

# **Recruitment Management System**

**S. M. Edirisinghe**  
**2020**



# **Recruitment Management System**

**A dissertation submitted for the Degree of Master of  
Information Technology**

**S. M. Edirisinghe  
University of Colombo School of Computing  
2020**



## Abstract

The system implemented in the final project is named as 'Collab Infinite'. Collab Infinite addresses the problems in the recruitment and career development of the Information Technology industry. The objective of this project is to create a fully automated system for recruitment management and career development of IT university students and fresh graduates to land their first job. Collab is a web application developed to address this problem.

Recruitment management and career development systems are very useful, especially when it is specifically created for the industry of Information Technology as there is a gap between demand and the supply of IT vacancies due to the mismatched skill set or lack of actual experience. Most of the time it is clearly proven that after finishing the degree, students are unable to find suitable jobs as they are not directed enough for the real industry. So, the directions and proper agents to connect to the system is much needed. Collab Infinite acts as a solution for the above problem by being the middle agent and the mediator.

The system built in this study consists of 03 independent components according to the particular user type which are candidates, recruiters and admin users. Recruiters have 02 separate components which are company users and typical recruiters which come under a particular company. So the system has 03 different layouts which is selected after the login process. Company users can add as much as recruiters to the system according to the availability of their access codes. Then recruiters can post vacancies, add candidates and schedule and conduct interviews for candidates. Candidates can add a number of skills to their profiles, check for available vacancies and communicate with recruiters, etc.

The first phase of the interviews are conducted through the system and the post first phase are announced via the system which is a special feature of Collab Infinite. This system has an attractive and user-friendly interface, so that it is easy for the users to use for the first time.

Access codes acts as the source of revenue for the Collab Infinite System. Access codes are sold by the system. But the candidates are free to use the full features and benefits of the system. The system will be hosted in the cloud as 03 separate components which are database, APIs and frontend (web layer).

We are able to demonstrate in this project that Collab Infinite significantly improves the efficiency and effectiveness of recruitment of candidates in the Software Industry. So that it will provide a good service by contributing a tremendous number of features to the recruitment of the Information Technology industry.

## Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: S. M. Edirisinghe

Registration Number: 2017/MIT/017

Index Number: 17550171

---

Signature:

Date:

This is to certify that this thesis is based on the work of

Mr. S. M. Edirisinghe

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Prof. G.K.A.Dias

---

Signature:

Date:

## **Acknowledgements**

First, I would like to thank University of Colombo School of Computing for the opportunity given me to study for the Master of Information Technology program.

Foremost I wish to thank Prof. G.K.A.Dias, my research project supervisor, for making the opportunity to explore my knowledge in the latest technology, and for encouraging me technically and spiritually during my studies. His guidance helped me in all the time of research.

In addition, I wish to thank all other staff members of the University of Colombo, especially the Computing Department, for their knowledge which directs us to achieve our goals in life.

I want to express my most profound gratitude to my parents and family for their encouragement and companionship.

Finally, I take this opportunity to express my gratitude to everyone who supported me throughout the course of this MIT project.

# Table of Contents

Chapter 1. Introduction	1
1.1 Background	1
1.1.1 Information systems in human resources management	1
1.1.2 What is a Recruitment System?	1
1.1.3 The Importance of information Technology in Recruitment Management	2
1.1.4 The advantages of Online Recruitment Systems	2
1.2 The Problem	3
1.2.1 Available traditional online recruitment system features	3
1.2.2 The problem in traditional online recruitment systems	3
1.3 Motivation	4
1.4 Scope and Limitation	4
Chapter 2. Background	6
2.1 Brief History on online recruitments systems	6
2.2 Evolution of the online recruitment process	7
2.3 Existing online recruitment systems	8
2.3.1 Analysis of the biggest Existing Online recruitment system	9
2.4 Summary	10
Chapter 3. Methodology	11
3.1 Design Overview	11
3.1.1 System Architecture	12
3.1.2 Technologies Used	13
3.1.3 Entity Relationships Diagram	14
3.1.4 API Documentation	15
3.2 Methodology and Implementation considerations	19
3.3 Work breakdown and project plan	20
3.4 Functional Requirements	21

3.4.1.1 User Registration	21
3.4.1.2 User Login	23
3.4.1.3 Candidate Profile Management	24
3.4.1.4 Vacancy Management	31
3.4.1.5 Candidate Applied Vacancies	36
Chapter 4. Evaluation	38
4.1 System feature evaluation	38
4.1.1 User Registration	38
4.1.2 User Login	41
4.1.3 Candidate Profile Management	43
4.1.4 Vacancy Management	46
4.2 Failures to achieve given objectives	49
4.3 Lessons learned during the project development	49
Chapter 5. Conclusion	51
5.1 Summary of Work	51
5.2 Improvements and future works	51
References	53
Appendix A. System Documentation	55
Appendix B. Test Results	56

