



College of Engineering Pune
(An Autonomous Institute of the Govt. of Maharashtra)

Software Engineering Mini Project II
Third Year Computer Engineering

Career Grip

Under the guidance of
Prof. Tanuja Pattanshetti

Adarsh Ninganur 111803115

Soumitra Tonapi 111803103

Rohit Kota 111803119

Batch: T6

Group: 110

Table of Contents

Table of Contents	Page number
1. Abstract	1
2. Introduction	1
3. SRS	2
3.1. Introduction	2
3.2. Overall Description	3
3.3. External Interface Requirements	4
3.4. System Features	5
3.5. Other Nonfunctional Requirements	7
3.6. Other Requirements	8
4. Literature Review	10
4.1 Tabular representation	10
4.2 Conclusion that summarizes the literature review.	16
5. Proposed Statement	17
6. Planning	18
7. Design	19
7.1 Data Flow Diagram	19
Level 0	19
Level 1	20
Level 2	21
7.2 Use Case Diagram	22
7.3 Activity Diagram	23
7.4 System Architecture	24
7.5 Class Diagram	25
7.6 Sequence Diagram	26
7.7 State Diagram	27
7.8 ER Diagram	28
7.9 Component Diagram	29
7.10 Deployment Diagram	30
8. Implementation	31
9. Testing	47
10. Github/ Lab Link	56
11. References	56

1. Abstract

Many newcomers to freelancing often struggle to kickstart their freelancing career compared to seasoned freelancers who don't have much difficulty finding work. In this competitive era, the education among the people is increasing, and their jobs are now decreasing. Moreover, the companies want the people who are the best in their fields. In that scenario, it becomes difficult to find people who are suitable for the company's position. Thinking about these problems, one can think about the process which can handle this problem and make the work less complicated.

The portal deals with the recruitment process here. The portal aims to simplify the process of hiring applicants. The recruiter can view the applicants for the job and their respective resumes. They can post jobs based on their requirement. This portal will allow a job seeker to apply for a job (with a resume) in the company for the interested vacancy available at the company. Moreover, the job seekers might receive Job alerts based on their interests.

2. Introduction

Career Grip is a platform that bridges the gap between Job Seekers and Companies where Recruiters can shortlist candidates who have the suitable skill set to handle the responsibilities efficiently. On the other hand, job seekers want a job to apply their skills and knowledge to grow their professional careers. Moreover, there are Freelancers who want to work for multiple clients and may want short-time work. The purpose of our software is to build a platform where job seekers can hunt for desired jobs and stay abreast of various job openings; companies can easily handle the recruitment process.

3. SRS

1. Introduction

1.1 Purpose

Career Grip is a platform that bridges the gap between Job Seekers and Companies where Recruiters can shortlist candidates who have the suitable skill set to handle the responsibilities efficiently. On the other hand, job seekers want a job to apply their skills and knowledge to grow their professional careers. Moreover, there are Freelancers who want to work for multiple clients and may want short-time work. The purpose of our software is to build a platform where job seekers can hunt for desired jobs and stay abreast of various job openings; companies can easily handle the recruitment process.

1.2 Document Conventions

The document was created based on the IEEE template for System Requirement Specification Documents.

1. Headings are identified with larger font sizes and the use of bold characters.
2. The font style used in this document is Times New Roman.
3. Hyperlinks are identified with blue font colors and underlines.

1.3 Intended Audience and Reading Suggestions

This SRS is intended for project managers, evaluators, users, or any other stakeholder which needs in-depth information about the functioning of the web application.

Reading Suggestions:

- The first section is intended for the Project Managers.
- The second and third sections are intended for the developers.

1.4 Product Scope

The web application aims to simplify the process of hiring applicants. The recruiter can view the applicants for the job and their respective resumes. They can post jobs based on their requirement. This software will allow a job seeker to apply for a job (with a resume) in the company for the interested vacancy available at the company. Moreover, the job seekers might receive Job alerts based on their interests.

1.5 References

1. The Potential for Using Online Job-Portal -
<http://documents1.worldbank.org/curated/en/370301486667802852/pdf/WPS7966.pdf>

2. Research Paper - <https://www.irjet.net/archives/V6/i4/IRJET-V6I433.pdf>
http://ijirt.org/master/publishedpaper/IJIRT144246_PAPER.pdf
3. Efficacy of Job Portals - <https://garph.co.uk/ijarmss/july2013/15.pdf>
4. IEEE SRS template - https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc

2. Overall Description

2.1 Product Perspective

This Web Application creates a platform for job seekers to connect with Recruiters. They can search for jobs with advanced filters and apply for multiple job listings. Also, it makes the hiring process seamless for recruiters.

This is a standalone and self-contained product. A web browser is sufficient to use this application.

2.2 Product Functions

- Efficient system to help job seekers find prospective employers.
- Job seekers can view job titles in various fields/technologies by applying multiple filters to help them narrow down their search.
- Job seekers can apply for multiple positions.
- Company recruiters can post/manage roles and jobs to which job seekers can apply.
- Personalized Dashboard for every user.

2.3 User Classes and Characteristics

- Jobseeker - They can search for jobs and apply for multiple positions.
- Recruiter - They can Post Jobs and view applicants and their resumes.

2.4 Operating Environment

The software is designed to work in any operating system interface supporting Internet Explorer 9 and above(inclusive of all parallel versions of Chrome, Firefox, Chromium, etc.)
Supported Operating Systems:

- Windows 2000
- Windows XP
- Windows Vista
- Windows 7
- Windows 8
- Windows 10
- Mac OS X
- Linux. (Ubuntu, Debian, Arch-Linux, etc.)

2.5 Design and Implementation Constraints

Since the Web application is built using JavaScript and JavaScript-based libraries, including React.js and Express.js, any platform on which it has to be tested by developers must have a JavaScript runtime environment like Node.js. The application uses a hosted MongoDB Atlas database to ensure remote accessibility. Moreover, javascript must be enabled on the web browser.

2.6 User Documentation

A user manual will be provided, which would include instructions on how to navigate within the application.

The Web application will be fairly intuitive to use for an internet user.

2.7 Assumptions and Dependencies

For smooth functioning of the application, it is assumed that:

- The user has a modern web browser installed on their system.
- The user system has enough memory, processing power, and internet connectivity.
- The MongoDB database is up and running.
- Javascript is enabled in the web browser.

3. External Interface Requirements

3.1 User Interfaces

The main components of the user interface can be listed as follows:

- 1. Home Page:** The user will be greeted with this page when the web application is accessed. The user can either choose to Login to their existing account or choose to Sign Up based on whether they are a Job Seeker or Recruiter.
- 2. Dashboard:** Users will be presented with a personalized dashboard depending on whether they are Job seekers or Recruiter.
- 3. Search tool:** Job seekers will have the Search tool feature, which will enable them to search for relevant jobs based on their interests, skills, or experience.
- 4. Profile Information:** Every user can maintain their profile information. In the case of a recruiter, they need to maintain their company profile. They can update their profile, for example, skills, projects, or personal information, whenever required.
- 5. Other UI Components:** We may need to include some components that are not stated here if deemed critical to the functioning of the application.

3.2 Hardware Interfaces

- The application will be a web application; hence it will be supported on modern web browsers like Google Chrome, Mozilla Firefox, Safari.
- The application uses a hosted MongoDB Atlas database to ensure remote accessibility.
- The Web application may be either used on a desktop or a mobile phone.
- CPU: As the application is web-based so there is no need for a specialized CPU. A general CPU that can run web-browsers is enough
- Stable Internet connectivity in the device.

3.3 Software Interfaces

- **Runtimes**

The Node.js runtime environment is central to the functioning of our application. We are using Node.js as the primary way to run JavaScript code.

- **Packages and Libraries**

Libraries, including React.js and Express.js, are essential for the functioning of the frontend and backend, respectively. Our application would also rely on NPM packages for certain features. But, all these dependencies would be bundled with the application as one complete package.

- **APIs**

Our application uses the database by abstracting the database functionality through a software API.

3.4 Communications Interfaces

- **HTTPS:** The web application will use HTTPS to enable secure communication between client and server.
- **SMTP:** We will be using a SMTP protocol-based Node js module to send emails for email verification and job alerts.

4. System Features

4.1 User Sign Up and Login

4.1.1 Description and Priority

This is a feature of high priority. Job seekers must have an account and login to apply for jobs. Similarly, Recruiters must have their company profile to view the applicants. Existing users must be able to login to access their accounts.

4.1.2 Stimulus/Response Sequences

Home Page → Sign Up / Login Page → Dashboard

4.1.3 Functional Requirements

- REQ-1: A form should be rendered on the user interface where the user can enter his/her details depending on Sign Up/ Login.
- REQ-2: The server should be able to connect to the database and be able to fetch the required information from the database.
- REQ-3: If a user already exists with the given email-id during Sign-up, it should prompt the user with the error.
- REQ-4: If a user enters incorrect login credentials, the user must be prompted with the error message.

4.2 Job Search

4.2.1 Description and Priority

This is a feature of high priority. Job seekers should be able to search for jobs based on their skills/ experience. They can search using advanced filters such as role, salary, or technology. The relevant job postings will be displayed.

4.2.2 Stimulus/Response Sequences

SignUp / Login Page → Dashboard → Search Job → Apply filters → Show Job Postings

4.2.3 Functional Requirements

- REQ-1: A search button should be displayed on the home page, which Job seekers can use to search for jobs.
- REQ-2: Job seekers should be able to apply advanced filters such as role, salary, duration, technology, experience, or skills needed.
- REQ-3: The server should be able to connect to the database and be able to fetch the required jobs from the database depending on the filters.
- REQ-4: The search result may return zero listings if the searched query is not found in the listings or if too many filters are applied.

4.3 Apply for Job

4.3.1 Description and Priority

This is a feature of high priority. After searching for jobs and finding relevant job listings, a Job Seeker might want to apply for the job. The Job Seeker can apply using a custom resume to apply for the position.

4.3.2 Stimulus/Response Sequences

Search Job → Apply filters → Show Job Postings → Upload resume → Apply for Job.

4.3.3 Functional Requirements

- REQ-1: Jobseeker can upload his/ her resume in their profile details
- REQ-2: Job seekers should be able to apply for a particular job listing.
- REQ-3: A Job Seeker can apply for multiple jobs.

4.4 Manage Job Listings

4.4.1 Description and Priority

Recruiters should be able to post job listings and be able to view applicants for the listing. This will help them to shortlist applicants based on requirements. This is a feature of high priority.

4.4.2 Stimulus/Response Sequences

Post Job Listing → View Applicants → View their resumes → Shortlist Applicants.

4.4.3 Functional Requirements

REQ-1: Recruiters should be able to Post new Jobs from their profile.

REQ-2: Recruiters should be able to view the applicants for their job listings and their respective resumes.

REQ-3: Recruiters should be able to shortlist applicants based on their requirements.

4.5 Job Alerts

4.4.1 Description and Priority

Job seekers should receive Job Alerts based on their skills, interests, or roles. This is a feature of medium priority.

4.4.2 Stimulus/Response Sequences

Company Posts Job Listing → Send Job Alert (to relevant users).

4.4.3 Functional Requirements

REQ-1: Job seekers should receive Job Alerts on their email regularly.

REQ-2: The Job Alerts should be relevant to the Jobseeker which should be based on their interests, skills, technology, or roles.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. The performance requirement of the application is that there should be no visible lag when switching between web pages, scrolling the page, opening the application, searching for jobs using multiple filters, and applying to jobs.
2. The Recruiters should be able to see all the applicants and their respective resumes.

5.2 Safety Requirements

1. It will be clearly stated that the designers of the application will not be responsible for any damage caused by the use of the application

2. Also, Jobseekers must exercise caution while applying for jobs to prevent any fraudulent activity.

5.3 Security Requirements

1. Every user (whether a Jobseeker or a Recruiter) will have to login with a verified email address and a password when using the application.
2. The profiles of the users will be stored in password-protected databases in the application's backend.
3. The email sending service will only send encrypted emails to users.

5.4 Software Quality Attributes

1. Reliability

The Recruiters are approved only by the Admin so that the platform will be reliable for job seekers.

2. Portability

The application should be usable from any device supporting modern web browsers.

3. Usability

The application will be easy enough to use for any first-time user. A user manual will also be provided, which will include instructions.

4. Flexibility

A user should be able to login into our application from any supported device - Desktop, Laptop, or mobile device.

5. Robustness

As the MongoDB database will be used for the application, it is fairly robust against possible errors.

5.5 Business Rules

- Recruiters can post job listings with Job details.
- Moreover, users have to enter their email addresses in the correct format to use the platform.
- Jobseekers can apply with a Resume which makes it easier for recruiters to
- Furthermore, Job alerts enable them to view new or relevant jobs without opening the web application in the first place.

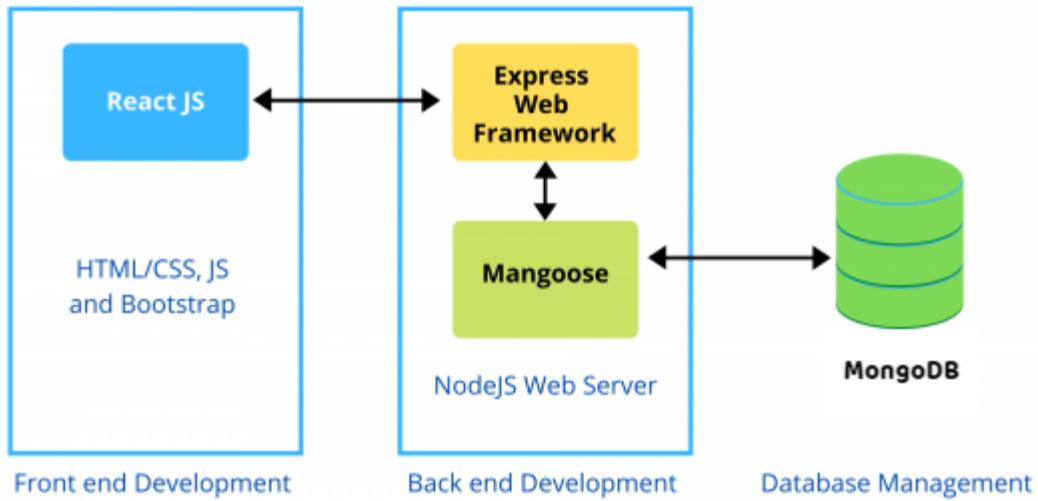
6. Other Requirements

Appendix A: Glossary

- Node.js: Node.js is a back-end JavaScript runtime environment that executes JavaScript code outside a web browser.
- Express.js: Express.js is a back end web application framework for Node.js
- React - React is an open-source, front-end, JavaScript library for building user interfaces or UI components.

