Group Project Log

Note: *all information must be filled out. You must hand in the project log along with each group project deliverable for this course (e.g., milestones, proposals, reports). The percentage of work allocated to each group member must add up to 100%.*

Group Name:	Web Group-21

Group Members:	6

Deliverable:	TrackMyJobs Technical Report

Group Member Name	Banner ID	Work Done
Parth Bagaria	B00839783	16.66%
Jan Chayopathum	B00837398	16.66%
Anudish Jinturkar	B00839915	16.66%
Roshan K. Patel	B00853917	16.66%
Tejas Patel	B00846296	16.66%
Zankrut B.Thakkar	B00856858	16.66%
Total Contribution		100%

TECHNICAL REPORT

TRACKMYJOBS

Members and Contributors

Parth Bagaria
B00839783
pr270476@dal.ca

Jan Chayopathum
B00837398
nw814986@dal.ca

Anudish Jinturkar
B00839915
an455254@dal.ca

Roshan K. Patel
B00853917
roshan.patel23@dal.ca

Tejas Patel
B00846296
tejas.patel@dal.ca

Zankrut B.Thakkar
B00856858
zn834201@dal.ca

Faculty of Computer Science
Dalhousie University

07th August 2020

ABSTRACT

The purpose of TrackMyJobs is to help the users search their dream job and keep track of their job applications to help users manage their job applications more effectively. Moreover, TrackMyJobs targets to help users manage and organize activities related to job applications by using job boards, activities, notes, and notification, which can keep them notified and easy to follow up with their job applications. TrackMyJobs also provide blog which will help users to find tips for job application. It is a feature that users can share their experience with others.

KEYWORDS

Angular Application, Activities, Analytics, Blogs, Job Application, Job Notification, Job Search, Job Tracking, Profile Management, Session Management, Web Application Development

CONTENTS

TABLE OF FIGURES

1. Introduction

TrackMyJobs is a web application for users to keep track of their job application. For current version, users can search job availability in Canada. Moreover, users can add activities, notes, and notification for tracking the job application. Users can share their experience as well as get tips from others with blog feature that provided within web application. This report is developed from project proposal [1].

1.1 Live Project URL

Live project URL: https://track-my-jobs.herokuapp.com/home

GitHub repository (Front-end): https://github.com/Zankrut97/grp21-trackmyjobs

GitHub repository (Back-end): https://github.com/parth1229/web-project-jobtracker

Background

1.2 Competitive Landscape

According to [2], there are variety of alternative in organizing job search. We categorize our competitors into direct and indirect competitors.

The direct competitor is job search organizer website. The job search organizer website helps users to organize their job application. For example, Huntr [3] provides user-friendly user interface (UI) and job search in one place. Users can manage their activities and analyze their job search. It also provides inspired talks to motivate and share their stories.

The indirect competitor is job search site. The job search site is a source of information for job search. Users search their interested jobs in here. Site retrieves the list of job currently available for applying. The well-known job search sites are LinkedIn [4], Indeed [5], and Monster [6]. However, users cannot manage and organize their job application in these sites.

1.3 Problem and Approach **[1]**

The primary purpose of the web application is to create an easy to manage job application tracking system. The web application will help to manage and track the user’s job applications. The application will allow users to move jobs across various stages (Resume Submitted, Interview scheduled, Offer Acceptance).

Let us imagine a scenario, Bob (A user) is actively looking for a web developer job. Bob applies to some companies for that position. However, it becomes more difficult for bob to manage his job application status and interview dates. He tries to use google sheets, but it takes lots of time to format and add contextual information. Due to this, one day, he missed a significant interview. To solve this kind of problem, we are developing this web application in which bob can manage his job application, and bob will be notified via pop notification whenever any deadline for the activity is near.

Let us take one more example to have a clear idea about our intention for creating this web application. Imagine, Sam is a business analyst at a small company in Toronto. He wants to change his company for some reason. So that he started looking for an employer who is currently hiring; however, it takes him lots of time to go to the individual company’s website and check whether they are hiring or not. We want to reduce this searching time by creating a platform where users can filter their job search and find all the list of employers who are currently hiring.

Our web application will help the user manage interview dates and deadlines (i.e. Document submission, exp. letter). The web application will also populate various jobs from the website, making it easier for them to apply for a job and manage job status. Our main motive for creating this application is to make a one-stop website for all users looking for a job in a domain and who wants to manage and track their web application.

2. Application Details

TrackMyJobs has 12 features which are profile management, job search, job discovery, job application management, job board management, job application analytics, blog management, blog creation, job application activities, notification management, job application notes, and contact management.

2.1 Target User Insights

The primary target users are millennials (age around 22–38) which familiar with technology. Their education is post-secondary education. They are looking for professional career in well-known company in their desired industry. Some people just enter the industry. Therefore, they will need tool to help them manage their job application so they will have more time to prepare for an interview. [1]

On the other hand, some people have many responsibilities such as job and young family. Therefore, they will need tool to help them organize their schedule so they can manage their time more effectively.

2.2 User-Centered Design Approach

While creating the web pages, our focus was on making the web pages visually appealing and not to clustered. To make the website looks more visually and aesthetically appealing, we used consistent color, image, and typography. Apart from this, we have also incorporated angular material design and bootstrap in the web pages. For better user experience, the form and the input text used in all the web pages are responsive and provide front-end validation.

Nowadays, it becomes an industry standard to create a responsive web design. Responsive web design helps to create web pages which reacts to the size of user screen. It also helps to optimize user's browsing experience by creating a flexible and responsive web pages. The list below provides our design consideration for web application:

Use of White Space: We make sure that we use the white space of the web page effectively. Because whenever a user looks at the design with well-composed space, the user can effectively evaluate the design and focus on the important text which improve ability to scan through web page. [1]

Feature Exposure: While creating the overall structure of home page, we make sure that the user can see all the feature of web application without going into sub section of the pages. we have done this using images and icons to show the feature to the user.

Use of Heading: We make our layout skimmable. To achieve this, we use various heading and sub-heading to get the user attention on a specific part. [1]

Simplicity: We make the website look simple and easy-to-use. Because sometimes adding too much text or animation creates a confusing website. Moreover, they are hard to understand. Therefore, we create a web application with minimal text and animation.

Navigation: This is possibly the most important part of all the web pages. We make sure that the web page navigation is easy to find and comprehend. We put a consistent navigation bar as a header of all the web pages which will help the users to easily navigate in the web pages.

Familiarity & Consistency: Using similar layout (navigation bar always at the top of the web application) and applying same color scheme for entire web application.

2.2.1 Information Architecture

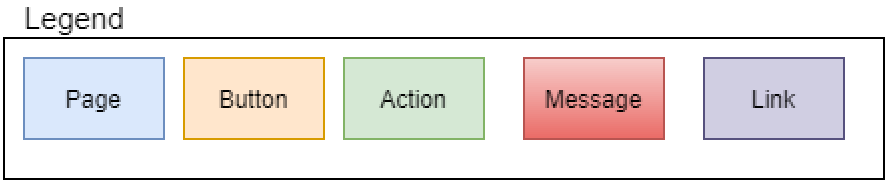


Figure 1 Sitemap legend [7]

As you can see in the above Figure 1, We have used the above legend in the sitemap. The webpage is depicted using blue color box. Various user actions like input and button clicking are represented using green box. Button that are present on the web page are depicted using orange box. The gray box represents any external links.

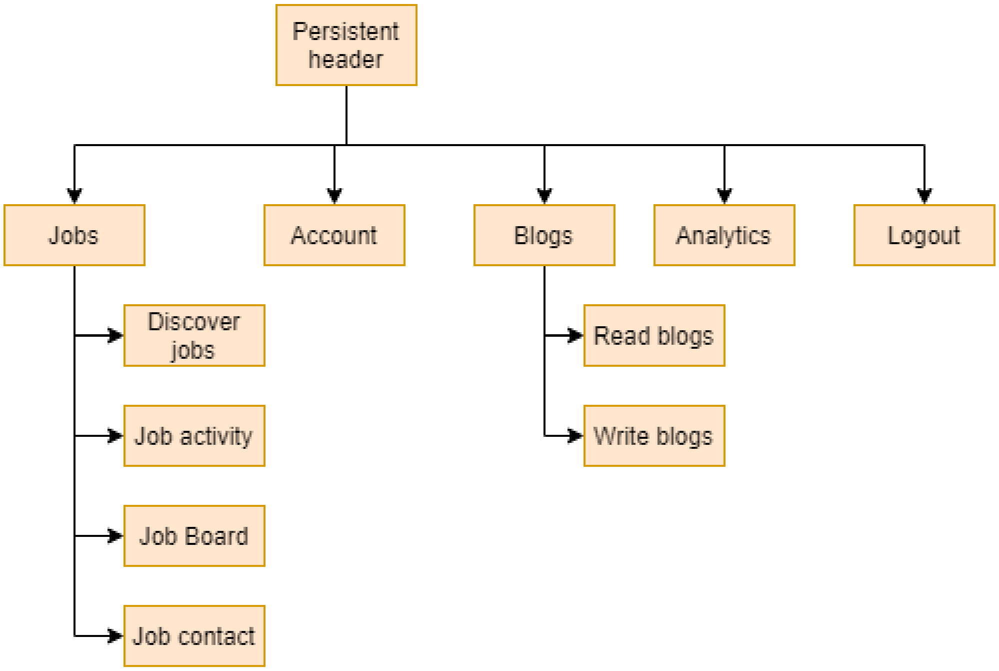


Figure 2 Site-map persistent header [7]

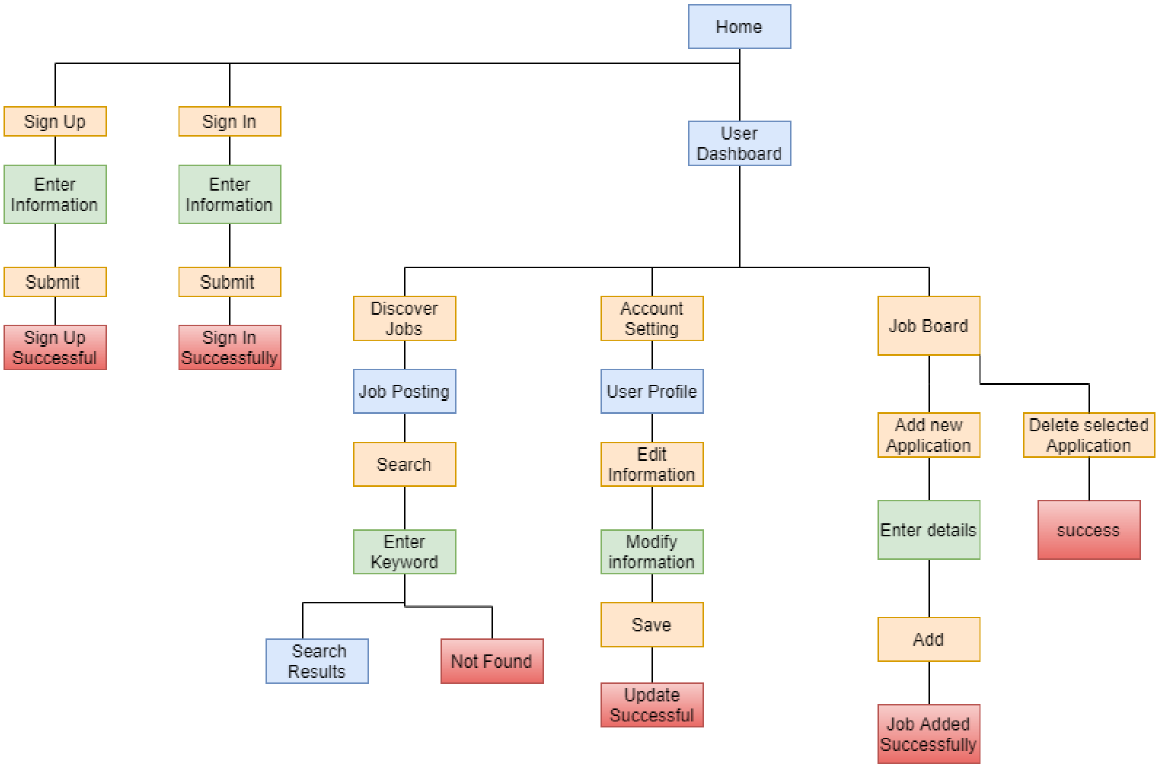


Figure 3 site-map Part 1[7]

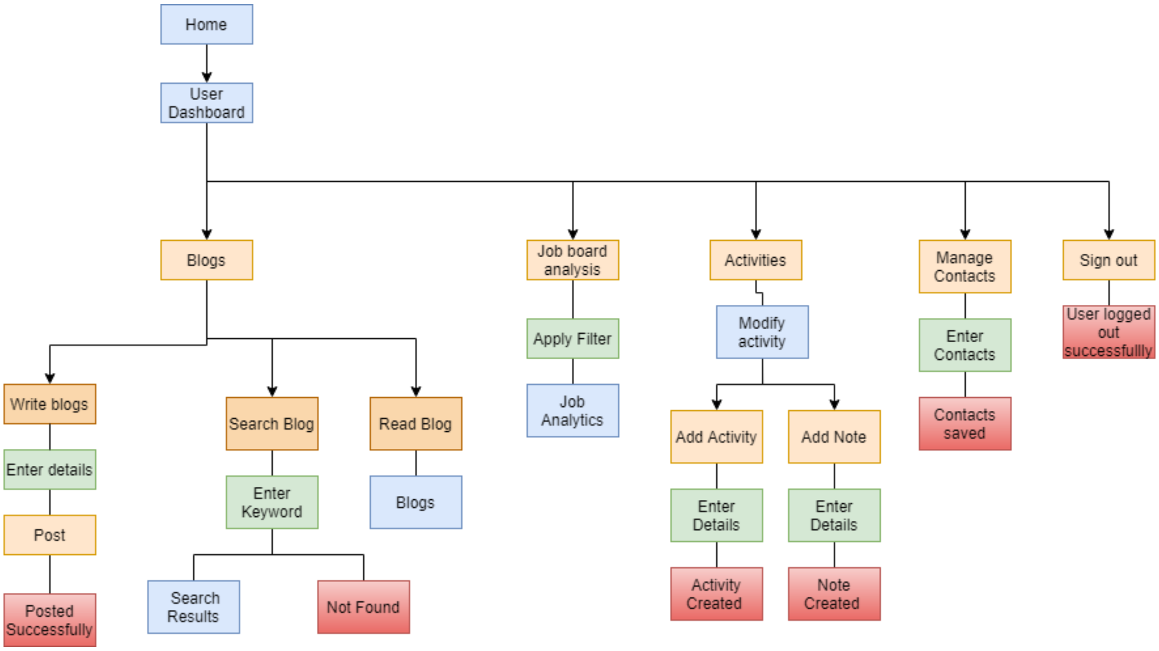


Figure 4 Site-map part 2 [7]

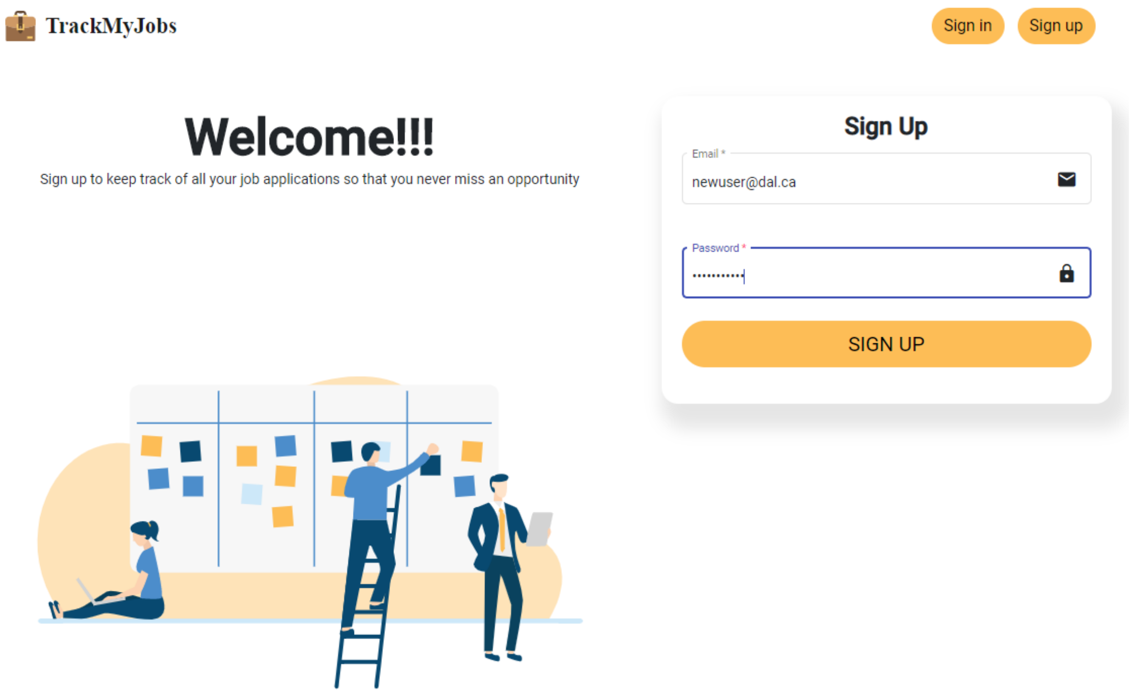
2.2.2 Design and Layout

We provide high-fidelity prototype of 12 features of TrackMyJobs as below:

Profile Management

Sign Up Page:

For any activities or a Web Application interaction, the major tasks and significant role is played by session management. [8] There, it is very much mandatory to host session management for a website to be complete and provide users their individual view persona of the Web Application. The origin for this begins with new User Registration. Therefore, we developed a User Sign-Up page that allows users to sign up for our web application through email.



Welcome!!!

Sign up to keep track of all your job applications so that you never miss an opportunity

Sign Up

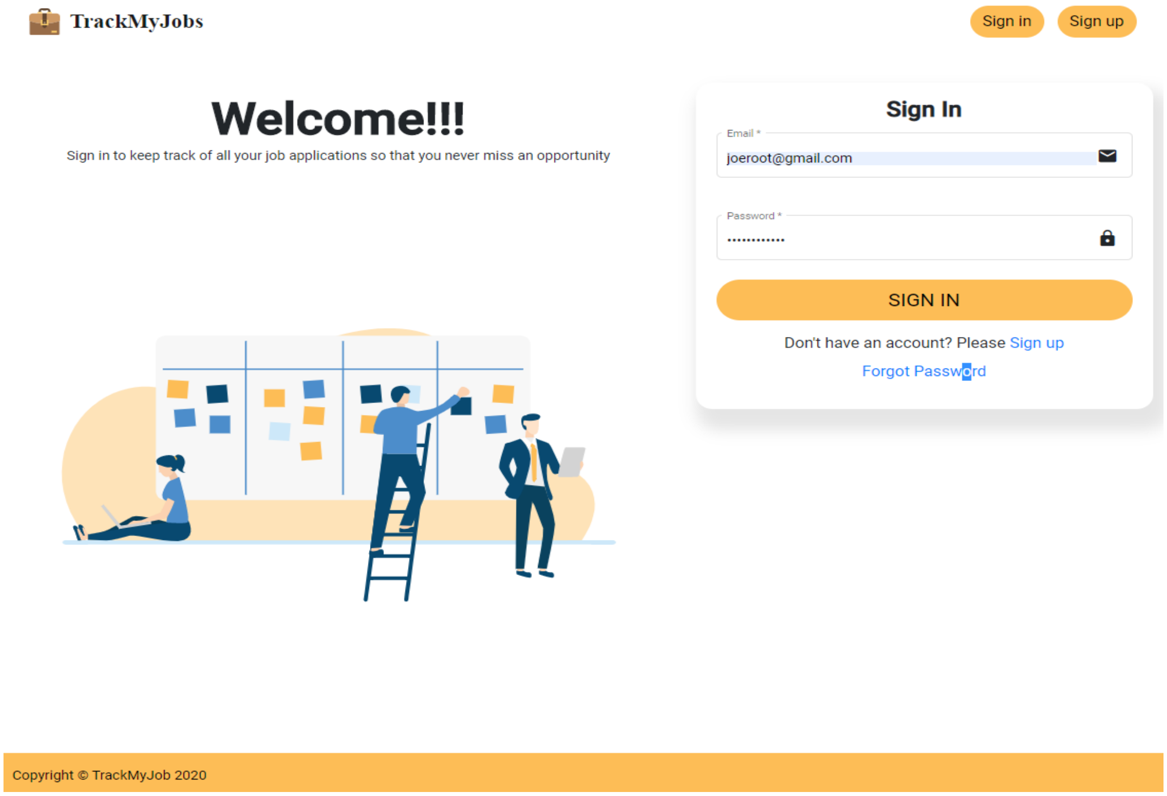
Email * newuser@dal.ca

Password * ···········

Figure 5 Screenshot for signup page [9]

Sign in Page:

As a part of user profile management and Application management based on every individual user, we need a User Login. [8] Therefore, we developed a Sign-in page that allows user to sign-in to the application and system allows to login after necessary validations.