## List of Figures

Figure 2.1: Evolution of online recruitment system process	7
Figure 2.2: Data flow diagram – LinkedIn	9
Figure 2.3: Activity Diagram: LinkedIn Login	9
Figure 2.4: Basic Functionalities of LinkedIn	10
Figure 3.1: System architecture overview	11
Figure 3.2: System architecture	12
Figure 3.3: Entity Relationship Diagram	14
Figure 3.4: Swagger API documentation overview	15
Figure 3.5: Company Controller API documentation overview	16
Figure 3.6: Get companies GET REST service request	16
Figure 3.7: Get companies GET REST service response	17
Figure 3.8: Create vacancy POST Rest service	17
Figure 3.9: Candidate controller API overview	18
Figure 3.10: Get all candidates GET REST service	18
Figure 3.11: Get all candidates GET REST service	19
Figure 3.12: Work breakdown	20
Figure 3.13: Project plan	20
Figure 3.14: User registration page	22
Figure 3.15: Company registration page	22
Figure 3.16: Login page	24
Figure 3.17: Basic information tab	27
Figure 3.18: Edit basic info modal window	27
Figure 3.19: Technical skills tab	28
Figure 3.20: New technical skill modal window	28



Figure 3.21: Educational qualifications tab	29
Figure 3.22: New educational qualification modal window	29
Figure 3.23: Work history tab	30
Figure 3.25: New work history modal window	30
Figure 3.26: Recruiter vacancies page	33
Figure 3.27: New vacancy creation page	34
Figure 3.28: New vacancy creation add roles and responsibilities	34
Figure 3.29: Candidate available vacancies	35
Figure 3.30: Candidate detailed vacancy page	35
Figure 3.31: Candidate applied vacancies	37
Figure 3.32: Candidate applied vacancy detail	37
Figure 4.1: Test case 02	39
Figure 4.2: Test case 03	40
Figure 4.3: Test case 04	40
Figure 4.4: Test case 07	41
Figure 4.5: Test case 09	42
Figure 4.6: Test case 10	42
Figure 4.7: Test case 11	43
Figure 4.8: Test case 45	48
Figure 4.9: Test case 48	48
Figure 4.10: Test case 52	48
Figure 4.11: Test case 56	49

## List of Tables

Table 3.1: User stories for user registration	21
Table 3.2: User stories for user login	23
Table 3.3: User stories for candidate profile management	24
Table 3.4: user stories for vacancy management	33
Table 3.5: Candidate applied vacancies user stories	36
Table 4.1: Test cases for user registration	39
Table 4.2: Test cases for user login	41
Table 4.3: Test cases for candidate profile management	43
Table 4.4: Test cases for vacancy management	46

# **Chapter 1. Introduction**

The problem which this project tries to address, motivation to come up with a solution, objectives and the scope of the implemented solution are described by this chapter. This solution can be used for any client who is needed to collaborate with their team members in the Software Industry.

## **1.1 Background**

Throughout the past few years, the human resource management (HRM) process has been changed significantly. The involvement has been changed from the administrative partner to strategic manager of an organization, mostly along with the strong support of IT evolution in this area. The extended use of information systems has a strong effect on how HRM is managed currently. It introduces a huge transformation of human resources procedures within organizations, on how they gather, use, store, and share information. Multiple HRM processes have become more effective and the result of this progress allowed a greater contribution of HR in the business strategy. This new aspect in business strategy adds significant value to strategic human resources function and to its professionals.

### **1.1.1 Information systems in human resources management**

Human resources information system shapes integration between human resource management and IT. HRIS has specifically been designed to provide managers with information to make HR related decisions and it lets you have a track of all your human resources and information about them. It is normally done in a database, or commonly in a set of interconnected databases.

### **1.1.2 What is a Recruitment System?**

Using a recruitment system, employers can effectively and efficiently manage their candidate flow and recruitment processes. It enables you to gather, edit, create and store a database of candidates, vacancy management, application processing, accepting rejecting applications and much more.

### **1.1.3 The Importance of information Technology in Recruitment Management**

This allows in optimizing the hiring process and increasing the recruitments of most fitted candidates. With the help of this, Recruitment agencies can ensure that candidate applications are processed accurately and efficiently during the recruiting period. It allows you to track and search for the most qualified candidates effectively.

It makes the selection process much easier for the candidate. A good recruitment system serves both the employer and the candidate. A well-designed recruitment system mainly focuses on a positive candidate experience. It allows the application process to be simpler to understand and complete. Creating a quality candidate experience will favorably affect your company and in return, enable you to stand out among future employment.