- MongoDB: MongoDB is a cross-platform document-oriented database program. It is classified as a NoSQL database program.
- HTML: HyperText Markup Language is the standard markup language for documents designed to be displayed in a web browser
- CSS: Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.
- CPU: Central processing unit is the electronic circuitry within a computer that executes instructions that make up a computer program.

Appendix B: Analysis Models



4.**Literature Review**

Year of Paper	Title of the Paper	Journal/ conference details	Methodology	Proposed idea	Advantages/ achieved objectives in paper	Disadvantages/ Limitations
2013	Efficacy of job portals and social media on Organizational business	International Journal of Advanced Research in Management and Social Sciences	Study of job portals in recruitment efficacy with relation to managing selection and retention of employees. It is based upon descriptive analysis and based on secondary data. Data sources would be internet portals, websites, newspapers, research articles, journals and papers, business magazines, etc.	Analysis of pros and cons of e-recruitment for organization and for employees and to analyze how e-recruitment can affect organizational ability to select and retain staff.	<ul style="list-style-type: none"> ➤ Built knowledge and understanding of the technology options available. ➤ Ensuring candidate- and user-friendly interfaces on their systems. ➤ Understanding of Internet access and proficiency levels amongst target groups. 	<ul style="list-style-type: none"> ➤ Loss in staff productivity, data leakage from staff gossiping freely in an open environment, damage to a company's reputation, scams practiced by cyber crooks. ➤ Computer servers can only process so much information at one time. The use of social networking websites, alongside email and company computer programs, slows down the servers.

2013	A Study on the Freelancing Remote Job Websites	International Journal of Business Research and Management	<p>A study of various existing freelance marketplaces was done and data was obtained. Over 20 different features from each of these sites were taken into account. These marketplaces were compared and accordingly factors improving the chances of getting job seekers work were given more focus.</p>	<p>Analysis of data in order to find solutions to improve the rate at which these job seekers on freelance websites get work.</p>	<ul style="list-style-type: none"> ➤ Critical analysis of job portals by comparing various distinct demographics and areas (like the US and South East Asia) helps to better pinpoint areas of growth in these regions. ➤ Labor laws can be easily bypassed if both parties feel the deal is mutually beneficial. 	<ul style="list-style-type: none"> ➤ Does not consider how competitive the economy is for job seekers in these various regions.
2013	Job Portal-A Web Application for Geographically Distributed Multiple Clients	First International Conference on Artificial Intelligence, Modelling & Simulation	<p>This paper proposes a design of an online recruitment system, that allows employers to post their job advertisements, which job seekers can refer to when looking for jobs. This job portal is able to capture job requirements based on industry needs.</p>	<p>To identify the needs of the companies and jobseekers which will be incorporated into a knowledge management system with updated information. Also, to design a system that allows employers to post their job advertisements, which job seekers can refer to when looking for jobs.</p>	<ul style="list-style-type: none"> ➤ The knowledge portal can be a suitable medium for job seekers to establish a link with employers or HR managers. ➤ The Admin can regulate the platform. The Admin manages, controls, and views the whole process in the system. 	<ul style="list-style-type: none"> ➤ The application was developed and tested on a legacy platform (Windows XP). ➤ The application is not user-friendly.

2014	Development of a Job Web Portal to Improve Education Quality	International Journal of Computer Theory and Engineering	<p>A questionnaire was created for a survey that was handed out to students at FCSIT, UM.</p> <p>The data from the sample population was used to try to figure out where students would try to go post-graduation. Data Analytics was carried out afterward on this obtained output of the surveys.</p>	<p>With the results of the data analytics, the team was able to identify critical interests for the students who were planning to work in the freelance field. So, a system to link all these factors together could be created.</p>	<ul style="list-style-type: none"> ➤ Helps to better determine the factors that a job seeker will consider while applying for jobs, making filters for companies better. ➤ Implementation of a system schema integrates all the various features that were previously distinct or connected without an efficient portal. 	<ul style="list-style-type: none"> ➤ The study is limited to a very specific sample (Computer Engineering students) creates a huge bias in the data that has been collected.
2016	E-Recruitment and Its Impact upon on Job Seekers: A Contemporary Approach	Commerce & Management Department, Guru Kashi University	<p>The study discusses the concept of e-recruitment and the main purpose of this particular research to know about the general impacts of e-recruitment especially in the context of job seekers. It does so by conducting a survey with a list of questions related to the recruitment process.</p>	<p>To study the impact of online recruitment on job seekers and identify functionalities or tools to help them for job search</p>	<ul style="list-style-type: none"> ➤ Job seekers can find job postings easily on websites, job boards, and portals and apply using resume-building features. ➤ It is a time-saving and cost-effective method for job seekers compared to the conventional approach. 	<ul style="list-style-type: none"> ➤ The data collected about the candidate's responses were for a limited audience, thereby generalizing the answers to be a common norm.

2017	A Review on Job Portal- A Web Application for Distributed Clients	International Journal of Advanced Research in Computer and Communication Engineering	Study of distributed application structure that conceptualizes the relationship between the web portal, recruiters, and job seekers.	The proposed system is two-sided: it would be candidate-oriented or recruiter-oriented. The Jobseeker can view the requirements of available job positions and can apply for the same by submitting their CV.	<ul style="list-style-type: none"> ➤ Save time and workload for recruiters and job seekers. ➤ The responsibilities of the Admin, recruiter and Jobseeker are clearly stated in the paper. 	<ul style="list-style-type: none"> ➤ The application is proposed to be used on a legacy system. ➤ The security standards are outdated compared to current developments.
2017	A Online Job Portal Management System	International Journal of Innovative Research in Technology	A study regarding previously existing systems was read. Accordingly, a new system was devised. User functions split across multiple modules (Job Seeker, Resumes, Upload/Download, Employer) which can handle all the data that the user is giving to the system.	Creation of a common platform for jobseekers and employers. Easy for a novice user to use and a simple GUI to implement. Simple login structure and a large number of job profiles to view.	<ul style="list-style-type: none"> ➤ Highly secure and portable, a wide range of services available under one tool. ➤ Efficient search mechanisms for users to find their target jobs with relative ease. 	<ul style="list-style-type: none"> ➤ While very efficient for job seekers to use, it lacks the same nuances for Employers. ➤ Difficult for employers to shortlist and make decisions.

2018	A Study on Factors Affecting the Effectiveness of Job Portals from Job Seekers Perspective	International Journal of Scientific Research and Modern Education	The study has collected data provided by job portals and it was analyzed using statistical software and hypothesis testing was done to understand comparatively important factors in the job search.	The purpose of this study was to identify important aspects of job portals by observing what factors the job seekers gave maximum priority to.	<ul style="list-style-type: none"> ➤ Identified factors that majorly influence candidate choice of a job portal for job search. ➤ Inferred the effectiveness of job portal as a tool for Recruitment. 	<ul style="list-style-type: none"> ➤ The practicability of the identified important aspects of jobs portals is not analyzed. ➤ While an in-depth analysis of factors affecting job seekers is done, factors affecting the recruiter's decision are not considered; which would ensure a smooth recruitment process.
2019	Smart Job Recruitment Automation: Bridging Industry and University	2019 Artificial Intelligence for Transforming Business and Society (AITB)	This paper proposes and highlights the need for an online job board system for colleges and its effectiveness in bridging the gap between college students and career opportunities. A questionnaire involving 300 students was conducted to gain knowledge about how they perceive the job board system.	The overall purpose of this study is to bring students and companies on a single platform and replace the traditional way of hiring students on campuses.	<ul style="list-style-type: none"> ➤ The portal would automate the job requirement process and help students find better jobs for themselves. ➤ It would help companies to get better candidates suited for the job. 	<ul style="list-style-type: none"> ➤ Although the study states that it is an automated system, it does not mention which all features will be automated for the students or the company.

2019	ONLINE JOB PORTAL	International Research Journal of Engineering and Technology (IRJET)	The main scope of this study concerns the selection of tools used and the requirements to be met by the system which affects the development of a knowledge system to implement E-recruitment. The platform attempts to provide a new link between students, unemployed graduates, and the IT industry.	Idea was to share information with students and to assist graduates in their job search by establishing links between the faculty and organizations. This means that this knowledge system should serve as an online recruitment system.	<ul style="list-style-type: none">➤ Employers can identify a large number of eligible job seekers and get their information easily.➤ Companies or organizations can extend the search domain, hence, they will have a better prospect of selecting the most qualified candidates.	<ul style="list-style-type: none">➤ The portal industry is several years old, and vendors come into and out of the market every month. Since typical licensing and development costs are more, vendor selection is high risk.
------	-------------------	--	---	--	--	---

Conclusion:

From the literature study, it is clear that there is a need for a new portal that considers all of the weaknesses that have been seen in previous implementations of the job portals for freelancers. If a proper implementation is created, job seekers and employers will find it easier to communicate without the many difficulties that exist in the current platform.

Throughout the research papers, we have gathered that while many applications exist that cover small functionalities, there is no highly efficient and widely used platform that a very large majority of people use. Many of the attempted implementations of such a system are lacking in some or the other department, such as having a poor interface for job employers or without proper filter functions for job seekers to find their target jobs.

Along with this, the literature review made clear that a simple to operate GUI is absolutely essential in order to get movie users invested in the platform. It should be easy to use for anyone, while at the same time have enough features that both employers and freelance job seekers will be able to access all the possible functionalities without facing too many issues. In order to improve the application experience for both types of users, we plan to make the GUI as smooth as possible.

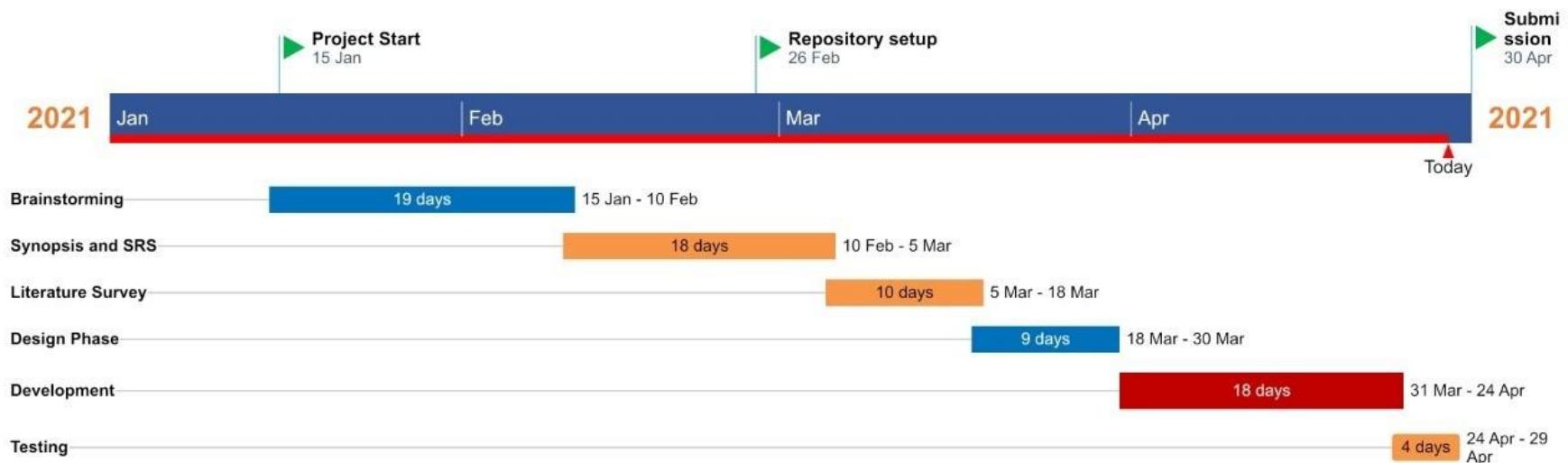
Taking into account all of these factors, we will develop a Job Portal for Freelance Workers accordingly.

5. Proposed statement

To design a web-based Job Portal where job seekers can hunt for desired jobs and stay abreast of various job openings, companies can easily handle the recruitment process.