Sign in to keep track of all your job applications so that you never miss an opportunity

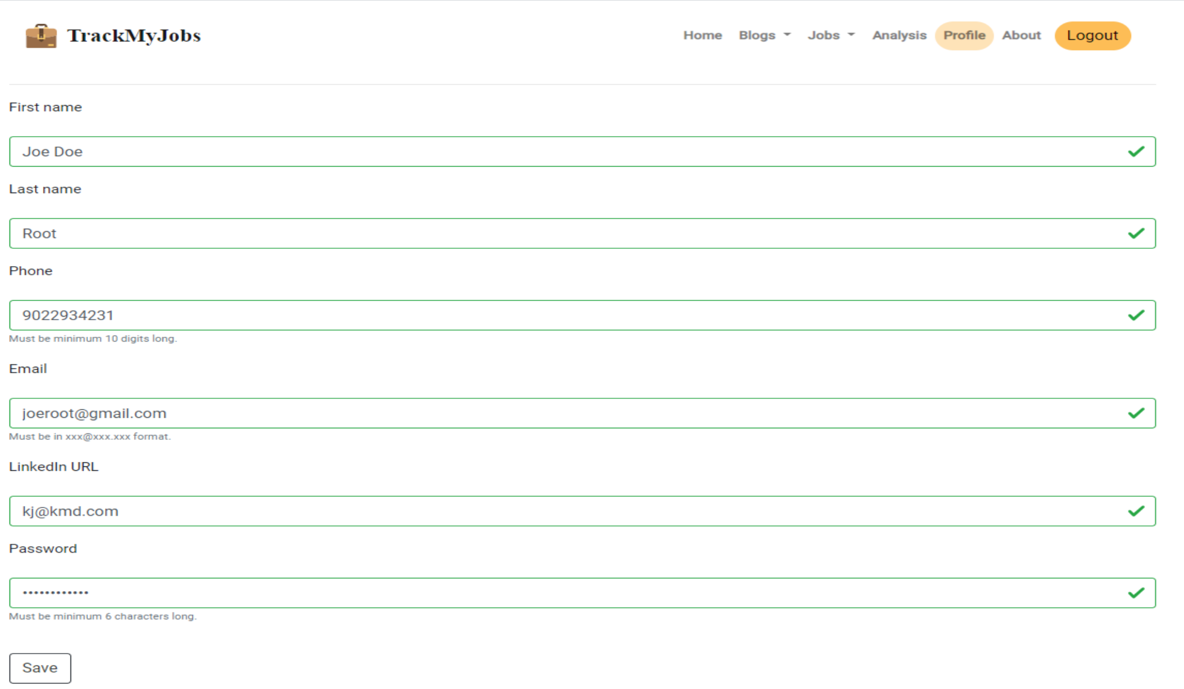
joeroot@gmail.com

Don't have an account? Please Sign up

Figure 6 Screenshot for sign in page [9]

Profile Page:

For our application, the main task is the maintain user's profile which will consist the details of the user namely user's Name, Email ID, Contact Number, LinkedIn profile URL. [8] Thus, we developed a user profile page where the user can view and edit the profile and their personal information. [10]



First name

Joe Doe ✓

Last name

Root ✓

Phone

9022934231 ✓

Must be minimum 10 digits long.

Email

joeroot@gmail.com ✓

Must be in xxx@xxx.xxx format.

LinkedIn URL

kj@kmd.com ✓

Password

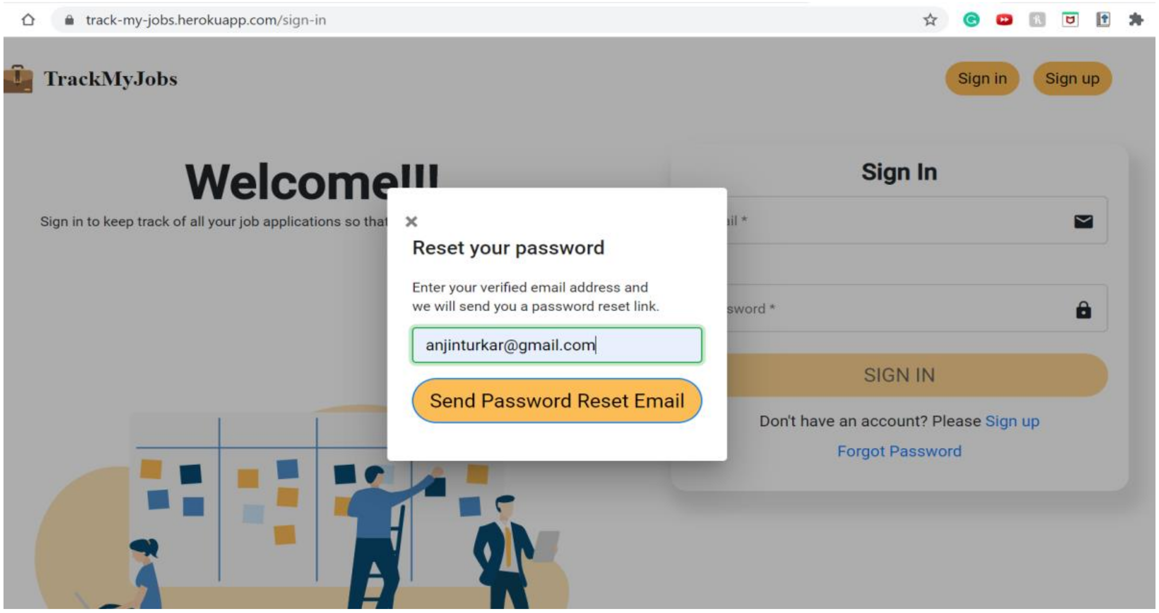
•••••••••••• ✓

Must be minimum 6 characters long.

Figure 7 Screenshot for profile page [9]

Forgot Password Page:

As a part of user authentication functionality, it is very much important to guide user to have access to reset their password if by chance they forget it. [8] Therefore, we implemented Forget Password functionality where the user can reset the password if they will provide the valid email address.



Sign in to keep track of all your job applications so that

Don't have an account? Please Sign up

Reset your password

Enter your verified email address and we will send you a password reset link.

anjinturkar@gmail.com

Figure 8 Screenshot for forgot password page [9]

Job Search

From Figure 9[9], job search page has minimal elements so that users can focus on the main task which is to search for job availability. The job search in TrackMyJobs is based on Canada.

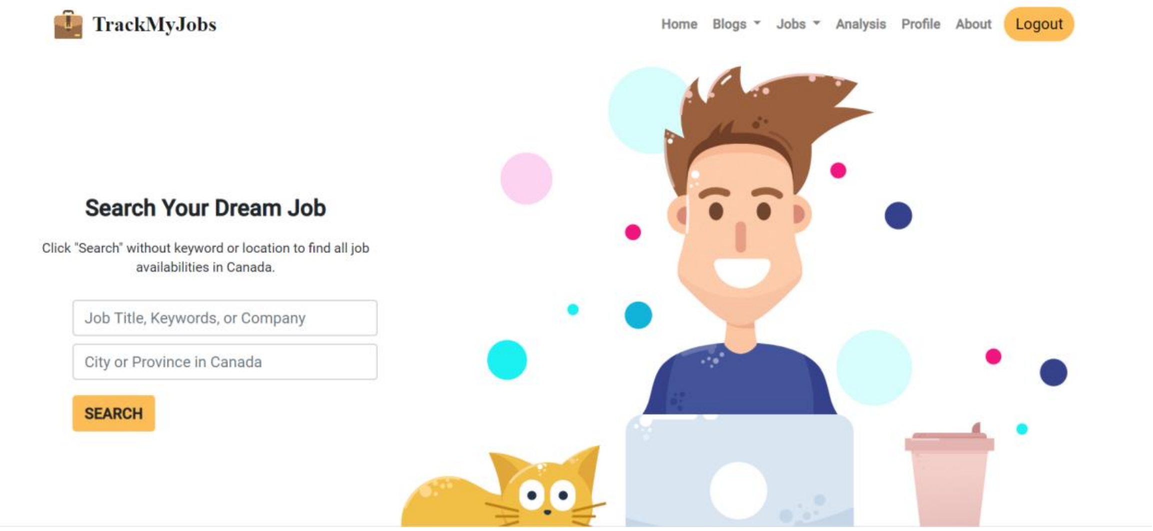
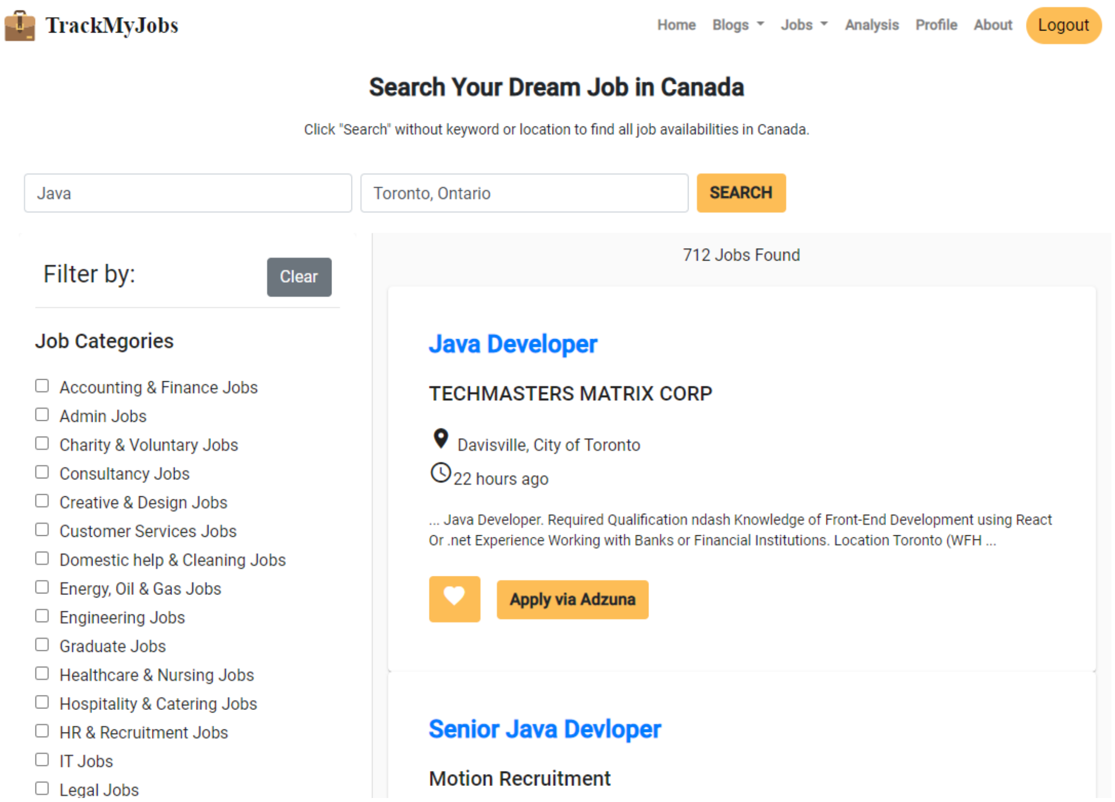

TrackMyJobs
Home
Blogs
Jobs
Analysis
Profile
About
Logout
Search Your Dream Job
Click "Search" without keyword or location to find all job availabilities in Canada.
Job Title, Keywords, or Company
City or Province in Canada
SEARCH

Figure 9: High-fidelity prototype of job search page [9]

From Figure 10 [9], job search result page has search bar on the top. Users can provide different keyword and location by going to the top of the page. On the left side is the filter to filter job categories, job type, and sort by. The filter helps users to narrow their search. The right side is the search result. If the users click on job name, it will redirect users to that job full detail in new tab.



Search Your Dream Job in Canada

Click "Search" without keyword or location to find all job availabilities in Canada.

Java

Toronto, Ontario

Filter by:

Job Categories

- □ Accounting & Finance Jobs
- □ Admin Jobs
- □ Charity & Voluntary Jobs
- □ Consultancy Jobs
- □ Creative & Design Jobs
- □ Customer Services Jobs
- □ Domestic help & Cleaning Jobs
- □ Energy, Oil & Gas Jobs
- □ Engineering Jobs
- □ Graduate Jobs
- □ Healthcare & Nursing Jobs
- □ Hospitality & Catering Jobs
- □ HR & Recruitment Jobs
- □ IT Jobs
- □ Legal Jobs

712 Jobs Found

Java Developer

TECHMASTERS MATRIX CORP

Davisville, City of Toronto

22 hours ago

... Java Developer. Required Qualification ndash Knowledge of Front-End Development using React Or .net Experience Working with Banks or Financial Institutions. Location Toronto (WFH ...

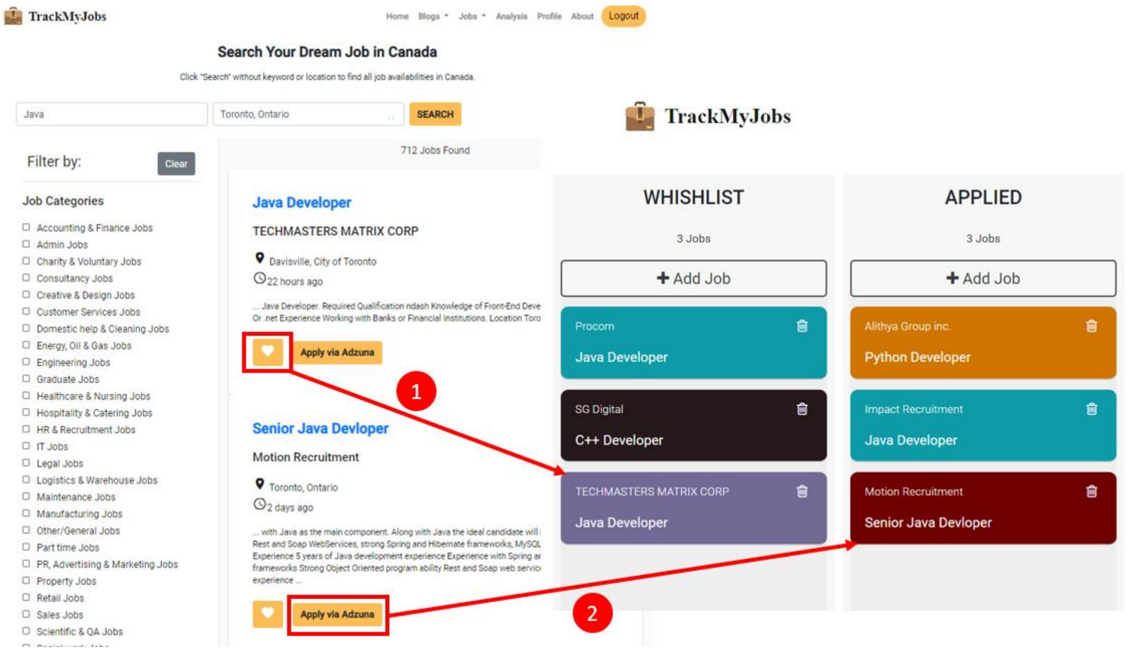
Senior Java Devloper

Motion Recruitment

Figure 10: High-fidelity prototype of job search result page [9]

Job Discovery

From Figure 11 [9], the job discovery feature is the interaction between job search and job application management. If users click on “Heart” button on job search result page, it will automatically add that job with status “wishlist” to the job database. If users click on “Apply via Adzuna” button on job search result page, it will automatically add that job with status “applied” to the job database. When users display their job board, they will find that job display on the appropriate status.



Search Your Dream Job in Canada

Click "Search" without keyword or location to find all job availabilities in Canada.

Java | Toronto, Ontario | SEARCH

712 Jobs Found

Filter by: Clear

Job Categories

- □ Accounting & Finance Jobs
- □ Admin Jobs
- □ Charity & Voluntary Jobs
- □ Consultancy Jobs
- □ Creative & Design Jobs
- □ Customer Services Jobs
- □ Domestic help & Cleaning Jobs
- □ Energy, Oil & Gas Jobs
- □ Engineering Jobs
- □ Graduate Jobs
- □ Healthcare & Nursing Jobs
- □ Hospitality & Catering Jobs
- □ HR & Recruitment Jobs
- □ IT Jobs
- □ Legal Jobs
- □ Logistics & Warehouse Jobs
- □ Maintenance Jobs
- □ Manufacturing Jobs
- □ Other/General Jobs
- □ Part time Jobs
- □ PR, Advertising & Marketing Jobs
- □ Property Jobs
- □ Retail Jobs
- □ Sales Jobs
- □ Scientific & QA Jobs

Java Developer

TECHMASTERS MATRIX CORP

Davisville, City of Toronto

22 hours ago

... Java Developer. Required Qualification ndash Knowledge of Front-End Deve Or .net Experience Working with Banks or Financial Institutions. Location Toro

Apply via Adzuna

Senior Java Devloper

Motion Recruitment

Toronto, Ontario

2 days ago

... with Java as the main component. Along with Java the ideal candidate will Rest and Soap WebServices, strong Spring and Hibernate frameworks, MySQL Experience 5 years of Java development experience Experience with Spring ar frameworks Strong Object Oriented program ability Rest and Soap web servic experience ...

Apply via Adzuna

WHISHLIST

3 Jobs

+ Add Job

Procom
Java Developer

SG Digital
C++ Developer

TECHMASTERS MATRIX CORP
Java Developer

APPLIED

3 Jobs

+ Add Job

Alithya Group inc.
Python Developer

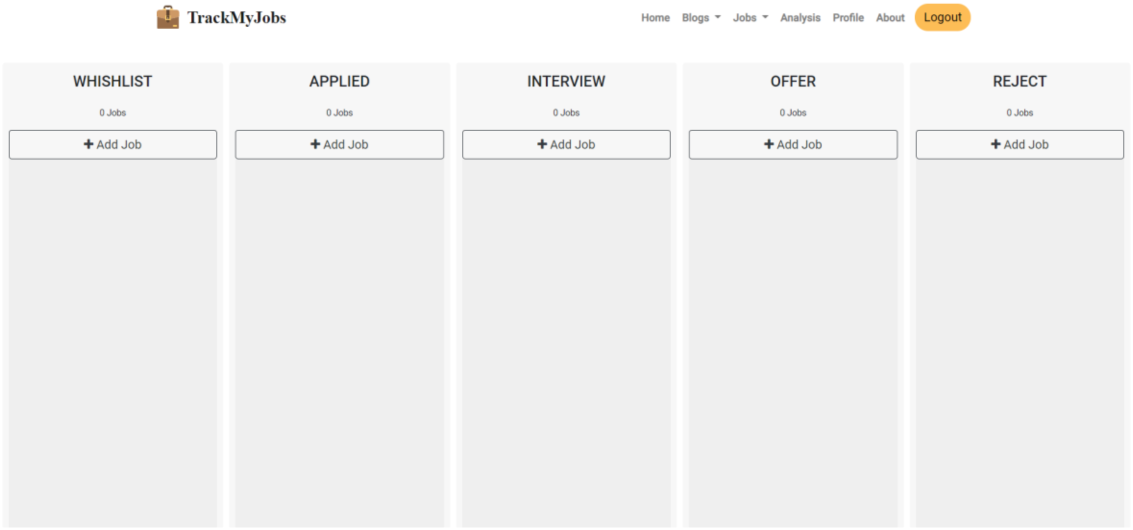
Impact Recruitment
Java Developer

Motion Recruitment
Senior Java Devloper

Figure 11: Interaction of Job Discovery Feature [9]

Job Board Management

Figure 12 [9] illustrate the design of a job board of TrackMyJobs portal. A job board has default partition that helps users to organize their job applications. The partition is based on different status of job application viz. wishlist, applied, interview, offer and reject. A simple design of partitioned section with relevant information helps users to easily identify category of a job application resides on that partitioned section.