### **1.1.4 The advantages of Online Recruitment Systems**

Online recruitment services provide a large variety of tools such as pre-employment screening, assessing personalities and testing to screen candidates to allow you to select suitable candidates who fit your organization's values with automating human interaction. Many recruiting software offer multiple services similar to this, where you can personalize your organization's specific requirement for every job.

This reaches a vast community of potential employees than other tools. By allowing you to reach job seekers anywhere, you can find candidates with unique skills and compatibility. Many online recruiters are targeted for specific industries. Also, many organizations help employers to post job vacancies on their websites. This ensures employers can find people with very specific skills and it can also attract candidates who are not looking for new job opportunities.

In most companies, hiring costs are one of the highest expenses. Recruiting involves both actual costs of hiring an employee, as well as the costs of retaining and training. Each employee needs to receive training and orientation and a time period to learn the pattern of his job. This can be considered as an investment for many organizations. If an employee leaves before the organization pays off those costs, the recruitment process has to begin again, which is considered as a financial loss. Properly planning online recruitment and selection procedures can significantly reduce costs and minimize employee turnover, which is very favorably effects on the revenue.

## **1.2 The Problem**

There is always a gap between demand and supply for IT professionals such as due to lack of skilled IT professionals and skilled people unable to find appropriate jobs. It is a big issue in Sri Lanka that there are very less portals for both job seekers and recruiters.

### **1.2.1 Available traditional online recruitment system features**

#### Video Interview Tools

- To standardize the interview process for both employer and candidate free video interview tools like Skype [1] can be used.

#### Applicant Tracking Systems

- Applicant Tracking Systems that gives the ability to evaluate candidates more effectively, like Greenhouse [2], helps to speed up the candidate selections.

#### Learning Tools

- Companies can boost their employer brand by providing learning tools like Lessonly [3] and with the online tools good employees can hit the ground running at your organization.

### **1.2.2 The problem in traditional online recruitment systems**

Many applicants attest that searching jobs online, doesn't have a personal touch. Job seekers need the ability to communicate with someone to clear out the questions regarding the company and the job they are interested in. Job seekers commonly complain that employers are so unreachable and have very little or no human interaction. Expired job postings are the next significant disadvantage of online hiring that job applicants experience. Many applicants say that they have applied for job posting which currently appears in vacancies but find out weeks later that the position had already been filled a long time ago. Company websites that break down when trying to submit their recruitment materials is one of the major complaints that applicants do. Others are saying that some website interfaces are so complex and find it difficult to route through the site.

Employers trust the applicant tracking system to store resumes that include relevant indexes. Some candidates purposely include their resume with keywords, so their resume is getting selected by the system. So, recruiters tend to call applicants for interviews that do not meet the job requirements.

### **1.3 Motivation**

The aim of this project is to create an online portal for both job seekers and recruiters, to offer jobs and find jobs, communicate with each other and conduct interviews through the portal. Also, people who have the same skill set can network together. This system is focused on the IT industry to fill the gap between job seeking and recruitment; most of these tasks are automated through the system. So, recruiters can post their jobs in the system and candidates can seek jobs via the system. Candidates and recruiters both will be ranked according to their performance, reliability etc. via the system.

Currently in Sri Lanka it is a good opportunity to build a system to prepare candidates for interviews. This system will provide preparation tests for interviews in a given timeframe. So, candidates can practice for interviews through the system and will provide all the relevant study materials before the interview. Also, it is a huge problem that there is no online system for freelance work in Sri Lanka. So, via this system employers can offer freelance jobs. Freelancers also are ranked according to their performance through the system.

### **1.4 Scope and Limitation**

The scope of this project is to build a working solution of recruitment management and career guidance system in respective iterations. System will be web based along with APIs for integration with other systems.

The system consists of mainly 03 types of users which are recruiters, candidates and admin users. Also, candidates can act like freelancers. The system facilitates to register users as a candidate or a recruiter. If a user registers as a recruiter, there will be a special access code which the user has to obtain to register under a company. Recruiters can post job openings, view candidate profiles, schedule interviews, etc. This system will automate the hiring process and therefore recruiters must do only minimal work for that. The system will generate time wise reports to recruiters regarding candidate information, job postings and so on. Also, the system will generate special access codes for recruiters when they want to register under a company. The system will automate the candidate selection process and therefore select the suitable candidates with minimal effort.

When a user is registered as a candidate, the system will allow you to add skills, education qualifications, professional qualifications and employment history to your profile. The system will allow candidates to apply for jobs. Also, the system will notify candidates when a relevant job vacancy is posted. The system will group candidates with similar sets of skills and network them via the system. System will provide study materials to candidates for preparing interviews. The other important benefit of this system for candidates is the system will generate CV and

portfolio according to the candidate's skills. Also, if a candidate is registered as a freelancer, there are freelance jobs available in the system.