6. Planning (Gantt chart)

Career Grip



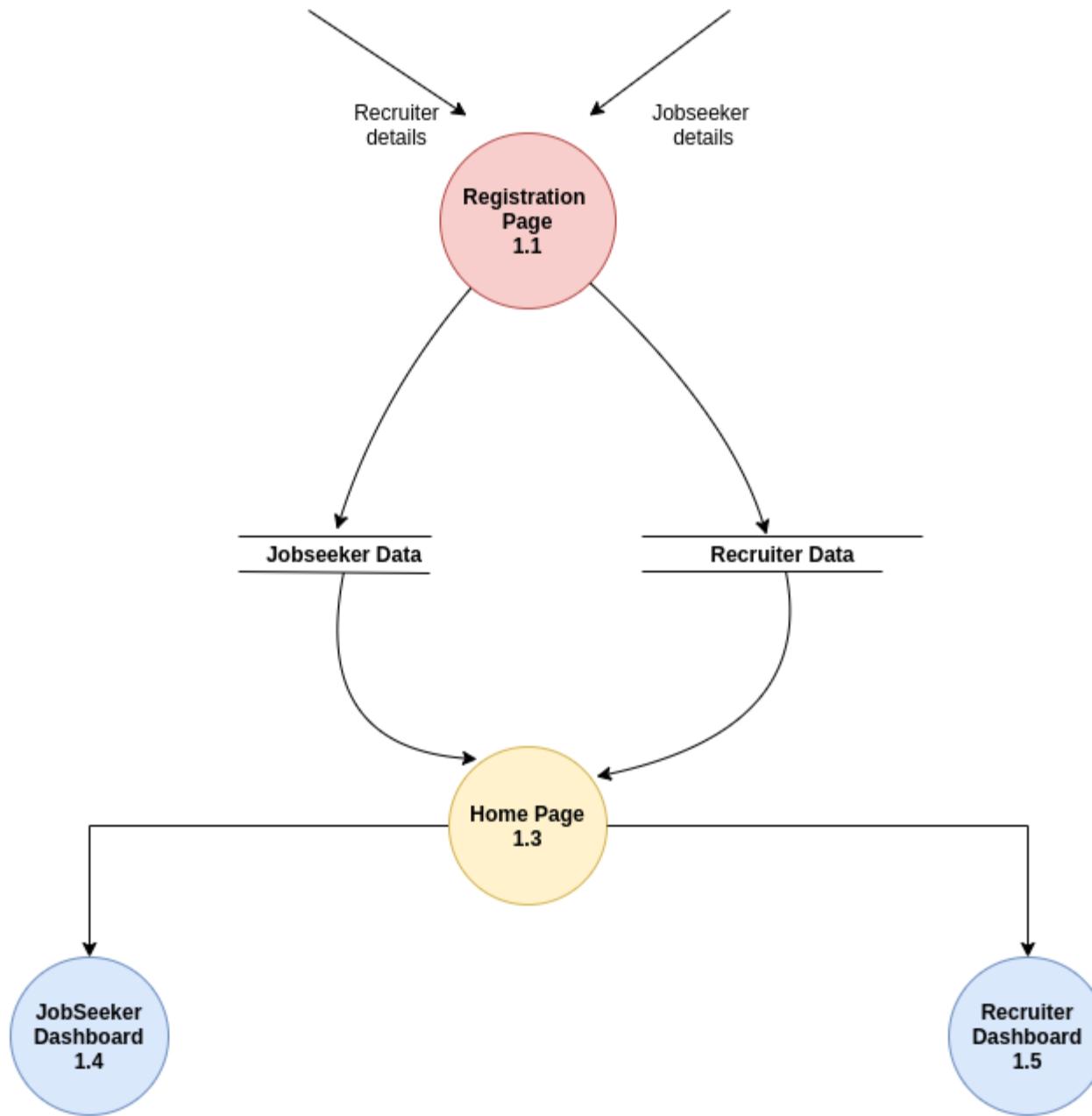
7. Design

7.1 Data Flow Diagram

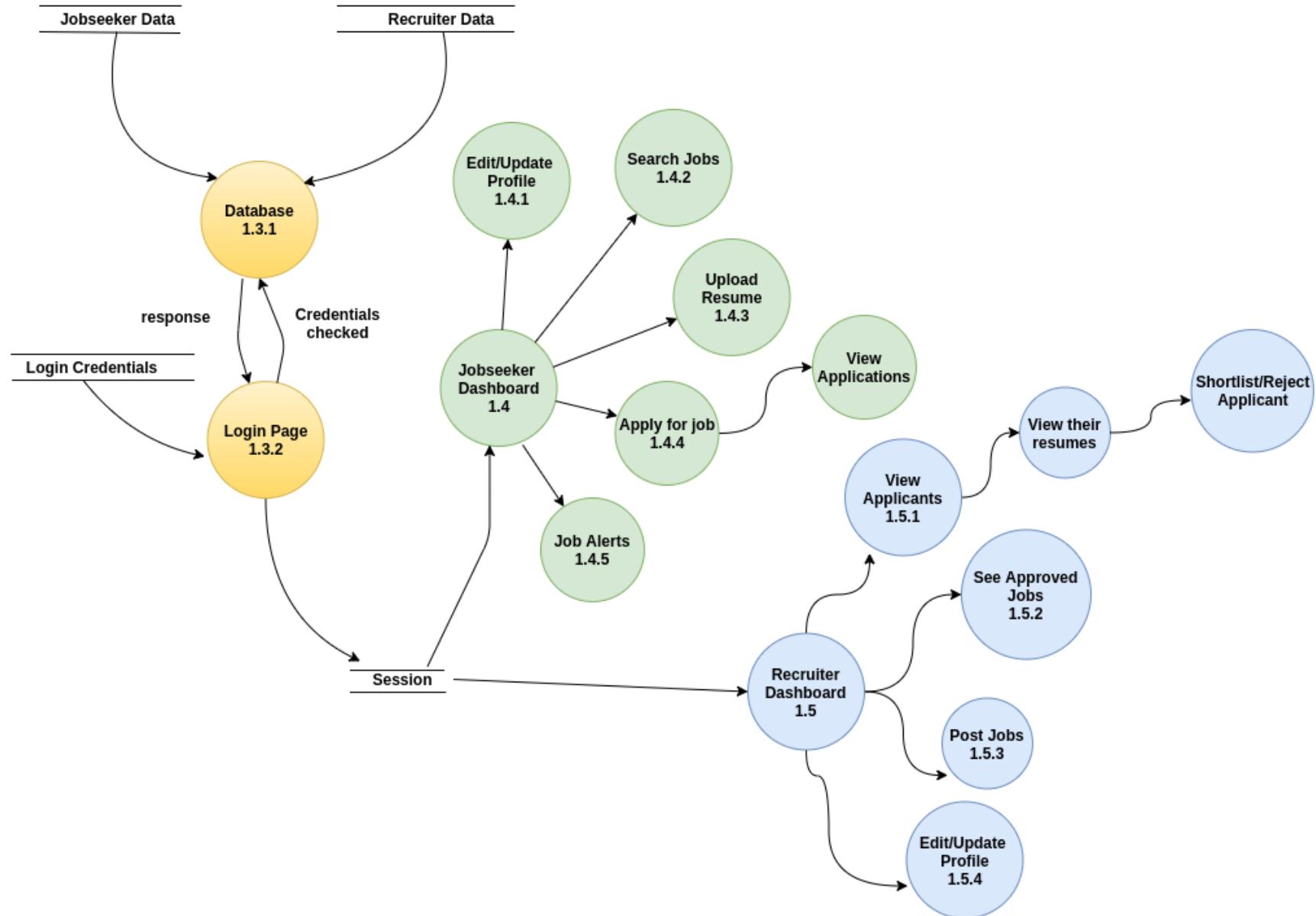
Level 0



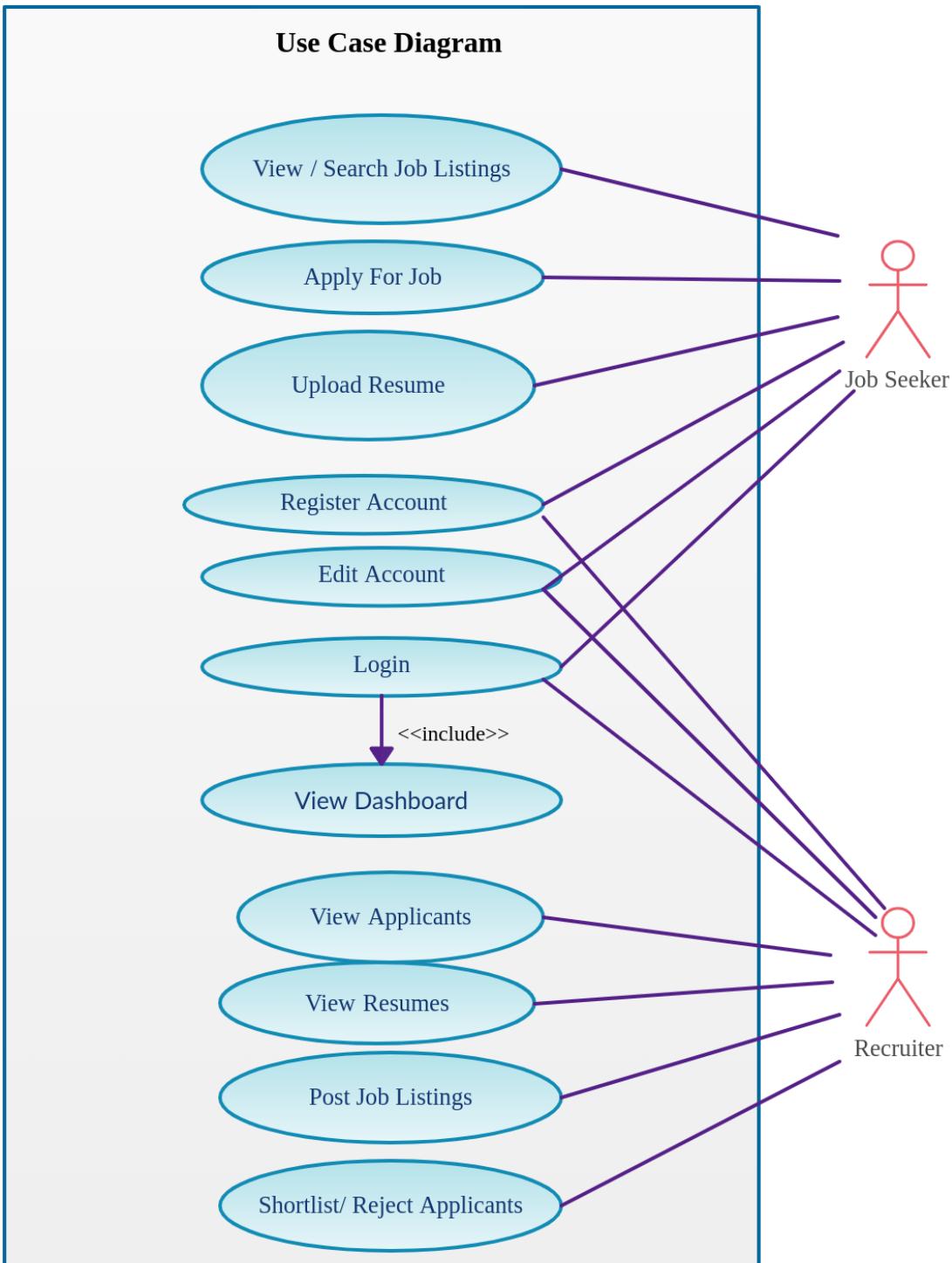
Level 1



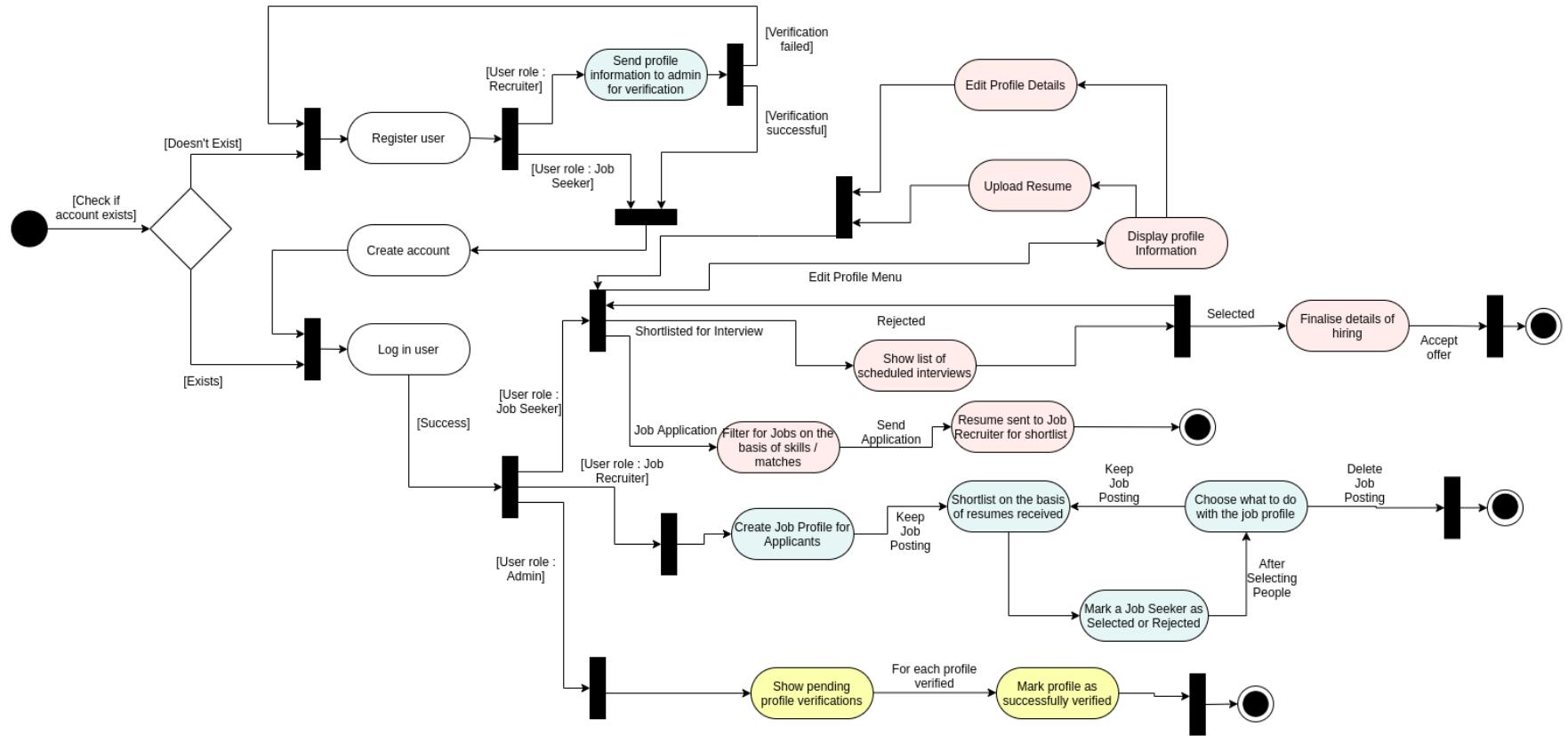
Level 2



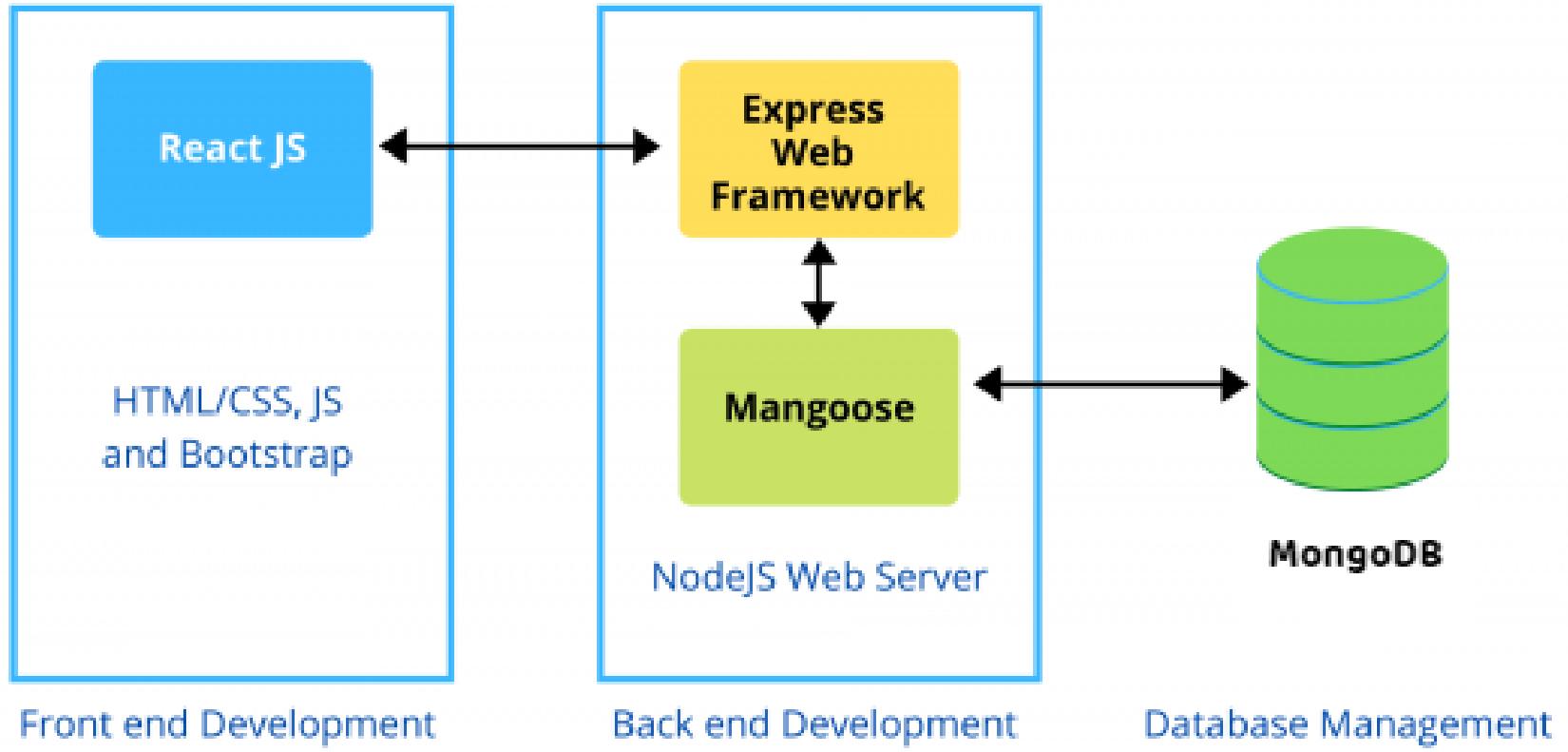
7.2 Use Case Diagram



7.3 Activity Diagram

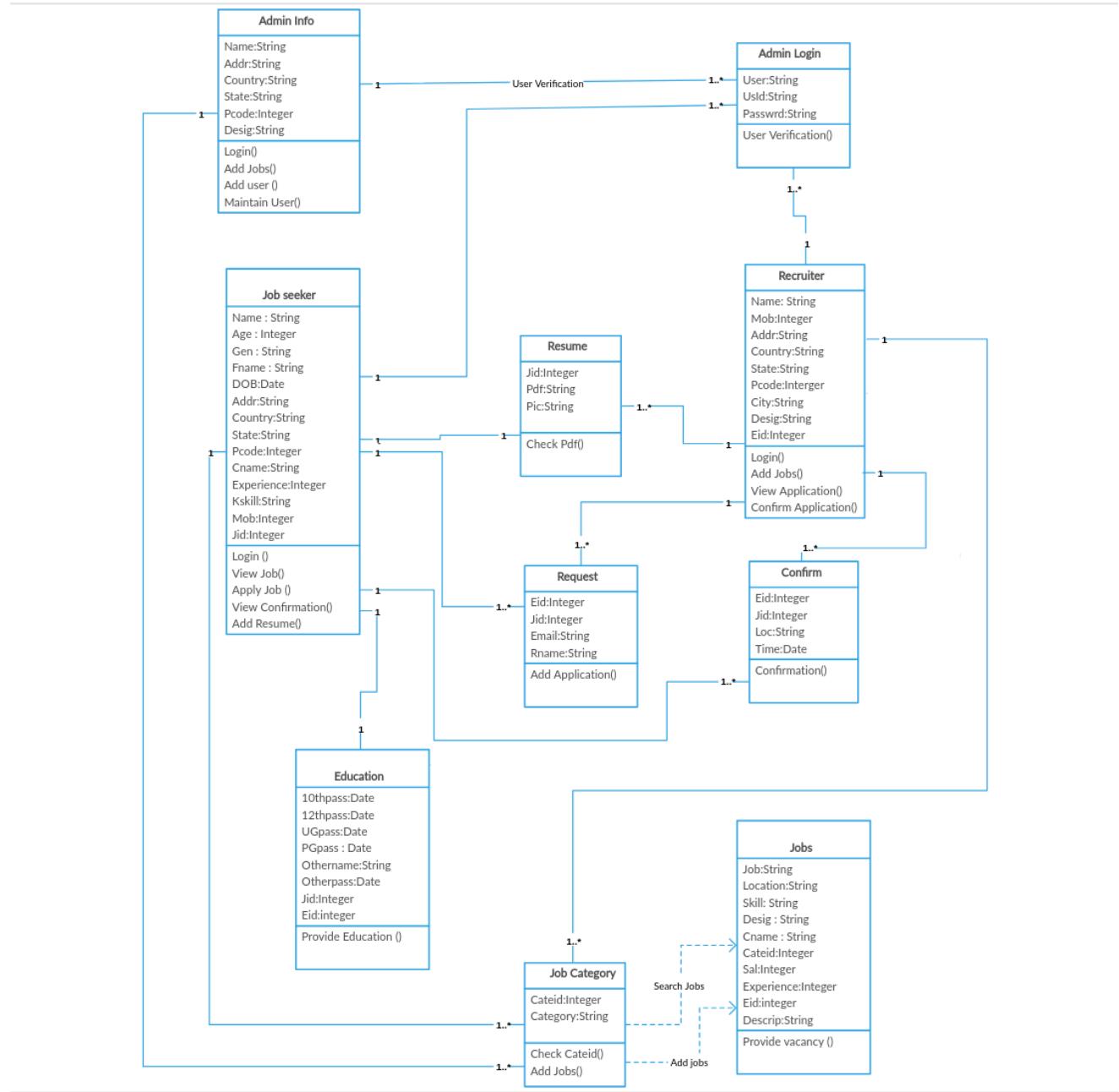


7.4 System Architecture

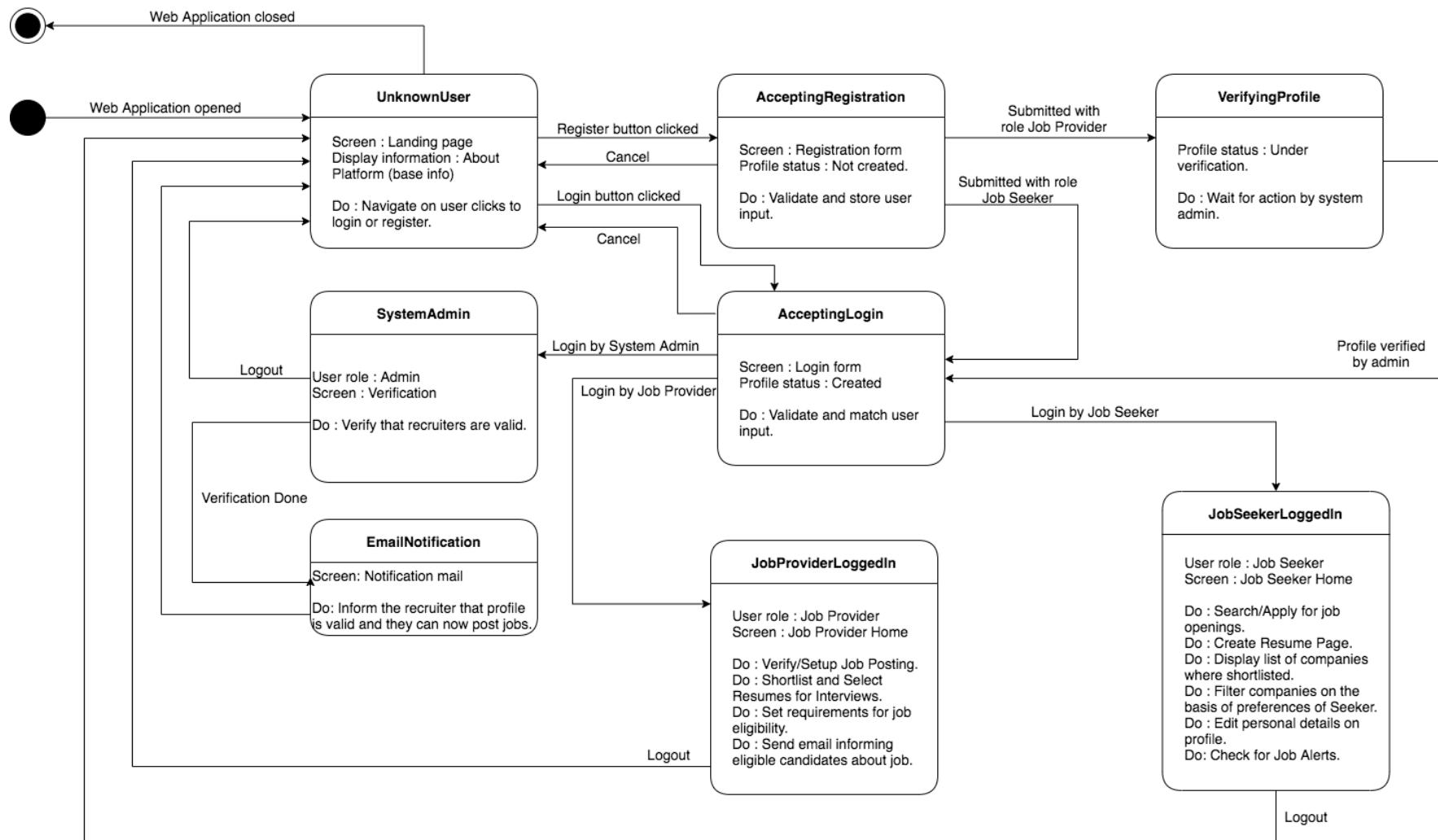


MERN STACK DEVELOPMENT

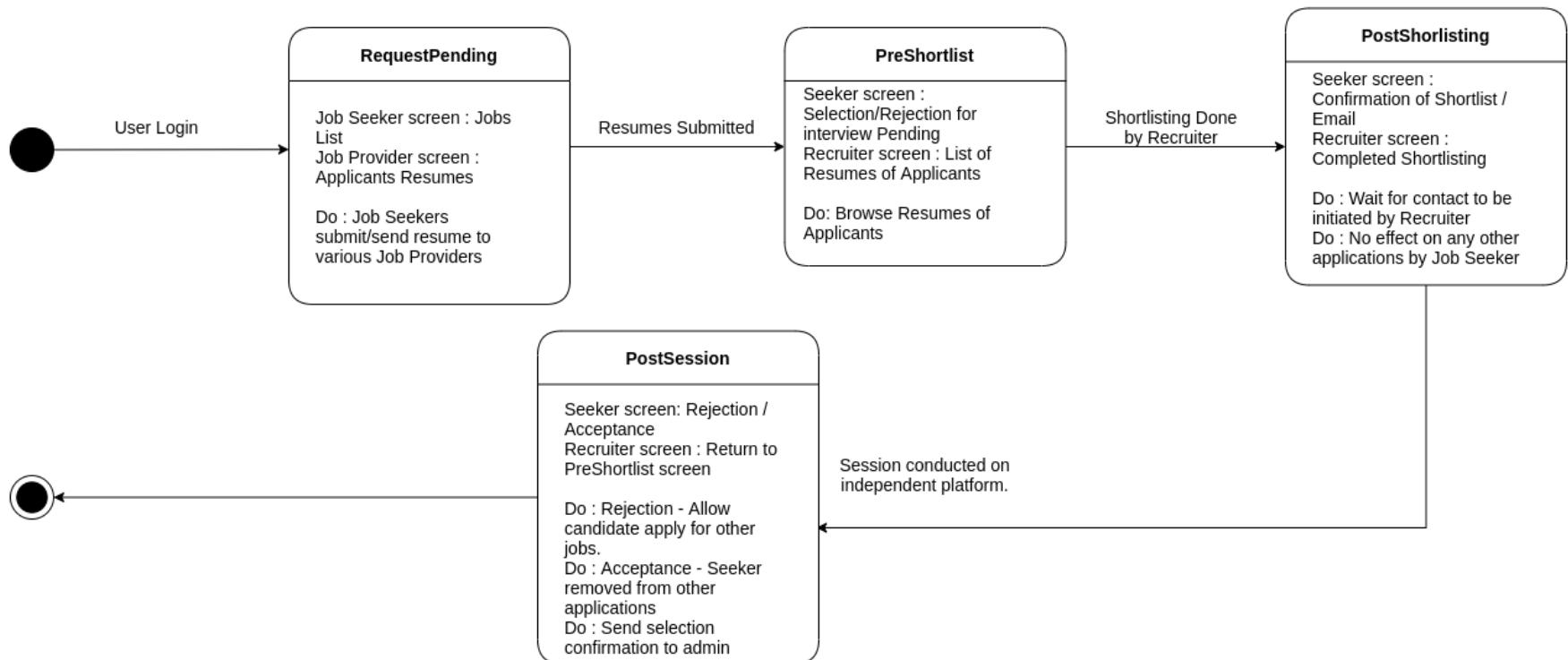
7.5 Class Diagram



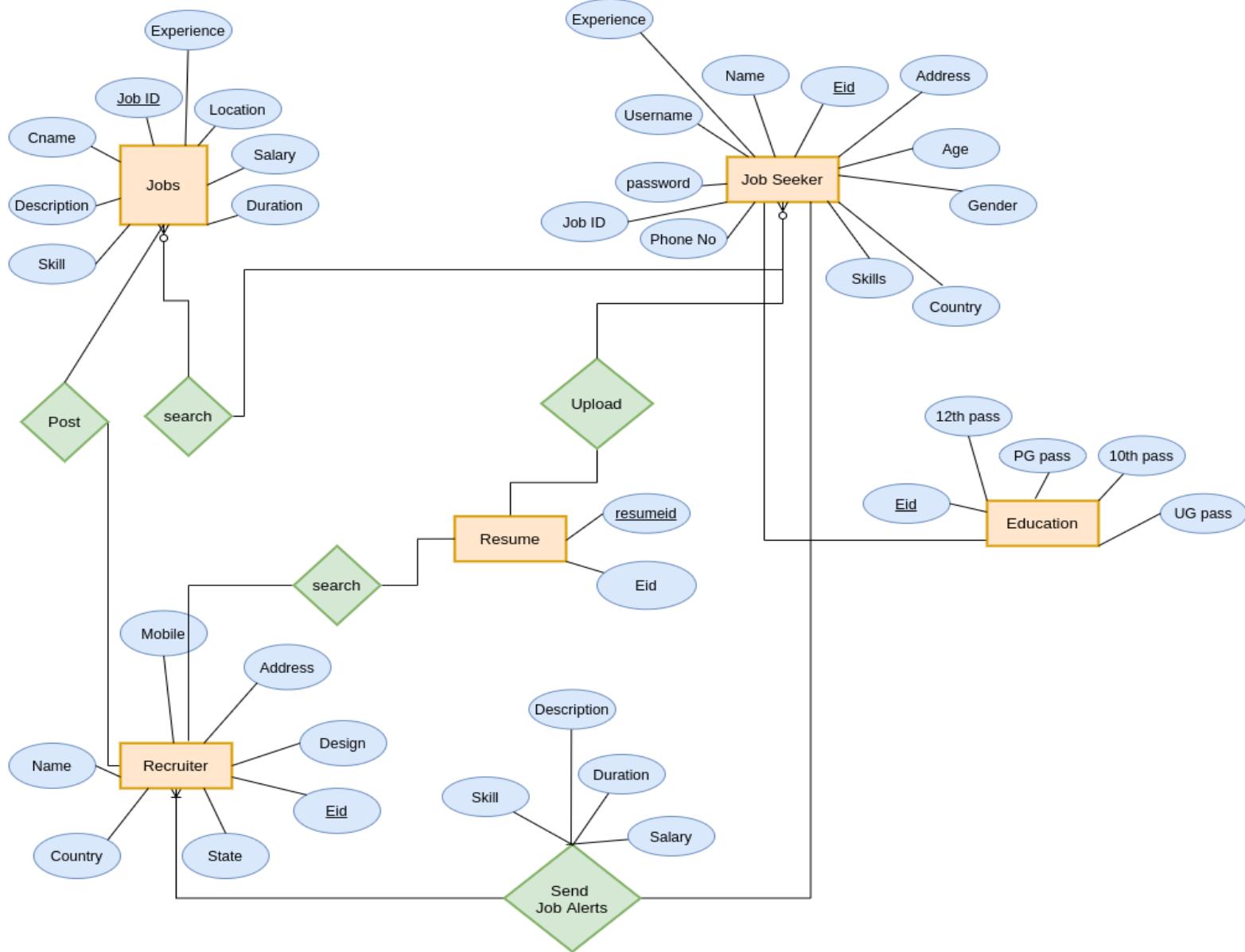
7.6 Sequence Diagram



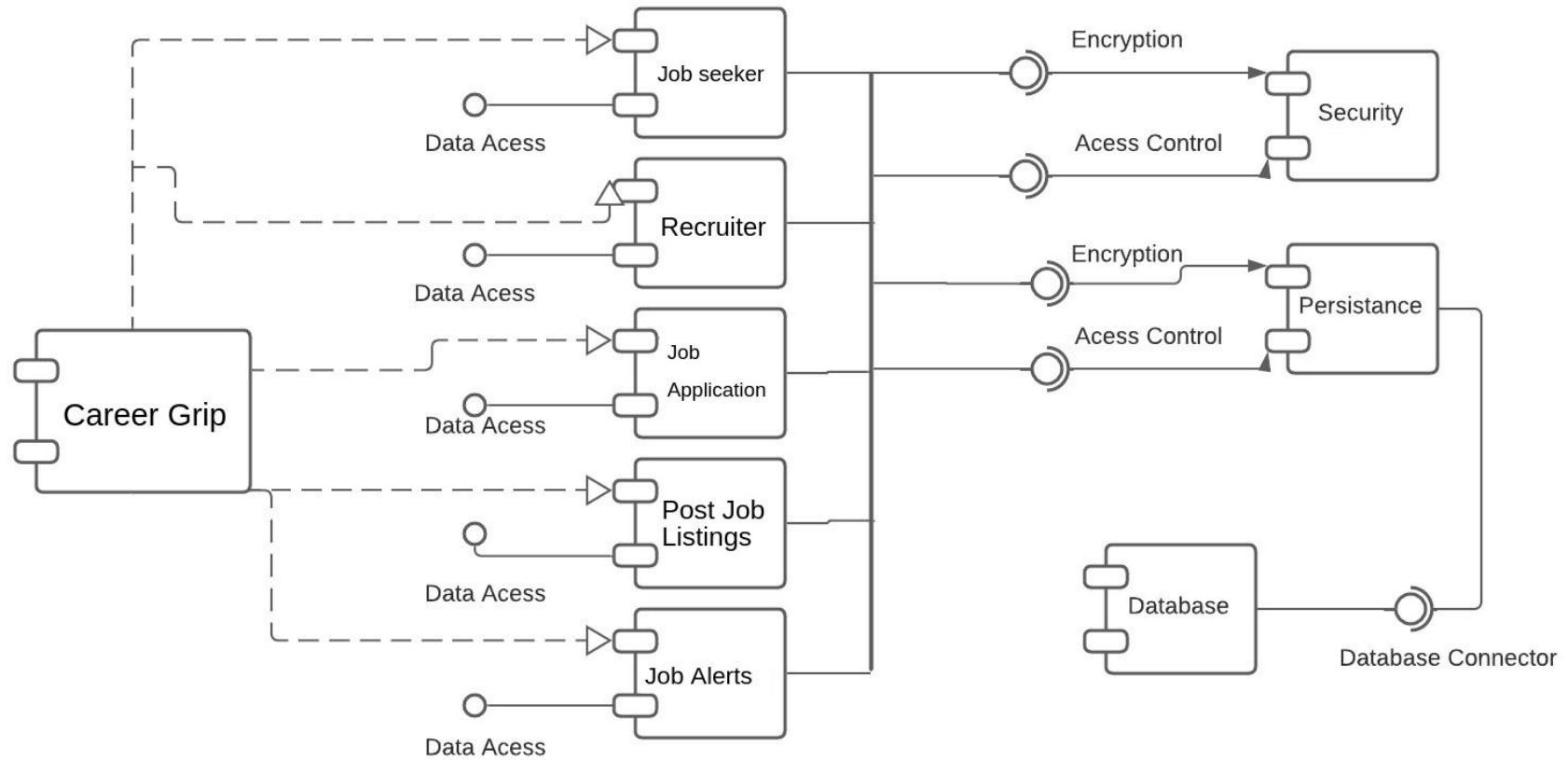
7.7 State Diagram



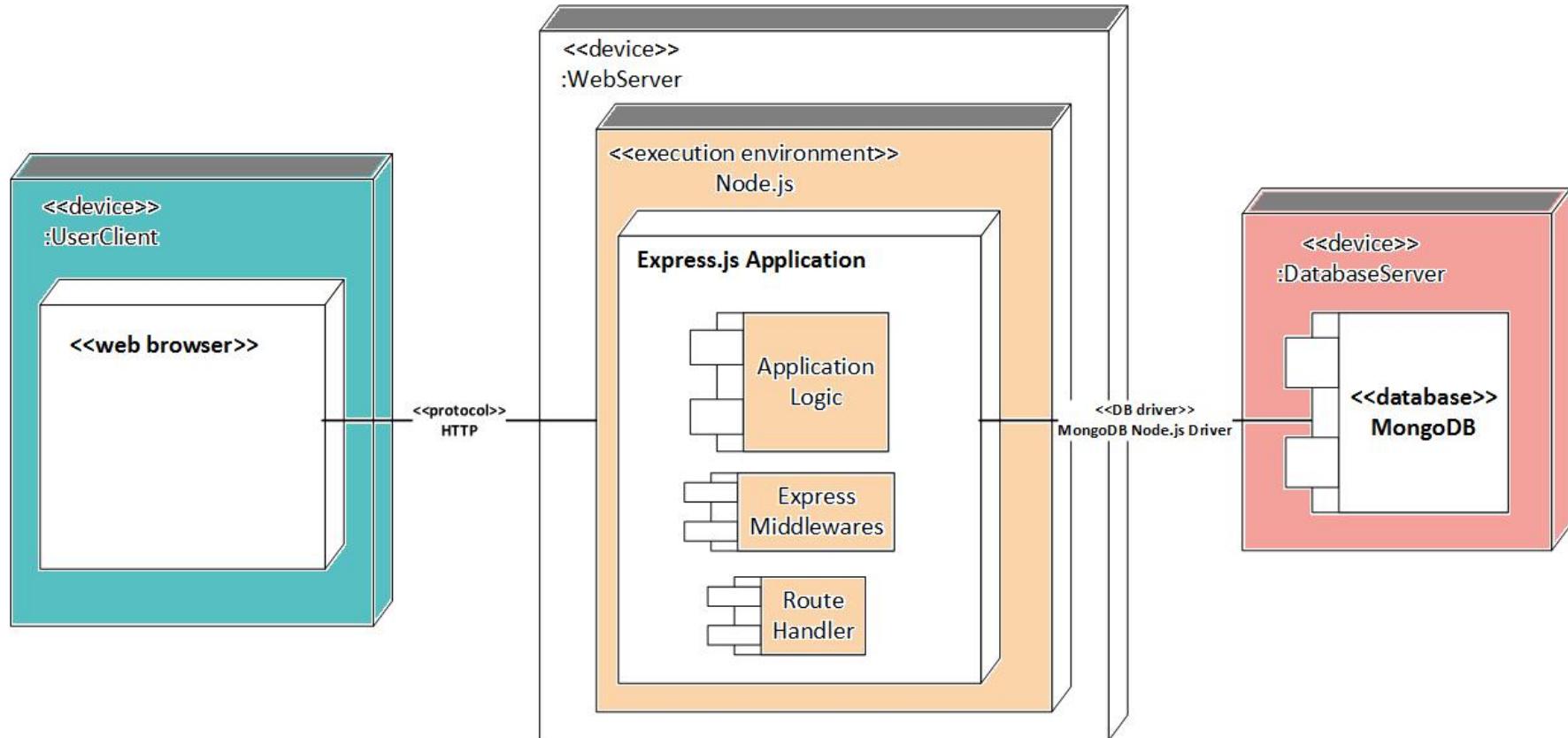
7.8 ER Diagram



7.9 Component Diagram



7.10 Deployment Diagram



8. Implementation

Coding Screenshots:

- Code Snippet: **Welcome.js**

```
1  import { Grid, Typography } from "@material-ui/core";
2  import './Welcome.css';
3
4  const Welcome = (props) => {
5    return (
6      <Grid
7        container
8        item
9        direction="column"
10       alignItems="left"
11       justify="center"
12       style={{ padding: "5%", minHeight: "93vh" }}
13     >
14       <Grid item>
15         <Typography variant="h2"><div class="myclass">Welcome to</div></Typography>
16         <Typography variant="h2"><h2><div class="myclass">Career Grip</div></h2></Typography>
17
18       </Grid>
19     </Grid>
20   );
21 };
22
23 export const ErrorPage = (props) => {
24   return (
25     <Grid
26       container
27       item
28       direction="column"
29       alignItems="center"
30       justify="center"
31       style={{ padding: "30px", minHeight: "93vh" }}
32     >
33       <Grid item>
34         <Typography variant="h2">Error 404</Typography>
35       </Grid>
36     </Grid>
37   );
38 };
39
40 export default Welcome;
41
```

- Code Snippet: **JobApplications.js**

```
61  const FilterPopup = (props) => {
62    const classes = useStyles();
63    const { open, handleClose, searchOptions, setSearchOptions, getData } = props;
64    return (
65      <Modal open={open} onClose={handleClose} className={classes.popupDialog}>
66        <Paper