WHISHLIST	APPLIED	INTERVIEW	OFFER	REJECT
0 Jobs	0 Jobs	0 Jobs	0 Jobs	0 Jobs
+ Add Job	+ Add Job	+ Add Job	+ Add Job	+ Add Job

Figure 12 Job Board [9]

Job Application Management

Figure 13 illustrate the job applications management of a job board [9]. A job board consists of several job applications. A job application is designed as a simple card that holds information about the job application such as job title and company name. A delete button is also added to the job application card to delete it from the job board.

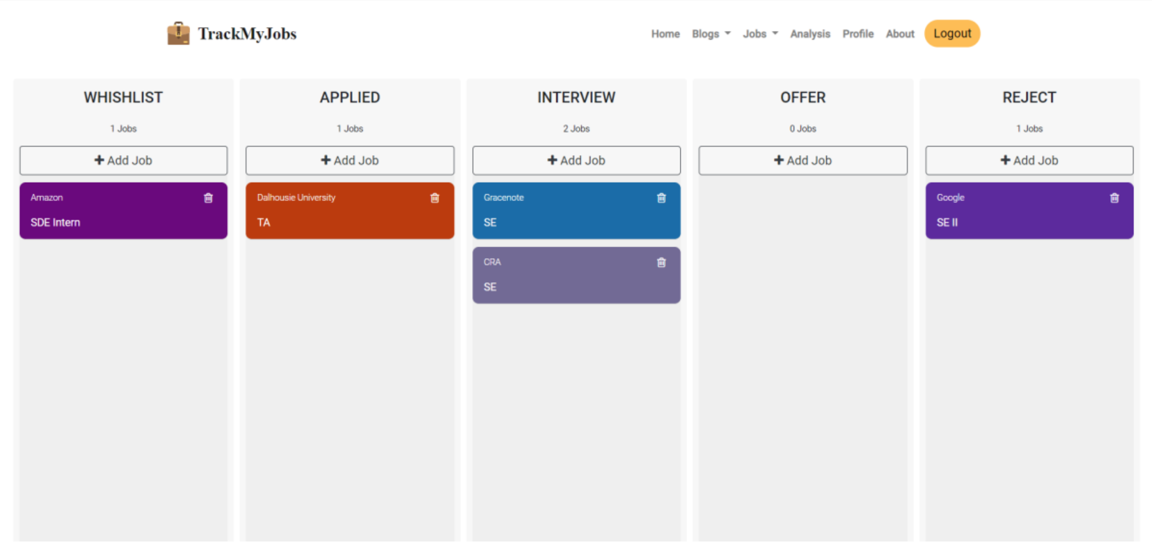
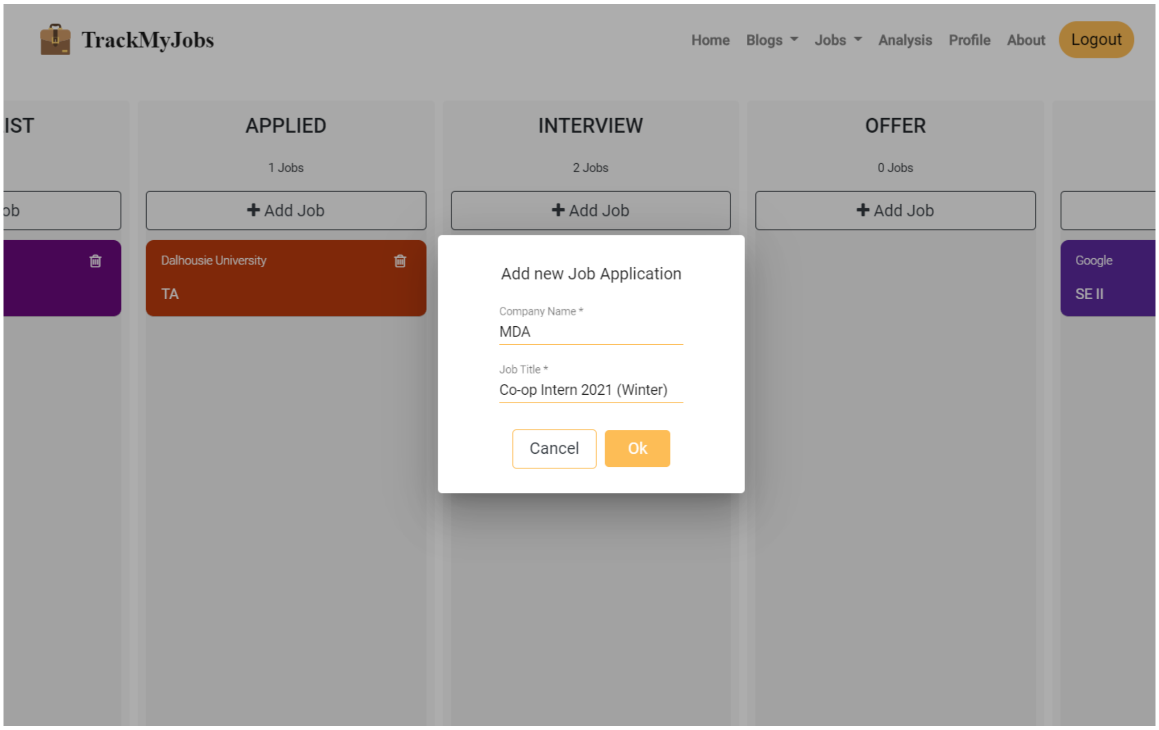


Figure 13 Job Application [9]

Click on a add job button to add new job application on a specific partitioned section of the job board. A dialog box with simple design will be displayed as depicted in Figure 14. Figure 15 illustrate the result after adding a new job application in the Interview section.

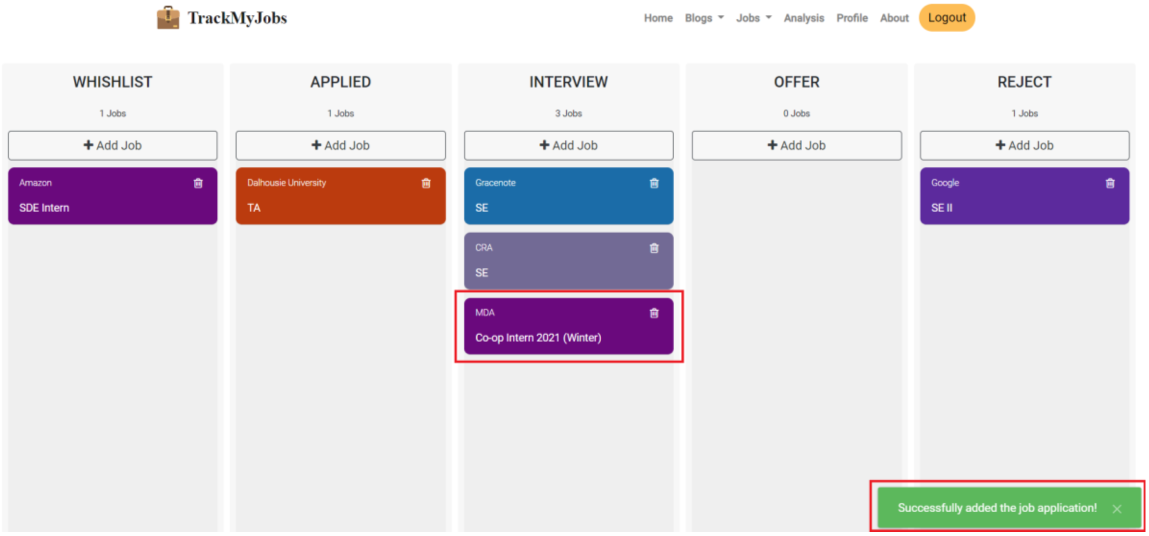


Add new Job Application

Company Name * MDA

Job Title * Co-op Intern 2021 (Winter)

Figure 14 Add job application [9]



Successfully added the job application!

Figure 15 Job board after adding new job application [9]

Job Board Analysis

Job board analysis page will give provide the analysis of the various data to the user. We have visualized the data as shown in figure 16 . We have provided three types of graph. First one is written blog analysis which provided the written blogs analysis (it means how many blogs the user has written in a month). The second analysis is job application analysis. These analytics will help the user to understand how many jobs they applied in the certain period and it will visually show the status of each application.[14]

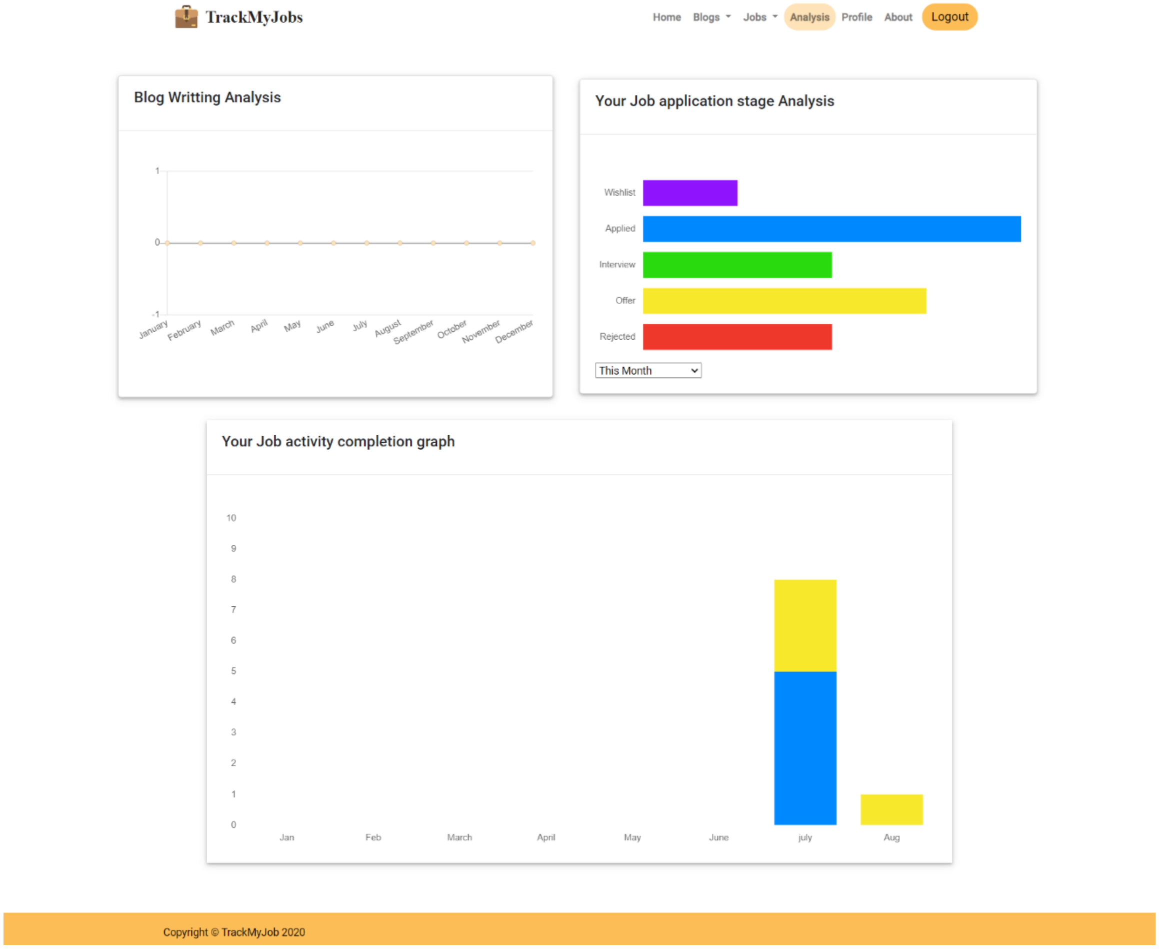
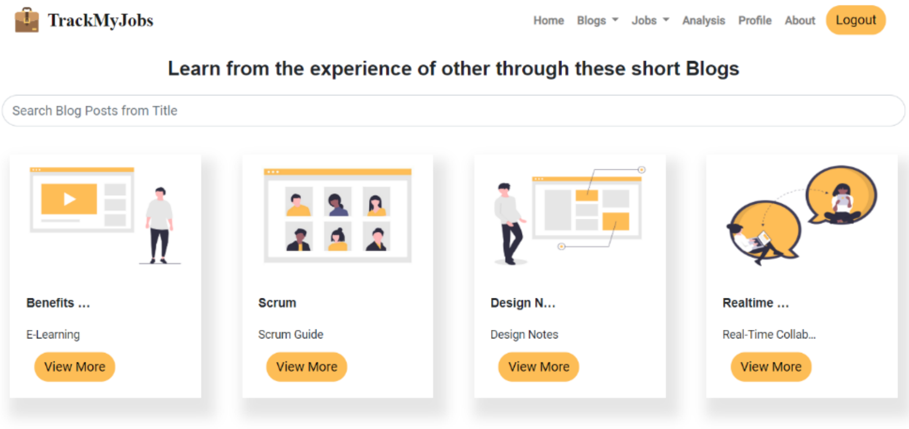


Figure 16 Job board analysis [9]

Blog Management

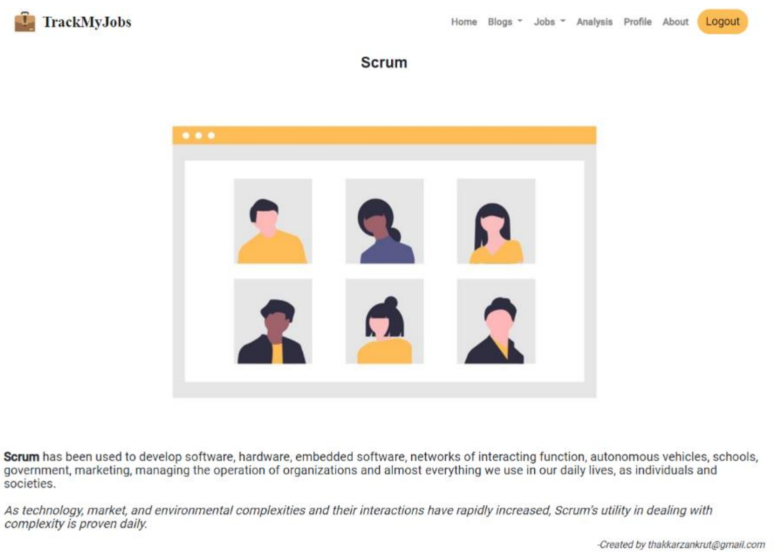
The blog management feature provides users with the functionality to see all blog posts, filter blog posts using the search feature, and read a specific blog post. Figure 17 [9] shows the UI design of the read blogs page, here as you can see, all the blog posts are shown in the form of a card. The primary reason behind using the card to show different blog posts is that users can easily focus on essential elements of blogs like title and sub-title and can easily navigate all the different blog posts. [11] Additionally, the position of the search bar is kept above the blog post so that user can easily search the required blog post and see the result of search below. Also, each blog post has View More button users can read a specific blog using this button. [12]



Learn from the experience of other through these short Blogs

Figure 17: Read Blogs Page [9]

When users click on View More button, they are navigated to a specific blog page, as shown in Figure 18 [9]. Here user can read the content of the blog post. All the information shown on this page is in chronological order. On top of the page, the user can see the blog's title, followed by the blog image, main content, and author of the blog. Additionally, if it is the user's blog, then the user will be shown a delete button at the bottom of the page. Using this delete option, users can delete the blogs written by them. Moreover, this page's aesthetics match the overall design of the web application, making it consistent.



Scrum

Scrum has been used to develop software, hardware, embedded software, networks of interacting function, autonomous vehicles, schools, government, marketing, managing the operation of organizations and almost everything we use in our daily lives, as individuals and societies.

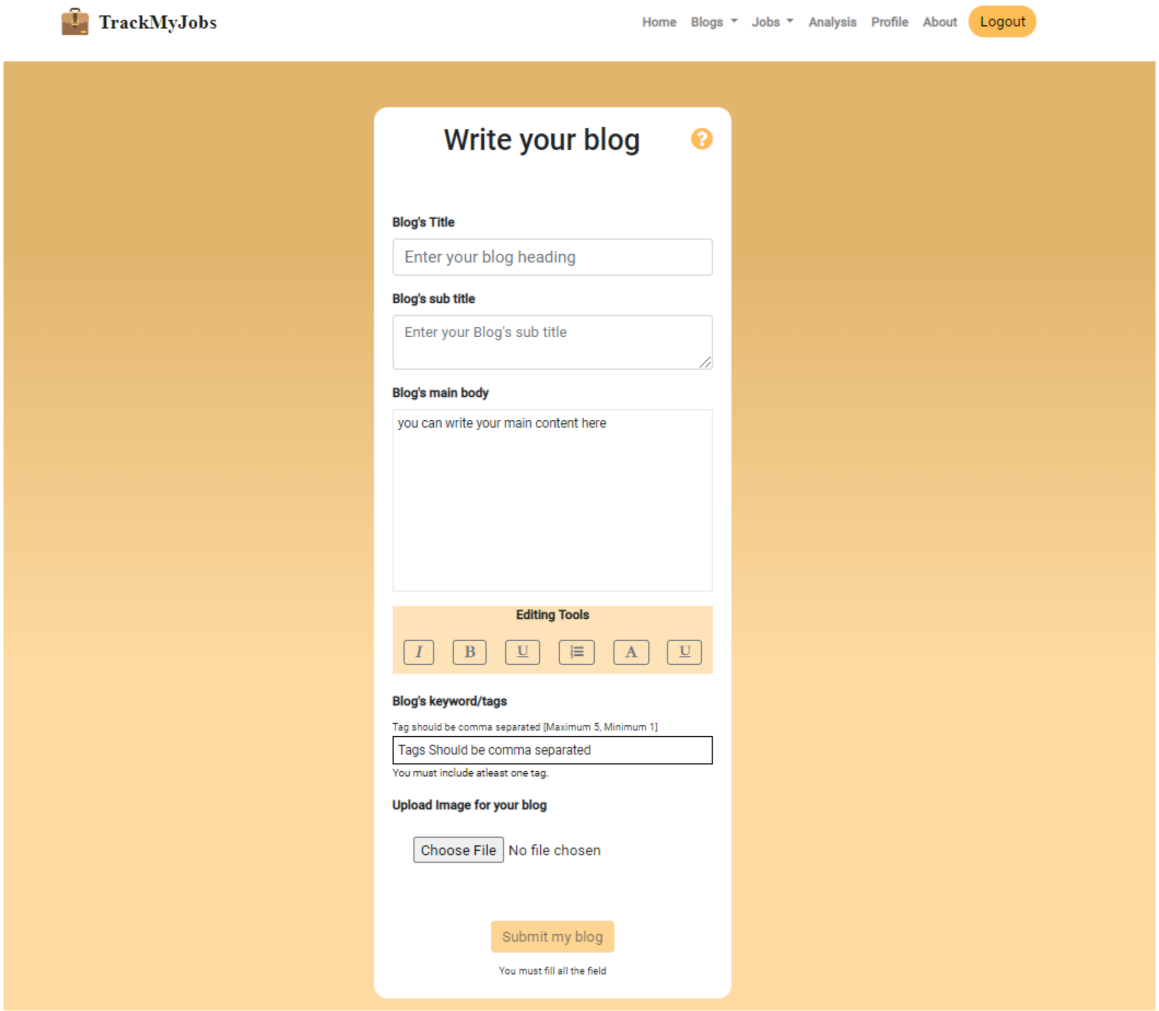
As technology, market, and environmental complexities and their interactions have rapidly increased, Scrum's utility in dealing with complexity is proven daily.

-Created by thakkarzankrut@gmail.com

Figure 18: Read Specific Blog Post [9]

Blog Creation

The blog creation feature allows user to share their knowledge regarding the application process in the form of short blog post. The Figure 19 [9] show the write blogs page. The primary idea behind the design of this page was to give user a feel of actual text editor. Here, we have tried to provide user with the functionality of text editor which match the overall design theme of the web application. Also, all the elements are placed in the chronological order like title, sub-title, content, keyword, and blog image. Users can also add the image of the blog. Additionally, user will not be able to submit the blog until all the required information is filled and if required information is not filled it will highlight the part which is required to submit the blog. Moreover, if user wants to apply any style to the blog's content such as bold, italics, underline, or list then they can easily apply it to using the editing tools. [11] The placement of the editing tools is strategic, and user can select the content from body and apply the style filter. Furthermore, after clicking on the submit button user will be prompted will the confirmation dialogue box indicating that the blog is posted successfully and will be redirected to the read blogs page. [11]



Write your blog

Blog's Title

Enter your blog heading

Blog's sub title

Enter your Blog's sub title

Blog's main body

you can write your main content here

Editing Tools

Blog's keyword/tags

Tag should be comma separated [Maximum 5, Minimum 1]

Tags Should be comma separated

You must include atleast one tag.