The system provides a messaging service to communicate between recruiters and candidates. Therefore, every recruiter and candidate has his own message box. The system will conduct online interviews and practice tests for candidates. Also, the system will rank candidates and recruiters according to their performance.

There are also admin users in the system. The system will allow admin users to configure the system. Also, admin users have the privilege to add companies to the system and generate access tokens for recruiters under a particular company and admin users can manage other user roles. One of the most important features of the system is that the system will generate various reports for admin users, recruiters and candidates. It will help to track candidate and recruiters' performance, number of job postings and so on.

The high level objectives of the system are stated below.

1. Provides the environment for recruiters to offer their job postings.
2. Candidates will be able to find jobs via the system.
3. Freelancers will be able to find freelance jobs through the system.
4. Rank and evaluate candidates, recruiters and freelancers according to their performance.
5. Automate hiring process and conduct online interviews.
6. Automate candidate selection process through their rank and overall score.
7. Communicate between recruiters and candidates via messaging service provided by the system.
8. Provide reports and statistics to recruiters timewise.

## **Chapter 2. Background**

Online recruitment systems are one of the most active research areas in software engineering and different approaches have been proposed by the research community. This chapter describes major approaches in ATS.

### **2.1 Brief History on online recruitments systems**

Today, it's hard for many recruiters and candidates to realise a recruitment area that does not have the Internet as its core. Companies as Web recruit [4] make huge use of the internet in their functions, and also, it started functioning recently. At the end of the 20th century Monster.com [5] and Jobserve.com [6] had become functioning for candidates and recruiters in the international market. By the introduction of the e-based systems it allows the candidates to go for the most suitable jobs and communicate real-time via chat.

There were lots of benefits for those who are searching for recruiters, as they can easily receive job applications, provide a response, payroll update with new employee details, without wasting time and money. By the 2000s, it was proven that online recruitment systems were becoming more advanced. Companies that promote social media in their recruitment have realized that they are with an advantage over the competition, mostly engaging with job seekers especially candidates who don't have an interest in joining the company. Even though the success of social media recruitment in the near past can be dependent on many reasons, including its capability to provide sustainability and its advantage in maintaining of talent pools.

Webrecruit [4] is the best example in the recent past.



## 2.2 Evolution of the online recruitment process

Recruitment is a good target for innovation. Millions of dollars have been spent every year, with unsuccessful business models and widely felt failures. But with the hundreds of startups taking a huge profit, more modern innovations can be mapped on a single event.

Today, like many years ago, we still sort applicants, run them through a series of interviews. Also, selection needs many people to manage that process.

The framework below shows the most important change across three different periods of innovation. The arrows show the “direction” of innovation, from which novel methods were introduced. For example, job aggregators emerged in 2005 to innovate online job boards.

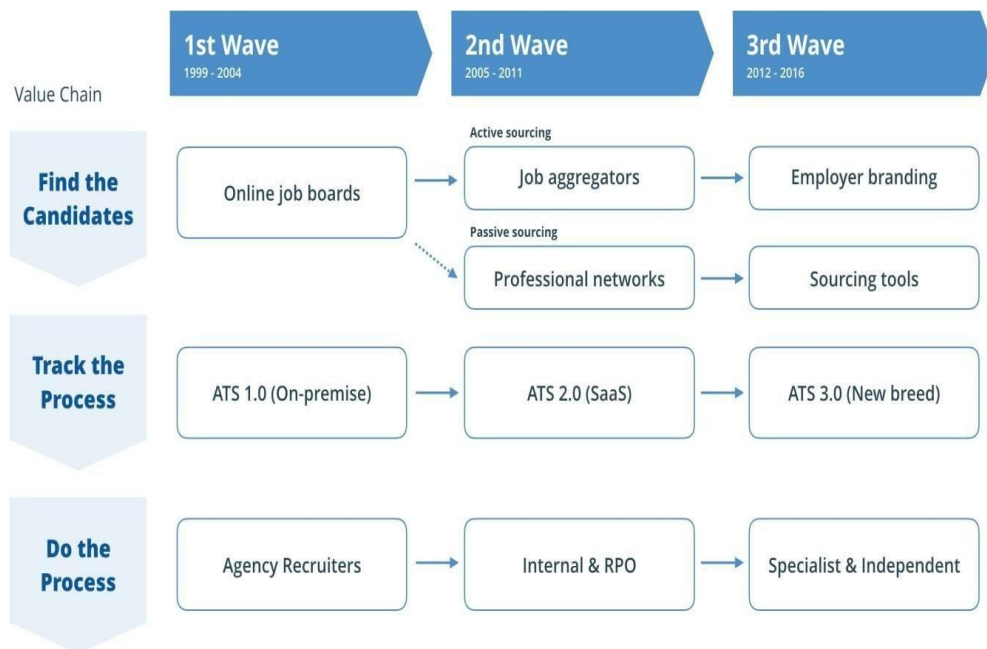


Figure 2.1: Evolution of online recruitment system process - SmartRecruiters [21]

## 2.3 Existing online recruitment systems

**Job Aggregators:** Job boards and job recruiters are key to the recruiting process because direct applicants make up half of the total hires which are in place. E.g.: Indeed [12], CareerBuilder [13], Google [14] for jobs.

