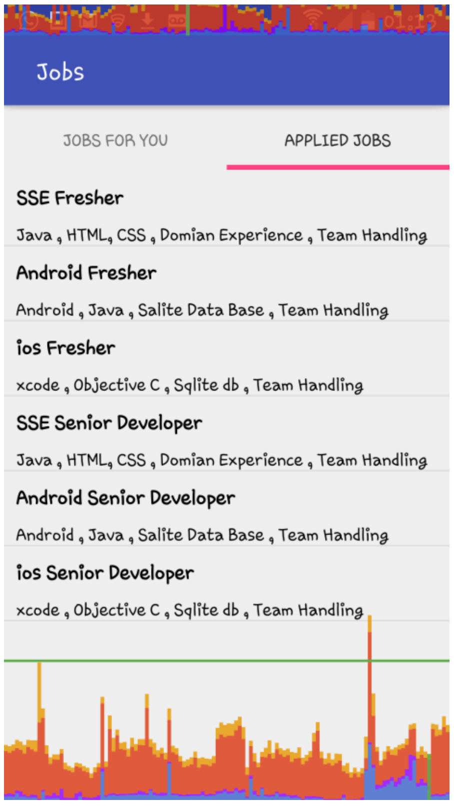
Apply view

Android Senior Developer

Android , Java , Salite Data Base , Team Handling

Apply view

Figure 21: Profile GPU Rendering for Jobs for you screen



Jobs

JOBS FOR YOU

APPLIED JOBS

SSE Fresher

Java , HTML, CSS , Domian Experience , Team Handling

Android Fresher

Android , Java , Salite Data Base , Team Handling

ios Fresher

xcode , Objective C , Sqlite db , Team Handling

SSE Senior Developer

Java , HTML, CSS , Domian Experience , Team Handling

Android Senior Developer

Android , Java , Salite Data Base , Team Handling

ios Senior Developer

xcode , Objective C , Sqlite db , Team Handling

Figure 22: Profile GPU Rendering for Applied Jobs screen

Chapter 11 - Conclusion and Future Work

11.1 Conclusion

Job Search Portals stands as a revolutionizing element in the sphere of recruitment. They act as a communication bridge between applicants and recruiters facilitating their requirements. This application helps organizations to have a greater exposure to the candidate pool and also job seekers facilitating wide search of jobs matching their interests. The android application provides flexibility to the jobseekers to view the openings and applied jobs without the need to carry a laptop. This application provides an enhanced user experience for both employer and jobseeker. It provides user friendly interface which facilitates in reaching wide range of audience.

The application has achieved all the requirements that were initially set in the requirements gathering phase. This project taught me some best practices in the technology stack like Spring MVC, Hibernate ORM, Android development, RESTful web services. Starting from requirements elicitation to design, construction, implementation and testing, I have gained a very good experience working with various technologies at every phase. Development of this project boosted my confidence in mobile and web development.

11.2 Future Work

This project fulfills the primary requirements of the job seekers and employers. It can be extended in several ways – We can provide recommendations and email updates for new job postings based on the job seeker’s search history. Since, the job seekers might be interested in building a strong Resume, we can provide tips and information for the same. We can also provide templates for building the Resumes which might interest most applicants. The mobile application is developed fulfilling the functionalities of job seeker, it can be extended to support functionalities of Employer as well.

References

[1] “Overview of Spring Framework”

http://docs.spring.io/spring/docs/current/spring-framework-reference/html/index.html[Sep.10, 2016]

[2] “Hibernate Getting Started Guide”

https://docs.jboss.org/hibernate/orm/5.0/quickstart/html/ [Sep 15, 2016]

[3] “Hibernate-Architecture”

http://www.tutorialspoint.com/hibernate/hibernate_architecture.htm [Sep. 30, 2016]

[4] “An Overview of the Android Architecture”

http://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture [Oct. 10, 2016]

[5] “Android Runtime”

https://en.wikipedia.org/wiki/Android_Runtime [Oct. 12, 2016]

[6] “UML - Standard Diagrams”

https://www.tutorialspoint.com/uml/uml_standard_diagrams.htm [Oct. 15, 2016]

[7] “Apache JMeter”

http://jmeter.apache.org/ [Oct. 20, 2016]

[8] “Profile GPU Rendering Walkthrough”

https://developer.android.com/studio/profile/dev-options-rendering.html#WhatYouNeed [Nov. 01, 2016]

[9] “Platform Architecture”

https://developer.android.com/guide/platform/index.html[Nov. 05, 2016]