Upload Image for your blog

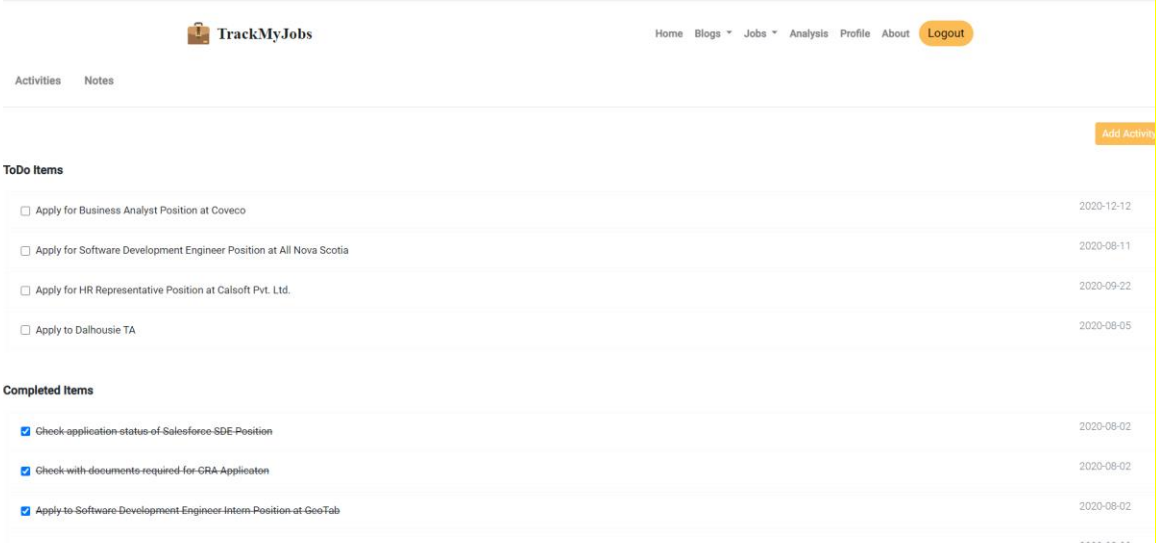
No file chosen

You must fill all the field

Figure 19: Write blogs page [9]

Job Application Activities

For user, it is vital if they are given a functionality where they can save the activities that they need to perform which can be related to any specific Job Application. Also, if they get a feature that they can see their To Do list with the activity deadline and the list of activities they have completed with the completion date. [1] Then, it will be very beneficial for the user to track the activities and add any new. Thus, keeping this in mind we developed a Job Activities feature which allows the user to deal with the activities for the jobs. [13]This feature also allows the user to add any new activity and maintain its important deadline and events. As shown in, we developed a list of activities which are divided into two categories which are "to do items" and "completed items." Initially on adding a new activity, it is by default assigned to "to do list items" along with the target end date. The users can also mark this activity as a completed activity when they complete the activity. As the Figure 20 Job Activities Page, we focused on keeping this page minimal so that users can fully focus on their activities. [14]



ToDo Items

- □ Apply for Business Analyst Position at Coveco 2020-12-12
- □ Apply for Software Development Engineer Position at All Nova Scotia 2020-08-11
- □ Apply for HR Representative Position at Calsoft Pvt. Ltd. 2020-09-22
- □ Apply to Dalhousie TA 2020-08-05

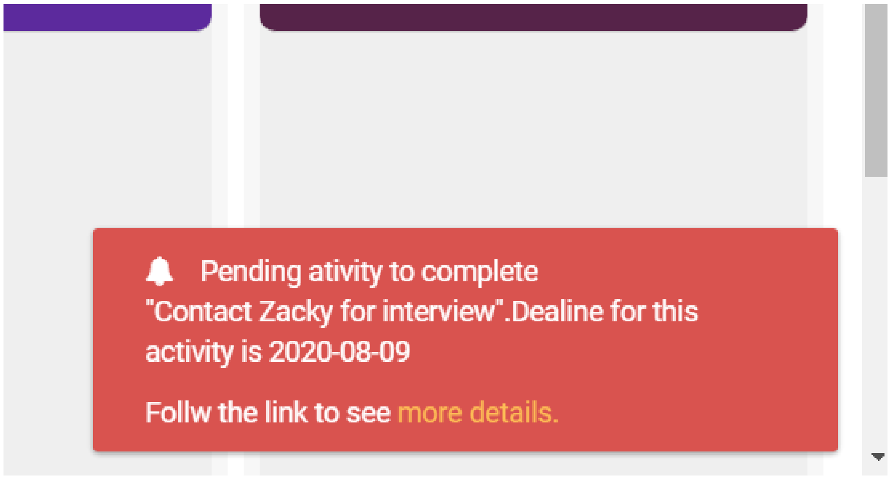
Completed Items

- ☑ ~~Check application status of Salesforce SDE Position~~ 2020-08-02
- ☑ ~~Check with documents required for CRA Applicaton~~ 2020-08-02
- ☑ ~~Apply to Software Development Engineer Intern Position at GeoTab~~ 2020-08-02

Figure 20 Job Activities Page [9]

Job Notification Management

Push notifications are a fast and efficient way to communicate with our target audience. There is multiple advantage of incorporating push notification in the web application. First, it increases the user engagement with the application.by providing push notification we can give real-time update to the user. Secondly, it reduces the efforts in customer journey. Due to these reasons we have decided to provide in-app notification to the user based on some condition. The user will be provided a deadline notification for a specific activity when the deadline for an activity is within 7 days from the todays date. As you can see in the below image, the user will be prompted with the notification when the user visits the job board page. The notification will be seen to user only for one time in that session. [15]



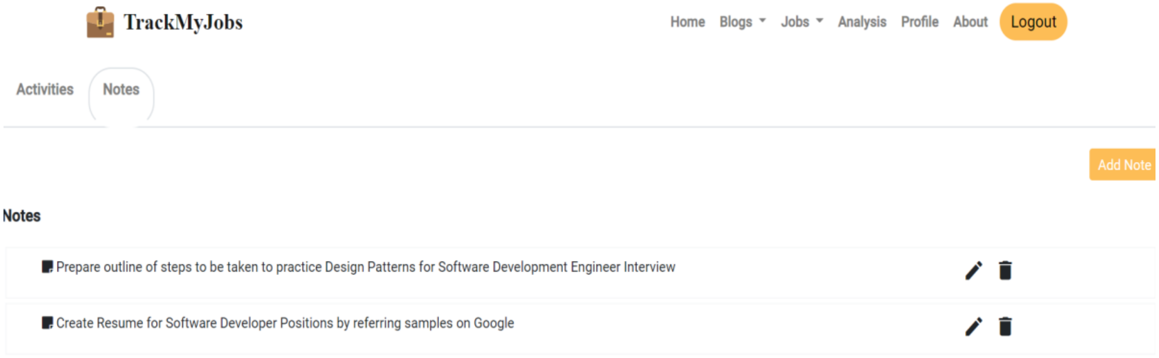
🔔 Pending ativity to complete "Contact Zacky for interview".Dealine for this activity is 2020-08-09

Follw the link to see more details.

Figure 21 Notification management [9]

Job Application Notes

For a user, if given an additional support of giving some feature to track notes related to any activity or anything that they need to explore, it helps them to save the notes regarding the same and check it later. [14]The Job Application Notes management feature allows the users to manage their notes so that they can keep notes of the important things which will help them in performing the activities and preparing for the Job Position Interviews or any required things. The user has options to add any note, or update, or any previously added notes. This gives user the overall feature of updating any old note saved based on new changes or deleting them when they are no more required. The Figure 22 Job Application Notes shows a simplistic design implemented based on above described functionality of the Notes Management. [14] [9] [1]



Notes

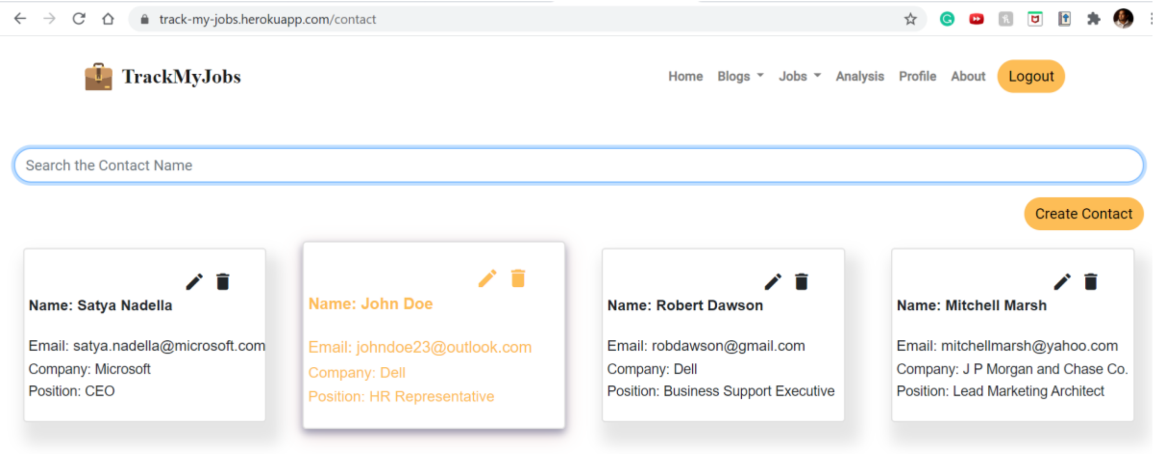
Prepare outline of steps to be taken to practice Design Patterns for Software Development Engineer Interview

Create Resume for Software Developer Positions by referring samples on Google

Figure 22 Job Application Notes [9]

Contact Management

For any user, it is very important to save the contacts of the important contact person they need to communicate to when applying for a job or while on-going the application process. It is tedious to remember the contact people. [1] To address this issue, we came up with an idea to let the users store the contact details of the people to whom they want to communicate further or the people whose contact they will need in future. Therefore, we implemented the Contact Management Functionality where user can save the contacts of the people. User also has an option to search specific contacts using the search bar and update or delete an existing contact. The Figure 23 Job Contacts Management Page shows the above discussed implementation. [14] [1]



Name: Satya Nadella
Email: satya.nadella@microsoft.com
Company: Microsoft
Position: CEO

Name: John Doe
Email: johndoe23@outlook.com
Company: Dell
Position: HR Representative

Name: Robert Dawson
Email: robdawson@gmail.com
Company: Dell
Position: Business Support Executive

Name: Mitchell Marsh
Email: mitchellmarsh@yahoo.com
Company: J P Morgan and Chase Co.
Position: Lead Marketing Architect

Figure 23 Job Contacts Management Page [9]

3. Application Workflow

The application workflow of 12 features are provided below:

3.1 Interaction Design

In this section we have tried to explain the user interaction as well the user persona for all the 12 features. Moreover, we have tried to explain the user interaction through the interaction diagrams which will give the clear idea of how a user can navigate and use that functionality.

3.1.1 Profile Management

User Persona and Scenario

User visits our website for the first time, the user will be landing on a login/Signup page where they can enter their credentials like email id and password for login. For signup they can enter the first name, last name, password, and email id. On successful login or signup, they will be redirected to their account dashboard.[8]

Use Case: Signup

1. User visits login/signup page. [user action]
2. User clicks on Signup. [user action]
3. System will display the Signup component. [system action]
4. User will enter a first name, last name, email id and password. [user action]
5. User clicks signup button. [user action]

System prompts the user that email Id already has been used. [system action]

- User will enter a different email id for signing up. [user action]
- User will enter the password again. [user action]
- User clicks signup button. [user action]

System prompts the user that password length should be minimum 6 character. [system action]

- User will enter the new password [user action]
- User clicks on the login button [user action]

6. System validates the user inputs. [system action]
7. System notifies user that signup is successful message. [system action]
8. System will redirect the user to their account's dashboard page. [system action]

Use Case: Login

1. User visits login/signup page. [user action]
2. User clicks on login. [user action]
3. System will display the login component. [system action]
4. User will enter an email id and password. [user action]
5. User clicks login button. [user action]
6. System validates the user inputs. [system action]
7. System notifies user that signup is successful message. [system action]
8. System will redirect the user to their account’s home page. [system action]

User Persona and Scenario

User visits our website and tries to login with their credentials. But the user for some reason is not able to recollect their password. System will ask for confirmation of their email id and then send a verification code to their email address and if the verification is successful then the user will be asked for creating new password. After entering the correct password, the user will be able to access the dashboard.[9]

Use Case forgot password/reset password

1. User visits login/signup page. [user action]
2. User clicks on login. [user action]
3. System will display the login component. [system action]
4. User will enter an email id. [user action]
5. Users click on the Reset password button. [user action]
6. System validate the email and will display the password reset email sent password alert. [system action]
7. User will open the reset password email. [user action]
8. User clicks on reset password link. [user action]
9. User will enter the new password code. [user action]
10. User clicks on the reset password button. [user action]
11. User will enter the email and new password on sign-in page. [user action]
12. User clicks on the sign-in button. [user action]
13. System validate the password. [system action]
14. System redirects the user to the home page. [system action]

3.1.2 Job Search

The front-end for job search feature is mainly to get input from users to extract data from job search API. Then, the system will retrieve the data and send the data back and render to users. Users see the result in user-friendly UI and users can click on a specific job to see the detail. The front-end for job search feature is mainly to get input from users to extract data from job search API. Then, the system will retrieve the data and send the data back and render to users. Users see the result in user-friendly UI and users can click on a specific job to see the detail.

User Persona and Scenario

William is a 22-year-old computer science student at Dalhousie University. He uses web application most of the time to simplify his life. Now, he is in his final year at Dalhousie University. He is going to graduate with computer science degree soon. He wants to be Java developer after graduation. Therefore, he decides to start job search in his last semester.

Use Case

William can use job search feature from anywhere and anytime, but internet connection is required. illustrated the interaction design of job search. The use case for job search are listed below:

1. Users sign in to TrackMyJobs.
    a. If users do not have an account, they have to sign up.
    b. After users successfully sign up, users can sign in to TrackMyJobs.
2. Users click on “jobs” and “discover jobs” on navigation bar. Then, the system will show the “job search” page. Users input keyword that related to their area of interested and click “search.”
3. After users click “search,” the web application will interact with controller.
4. Controller interacts with job service by sending keyword to job service.
5. Job service detect keyword and send intent request to job search API.
6. Job search API send back intent response to job service. The response contains the data about job search related to keyword.
7. Job service extract relevant data and send the data back to controller.
8. Job search result page renders to user with the relevant data. Users can navigate through the list of jobs.
    a. If no relevant data because no result is matched with keyword, job service will send the default message that there is no relevant data.

Noted: The numbered steps in Figure 24 and numbered list above is related.

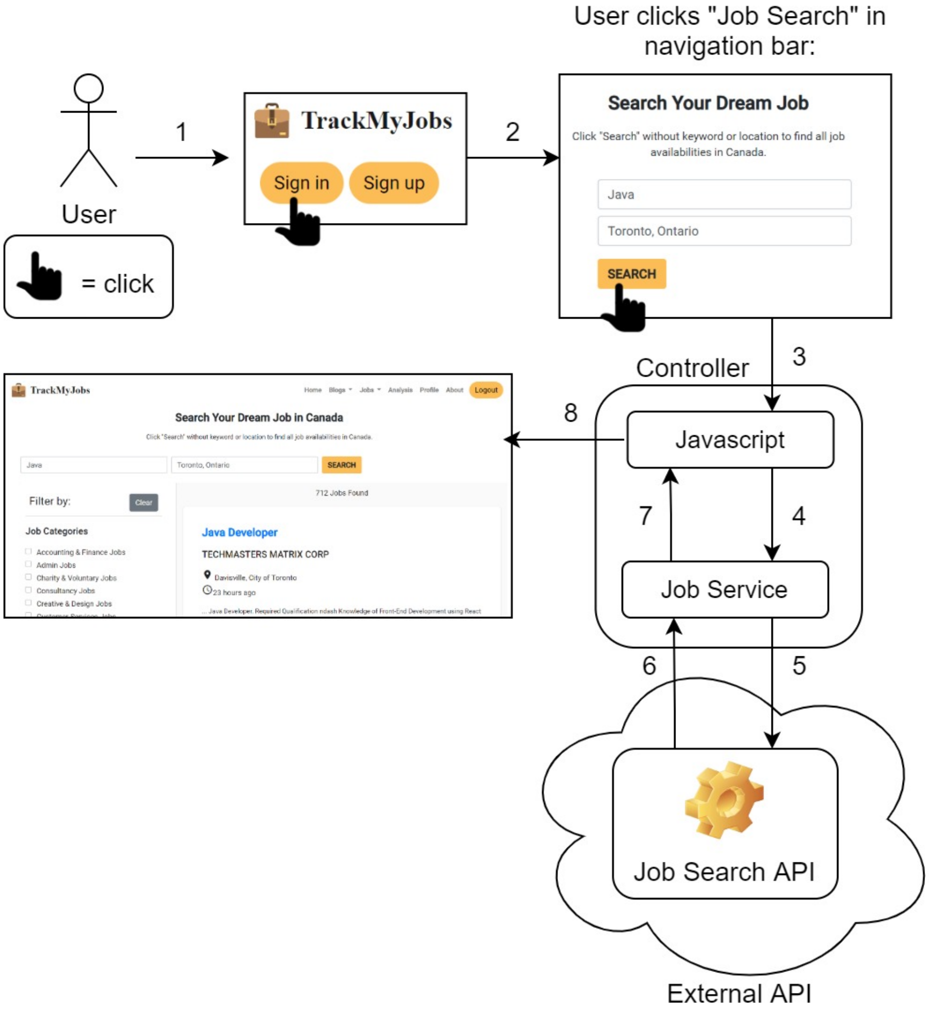


Figure 24: Interaction Design for Job Search Feature (drawn by [7])

3.1.3 Job Discovery

The front-end for job discovery feature is to apply and add a job to database by clicking on the button in job search result page. The system will automatically add a job with related status to database.

User Persona and Scenario

John is a 30-year-old experienced developer. He has been working in Bell for five years. Now, he thinks it is time to move on and look for a new job with higher responsibility and salary. However, he is still busy with current job. Therefore, he needs to search for a job and save some jobs in a wish list. He may apply some job right away if time allows. He needs a tool to help him track his job application.

Use Case and Interaction Design

As shown in Figure 25, there are two ways to add a job to database from job detail page which are "wish list" and "apply". The steps for each feature are listed below:

Wish List:

1. Users sign in to TrackMyJobs. Assume that user provide the correct credential, the system successfully authenticates users and sign in.
2. Users search for a job by providing related keyword and click "search." System retrieves the relevant job search result.
3. Users click on "heart" icon. The system will insert that job into database with status "wish list."

Users can remove a job from a wish list by clicking on "job board" in navigation bar. Then, users can delete the job from wish list by clicking on the "trash can" icon.

Apply:

1. Users sign in to TrackMyJobs. Assume that user provide the correct credential, the system successfully authenticates users and sign in.
2. Users search for a job by providing related keyword and click "search." System retrieves the relevant job search result.
3. Users click on "apply." The system adds that job into database with status "applied."

Users can remove a job from status "applied" by clicking on "job board" in navigation bar. Then, users can delete the job by clicking on the "trash can" icon.

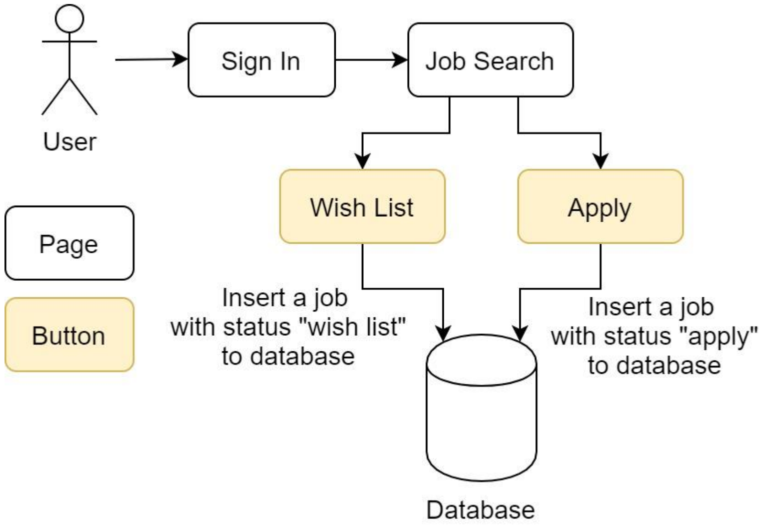

User
Sign In
Job Search
Wish List
Apply
Page
Button
Insert a job with status "wish list" to database
Insert a job with status "apply" to database
Database

Figure 25: Interaction Design for Job Discovery Feature (drawn by [7])

3.1.4 Job Application Management

The Job Board page provides simple user interface to interact and perform some operation to manage users job application.