**Testing & Evaluation:** As per Aberdeen, half of companies use pre-hire assessments to verify the knowledge, skills, and abilities as well as the other qualifications (KSAOs) of their candidates Eg: HackerRank [15], Pymetrics [16], Self-Management Group [17]

**Automation and Artificial Intelligence:** Majority of company owners believe machine learning can direct significant value in hiring. The emerging demands of improved hiring volume and reduce the wastage, Automation will become the main player of the recruitment in 2020.Eg: Bullhorn [18], GreenHouse [2], SAP [19] Success Factors, ICIMS [20], SmartRecruiters [21], Taleo [22]

**Recruitment CRMs:** By the rise of innovative hiring of recruitment chains, candidate relationship management (CRM) systems have become the brain for engaging, disrupting, and seasoning candidates. Eg : Smashfly [23] , Yello [24], Jibe [24]

**Video Conference Interviews:** By the results of recent research majority of HR managers having less nightmares of face to face interviews by interviewing them remotely via online systems. Eg : Convey IQ [25], SparkHire [26], Wepow [27]

### 2.3.1 Analysis of the biggest Existing Online recruitment system

Following Diagram shows how data flow through LinkedIn [7] system and how does it helps LinkedIn [7] to make decisions

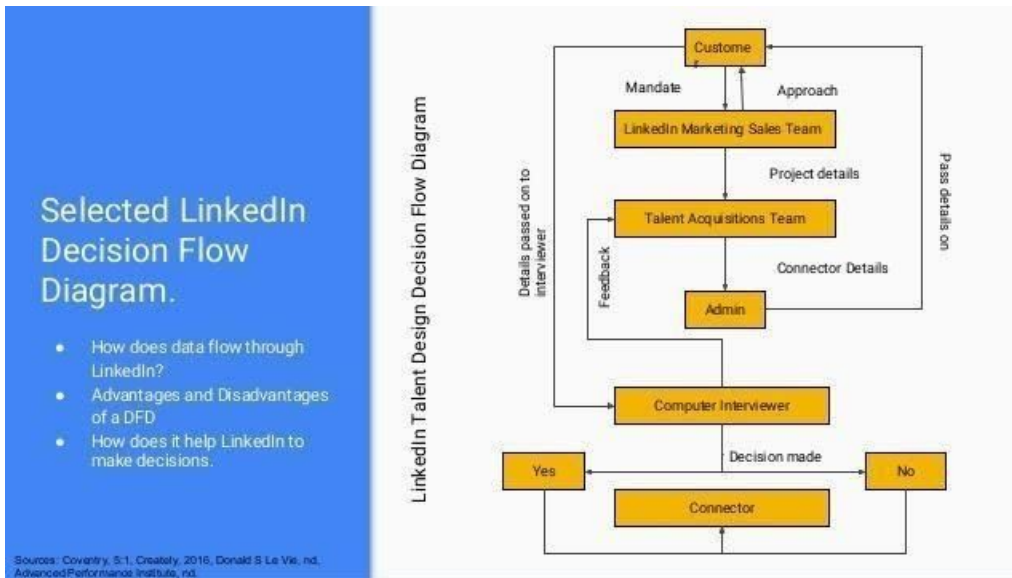


Figure 2.2: Data flow diagram – LinkedIn [7]