67          style={{
68            padding: "50px",
69            outline: "none",
70            minWidth: "50%",
71          }}
72        >
73          <Grid container direction="column" alignItems="center" spacing={3}>
74            <Grid container item alignItems="center">
75              <Grid item xs={3}>
76                Application Status
77              </Grid>
78              <Grid
79                container
80                item
81                xs={9}
82                justify="space-around"
83                // alignItems="center"
84              >
85                <Grid item>
86                  <FormControlLabel
87                    control={
88                      <Checkbox
89                        name="rejected"
90                        checked={searchOptions.status.rejected}
91                        onChange={({event}) => {
92                          setSearchOptions({
93                            ...searchOptions,
94                            status: {
95                              ...searchOptions.status,
96                              [event.target.name]: event.target.checked,
97                            },
98                          });
99                        })}
100                      />
101                    }
102                  label="Rejected"
103                />
104              </Grid>
105            </Grid>
106          </Grid>
107        </Paper>
108      </Modal>
109    );
110  }
111
```

- Code Snippet: **CreateJobs.js**

```

32  const CreateJobs = (props) => {
33    const classes = useStyles();
34    const setPopup = useContext(SetPopupContext);
35
36    const [jobDetails, setJobDetails] = useState({
37      title: "",
38      maxApplicants: 100,
39      maxPositions: 30,
40      deadline: new Date(new Date().getTime() + 10 * 24 * 60 * 60 * 1000)
41        .toISOString()
42        .substr(0, 16),
43      skillsets: [],
44      jobType: "Full Time",
45      duration: 0,
46      salary: 0,
47    });
48
49    const handleInput = (key, value) => {
50      setJobDetails({
51        ...jobDetails,
52        [key]: value,
53      });
54    };
55
56    const handleUpdate = () => {