User Persona and Scenario

Rob is an undergraduate student at Dalhousie University. He is in his final year of Computer Science. He wants to apply for an internship at Google, Facebook, Oracle, etc. He wants to manage his job applications, linked activities and contacts related to the job application. He also wants to manage the status of his job application, e.g. applied, interview, offer or reject.

Use Case and Interaction Design

Rob can use TrackMyJobs portal to manage his job applications. The portal provides the following tasks to be performed to manage job applications:

1. Create a job application
2. Delete a job application
3. Change status of a job application

The following is use case to perform the above tasks:

1. Users sign in to TrackMyJobs.
    a. If users do not have an account, they must sign up.
    b. After users successfully sign up, users can sign in to TrackMyJobs.
2. Users go to Job menu in navigation bar and select Job Board from the drop down.
3. System return a job board page with job applications of the user available in the database. They are group by status of the job applications.
4. User clicks on Add Job button of a specific job status partition of job board to create a new job application with that status.
5. System displays a dialog that asks the user to fill job application information.
6. System create a new job application in the database after user clicks on ok button with job application information.
7. System adds job application in front-end with success message notification.
8. User clicks on delete icon of a job application card to delete it.
9. System deletes the job application from the database and front-end and displays a success message notification.
10. User drag a job application card of one status partition and drop it to another status partition to change the status of the application.
11. System change the status of the job application in the database and return a success message notification to the user.

Figure 26 illustrates the user interaction diagram of job application management.

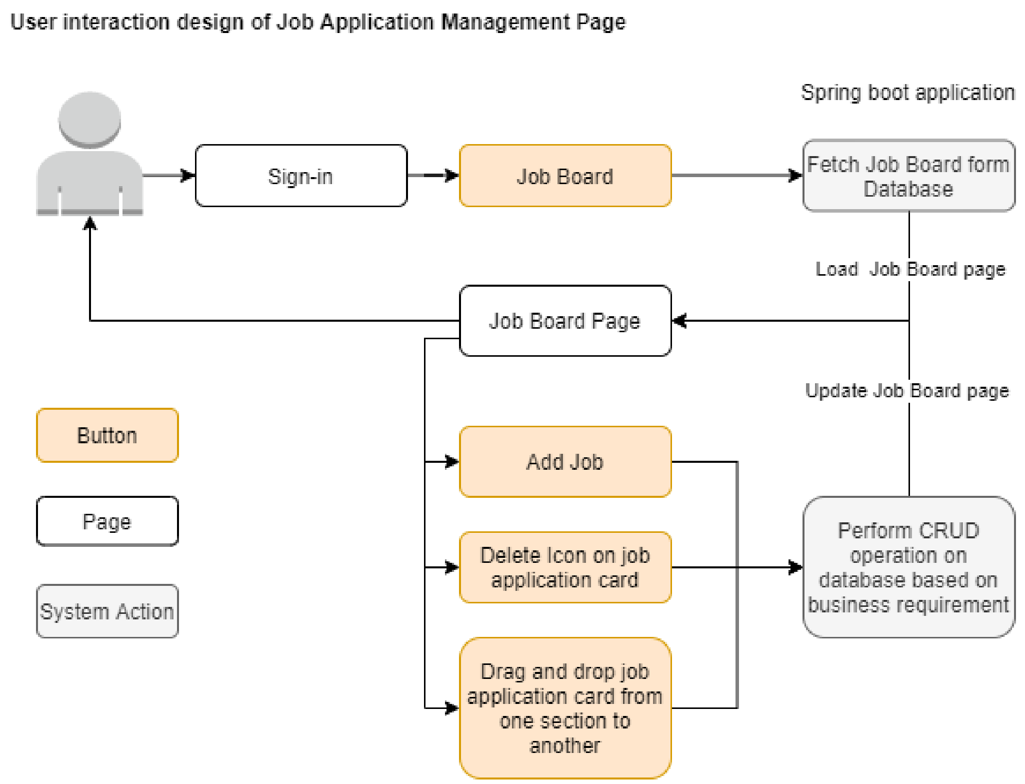

User interaction design of Job Application Management Page
Spring boot application
Sign-in
Job Board
Fetch Job Board form Database
Load Job Board page
Job Board Page
Update Job Board page
Button
Page
System Action
Add Job
Delete Icon on job application card
Perform CRUD operation on database based on business requirement
Drag and drop job application card from one section to another

Figure 26 Interaction Design of Job Application management [7]

3.1.5 Job Board Management

The job board page allows users to create job applications and manage them. System creates a job board in the database when users first time visit to Job Board menu in front-end after login.

User Persona and Scenario

Sushant is a 24-year-old graduate student. He is pursuing his master's in applied computer science at Dalhousie University. He wants to apply for software developer intern position to complete his next co-op term. He needs a tool to help him track his job application.

Use Case and Interaction Design

Sushant can use Job Board section of TrackMyJobs to manage his job applications. For that he needs to have a job board. The use case to create a job board are listed as below:

1. Users sign in to TrackMyJobs.
    a. If users do not have an account, they must sign up.
    b. After users successfully sign up, users can sign in to TrackMyJobs.
2. Users go to Job menu in navigation bar and select Job Board from the drop down.
3. System takes user ID and creates a new job board for the user in the database.
4. System return a job board page with predefined sections based on status of a job application.

Figure 27 illustrates the user interaction flow of job board management page.

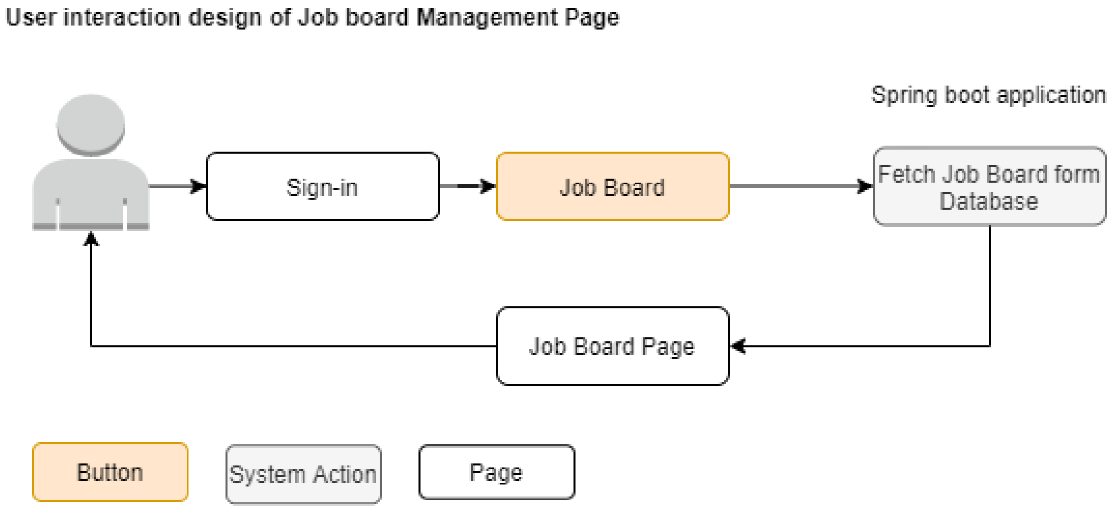

User interaction design of Job board Management Page
Spring boot application
Sign-in
Job Board
Fetch Job Board form Database
Job Board Page
Button
System Action
Page

Figure 27 Interaction Design of Job Board Management Page [7]

3.1.6 Job Application Analytics [14]

Sushant is currently our website’s existing user and wants to analyze all job application. This will help you to make your future decision.

Use case:

1. User clicks on my job board page

    1.1. System redirect the user to all job board analysis page

    1.2. System shows all the available job boards/created job boards

2. User selects one of the job board

3. User clicks on the visualization tool to visualize the data.

4. User change the job board

    4.1. System fetch the data related to that particular job board.

    4.2. System shows the data to the user

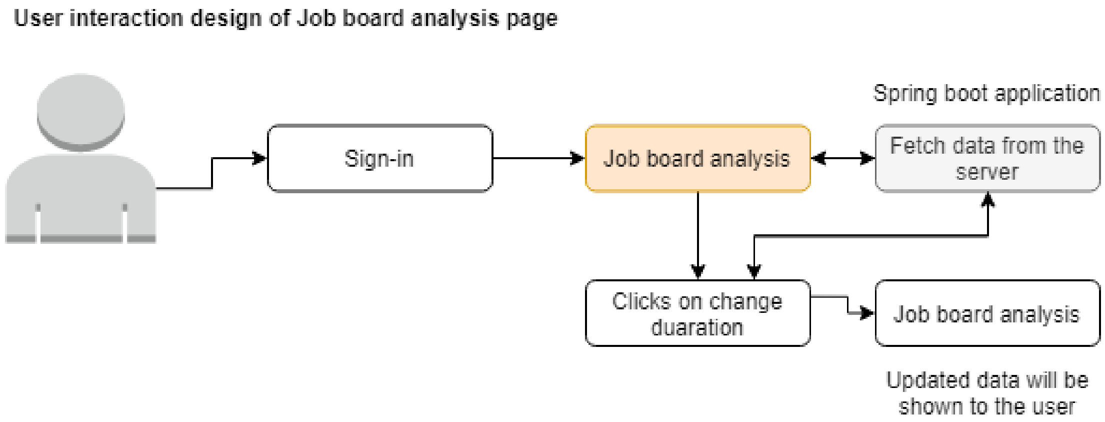

User interaction design of Job board analysis page
Spring boot application
Sign-in
Job board analysis
Fetch data from the server
Clicks on change duaration
Job board analysis
Updated data will be shown to the user

Figure 28 Interaction Design for Job board analysis (drawn by [7])

3.1.7 Blog Management

The user interface of the read blog page provides the functionality to view all the blog post additionally user can find the required blog by using the search functionality. Moreover, if user wants to read the content of the specific blog then they can click on the view more button.

User Persona and Scenario

Richard is 24 years old and he is pursuing his master's degree in Information Systems from Carleton University. As Richard will graduate this year, he is applying for the jobs in his field but as he is new to the application process, he wants to know some essential information regarding the building the resume from the experience person. And for this he will use the blogs functionality to find the required information.

Use Case and Interaction Design

Richard can use TrackMyJobs portal to read and search specific blog post. The portal provides the following features:

1. View all blog post
2. View specific blog post
3. Search blog post

The following is use case to perform the above tasks:

1. User accesses the home page of "TrackMyJobs". [user action]
2. User clicks on the "Sign In" button. [user action]
3. Systems shows the "Sign In" page, which requires user to enter the login credentials. [system action]
4. If user is not registered than they will click on the "Sign up" button to register. [user action]
5. User enters the login credentials. [user action]

6. System gives “Login Successful” feedback message and redirects the user to the home page. [system action]
7. User clicks on “Blogs” and selects “Read Blogs” from the drop-down menu. [user-action]
8. System displays blog post in form of card. [system action]
9. User enters keyword in the search bar. [user action]
10. System filters and displays the required blog post. [system action]

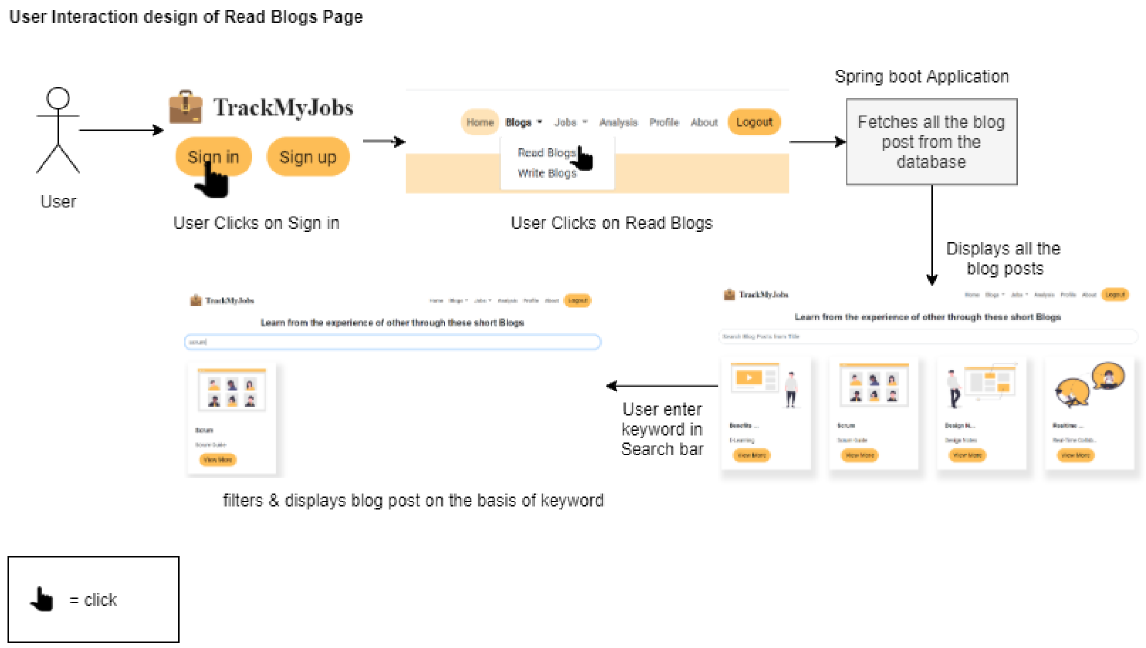

User Interaction design of Read Blogs Page
User
TrackMyJobs
Sign in
Sign up
User Clicks on Sign in
Home
Blogs
Jobs
Analysis
Profile
About
Logout
Read Blogs
Write Blogs
User Clicks on Read Blogs
Spring boot Application
Fetches all the blog post from the database
Displays all the blog posts
Learn from the experience of other through these short Blogs
User enter keyword in Search bar
filters & displays blog post on the basis of keyword
= click

Figure 29: Interaction Design for Read Blogs [7]

3.1.8 Blog Creation

The user interface of write blogs page provides the functionality to write and post the blog post with other users. User can also apply various formatting styles using the editing tools.

User Persona and Scenario

Caroline is 40 years old and she is currently working with the co-op office at Dalhousie University. She has experience and has a ton of knowledge in the field of job application process. She wants to share her knowledge with the other user. She will use the write blog feature provided by our application to share her knowledge in the form of small blogs.

Use Case and Interaction Design

Caroline can use TrackMyJobs to share her knowledge through short blog post and for that our portal provides following feature:

1. Write Blog post

The following is use case to perform the above tasks:

1. User accesses the home page of “TrackMyJobs”. [user action]
2. User clicks on the “Sign In” button. [user action]
3. Systems shows the “Sign In” page, which requires user to enter the login credentials. [system action]
4. If user is not registered than they will click on the “Sign up” button to register. [user action]
5. User enters the login credentials. [user action]
6. System gives “Login Successful” feedback message and redirects the user to the home page. [system action]
7. User clicks on “Blogs” and selects “Write Blogs” from the drop-down menu. [user-action]
8. System displays “Write blog” page. [system action]
9. User enters the title, sub-title, content, keywords, and image of the blog post. [user action]
10. After filling all the required information user clicks on “submit my blog” button. [user-action]
11. System displays the confirmation dialogue box and redirects to the read blogs page. [system-action]

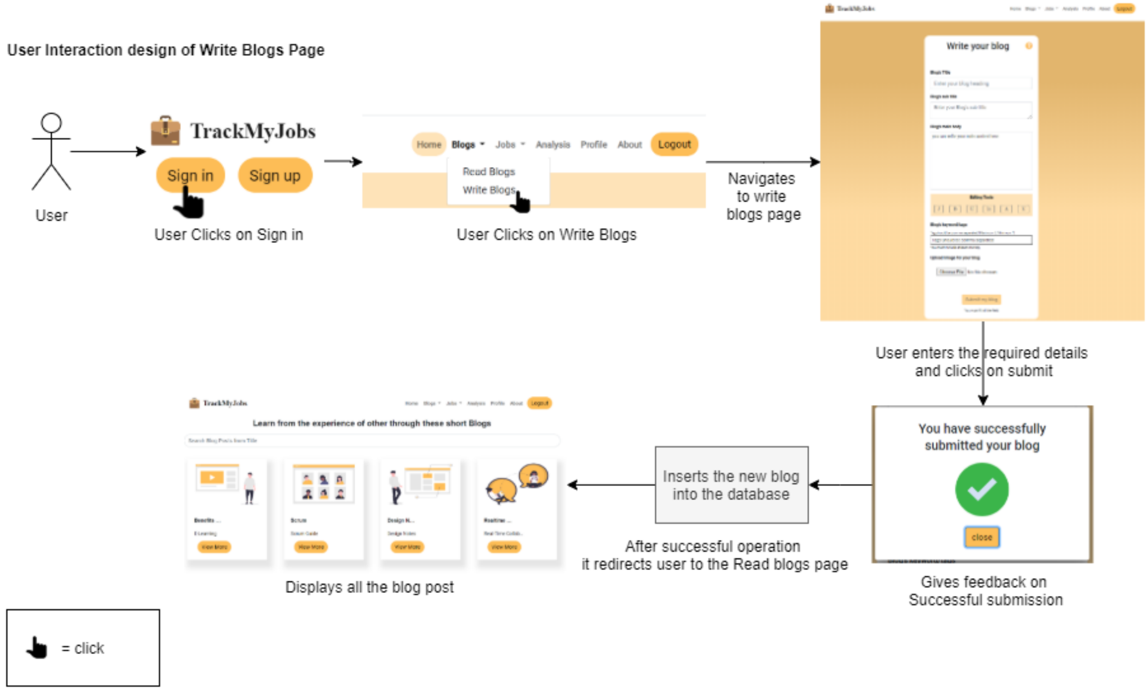

User Interaction design of Write Blogs Page
User
TrackMyJobs
Sign in
Sign up
User Clicks on Sign in
Home
Blogs
Jobs
Analysis
Profile
About
Logout
Read Blogs
Write Blogs
User Clicks on Write Blogs
Navigates to write blogs page
Write your blog
User enters the required details and clicks on submit
You have successfully submitted your blog
close
Gives feedback on Successful submission
Inserts the new blog into the database
After successful operation it redirects user to the Read blogs page
Learn from the experience of other through these short Blogs
Displays all the blog post
= click

Figure 30: Interaction Design for Write Blogs Page [7]

3.1.9 Job Application Activities

The user interface of the Job Application Activities page provides the functionality to view all the todo activities along with their deadlines and completed activities with their completion date which previously added by that user. Moreover, if user wants to add a new activity, it can be done by clicking on the Add Activity button and filling the activity details along with the Activity deadline.

User Persona and Scenario

Robertson is 28 years old and he is pursuing his master's degree in Applied Computer Science from Concordia University. As Robertson will graduate this year, he is applying for the jobs in his field but as he is new to the application process, he wants to keep track of all the activities he needs to do for applying to the jobs along with their deadlines to be completed.