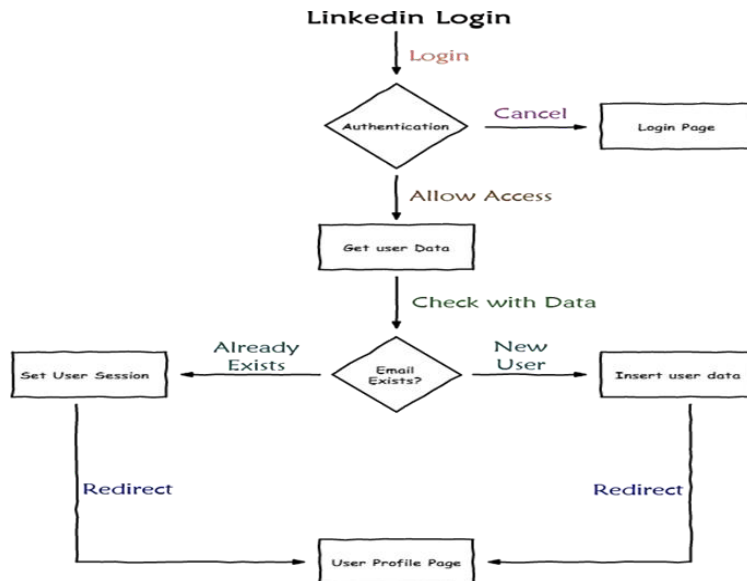


Figure 2.3: Activity diagram - LinkedIn user flow [7]

## BASIC FUNCTIONALITIES

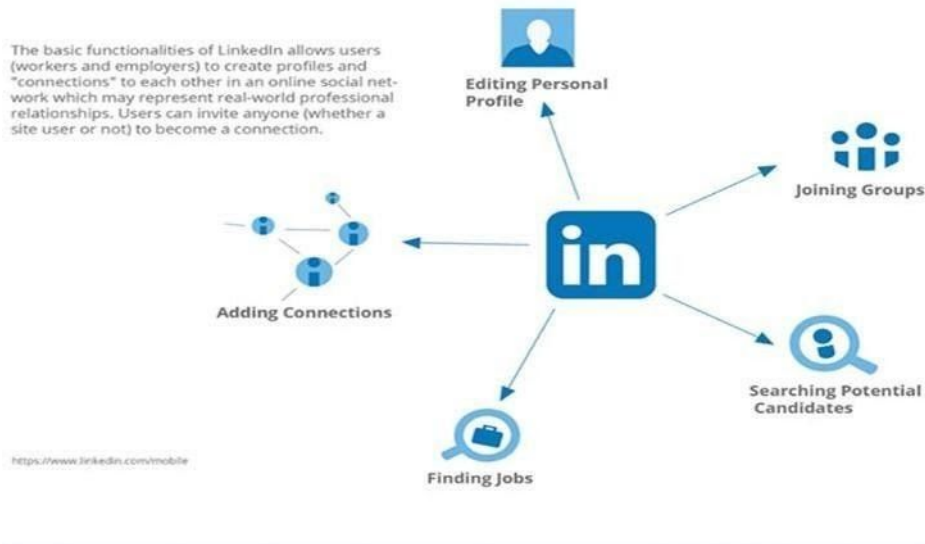


Figure 2.4: Basic Functionalities of LinkedIn [7]

## 2.4 Summary

Automated recruiting tools have become the brainiac to manage the HR functions and labor of a business. As the technology innovates and HR functions are distorted through all areas, there is a gap to develop automated software that can handle the process of recruitment and hiring.

The varying nature of business and the way that companies are recruiting has resulted in recruitment software acting as the key player. With the ideal recruitment software can change the way you hire, It saves your monitor expenses and timely effort during the recruiting process, and most importantly, ensures that you are hiring the ideal candidate.

It may even be reasonable to say that recruitment software has become a major role of the HR function significantly along the past few years when many continents have been fighting through a universal recession. Retaining, attracting and recruiting the suitable people have become the index to productivity of the effectiveness and efficiency of growth for many businesses, which has become a major industry level satisfaction.

Identification and reviewing of the similar systems are stated in section 2.3.

## Chapter 3. Methodology

### 3.1 Design Overview

Client-server architecture is one of the most popular system architecture methods. In simple terms, many clients can connect to the system concurrently and request and receive services from the server. So that client computer acts as an interface to send and receive requests and responses respectively. Server serves the responses as per the requests.

We live in a world of information technology and it has become the biggest player in every industry, mainly in business applications. It is essential to a business to be sustainable in its market position by restructuring existing organizational format and business to achieve their goals. The client-server model shows the logical perspective of distributed systems. In short, client-server models are built to adapt to changing technological and business requirements, by decentralizing the services provided.

In the Collab system, the architecture model used is client-server model. Following diagram represents the overview of the system architecture.

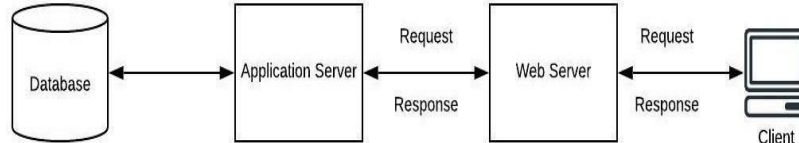


Figure 3.1: System architecture overview

Mainly the system consists of 04 components at the top level, which are Database, Application Server and the Web Server. Client machines connect to the system through the web server. Database handles the persistence storage of the system. Application server manages the connection with the database, handles the business logic, and serves http requests and http responses to the web server. In short, Application Server exposes REST services to the Web Server. Web server acts as the interface between the system and the client machines. Web server serves html pages to the client machines. In this architecture, it serves Ajax content.