57      console.log(jobDetails);
58      axios
59        .post(apiList.jobs, jobDetails, {
60          headers: {
61            Authorization: `Bearer ${localStorage.getItem("token")}`,
62          },
63        })
64        .then((response) => {
65          setPopup({
66            open: true,
67            severity: "success",
68            message: response.data.message,
69          });
70          setJobDetails({
71            title: "",
72            maxApplicants: 100,
73            maxPositions: 30,
74            deadline: new Date(new Date().getTime() + 10 * 24 * 60 * 60 * 1000)
75              .toISOString()
76              .substr(0, 16),
77            skillsets: [],
78          });
79        })
80      .catch((error) => {
81        setPopup({
82          open: true,
83          severity: "error",
84          message: error.message,
85        });
86      });
87    };
88  }
89
```

- Code Snippet: **Login.js**

```
54  const handleInput = (key, value) => {
55    setLoginDetails({
56      ...loginDetails,
57      [key]: value,
58    });
59  };
60
61  const handleInputError = (key, status, message) => {
62    setInputErrorHandler({
63      ...inputErrorHandler,
64      [key]: {
65        error: status,
66        message: message,
67      },
68    });
69  };
70
71  const handleLogin = () => {
72    const verified = !Object.keys(inputErrorHandler).some((obj) => [
73      return inputErrorHandler[obj].error;
74    ]);
75    if (verified) {
76      axios
77        .post(apiList.login, loginDetails)
78        .then((response) => {
79          localStorage.setItem("token", response.data.token);
80          localStorage.setItem("type", response.data.type);
81          setLoggedin(isAuth());
82          setPopup({
83            open: true,
84            severity: "success",
85            message: "Logged in successfully",
86          });
87          console.log(response);
88        })
89        .catch((err) => {
90          setPopup({
91            open: true,
92            severity: "error",
93            message: err.response.data.message,
94          });
95          console.log(err.response);
96        });
97    }
98  };
99
```