Use Case and Interaction Design

Robertson can use TrackMyJobs website to keep track of specific activities which he needs to while doing his job applications. The portal provides the following features:

1. Add Activity
2. Check/Uncheck ToDo and Completed Items.

The following is use case to perform the above tasks:

1. User accesses the home page of "TrackMyJobs". [user action]
2. User clicks on the "Sign In" button. [user action]
3. Systems shows the "Sign In" page, which requires user to enter the login credentials. [system action]
4. If user is not registered than they will click on the "Sign up" button to register. [user action]
5. User enters the login credentials. [user action]
6. System gives "Login Successful" feedback message and redirects the user to the home page. [system action]
7. User clicks on "Jobs" and selects "Job Activity" from the drop-down menu. [user-action]
8. System displays user's previously added list of activities in form of list. [system action]
9. User clicks on "Add Activity". [user action]
10. System gives a pop-up dialog for entering the activity details. [system action]
11. User enters the activity details and the deadline to complete it. [user action]
12. System saves the activity details and shows them in ToDo Items List. [system action]
13. User clicks on existing activity check box. [user action]
14. System moves the activity to Completed Items list. [system action]
15. User unchecks an activity from Completed Items List. [user action]
16. System moves the activity to ToDo Items List. [system action]

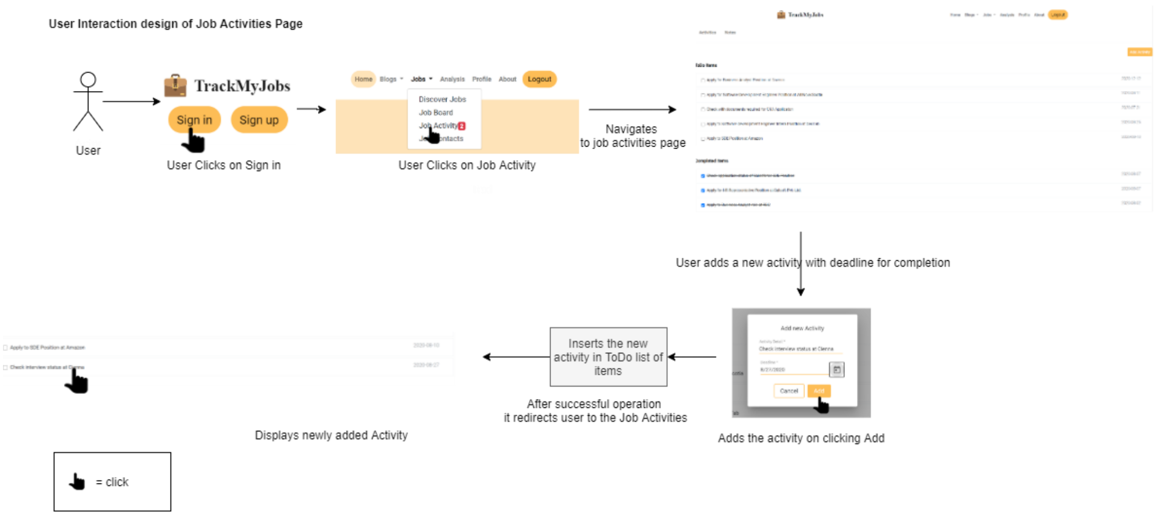

User Interaction design of Job Activities Page
User
TrackMyJobs
Sign in
Sign up
User Clicks on Sign in
Home
Blogs
Jobs
Analysis
Profile
About
Logout
Discover Jobs
Job Board
Job Activity
Job Contacts
User Clicks on Job Activity
Navigates
to job activities page
User adds a new activity with deadline for completion
Add new Activity
Cancel
Add
Adds the activity on clicking Add
Inserts the new activity in ToDo list of items
After successful operation
it redirects user to the Job Activities
Apply to SDE Position at Amazon
Check interview status at Cienna
Displays newly added Activity
= click

Figure 31 Interaction Diagram for Job Activities [7]

3.1.10 Job Application Notes

The user interface of the Job Application Notes feature provides the functionality to view all the notes. Moreover, if user wants to add a new note, it can be done by clicking on the Add Note button and filling the Note details. Also, if the user wants to edit or delete the previously added note, it can be done by clicking on the “edit” and “delete” icons respectively in the note tile.

User Persona and Scenario

Jessica is 24 years old and she is pursuing her master’s degree in Computer Science from University of Texas. As Jessica will graduate this year, she is applying for the jobs in her field, but she wants to keep track of all the notes which she needs to study and learn before she starts applying to the jobs.

Use Case and Interaction Design

Jessica can use TrackMyJobs website to keep track of specific notes which she needs to study or perform while doing her job applications. The portal provides the following features:

1. Add Note
2. Edit Note.
3. Delete Note.

The following is use case to perform the above tasks:

1. User accesses the home page of “TrackMyJobs”. [user action]
2. User clicks on the “Sign In” button. [user action]

3. Systems shows the “Sign In” page, which requires user to enter the login credentials. [system action]
4. If user is not registered than they will click on the “Sign up” button to register. [user action]
5. User enters the login credentials. [user action]
6. System gives “Login Successful” feedback message and redirects the user to the home page. [system action]
7. User clicks on “Jobs” and selects “Job Activity” from the drop-down menu. [user-action]
8. System displays user’s previously added list of activities in form of list. [system action]
9. User clicks on “Notes” tab. [user action]
10. System opens the notes details tab. [system action]
11. User clicks on the Add Note button. [user action]
12. System open the Add Note dialog. [system action]
13. System saves the note details and shows them in Note List. [system action]
14. User clicks on existing note edit icon. [user action]
15. System open the Update Note dialog. [system action]
16. User updates the note and clicks on Update note button. [user action]
17. System updates the note details and shows them in the Note List. [system action]
18. User clicks on the delete icon of the Note list. [user action]
19. System deletes the Note. [system action]

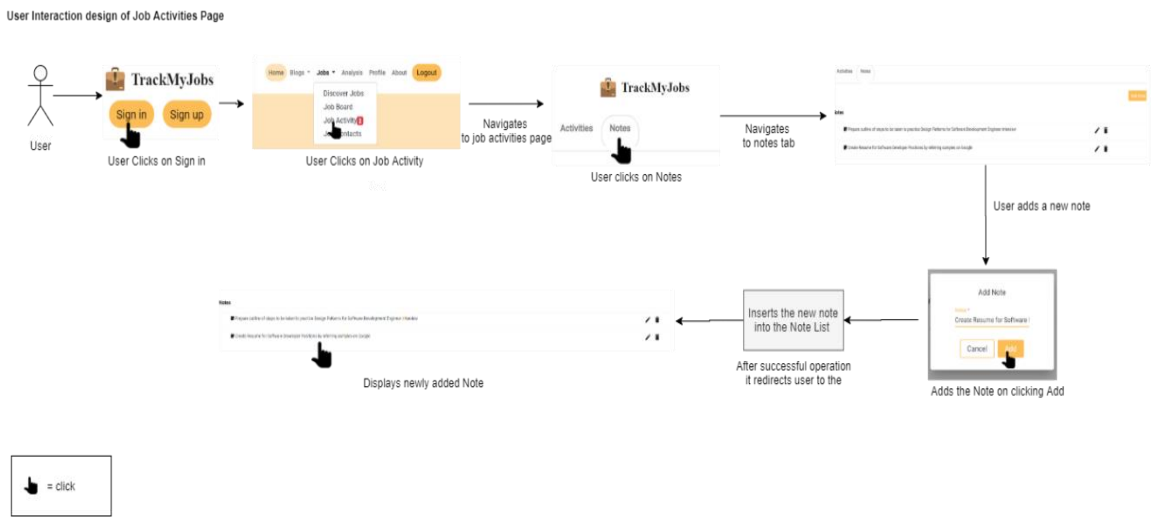


Figure 32 Interaction Diagram of Job Notes [7]

3.1.11 Contact Management

The user interface of the Job Contact Management feature provides the functionality to view all the contact cards. Moreover, if user wants to add a new contact, it can be done by clicking on the Add Contact button and filling the Contact details. Also, if the user wants to edit or delete the previously added contact, it can be done by clicking on the "edit" and "delete" icons respectively in the contact card.

User Persona and Scenario

Ileana is 22 years old and she is pursuing her bachelor's degree in Electrical Engineering from North Eastern University. As Ileana will graduate this year, she is applying for the jobs in her field, but she wants to manage contacts of all the people to whom she may need to communicate further.

Use Case and Interaction Design

Ileana can use TrackMyJobs website to manage the specific contacts of the people she needs to communicate further during the job applications. The portal provides the following features:

1. Add Contact
2. Edit Contact
3. Delete Contact

The following is use case to perform the above tasks:

1. User accesses the home page of "TrackMyJobs". [user action]
2. User clicks on the "Sign In" button. [user action]
3. Systems shows the "Sign In" page, which requires user to enter the login credentials. [system action]
4. If user is not registered than they will click on the "Sign up" button to register. [user action]
5. User enters the login credentials. [user action]
6. System gives "Login Successful" feedback message and redirects the user to the home page. [system action]
7. User clicks on "Jobs" and selects "Job Contacts" from the drop-down menu. [user-action]
8. System displays user's previously added contacts in form of cards. [system action]
9. User writes "Contact Name" in the search bar tab. [user action]
10. System filters out the contact cards on the entered Contact Name. [system action]
11. User clicks on the Add Contact button. [user action]
12. System open the Add Contact dialog. [system action]
13. System saves the Contact Details and shows them in Contact cards form. [system action]
14. User clicks on existing Contact Card "edit" icon. [user action]
15. System open the Update Contact dialog. [system action]
16. User updates the Contact details and clicks on Update Contact button. [user action]
17. System updates the Contact details and shows them in the Contact card. [system action]
18. User clicks on the delete icon of the Contact Card. [user action]
19. System deletes the Contact. [system action]

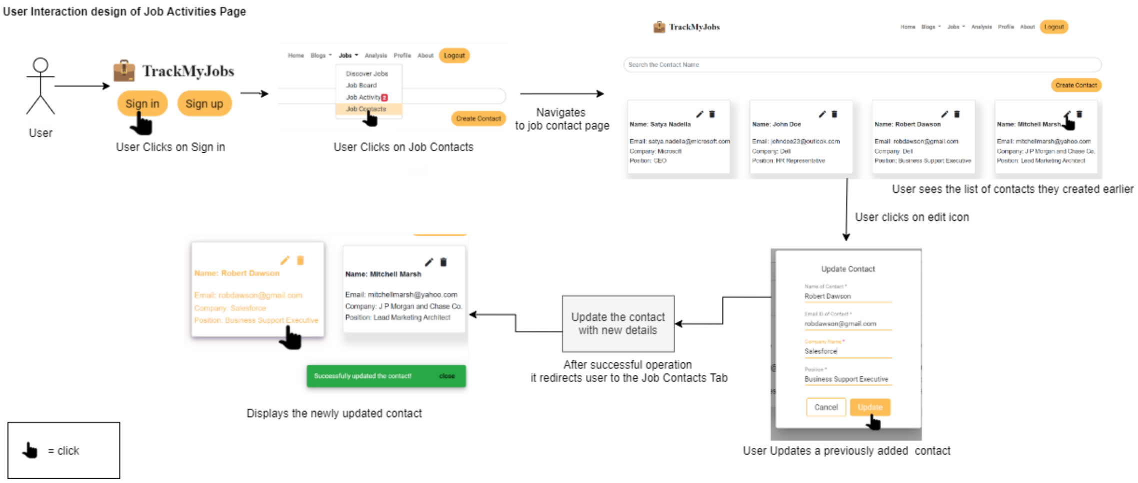
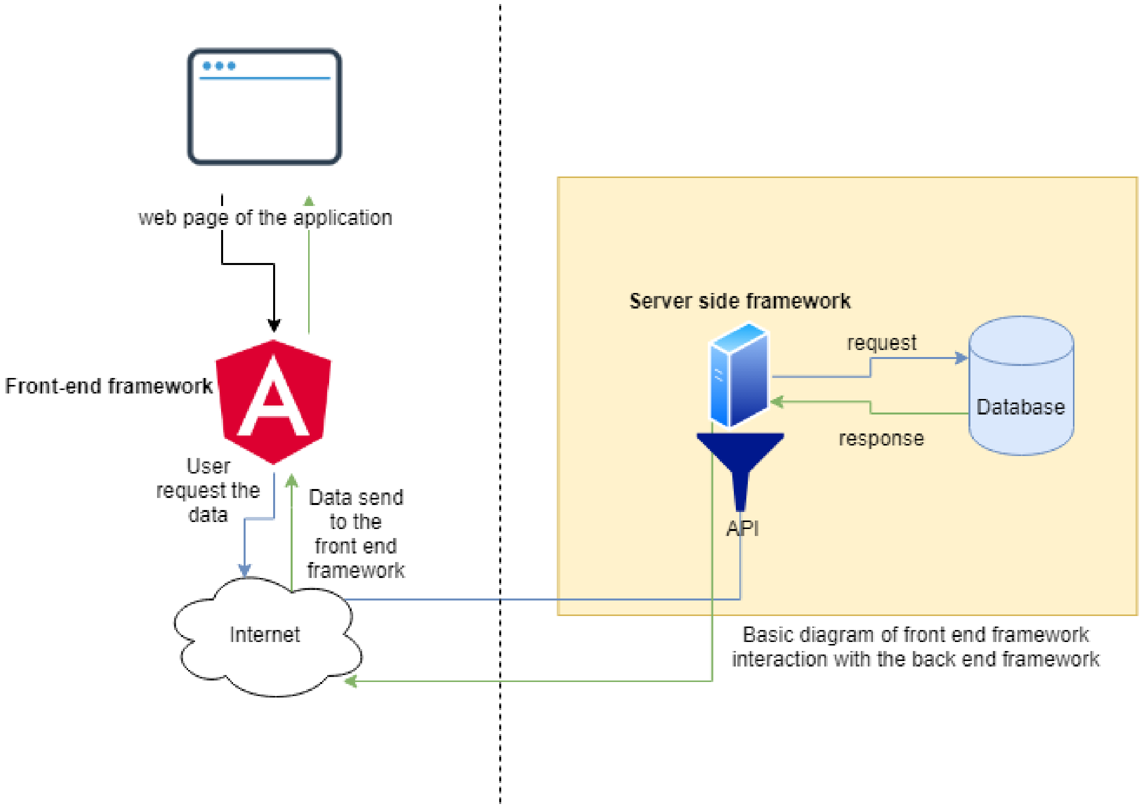

User Interaction design of Job Activities Page
User
TrackMyJobs
Sign in
Sign up
User Clicks on Sign in
Home Blogs Jobs Analysis Profile About Logout
Discover Jobs
Job Board
Job Activity
Job Contacts
Create Contact
User Clicks on Job Contacts
Navigates
to job contact page
TrackMyJobs
Search the Contact Name
Create Contact
Name: Satya Nadella
Email: satya.nadella@microsoft.com
Company: Microsoft
Position: CEO
Name: John Doe
Email: johndoe23@outlook.com
Company: Dell
Position: HR Representative
Name: Robert Dawson
Email: robdawson@gmail.com
Company: Dell
Position: Business Support Executive
Name: Mitchell Marsh
Email: mitchellmarsh@yahoo.com
Company: J P Morgan and Chase Co.
Position: Lead Marketing Architect
User sees the list of contacts they created earlier
User clicks on edit icon
Update Contact
Name of Contact *
Robert Dawson
Email ID of Contact *
robdawson@gmail.com
Company Name *
Salesforce
Position *
Business Support Executive
Cancel
Update
User Updates a previously added contact
Update the contact
with new details
After successful operation
it redirects user to the Job Contacts Tab
Name: Robert Dawson
Email: robdawson@gmail.com
Company: Salesforce
Position: Business Support Executive
Name: Mitchell Marsh
Email: mitchellmarsh@yahoo.com
Company: J P Morgan and Chase Co.
Position: Lead Marketing Architect
Successfully updated the contact!
close
Displays the newly updated contact
= click

Figure 33 Job Contacts Interaction Diagram [7]

3.2 Process and Service Workflow

Most of the features are supported by backend as shown in Figure 34. The data is retrieved from database for individual user.



web page of the application
Front-end framework
User request the data
Data send to the front end framework
Internet
Server side framework
request
response
Database
API
Basic diagram of front end framework interaction with the back end framework

Figure 34: System Architecture of TrackMyJobs [15] [7]

3.2.1 Profile Management

Sign-in/Sign-up

TrackMyJobs use Angular as web application framework. In sign-in and sign-up feature, the steps of backend for doing the login and registration in real time are listed below:

1. The angular front application will call the REST endpoint of the firebase real-time database.
2. The endpoint will require certain parameters for accepting the request and the validates the email id and password are valid and present.
3. The endpoint will return certain amount of data which will contain email and authentication token.
4. Then the communication will happen with the same token provided by the firebase server.

Reset Password

TrackMyJobs use Angular as web application framework. In password management feature, the steps of backend for doing the password management in real time are listed below:

1. The angular front application will call the REST endpoint of the firebase real-time database.
2. The endpoint will require certain parameters for accepting the request and validates the email id is valid and present.
3. The endpoint will send a reset password email on the provided email address and user will have to use the link provided for setting new password.
4. Then the communication will happen with the same token provided by the firebase server.

3.2.2 Job Search

TrackMyJobs use Angular as web application framework. In job search feature, the steps of backend for doing the job search in real time are listed below:

1. Controller (JavaScript) sends keyword to job service.

2. Job service sends the request to the job search API.

3. Job search API sends the response back to job service.

4. Job service extracts data and sends them back to the controller.

After these four steps, Angular renders the data to users on the UI. Users trigger job search feature by providing keyword and click “search.” Then, the system will process the request as elaborated above.

3.2.3 Job Discovery

Job discovery has two main features which are adding the job to wish list and applied. Wish list and apply features have similar process. Both features add a job to database with the related status. Users can trigger both features by clicking on the button (“apply via adzuna” or “wish list”). The method calls service to connect to MySQL and add data to the table. The steps of backend for doing the job discovery feature are listed below:

1. Controller (JavaScript) sends request to job app service.

2. Job app service parses request and make request to Spring Boot Server.

3. Spring Boot Server connect to database using JDBC.

4. MySQL process the request and add data to the table in database.

3.2.4 Job Application Management

A job application management has three major tasks which are to create a new job application, delete a job application and to change the status of a job application. The following steps are performed after any user interaction through user interface:

1. Angular service component makes an HTTP request to a back-end REST API endpoint for a specific job application management task.
2. Spring Boot application handles each request and communicates with MySQL database as per business requirements.
3. Spring Boot application sends response to the Angular service component that further parse the response and renders the front-end of the application.

Figure 35 illustrates the process workflow diagram of job application management.

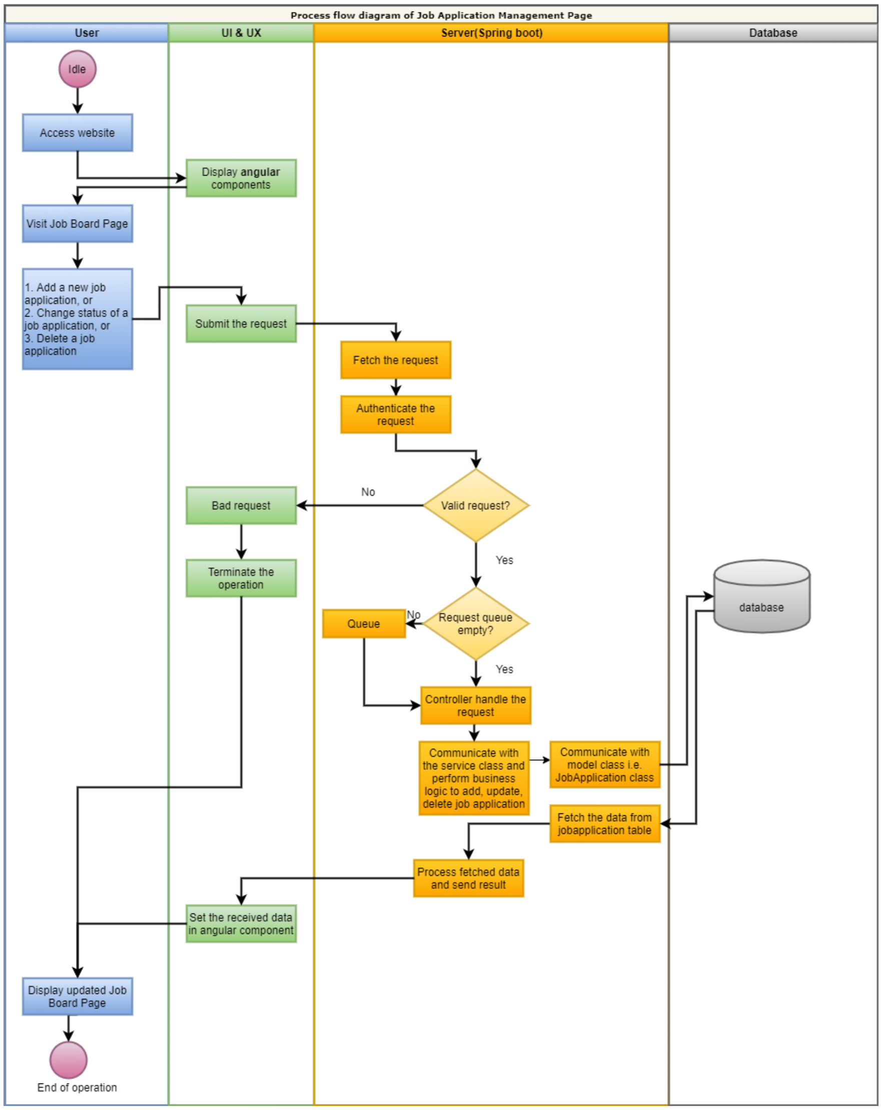

Process flow diagram of Job Application Management Page
User
UI & UX
Server(Spring boot)
Database
Idle
Access website
Display angular components
Visit Job Board Page
1. Add a new job application, or
2. Change status of a job application, or
3. Delete a job application
Submit the request
Fetch the request
Authenticate the request
Valid request?
No
Bad request
Terminate the operation
Yes
Request queue empty?
No
Queue
Yes
Controller handle the request
Communicate with the service class and perform business logic to add, update, delete job application
Communicate with model class i.e. JobApplication class
database
Fetch the data from jobapplication table
Process fetched data and send result
Set the received data in angular component
Display updated Job Board Page
End of operation

Figure 35 Process flow diagram of Job Application Management [7]

3.2.5 Job Board Management

System creates a job board page for each user in the database when user visits the job board menu for the first time. The following steps are to perform to create a job board page.