Alternative design strategy to implement for this system is to use MVC pattern. Even though the MVC design pattern is somewhat old, it is widely used in the software industry. It clearly addresses the separation of layers and easily maintainable.

### 3.1.1 System Architecture

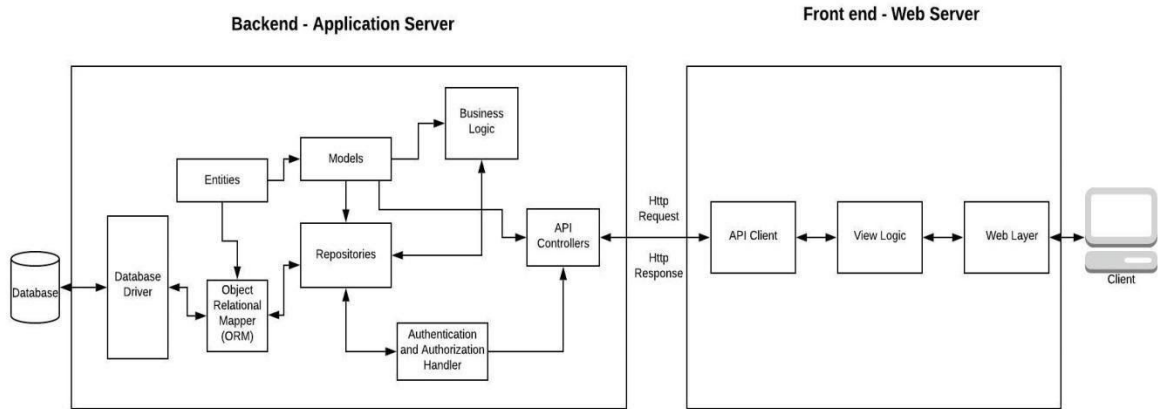


Figure 3.2: System Architecture

The above figure represents the Detailed System Architecture of the system. Application server and Web server are divided further to smaller components. As mentioned in the previous chapter, Database manages the persistence state of the system.

Backend (Service Layer) of the system is modularized into 08 components as shown in figure 3.2. Database driver creates and manages the connections with the database.

Database tables are mapped to Entities via Object Relational Mapper. Entities are further mapped to Model Classes for the use of other components in the Service Layer. Repositories handle the database operations in the system. It returns Model classes and supplies them to the Business Logic Layer. Business Logic Layer caters all the business logic, calculations, filtrations and every other logic. It connects to the Repository Layer and the Controller Layer in both ways. API Controller Layer manages sending HTTP responses and receiving HTTP requests. Basically API Controller Layer consists of REST services. Authentication and Authorization Handler is injected to API Controller Layer for the security purposes. It caters authorization and authentication of the system. Also, Cross Origin Resource Sharing (CORS) filter is applied in the API Controller Layer, so that the API Client of the front end can access REST Services of the Backend Layer.

Frontend consists of 03 main components. API client uses Ajax to send and receive HTTP requests and responses. View Logic processes the result of these responses and renders them to the Web Layer. Client machines can see the Web Layer through a web browser. Web layer consists of a set of web pages which acts as the interface between the system and the user.

### 3.1.2 Technologies Used

MySQL database is used for the persistence storage of the system. It is highly available, free and easy to use, configure and connect.

The Service Layer of the system is written in Java, as Java is portable and can be used in any platform whether it is Windows, Linux or Mac OS. Also Java has a lot of community support and a huge amount of available libraries. Spring Boot is used as the application framework. Spring Boot is the most popular modern Java framework to cater REST web services. It uses convention over configuration. So it takes a few minutes to up and run a Spring Boot application. JPA technology is used for persistence layer and Hibernate is used as the implementation of JPA layer. The build and packaging tool used in the Spring Boot application is Maven, as it is very easy to use with a pom.xml file and with simple Maven commands. As the application server, Embedded Tomcat Server is used. Tomcat server is the most widely used Java application server.

ReactJS is used for the web application of the system. It runs on a built-in Node server. So it is very easy to up and run the server. HTML5, CSS, Javascript and Bootstrap are used for implementing the web application.

Git is used as the version control system of the whole project. There are 02 GIT projects, one is for the service layer and the other one is for the web application.

As an alternative technology, .NET core web API technology can be used with Entity Framework as ORM layer and MS SQL Server as the database. For the frontend Angular or VueJs can be used.