- Code Snippet: **apiRoutes.js**

```

272  // to get info about a particular job
273  router.get("/jobs/:id", jwtAuth, (req, res) => {
274    Job.findOne({ _id: req.params.id })
275      .then((job) => {
276        if (job == null) {
277          res.status(400).json({
278            message: "Job does not exist",
279          });
280          return;
281        }
282        res.json(job);
283      })
284      .catch((err) => {
285        res.status(400).json(err);
286      });
287  });
288
289 // to update info of a particular job
290 router.put("/jobs/:id", jwtAuth, (req, res) => {
291   const user = req.user;
292   if (user.type != "recruiter") {
293     res.status(401).json({
294       message: "You don't have permissions to change the job details",
295     });
296     return;
297   }
298   Job.findOne({
299     _id: req.params.id,
300     userId: user.id,
301   })
302     .then((job) => {
303       if (job == null) {
304         res.status(404).json({
305           message: "Job does not exist",
306         });
307         return;
308       }
309       const data = req.body;
310       if (data.maxApplicants) {
311         job.maxApplicants = data.maxApplicants;
312       }
313       if (data.maxPositions) {
314         job.maxPositions = data.maxPositions;
315       }
316       if (data.deadline) {
317         job.deadline = data.deadline;

```

- Code Snippet: **Jobs.js**

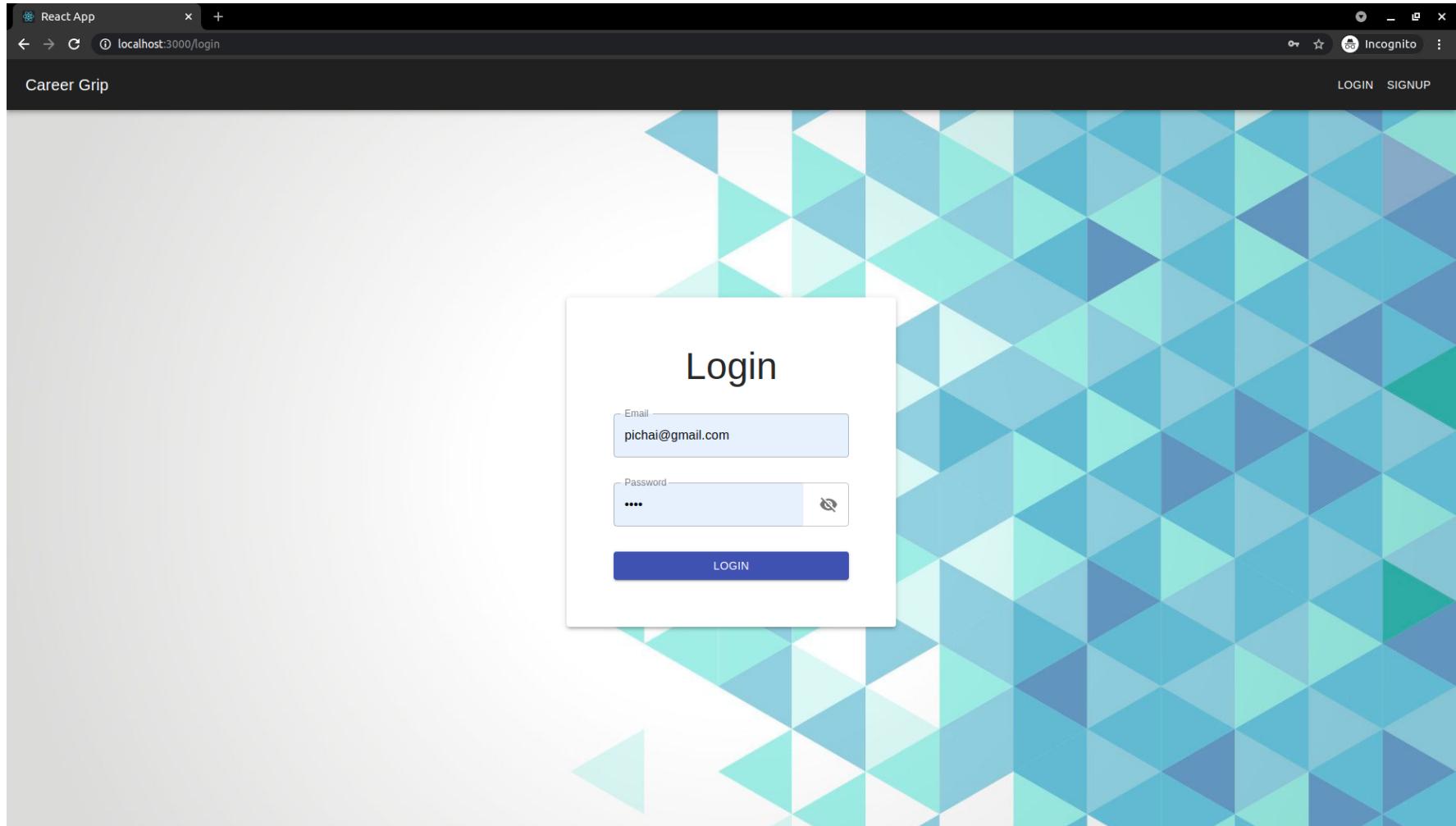
```
1  const mongoose = require("mongoose");
2
3  let schema = new mongoose.Schema(
4  {
5    userId: {
6      type: mongoose.Schema.Types.ObjectId,
7      required: true,
8    },
9    title: {
10      type: String,
11      required: true,
12    },
13    maxApplicants: {
14      type: Number,
15      validate: [
16        {
17          validator: Number.isInteger,
18          msg: "maxApplicants should be an integer",
19        },
20        {
21          validator: function (value) {
22            return value > 0;
23          },
24          msg: "maxApplicants should greater than 0",
25        },
26      ],
27    },
28    maxPositions: {
29      type: Number,
30      validate: [
31        {
32          validator: Number.isInteger,
33          msg: "maxPositions should be an integer",
34        },
35        {
36          validator: function (value) {
37            return value > 0;
38          },
39          msg: "maxPositions should greater than 0",
40        },
41      ],
42    },
43    activeApplications: {
44      type: Number,
45      default: 0,
46      validate: [
47        {
48          validator: Number.isInteger,
```

Result Screenshots:

- Home Page:



- Login Page:



- Sign-Up Page

The screenshot shows the 'Signup' page of a web application. The page has a dark header with the text 'Career Grip' on the left and 'LOGIN' and 'SIGNUP' on the right. The main content area has a light gray background. At the top center is the word 'Signup'. Below it is a form with the following fields:

- Category: A dropdown menu set to 'Applicant'.
- Name: An input field containing 'Rahul Patel'.
- Email: An input field containing 'rahul@gmail.com'.
- Password: An input field containing '****'.
- Institution Name #1: A dropdown menu set to 'IIT Delhi'.
- Start Year: A dropdown menu set to '2016'.
- End Year: A dropdown menu set to '2020'.

Below the form is a red button labeled 'ADD ANOTHER INSTITUTION DETAILS'. Underneath this button is a section for 'Skills' with four items: C, C++, Java, and Python, each with a delete icon. There is also a placeholder text 'Press enter to add skills'. Below this are two file upload fields: one for 'Resume (.pdf)' containing 'Resume.pdf' and another for 'Profile Photo (.jpg/.png)' containing 'pic.png'. At the bottom is a large blue 'SIGNUP' button.

- Edit Profile

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/profile". The page has a dark header with the "Career Grip" logo and navigation links for "HOME", "APPLICATIONS", "PROFILE", and "LOGOUT". The main content area is titled "Profile" and contains a form for editing profile details. The form includes fields for Name (Shruti Patel), Institution Name #1 (COEP) with Start Year (2015) and End Year (2019), and Institution Name #2 (Indian Institute of Science) with Start Year (2019) and End Year (2021). There is a button to "ADD ANOTHER INSTITUTION DETAILS". Below this, there is a skills section with three selected items: C, C++, and JS, each with a delete icon. A placeholder text "Press enter to add skills" is visible. There are two file upload fields: one for "Resume (.pdf)" and another for "Profile Photo (.jpg/.png)". Each upload field has a browse button (document icon) and a cloud storage icon. At the bottom of the form is a blue "UPDATE DETAILS" button. The background features a large, abstract geometric pattern of triangles in shades of blue and green.

Name
Shruti Patel

Institution Name #1
COEP Start Year
2015 End Year
2019

Institution Name #2
Indian Institute of Science Start Year
2019 End Year
2021

Skills
C C++ JS

Press enter to add skills

Resume (.pdf)

Profile Photo (.jpg/.png)

UPDATE DETAILS

- Post new Job Listing by Recruiter

The screenshot shows a web application titled "React App" running on "localhost:3000/addjob". The page has a header with "Career Grip" and navigation links for "HOME", "ADD JOBS", "MY JOBS", "EMPLOYEES", "PROFILE", and "LOGOUT". A "Incognito (2)" tab is open. The main content area is titled "Add Job" and contains a form with the following fields:

- Title: Software Developer
- Skills: C, C++, Java (each with a delete icon)
- Job Type: Full Time
- Duration: 5 Months
- Salary: 200000
- Application Deadline: 09/05/2021, 19:44 (with a calendar icon)
- Maximum Number Of Applicants: 100
- Positions Available: 2

A blue "CREATE JOB" button is located at the bottom center of the form.

- Jobs Posted by the Logged-in Recruiter

The screenshot shows a web application interface for managing jobs. At the top, there's a navigation bar with links for HOME, ADD JOBS, MY JOBS, EMPLOYEES, PROFILE, and LOGOUT. Below the navigation is a large banner with the text "My Jobs" and a search bar labeled "Search Jobs".

Job Posting 1: Junior Dev

- Role : Full Time
- Salary : ₹ 21200 per month
- Duration : 2 month
- Date Of Posting: 19/04/2021
- Number of Applicants: 100
- Remaining Number of Positions: 12

Tags: dev

Job Posting 2: Software developer

- Role : Full Time
- Salary : ₹ 56466 per month
- Duration : 4 month
- Date Of Posting: 23/04/2021
- Number of Applicants: 100
- Remaining Number of Positions: 1

Tags: C, C++

To the right of each job listing is a blue rectangular box containing three buttons:

- VIEW APPLICATIONS
- UPDATE DETAILS
- DELETE JOB

- View Applicants for the Job - for Recruiter

The screenshot displays a web application interface for managing job applications. At the top, a navigation bar includes links for HOME, ADD JOBS, MY JOBS, EMPLOYEES, PROFILE, and LOGOUT. The main title 'Applications' is centered above a grid of applicant profiles.

Tanya Patel Application:

- User icon: Placeholder profile picture.
- Name: Tanya Patel
- Rating: ★★★★★
- Applied On: 30/04/2021
- Education: IIT Kharagpur (2017-2021)
- SOP: i'm Interested
- Skills: C, C++, Java

Action Buttons:

- DOWNLOAD RESUME (blue button)
- ACCEPT (green button)
- REJECT (red button)

Rahul Singh Application:

- User icon: Placeholder profile picture.
- Name: Rahul Singh
- Rating: ★★★★★
- Applied On: 23/04/2021
- Education: COEP (2018-Ongoing)
- SOP: I have the relevant skills.
- Skills: C, C++

Action Buttons:

- DOWNLOAD RESUME (blue button)
- REJECTED (red button)

- Jobs visible to Job Seekers on the Home page

The screenshot shows a web browser window for a React App at localhost:3000/home. The page has a dark header with 'Career Grip' on the left and navigation links for HOME, APPLICATIONS, PROFILE, and LOGOUT on the right. A large blue and teal geometric background graphic is present. At the top center, the word 'Jobs' is displayed above a search bar labeled 'Search Jobs' with a magnifying glass icon. Below the search bar, two job listings are shown in white boxes:

Sr. Manager
★★★★★
Role : Full Time
Salary : ₹ 320000 per month
Duration : Flexible
Posted By : Karan Kumar
Application Deadline : 14/05/2021
Mangement Problem-solving

Software developer
★★★★★
Role : Full Time
Salary : ₹ 56466 per month
Duration : 4 month
Posted By : Sundar Pichai
Application Deadline : 03/05/2021
C C++

Each job listing includes an 'APPLY' button in a blue box on the right side.

- Job Seeker can view all their Applications

The screenshot shows a web application interface for managing job applications. At the top, a navigation bar includes links for HOME, APPLICATIONS, PROFILE, and LOGOUT. The main title 'Applications' is centered above a grid of job listings. Each listing is presented in a card format with a blue or orange status indicator on the right.

SWE
Posted By: Karan Kumar
Role : Full Time
Salary : ₹ 200000 per month
Duration : Flexible
C C++
Applied On: 30/04/2021

Software developer
Posted By: Sundar Pichai
Role : Full Time
Salary : ₹ 56466 per month
Duration : 4 month
C C++
Applied On: 30/04/2021

APPLIED

SHORTLISTED

- Automated Email to Job Seekers - Job Alerts

Career Grip - New Position for Software developer Inbox ×

 no.reply.careergrip@gmail.com
to me ▾

Check out this new position for Software developer

Deadline for Applying: Mon May 03 2021 06:09:00 GMT+0530 (India Standard Time)

Duration: 4 months

Salary: 56466

Skills: C,C++

Job Type: Full Time

Only 1 Position(s) available!