1. User visits the Job Board menu in Job section of the navigation bar.
2. Angular service component makes an HTTP request to a back-end REST API endpoint for a specific job application management task.
3. Spring Boot application handles each request and communicates with MySQL database as per business requirements.
4. Spring Boot application sends response to the Angular service component that further parse the response and renders the front-end of the application with a new job board page.

Figure 36 illustrates the process flow diagram of Job Board.

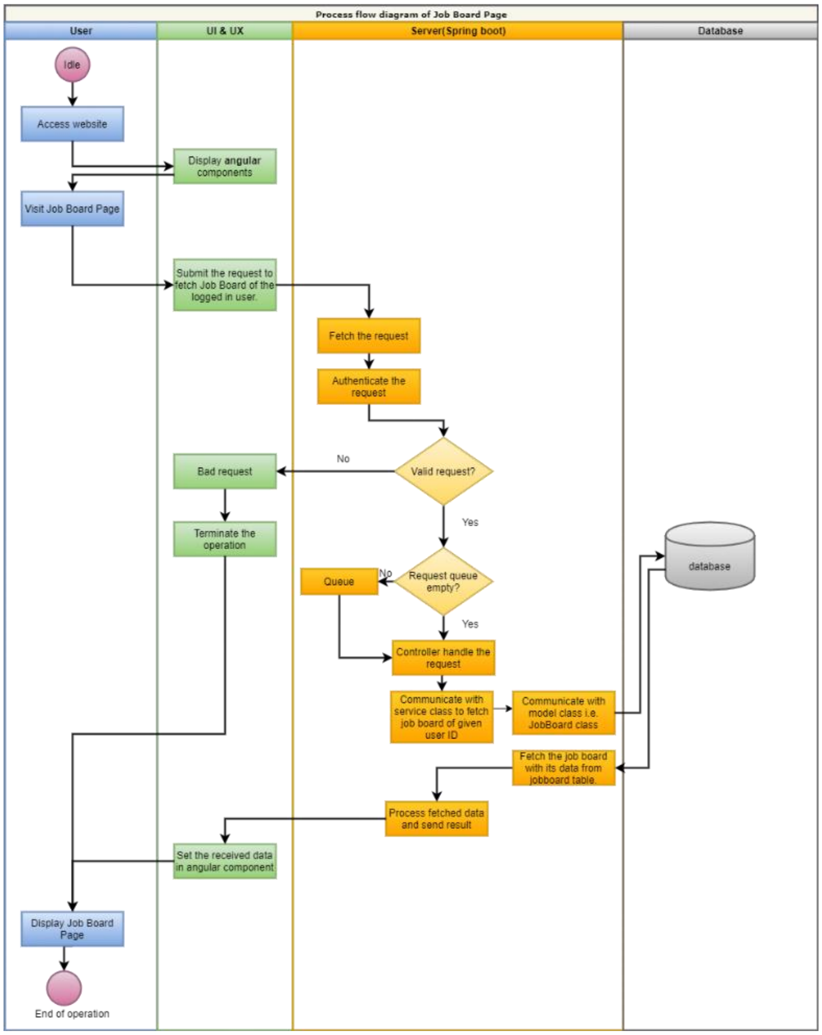

Process flow diagram of Job Board Page
User
UI & UX
Server(Spring boot)
Database
Idle
Access website
Display angular components
Visit Job Board Page
Submit the request to fetch Job Board of the logged in user.
Fetch the request
Authenticate the request
No
Valid request?
Bad request
Terminate the operation
Yes
database
Queue
No
Request queue empty?
Yes
Controller handle the request
Communicate with service class to fetch job board of given user ID
Communicate with model class i.e. JobBoard class
Fetch the job board with its data from jobboard table.
Process fetched data and send result
Set the received data in angular component
Display Job Board Page
End of operation

Figure 36 Process flow diagram of Job Board [7]

3.2.6 Job Application Analytics

The data of job board analysis page will be preloaded when the user clicks on the job board analysis page. This page only makes a get request to get all the data required to show the analysis to the user. We have only allowed the view permission of the graph and that is why it is the view only web page. User is not allowed to write any input or modify the data.

1. User visits the job board analysis page.
2. Angular component calls the service method. which makes the http request to a back-end REST API endpoint.
3. Here as we have mentioned the API will be GET request.
4. Spring boot application fetch the data from the data base based on the user ID.
5. Spring boot application sends the response to the angular service component that further parse the response and render the front-end of the application with the analysis data.

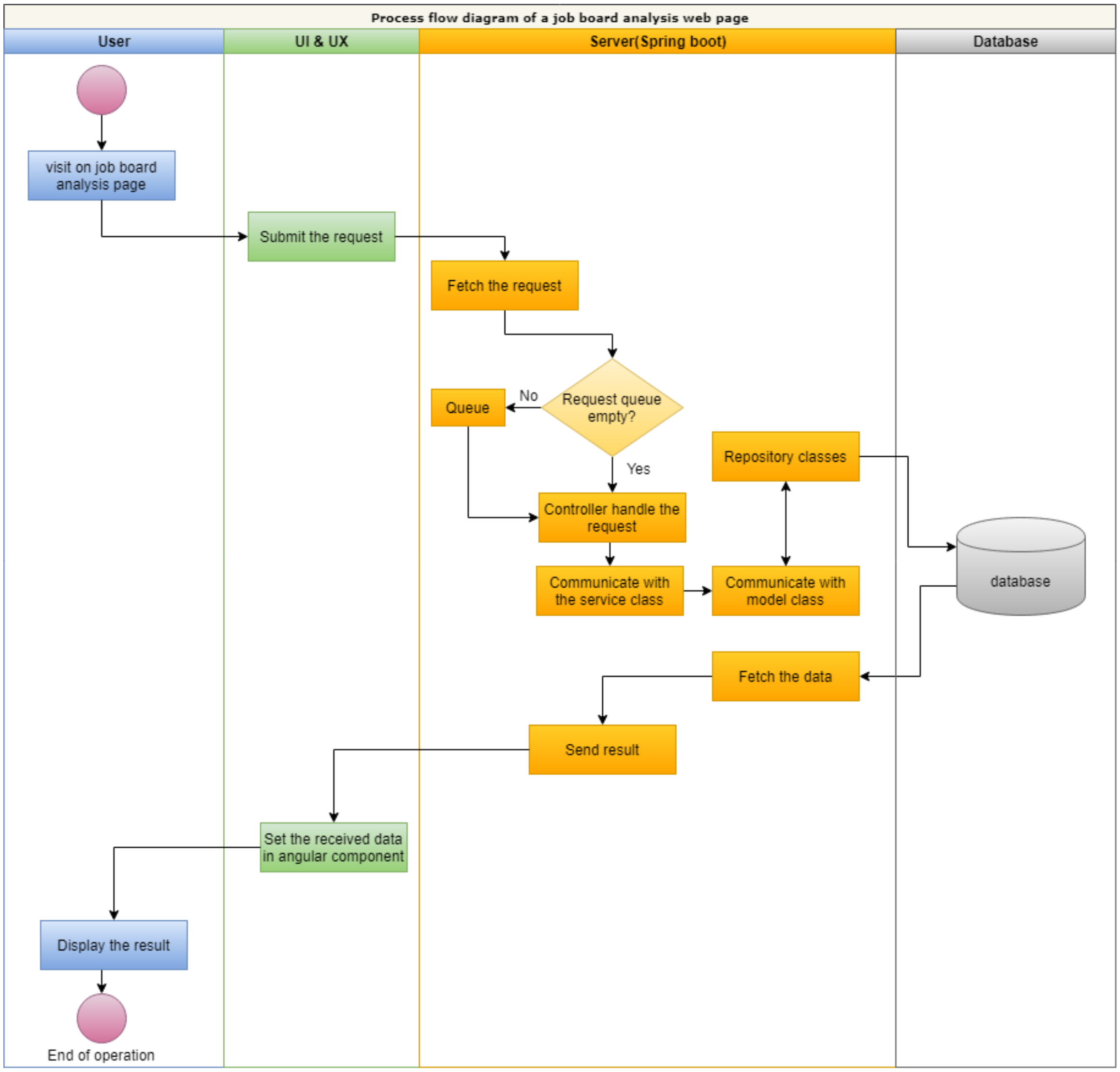

Process flow diagram of a job board analysis web page
User
UI & UX
Server(Spring boot)
Database
visit on job board analysis page
Submit the request
Fetch the request
Queue
No
Request queue empty?
Yes
Repository classes
Controller handle the request
Communicate with the service class
Communicate with model class
database
Fetch the data
Send result
Set the received data in angular component
Display the result
End of operation

Figure 38 Process flow diagram of Job board analysis [7]

3.2.7 Blog Management

The Read Blogs page fetches all the blog post from the database when user clicks on the “Read Blogs” page. Using this page user can look at all the blog posts. Additionally, user can also search specific blog post using the search functionality as well as read content of specific blog post by clicking on the view more or blog post card. The below given steps shows how the frontend and backend work together to give user the desired results.

1. When user access the Read blogs page, the angular component makes an HTTP request to the Backend REST API created in spring boot.
2. All the Request are handled by the backend application and required operations are performed on the MySQL database.
3. After the required operation is performed on the database it sends the required response to the angular application where it displays the result.

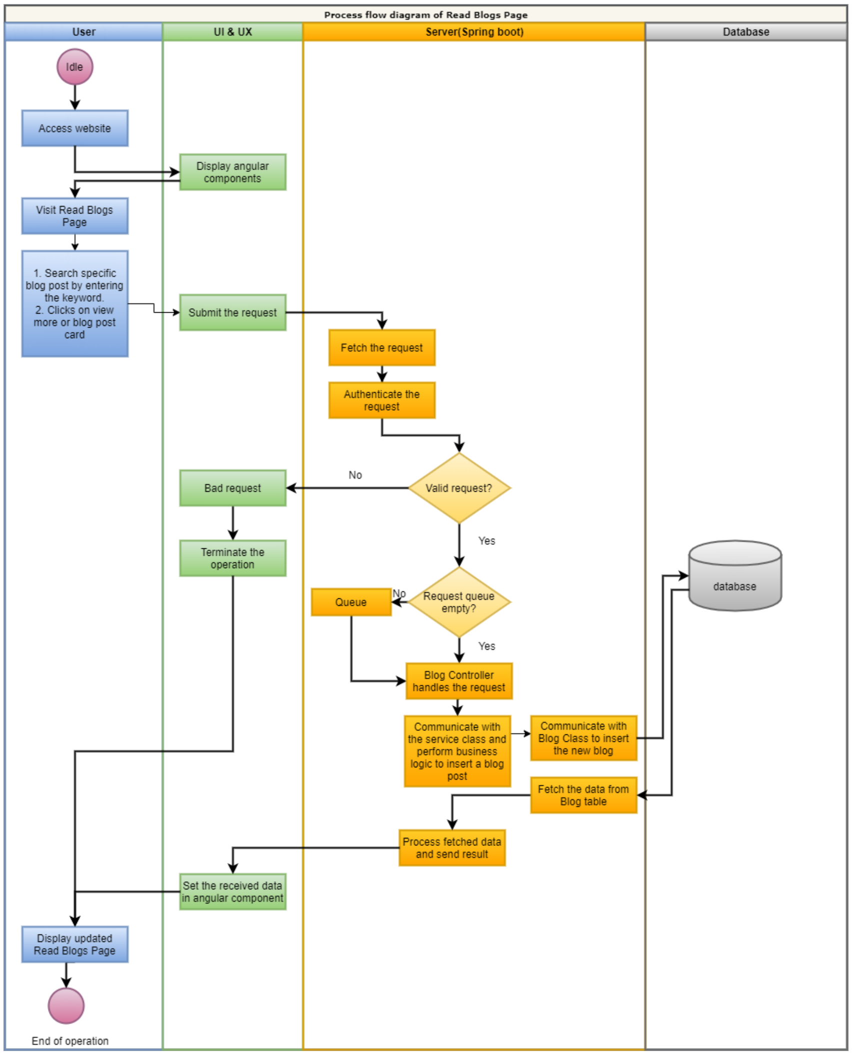

Process flow diagram of Read Blogs Page
User
UI & UX
Server(Spring boot)
Database
Idle
Access website
Display angular components
Visit Read Blogs Page
1. Search specific blog post by entering the keyword.
2. Clicks on view more or blog post card
Submit the request
Fetch the request
Authenticate the request
No
Valid request?
Bad request
Terminate the operation
Yes
database
Queue
No
Request queue empty?
Yes
Blog Controller handles the request
Communicate with the service class and perform business logic to insert a blog post
Communicate with Blog Class to insert the new blog
Fetch the data from Blog table
Process fetched data and send result
Set the received data in angular component
Display updated Read Blogs Page
End of operation

Figure 39: Process flow diagram of Read Blogs page [7]

3.2.8 Blog Creation

Blog creation functionality provides user the feature to share their knowledge in the form of short blogs. The below given Figure 39 [7] shows the process flow of how the HTTP POST request is handled and new blog is inserted into the database. The below given steps explains the process in brief.

1. After User sign’s in successfully and accesses the Write Blogs page.
2. User fills all the required data such title, sub-title, content, keywords, and image of the blog post.
3. After user clicks on the submit button the angular component makes the HTTP POST request which is handled by the spring boot. It performs the insert operation on the database. And After the operation is performed successfully it redirects the user to Read blogs page and all the blog posts.

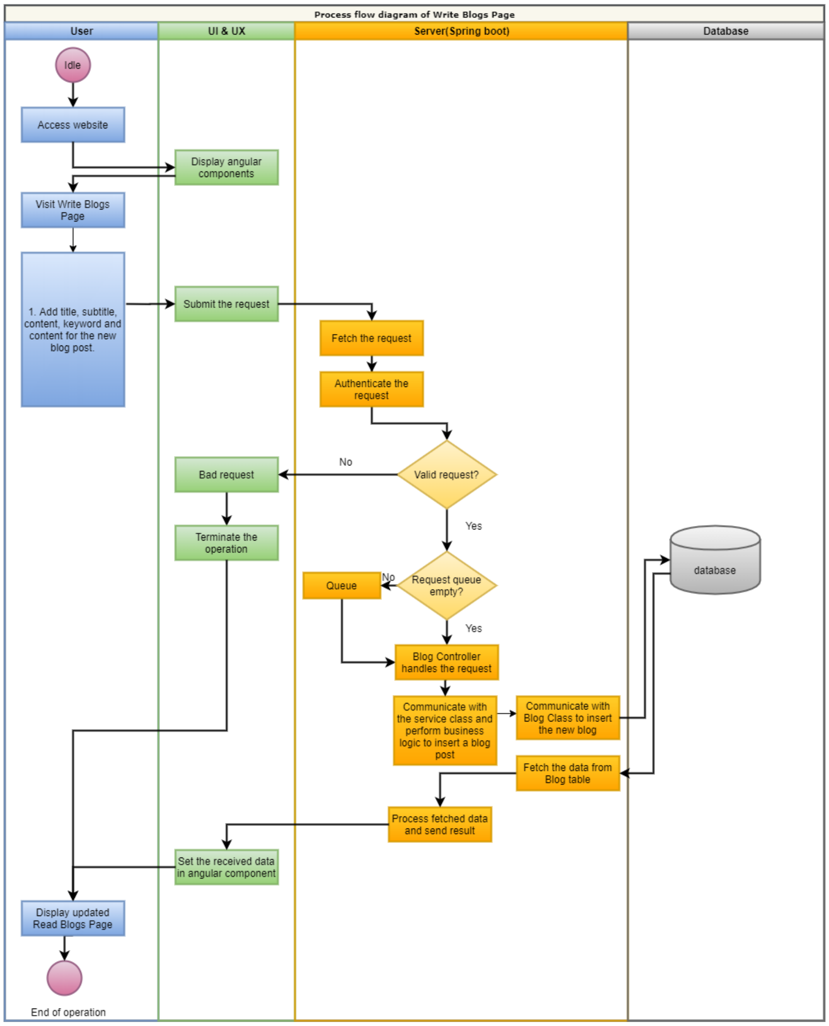

Process flow diagram of Write Blogs Page
User
UI & UX
Server(Spring boot)
Database
Idle
Access website
Display angular components
Visit Write Blogs Page
1. Add title, subtitle, content, keyword and content for the new blog post.
Submit the request
Fetch the request
Authenticate the request
No
Valid request?
Bad request
Terminate the operation
Yes
database
Queue
No
Request queue empty?
Yes
Blog Controller handles the request
Communicate with the service class and perform business logic to insert a blog post
Communicate with Blog Class to insert the new blog
Fetch the data from Blog table
Process fetched data and send result
Set the received data in angular component
Display updated Read Blogs Page
End of operation

Figure 40: Process flow diagram of Write Blogs page [7]

3.2.9 Job Application Activities

The Job Application Activities page fetches all the Job Activities related to the user from the database when user clicks on the “Job Activity” Tab. Using this page, user can look at its ToDo Items List and Completed Items List. User can add new activity. User can also check or uncheck activity from ToDo Items List or Completed Items List, respectively. The below given steps shows how the frontend and backend work together to give user the desired results.

1. When user access the Job Activities page, the angular component makes an HTTP request to the Backend REST API created in spring boot.
2. All the Request are handled by the backend application and required operations are performed on the MySQL database.
3. After the required operation is performed on the database it sends the required response to the angular application where it displays the result.

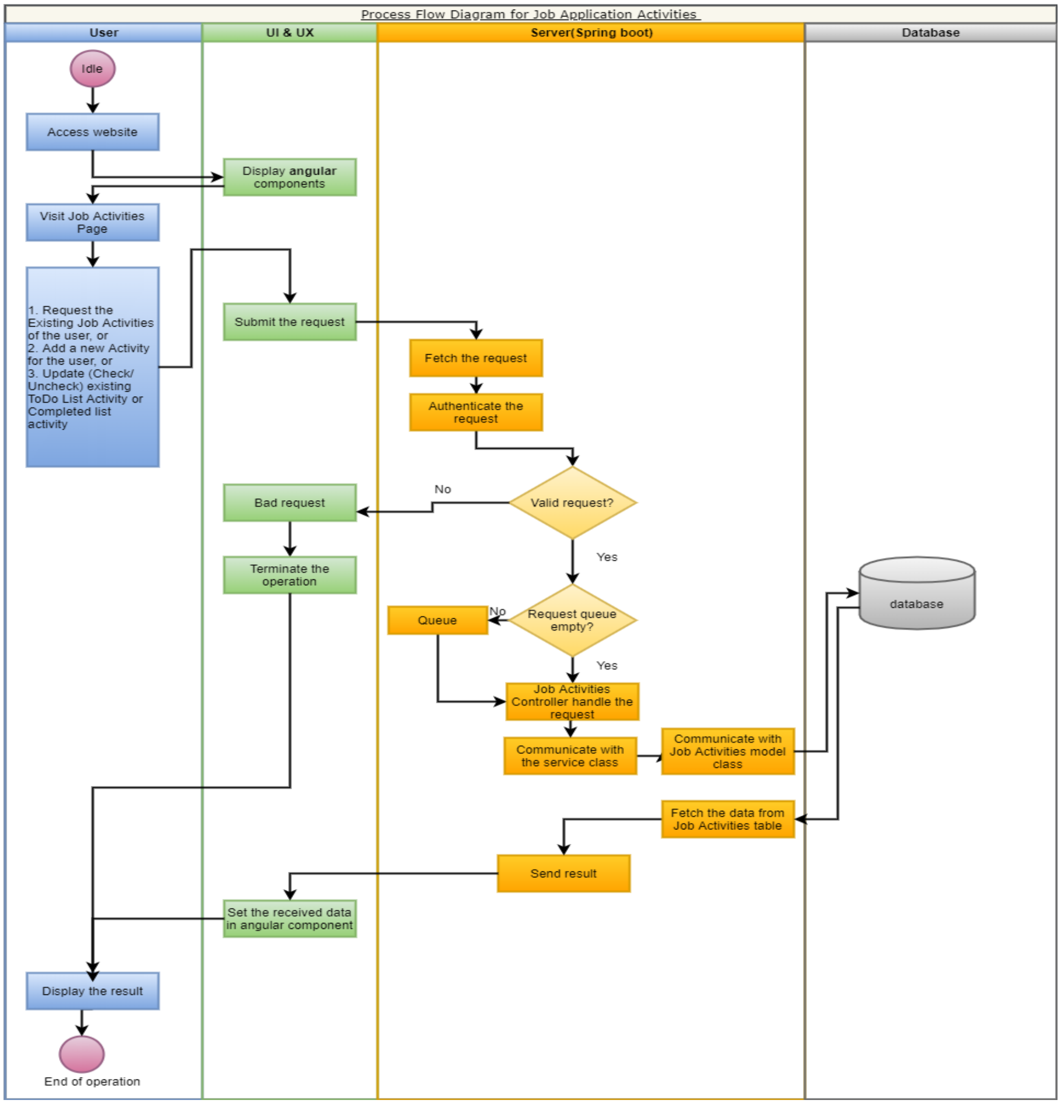

Process Flow Diagram for Job Application Activities
User
UI & UX
Server(Spring boot)
Database
Idle
Access website
Display angular components
Visit Job Activities Page
1. Request the Existing Job Activities of the user, or
2. Add a new Activity for the user, or
3. Update (Check/ Uncheck) existing ToDo List Activity or Completed list activity
Submit the request
Fetch the request
Authenticate the request
No
Valid request?
Bad request
Yes
Terminate the operation
database
Queue
No
Request queue empty?
Yes
Job Activities Controller handle the request
Communicate with the service class
Communicate with Job Activities model class
Fetch the data from Job Activities table
Send result
Set the received data in angular component
Display the result
End of operation

Figure 41 Process flow diagram of Job Activities [7]

3.2.10 Notification Management

The notification data will be loaded from the database when the user log-in. We have fetched the data initially because we are also providing notification to the user in the as a badge in the navigation bar. Once the user visits the job board page, if the user has any deadline for activity within 7 days, then the user will be prompted with a notification. If the user does not have any type of activity deadline within the next seven days then the user will not see any deadline. To improve the user-experience, the notification will only be prompted for one time in that session.

1. User log-in in the application.
2. Angular component calls the service method. which makes the http request to a back-end REST API endpoint.
3. Spring boot application fetch the data from the data base based on the user_id.
4. Spring boot application sends the response to the angular service component that further parse the response and render the front-end of the application with the analysis data.
5. Initially notification will be seen as a badge in the navigation bar.
6. User visits job board page.
7. User is prompted with deadline notification (If any).

3.2.11 Job Application Notes

The Job Application Notes tab fetches all the Job Notes related to the user from the database when user clicks on the “Job Notes” Tab. Using this tab, user can look at its notes list. User can add new notes. The user can also update or delete the previously added Notes. The below given steps shows how the frontend and backend work together to give user the desired results.

1. When user access the Job Notes tab, the angular component makes an HTTP request to the Backend REST API created in spring boot.
2. All the Request are handled by the backend application and required operations are performed on the MySQL database.
3. After the required operation is performed on the database it sends the required response to the angular application where it displays the result.

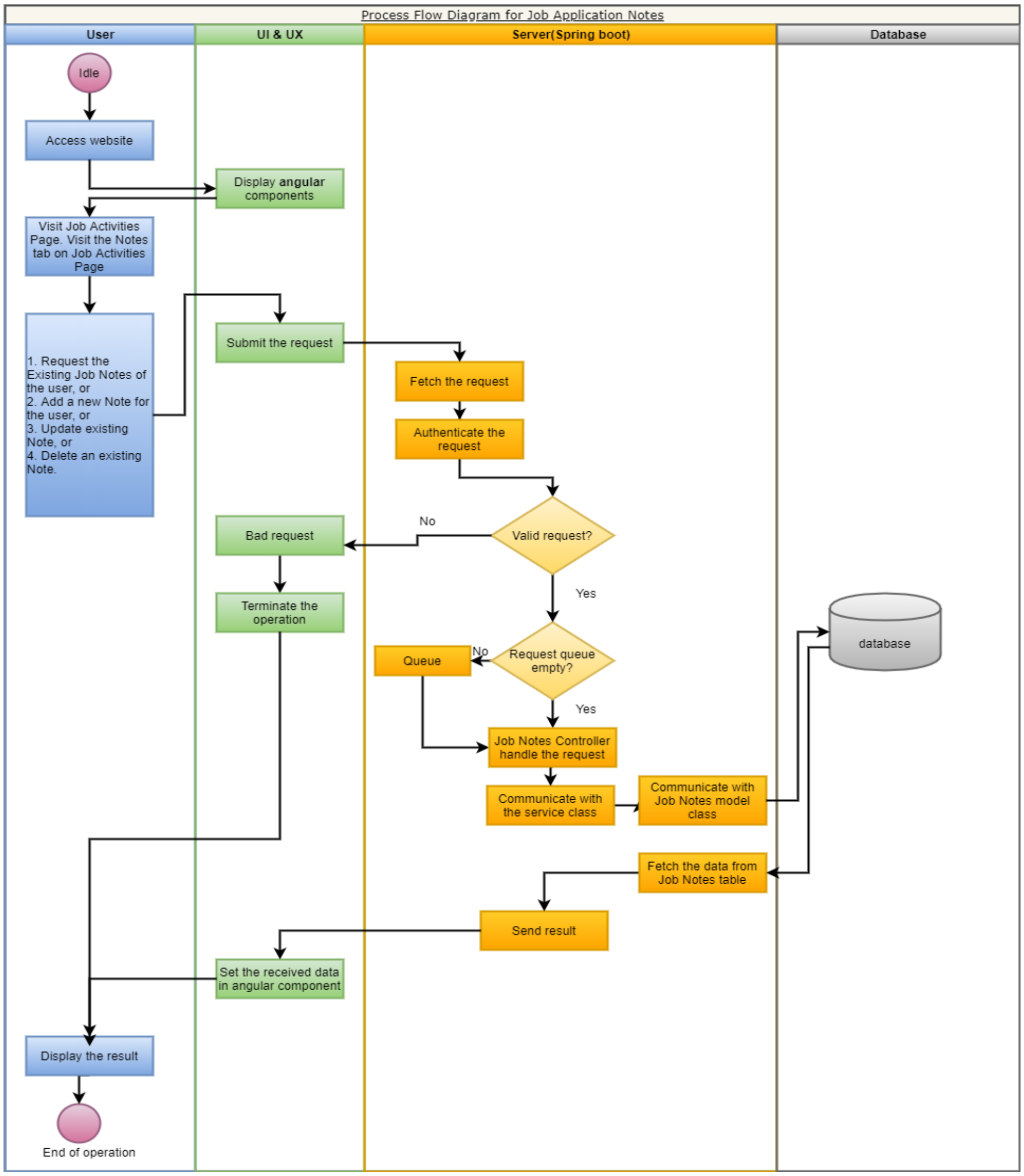

Process Flow Diagram for Job Application Notes
User
UI & UX
Server(Spring boot)
Database
Idle
Access website
Display angular components
Visit Job Activities Page. Visit the Notes tab on Job Activities Page
1. Request the Existing Job Notes of the user, or
2. Add a new Note for the user, or
3. Update existing Note, or
4. Delete an existing Note.
Submit the request
Fetch the request
Authenticate the request
Valid request?
No
Bad request
Terminate the operation
Yes
Request queue empty?
No
Queue
Yes
Job Notes Controller handle the request
Communicate with the service class
Communicate with Job Notes model class
database
Fetch the data from Job Notes table
Send result
Set the received data in angular component
Display the result
End of operation

Figure 42 Process flow diagram of Job Notes [7]

3.2.12 Job Application Contacts

The Job Application Contacts tab fetches all the Job Contacts related to the user from the database when user clicks on the “Job Contacts” Tab. Using this tab, user can look at its notes list. User can add new Contacts. The user can also update or delete the previously added Contacts. The below given steps shows how the frontend and backend work together to give user the desired results.

1. When user access the Job Contacts tab, the angular component makes an HTTP request to the Backend REST API created in spring boot.
2. All the Request are handled by the backend application and required operations are performed on the MySQL database.
3. After the required operation is performed on the database it sends the required response to the angular application where it displays the result.

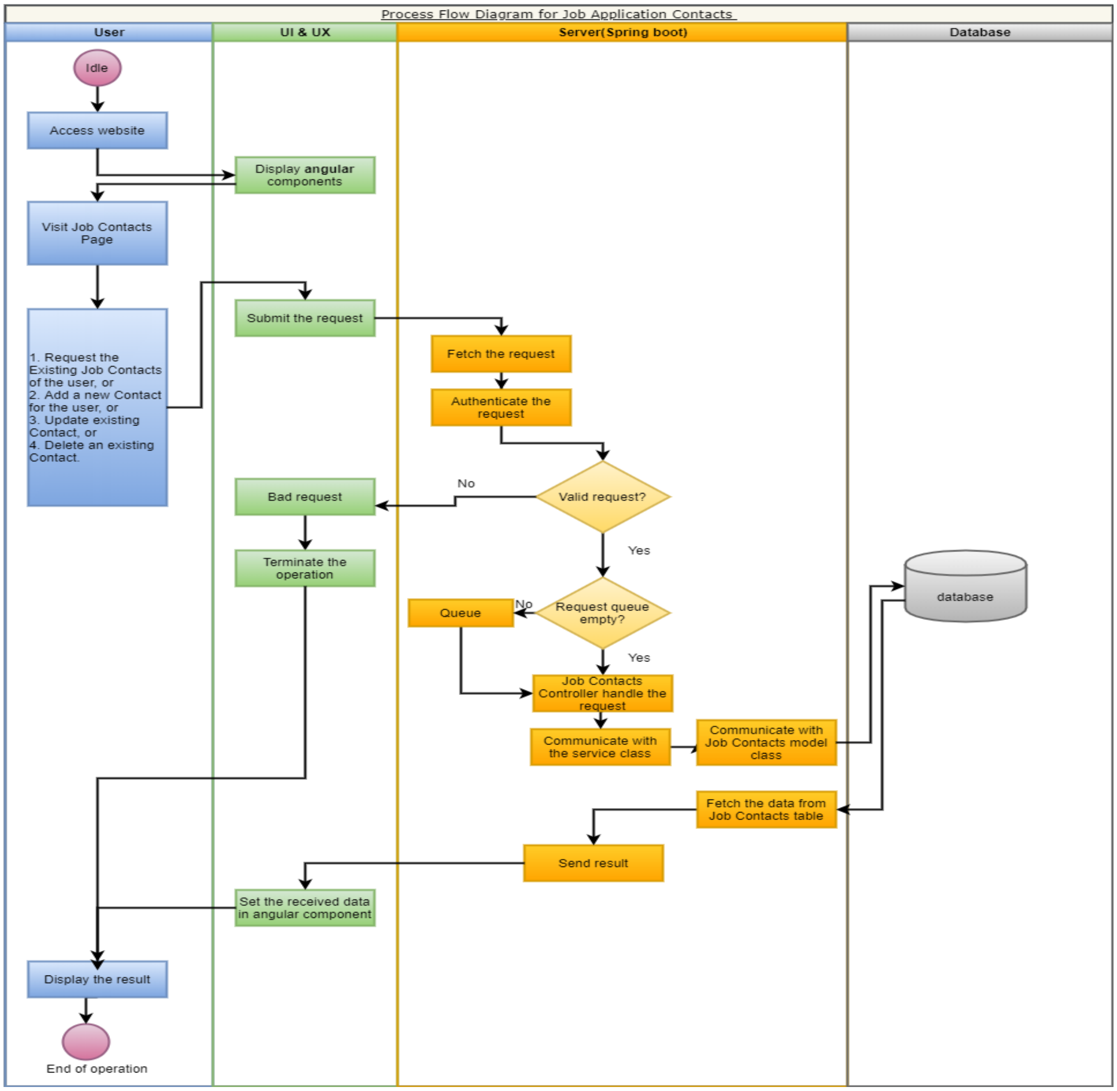

Process Flow Diagram for Job Application Contacts
User
UI & UX
Server(Spring boot)
Database
Idle
Access website
Display angular components
Visit Job Contacts Page
Submit the request
1. Request the Existing Job Contacts of the user, or
2. Add a new Contact for the user, or
3. Update existing Contact, or
4. Delete an existing Contact.
Fetch the request
Authenticate the request
Bad request
No
Valid request?
Yes
Terminate the operation
database
Queue
No
Request queue empty?
Yes
Job Contacts Controller handle the request
Communicate with the service class
Communicate with Job Contacts model class
Fetch the data from Job Contacts table
Send result
Set the received data in angular component
Display the result
End of operation

Figure 43 Process flow diagram of Job Contacts [7]

4. Application Testing

4.1 Testcases

4.1.1 Test Case 1 **[16]**

To test the job board page fetched for current sign in user.

1. After signing in, click on **Job Board** menu in the navigation bar.

Expected Result:

1. If you are a new user, then it will create a new blank job board in the database.
2. Display job board with partitions of different statuses, e.g. Wishlist, Applied, Interview, Offer and Reject.
3. It will also display job application cards for all job applications fetched from the application database if there are any.
4. Each job application contains the company name, job title and a delete button.

4.1.2 Test Case 2 **[16]**

To test add a new job application on a job board.

1. After signing in, click on **Job Board** menu in the navigation bar.
2. Click on **Add Job** button of a job status partition (e.g. Applied, Wishlist, etc.) in which you want to add a job.
3. A dialogue will open and ask to fill job application detail.
4. Enter a job title and a company detail in the dialogue box.
5. Click **Ok** button to create a new job.

Expected Result:

1. After clicking on the Job Board button, all job applications will be displayed by status wise on the job board.
2. After filling data and clicking on the Ok button, a REST API is called to create a new job application to the database. The spinner will be displayed while calling the API.
3. A notification with a success message will be displayed to the user.
4. The newly created job application card will be displayed on that job status partition.

4.1.3 Test Case 3 **[16]**

To test delete a job application from the job board.

1. After signing in, click on **Job Board** menu in the navigation bar.
2. All job applications will be loaded from the database and will be displayed on a card.
3. Click on **delete icon** button to delete the job application from the database.

Expected Result:

1. After clicking on the Job Board button, all job applications will be displayed by status wise on the job board.
2. After clicking on a delete icon of a job application, a REST API is called to delete the job application from the database. The spinner will be displayed until the response of the call will be received.
3. The job application card will be removed from the job board and notification with a success message will be displayed.

4.1.4 Test Case 4 **[16]**

To test the changing status of a job application of a job board.

1. After signing in, click on **Job Board** menu in the navigation bar.
2. All job applications will be loaded from the database and will be displayed on a card.
3. Drag a job application whose status you want to change and drop it into the destination job status partition.

Expected Result:

1. After clicking on the Job Board button, all job applications will be displayed by status wise on the job board.
2. After drag and drop operation, a REST API is called to change the status of the job application in the database. The spinner will be displayed until the response of the call will be received.
3. The job application card will be moved to the destination partition and notification with a success message will be displayed to the user.

4.1.5 Test Case 5 **[16]**

To test the job board page fetched for current sign in user when the application server is down.

1. After signing in, click on **Job Board** menu in the navigation bar.

Expected Result:

1. A spinner will be displayed to the user for a while.
2. A notification with an error message will be displayed to the user.

4.1.6 Test Case 6 **[16]**

To test errors in the application server while deleting job applications from the job board.

1. After signing in, click on **Job Board** menu in the navigation bar.
2. All job applications will be loaded from the database and will be displayed on a card.
3. Click on **delete icon** button to delete the job application from the database.

Expected Result:

1. After clicking on the Job Board button, all job applications will be displayed by status wise on the job board.
2. After clicking on a delete icon of a job application, a REST API is called to delete the job application from the database. The spinner will be displayed until the response of the call will be received.
3. The job application card will be removed from the job board, but a notification with an error message will be displayed.
4. The job application will be displayed in that job status partition once the user will refresh the portal.

4.1.7 Test Case 7 [16]:

This test case is to check the written blog summary for the logged-in user

1. Login to the web page
2. Navigate to the job-board analysis page

Expected result:

The user will be able to see his written blog summary for that specific year in the form of a line graph

Failed result:

If there is an error in the database or the value is not fetched properly, the user will see an error message

4.1.8 Test Case 8 [16]:

This test case is to check the job application summary for the logged-in user

1. Login to the web page
2. Navigate to the job-board analysis page

Expected result:

The user will be able to see his job application summary in the form of a horizontal bar graph.

Failed result:

If there is an error in the database or the value is not fetched properly, the user will see an error message

4.1.9 Test Case 9 [16]:

This test case is to check job activity summary for the logged-in user

1. Login to the web page
2. Navigate to the job-board analysis page

Expected result*:*

The user will be able to see his job activity summary in the stacked bar graph.

Failed result:

If there is an error in the database or the value is not fetched properly, the user will see an error message

4.1.10 Test Case 10[16]:

This test case is to check the job application summary of the past 6 months.

1. Login to the web page
2. Navigate to the job-board analysis page
3. Change the duration in job application analysis to six months.

Expected result*:*

The user will be able to see the job application summary of the last six months.

Failed result:

If there is an error in the database or the value is not fetched properly, the user will see an error message

4.1.11 Test Case 11

This test case is to check the job application summary of the past 6 months.

4. Login to the web page
5. Navigate to the job-board analysis page
6. Change the duration in job application analysis to six months.

Expected result*:*

The user will be able to see the job application summary of the last six months.

Failed result:

If there is an error in the database or the value is not fetched properly, the user will see an error message

5. Conclusion

We had a great experience while working on this project. It was an excellent opportunity to enhance our technical, time and project management skills. We learnt that there are many more aspects of software development other than just code development. Planning, testing, and design all play a very crucial role in the success of an application. Using all the guidelines specified within the course structure, we were able to succeed in developing a working model of our product TrackMyJobs.

6. Recommendations

For job search feature, the current version only uses one job search API. To get the robust search result, the application should use multiple job search APIs to get as much job as possible for users to search. With having multiple job search APIs, the backend will need to standardize the data across different APIs. Moreover, if the application has more users, the application can start to be the platform for HR or employers to post the job on the web application.

7. References

[1] W. Group-21, "Project Proposal," Dalhousie University, 2020.

[2] A. Doyle, "10 Easy Ways to Organize Your Job Search," The Balance Careers, 3 March 2020. [Online]. Available: https://www.thebalancecareers.com/organize-your-job-search-2060710. [Accessed 22 June 2020].

[3] Huntr, "Huntr," Huntr, [Online]. Available: https://huntr.co/. [Accessed 22 June 2020].

[4] LinkedIn, "LinkedIn," [Online]. Available: https://ca.linkedin.com/. [Accessed 22 June 2020].

[5] Indeed, "Indeed," [Online]. Available: https://ca.indeed.com/. [Accessed 22 June 2020].

[6] Monster, "Monster," [Online]. Available: https://www.monster.ca/. [Accessed 22 June 2020].

[7] "draw.io," draw.io, [Online]. Available: https://app.diagrams.net/. [Accessed 15 May 2020].

[8] P. Bagaria, "CSCI 5709 ASSIGNMENT 3".

[9] W. G.-W. A. 21. [Online]. Available: https://track-my-jobs.herokuapp.com/home. [Accessed 7 August 2020].

[10] P. Bagaria, "CSCI 5709 ASSIGNMENT 1".

[11] Z. Thakkar, "A3_Thakkar_Zankrut".

[12] Z. Thakkar, "A1_Thakkar_Zankrut".

[13] A. Jinturkar, "A1_Jinturkar_Anudish".

[14] A. Jinturkar, "A3_Jinturkar_Anudish".

[15] R. Patel, "A3_Patel_Roshan_kirtikumar".

[16] T. Patel, "CSCI 5709 Assignment 04", Halifax: Dalhousie University, 2020.

[17] R. Patel, "A1_Patel_Roshan_kirtikumar".

[18] R. Patel, "Assignment-04.md".