Checkout the [website](#) for applying

[I am interested in this position.](#) [Hi, I am interested.](#) [Thanks for the mail.](#)

 [Reply](#)  [Forward](#)

9. Testing

A) Black Box Testing:

1. Login Module:

Module 1 : Login Module											
Equivalence Classes		Invalid	Valid	Password	Invalid	Valid					
Expression - I(((A-Za-z){3,9})(?:\. /)?(?:[-:&=!@\$.!w]+ @) ?A-Za-z0-9.-)(? :WWW. [-:&=!@\$.!w]+ @)(A-Za-z0-9.-)+)(?: V[!~@!W-]?)? (?: V[!~@!W-]?)? (?: [-:&=!@\$.!w-]?)# (?:[!w"]?)")"	Does not follow expression (Including NULL)	Follows expression			Password not in the database	Password in the database					
Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status	
TS(CG)_001	Verify the login functionality of Career Grip	TC(CG)_Login_001	Enter an email which does not follow the expression and correct password	1. Enter an email which follows expression 2. Enter a Password 3. Click a Login Button	valid test url & server running	Email : rohit@ Password: abc	User should be in the same login page and be able to see "Incorrect email format"	Incorrect email format	Incorrect email format	PASS	
TS(CG)_001	Verify the login functionality of Career Grip	TC(CG)_Login_002	Enter an email which does not follow the expression and incorrect password	1. Enter an email which follows expression 2. Enter a Password 3. Click a Login Button	valid test url & server running	Email: rohit Password: lkjdfkasjdfklksadjf#4223lkjdf4	User should be in the same login page and be able to see "Incorrect email format"	Incorrect email format	Incorrect email format	PASS	
TS(CG)_001	Verify the login functionality of Career Grip	TC(CG)_Login_003	Enter an email which follows expression and incorrect password	1. Enter an email which follows expression 2. Enter a Password 3. Click a Login Button	valid test url & server running	Email : rohit@gmail.com Password: abc	User should be in the same login page and be able to see "Password is incorrect."	Password is incorrect.	Password is incorrect.	PASS	
TS(CG)_001	Verify the login functionality of Career Grip	TC(CG)_Login_004	Enter an email which follows expression and correct password	1. Enter an email which follows expression 2. Enter a Password 3. Click a Login Button	valid test url & server running	Email : rohit@gmail.com Password: new@123	User should be redirected to the home page and be able to see "Logged in Sucessfully"	Logged in Sucessfully	Logged in Sucessfully	PASS	

2. Register Module:

Module 2 : Register Module										
Equivalence Classes										
Name	Invalid	Valid	Email	Invalid	Valid	Password	Invalid	Valid		
	Empty	Non-empty	/((([A-Za-z]{3,9};(?:\w{ })?) (?:[-\w+=\\$\w]+@)?[A- Za-z0-9.-]+ (? www.[:& =\!\$,\w]+@)(A-Za-z0-9. -+)(? V[\t-\r\n%V.\w-])?" ??(? [-\w+=\%@\w-])# ?(?:[w"]*))?)?	Does not follow expression (including NULL)	Follows expression		Empty	Non-empty		
Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status
TS(CG)_002	Verify the signup functionality of Career Grip	TC(CG)_Signup_001	Enter an email which does not follow the expression and new password	1. Click on "SIGNUP". 2. Upload resume and profile photo. 3. Fill your name. 4. Enter email which follows the expression 5. Create new password 6. Enter Institution details 6.Click on "Sign Up"	Valid url and Server running	Email : pichai@ Password: abc	User should be able to see "Incorrect email format"	Incorrect email format	Incorrect email format	PASS
TS(CG)_002	Verify the signup functionality of Career Grip	TC(CG)_Signup_002	Enter an email which does not follow the expression and empty password	1. Click on "SIGNUP". 2. Upload resume and profile photo. 3. Fill your name. 4. Enter email which follows the expression 5. Create new password 5. Enter Institution details 6.Click on "Sign Up"	Valid url and Server running	Email: pichai Password:	User should be able to see "Password is required" and "Incorrect email format"	Password is required and Incorrect email format	Password is required and Incorrect email format	PASS
TS(CG)_002	Verify the signup functionality of Career Grip	TC(CG)_Signup_003	Enter an email which follows expression and empty password	1. Click on "SIGNUP". 2. Upload resume and profile photo. 3. Fill your name. 4. Enter email which follows the expression 5. Create new password 5. Enter Institution details 6.Click on "Sign Up"	Valid url and Server running	Email : pichai@gmail.com Password:	User should be able to see "Password is required"	Password is required	Password is required	PASS
TS(CG)_002	Verify the signup functionality of Career Grip	TC(CG)_Signup_004	Enter an email which follows expression and new password	1. Click on "SIGNUP". 2. Upload resume and profile photo. 3. Fill your name. 4. Enter email which follows the expression 5. Create new password 5. Enter Institution details 6.Click on "Sign Up"	Valid url and Server running	Email : pichai@gmail.com Password: new@123	User should be able to see "Logged in successfully" and be redirected to the home page	Logged in Successfully	Logged in Successfully	PASS

Career Grip

3. Search Module:

Module 3: Search Module										
Equivalence Classes										
Search Box	Invalid	Valid								
Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status
TS(CG)_003	Verify the Search functionality of Career Grip	TC(CG)_Application_001	Search for "Software" Job which is already present in the database.	1. Applicant will Login. 2. Click on "Home" button on Nav Bar. 3. Search in the textbox.	Connected to server, Logged-in Applicant	"Software" is given as input in the textbox.	User will see a job for "Software Developer".	User will see relevant jobs.	User will see relevant jobs.	PASS
TS(CG)_003	Verify the Search functionality of Career Grip	TC(CG)_Application_002	Search for "Mining" Job which is not present in the database.	1. Applicant will Login. 2. Click on "Home" button on Nav Bar. 3. Search in the textbox.	Connected to server, Logged-in Applicant	"Mining" is given as input in the textbox.	User will be shown "No jobs found".	"No jobs found" as result	"No jobs found" as result	PASS
TS(CG)_003	Verify the Search functionality of Career Grip	TC(CG)_Application_003	Sort the Jobs to see Highest salary job first.	1. Applicant will Login. 2. Click on "Home" button on Nav Bar. 3. Sort by Salary in descending order.	Connected to server, Logged-in Applicant	Salary checkbox is marked.	User will see the jobs in descending order of their salary.	Jobs displayed in descending order of Salary.	Jobs displayed in descending order of Salary.	PASS

4. Profile Module:

Module 4: Profile Module										
Equivalence Classes										
Graduation Period	Invalid	Valid	Resume	Invalid	Valid					
	Start year < 1930	Start year > 1930 and graduation year < 2030		Any file format other than pdf	PDF file					
Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status
TS(CG)_004	Verify the Profile functionality of Career Grip	TC(CG)_Profile_001	Upload a pdf file in Resume part of Profile tab.	1. Applicant will Login. 2. Click on "Profile" button on Nav Bar. 3. Update resume. 4. Enter you College Name 5. Enter year of Entry and Graduation year of your college	Connected to server, Logged-in Applicant	Upload "resume.pdf"	User will see an alert stating File uploaded successfully.	Alert stating File uploaded successfully.	Alert stating File uploaded successfully.	PASS
TS(CG)_004	Verify the Profile functionality of Career Grip	TC(CG)_Profile_002	Upload a docx file in Resume part of Profile tab. (which is not allowed)	1. Applicant will Login. 2. Click on "Profile" button on Nav Bar. 3. Update resume. 4. Enter you College Name 5. Enter year of Entry and Graduation year of your college	Connected to server, Logged-in Applicant	Upload "resume.docx"	User will see an alert stating Bad request.	Alert stating Bad request.	Alert stating Bad request.	PASS
TS(CG)_004	Verify the Profile functionality of Career Grip	TC(CG)_Profile_003	Giving Start year greater than the end year	1. Applicant will Login. 2. Click on "Profile" button on Nav Bar. 3. Enter you College Name 4. Enter year of Entry and Graduation year of your college	Connected to server, Logged-in Applicant	Start year : 2016 and End year : 2012	User will see an alert stating that JobApplicantInfo validation failed: education.0.endYear: End year should be greater than or equal to Start year	Alert stating that JobApplicantInfo validation failed: education.0.endYear: End year should be greater than or equal to Start year	Alert stating that JobApplicantInfo validation failed: education.0.endYear: End year should be greater than or equal to Start year	PASS
TS(CG)_004	Verify the Profile functionality of Career Grip	TC(CG)_Profile_004	Giving Start year less than 1930	1. Applicant will Login. 2. Click on "Profile" button on Nav Bar. 3. Enter you College Name 4. Enter year of Entry and Graduation year of your college	Connected to server, Logged-in Applicant	Start year : 1730	User would be able to see an alert stating that JobApplicantInfo validation failed: education.0.startYear: Path 'startYear' (1730) is less than minimum allowed value (1930).	Alert stating that JobApplicantInfo validation failed: education.0.startYear: Path 'startYear' (1730) is less than minimum allowed value (1930).	Alert stating that JobApplicantInfo validation failed: education.0.startYear: Path 'startYear' (1730) is less than minimum allowed value (1930).	PASS

B) White Box Testing

1. Login Module

Module 1 : Login Module											
Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Code Snippet	Status	
CG_001	Verify the login functionality of Career Grip	CG_Login_001	If user enters a email that is not present in database	Connected to server	Email: john@gmail.com password: test	User will see an alert stating that the User does not exist	Alert stating that User does not exist	Alert stating that User does not exist	apiRoutes.js: router get - /user/part. if part for recruiter and else for applicant.	PASS	
CG_001	Verify the login functionality of Career Grip	CG_Login_002	If user is present and password doesn't match	Connected to server	Email: john@gmail.com password: test	User will see an alert stating that the Password is incorrect.	Unsuccessful Login.	Unsuccessful Login.	passportConfig.js: password.use() part , catch exception	PASS	
CG_001	Verify the login functionality of Career Grip	CG_Login_003	Email and corresponding password is right	Connected to server	Email: john@gmail.com password: test	User will be Logged in successfully and can access options in the nav bar.	Logged in successfully	Logged in successfully	Login.js: if part of handleLogin()	PASS	
CG_001	Verify the login functionality of Career Grip	CG_Login_004	The client is not connected to server or the server is unreachable	Disconnect the database server	Email: john@gmail.com password: test	User would not be able to login as Server is unreachable	Server is unreachable	Server is unreachable	Data fields missing, as mongod service is down	PASS	

2. Job Application Module

Module 2: Job Application Module											
Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Code Snippet	Status	
CG_002	Verify the Job Application functionality of Career Grip	CG_Application_001	Job Seeker applies for a Job listing.	Connected to server, Logged-in Applicant	Profile details of user, SOP	User will see an alert stating that Job application successful	Alert stating that Job application successful	Alert stating that Job application successful	apiRoutes.js: router post - /jobs/:id/applications part. if application.save() successful.	PASS	
CG_002	Verify the Job Application functionality of Career Grip	CG_Application_002	Job Seeker applies for a Job listing without SOP	Connected to server, Logged-in Applicant	Profile details of user	User will see an alert stating that Job application successful	Alert stating that Job application successful	Alert stating that Job application successful	apiRoutes.js: router post - /jobs/:id/applications part. if application.save() successful.	PASS	
CG_002	Verify the Job Application functionality of Career Grip	CG_Application_003	Job Seeker applies for a Job which was previously applied to.	Connected to server, Logged-in Applicant	Profile details of user	User will see an alert stating that You have already applied for this job	Alert stating that You have already applied for this job	Alert stating that You have already applied for this job	apiRoutes.js: router post - /jobs/:id/applications part. if (appliedApplication != null) is true.	PASS	
CG_002	Verify the Job Application functionality of Career Grip	CG_Application_004	View Applications - Button provided in NavBar	Connected to server, Logged-in Applicant	Authenticated Applicant	User would be able to view all the Jobs which were previously applied to.	List of Applications	List of Applications	apiRoutes.js: router get - /applications part. Displays all the applications.	PASS	

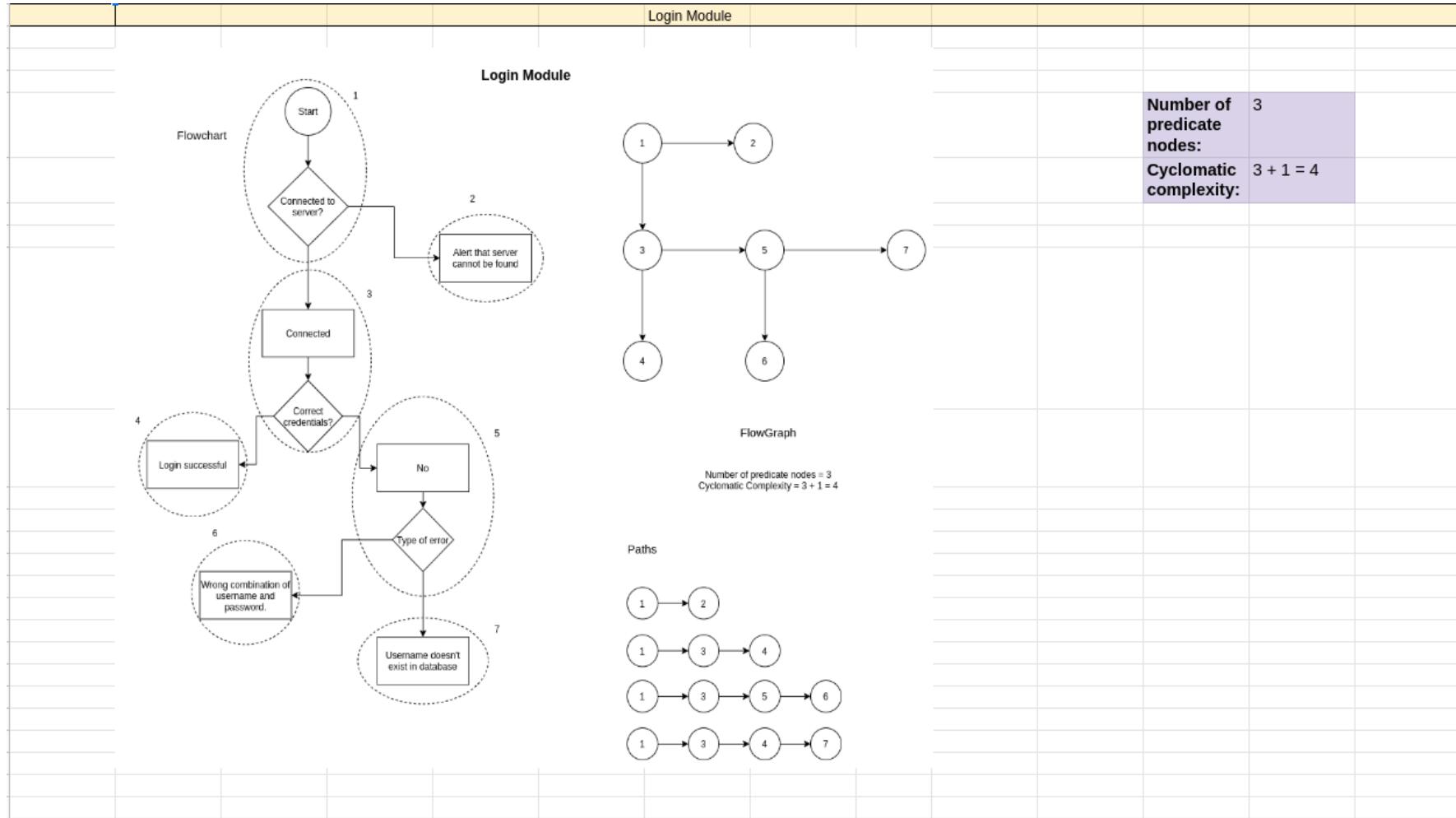
3. Recruiter Module

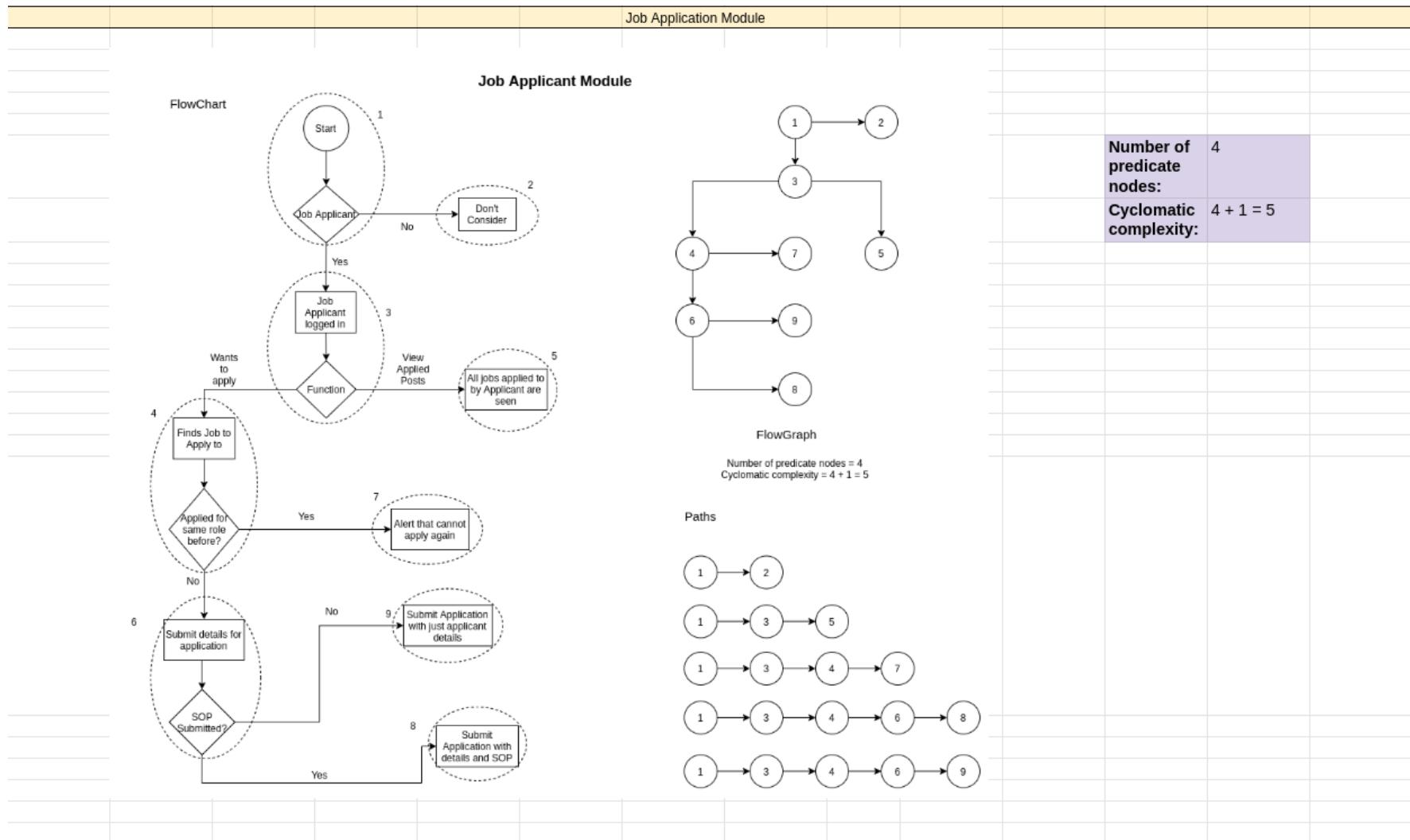
Module 3: Recruiter Module											
Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Code Snippet	Status	
CG_003	Verify the Recruiter functionality of Career Grip	CG_Recruiter_001	Recruiter posts a Job listing.	Connected to server, Logged-in Recruiter	Job Details - Title, Skills, Job Type, Duration, Salary, Deadline, Max Applicants, Available Positions	User will see an alert stating that Job added successfully to the database.	Alert stating that Job added successfully to the database	Alert stating that Job added successfully to the database	apiRoutes.js: router post - /jobs part. If job.save() is successful.	PASS	
CG_003	Verify the Recruiter functionality of Career Grip	CG_Recruiter_002	Posting with Incomplete Job Details - without Title	Connected to server, Logged-in Recruiter	Job Details without Title field	User will see an alert stating that some fields are required	Alert stating that jobs validation failed: title: Path 'title' is required.	Alert stating that jobs validation failed: title: Path 'title' is required.	apiRoutes.js: router post - /jobs part. If job.save() fails, exception occurred. catch part executes.	PASS	
CG_003	Verify the Recruiter functionality of Career Grip	CG_Recruiter_003	My Jobs - Button provided in NavBar. It is used to view all the jobs posted by the recruiter.	Connected to server, Logged-in Recruiter, Must have posted atleast one Job	Authenticated Recruiter	User would be able to view all the jobs posted by the recruiter previously.	List of all Job Listings with their Job Details.	List of all Job Listings with their Job Details.	apiRoutes.js: router get - /jobs part. Displays all the jobs posted by the recruiter.	PASS	
CG_003	Verify the Recruiter functionality of Career Grip	CG_Recruiter_004	Check if Recruiter can shortlist applicants.	Connected to server, Logged-in Recruiter, Applicant has already applied for the Job	Authenticated Recruiter	User can see the applicants for a particular job and Shortlist/Accept/Reject them.	List of all Applicants for the Job and option to shortlist them.	List of all Applicants for the Job and option to shortlist them.	JobApplications.js: in const updateStatus, status is set to shortlisted.	PASS	

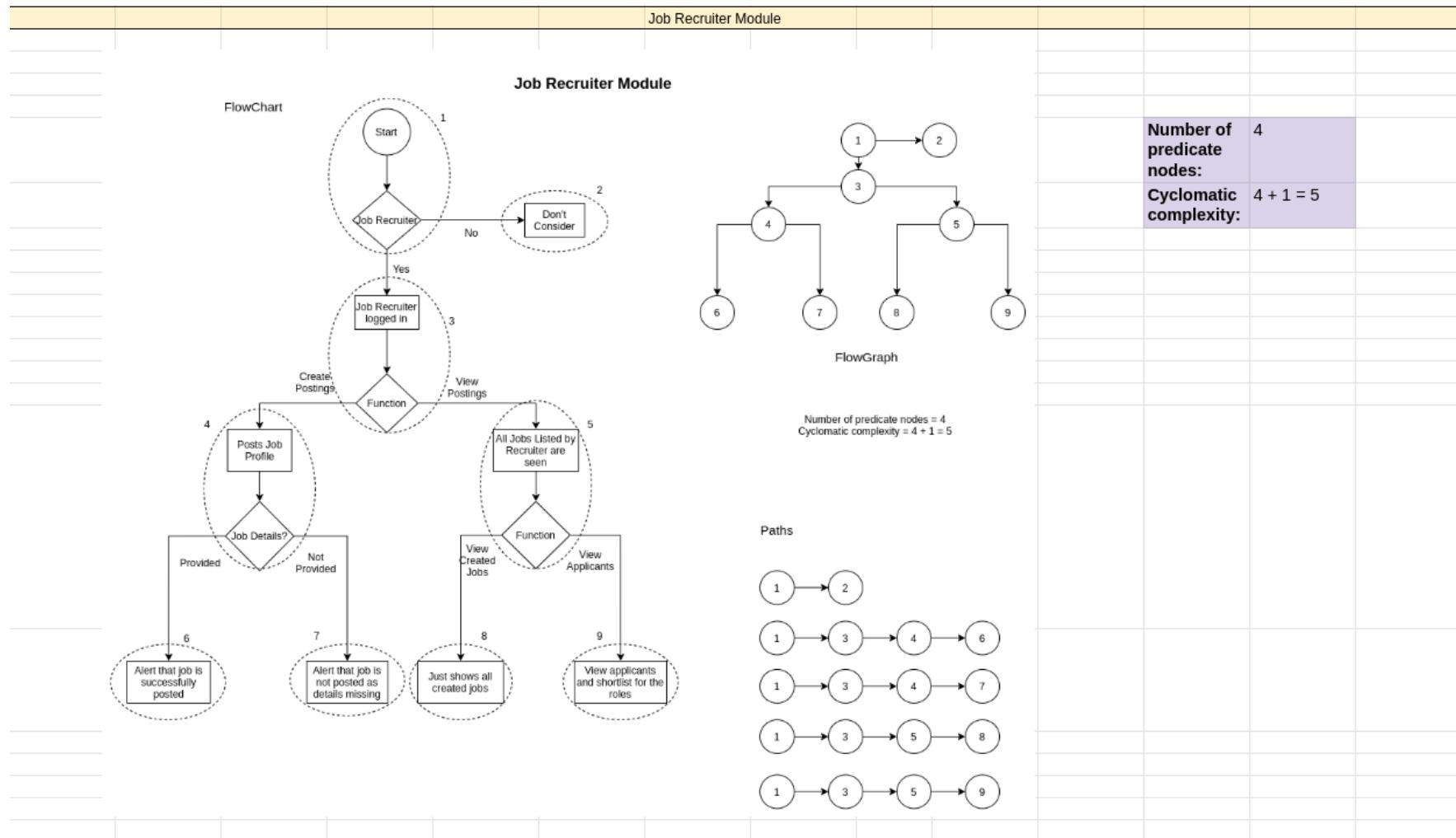
4. Job Alerts

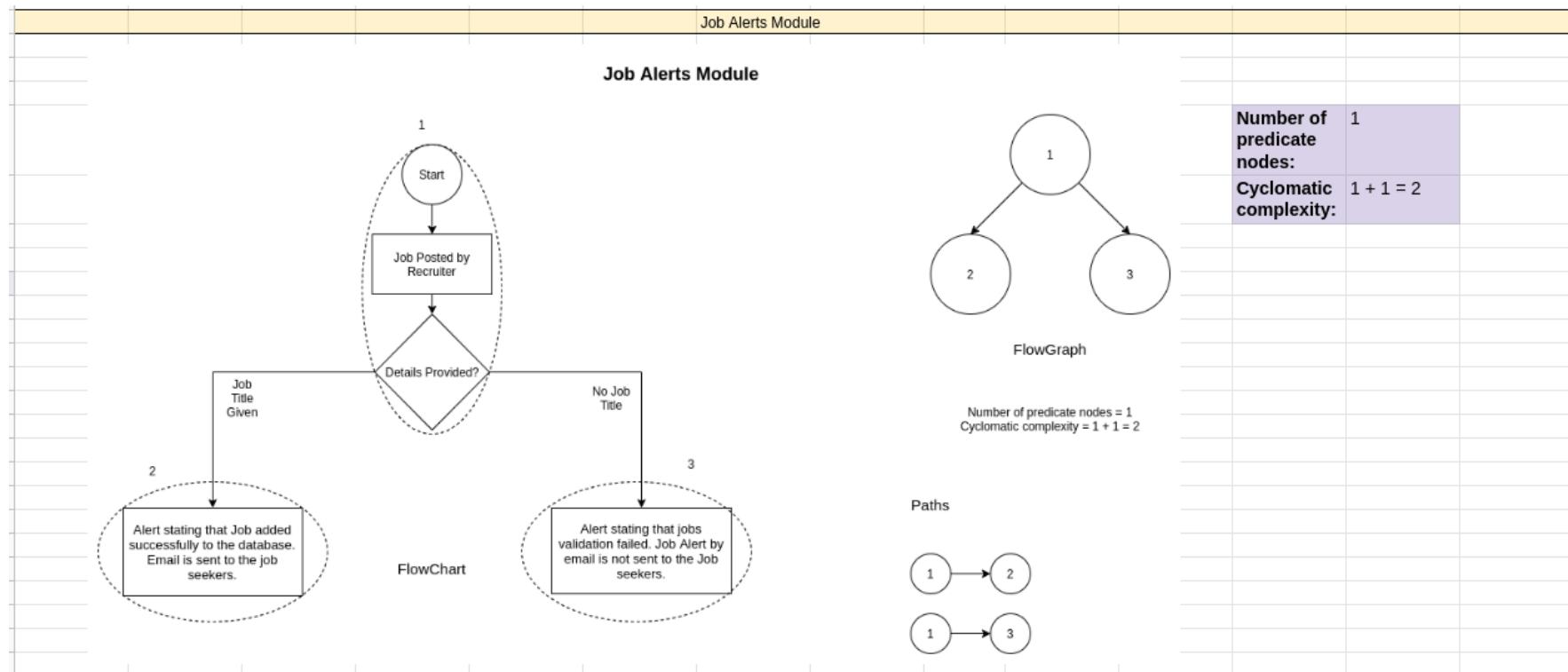
Module 4: Job Alerts											
Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Code Snippet	Status	
CG_004	Verify the Job Alerts functionality of Career Grip	CG_Alert_001	Check when Recruiter posts a Job listing. If Job Alerts are sent to the Applicants.	Connected to server, Logged-in Recruiter	Job Details - Title, Skills, Job Type, Duration, Salary, Deadline, Max Applicants, Available Positions	The Job seekers will receive an email with the Job Details in their Inbox.	Alert stating that Job added successfully to the database. Email is sent to the job seekers.	Alert stating that Job added successfully to the database. Email is sent to the job seekers.	apiRoutes.js: router post - /jobs part. After the job is successfully added, email is sent to the job seekers using nodemailer module.	PASS	
CG_004	Verify the Job Alerts functionality of Career Grip	CG_Alert_002	Posting with Incomplete Job Details - without Title and verifying that Job Alert is not sent.	Connected to server, Logged-in Recruiter	Job Details without Title field	User will see an alert stating that some fields are required	Alert stating that jobs validation failed: title: Path 'title' is required. Job Alert by email is not sent to the Job seekers.	Alert stating that jobs validation failed: title: Path 'title' is required. Job Alert by email is not sent to the Job seekers.	apiRoutes.js: router post - /jobs part. If job.save() fails, exception occurred. catch part executes.	PASS	

C) Path Diagrams









10. Github Link

- <https://github.com/adarshninin777/career-grip/>

11. References

- The Potential for Using Online Job-Portal -
<http://documents1.worldbank.org/curated/en/370301486667802852/pdf/WPS7966.pdf>
- Research Paper - http://ijirt.org/master/publishedpaper/IJIRT144246_PAPER.pdf
- Efficacy of Job Portals - <https://garph.co.uk/ijarmss/july2013/15.pdf>
- IEEE SRS template - https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc
- Mern Stack Tutorial - <https://www.geeksforgeeks.org/mern-stack/>
- Mern stack Tutorial - <https://blog.logrocket.com/mern-stack-tutorial/>
- Youtube Mern Stack Course - <https://www.youtube.com/watch?v=7CqJlxBYj-M&t=1328s>
- Node JS Docs - <https://nodejs.org/en/docs/>
- MongoDB database Documentation - <https://docs.mongodb.com/>