### 3.1.3 Entity Relationships Diagram

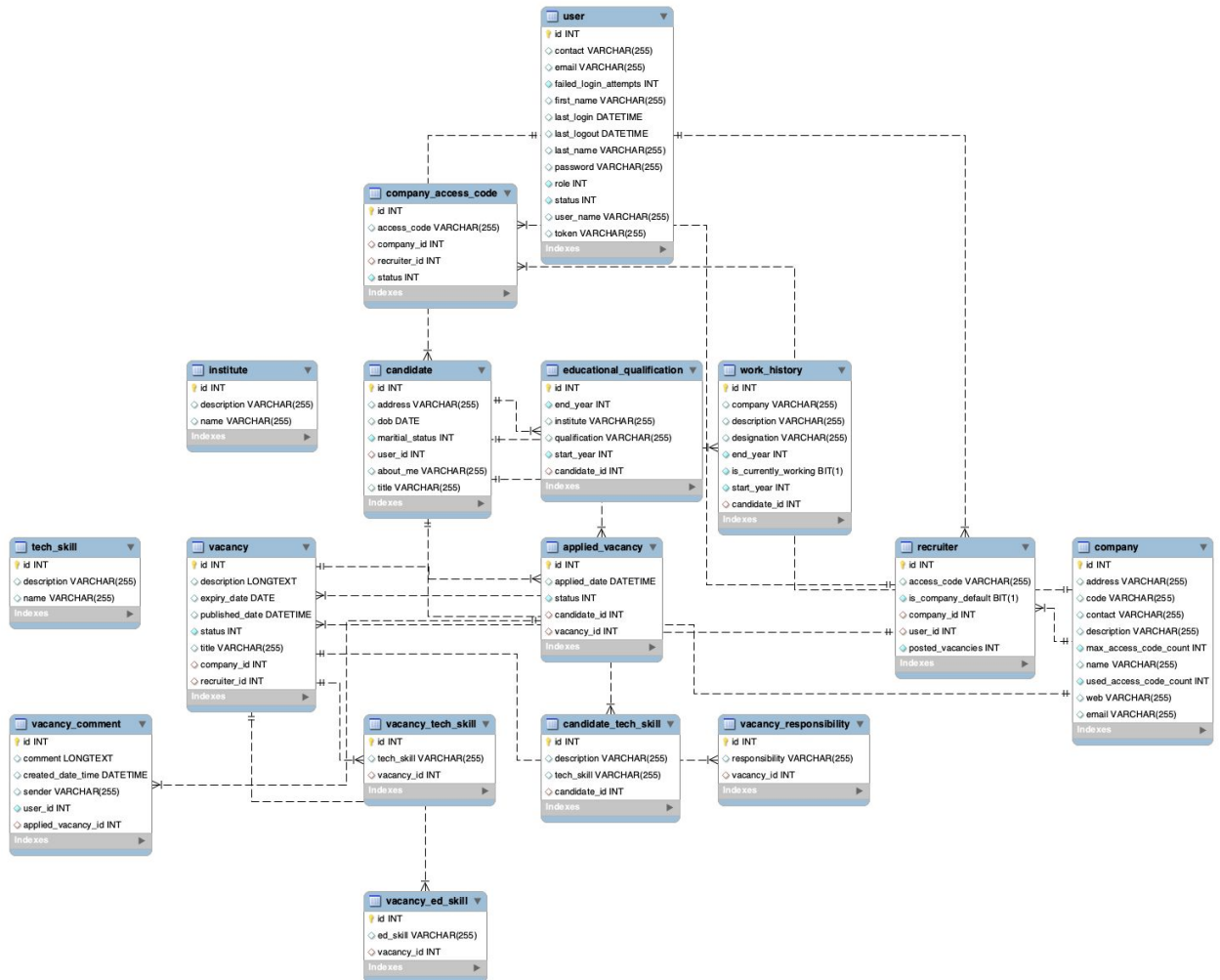


Figure 3.3: Entity Relationship Diagram

The above diagram represents the entities and their relationships in the system. The system consists of 03 types of users which are candidates, recruiters and admin users. Admin users are allowed to configure the system, manage users and manage other things. Recruiter entities can post job offerings on behalf of the Company entity which is represented by the Vacancy entity. A company entity can have one or more recruiters. Institute entity is a master data entity which keeps identifying the Candidate’s educational qualifications. Candidates can have one or more Educational Qualifications and Tech Skills. Candidate Applied Vacancy entity tracks the applied vacancies of candidates.

### 3.1.4 API Documentation



API documentation is implemented in the service layer of the Collab Infinite system. The documentation is configured in the 'application.properties' file and code level implementations are found in the utility package in the application.

Swagger is used as the API documentation technology in this project as it is well compatible with Java and SpringBoot. Some of the documentation are stated below.

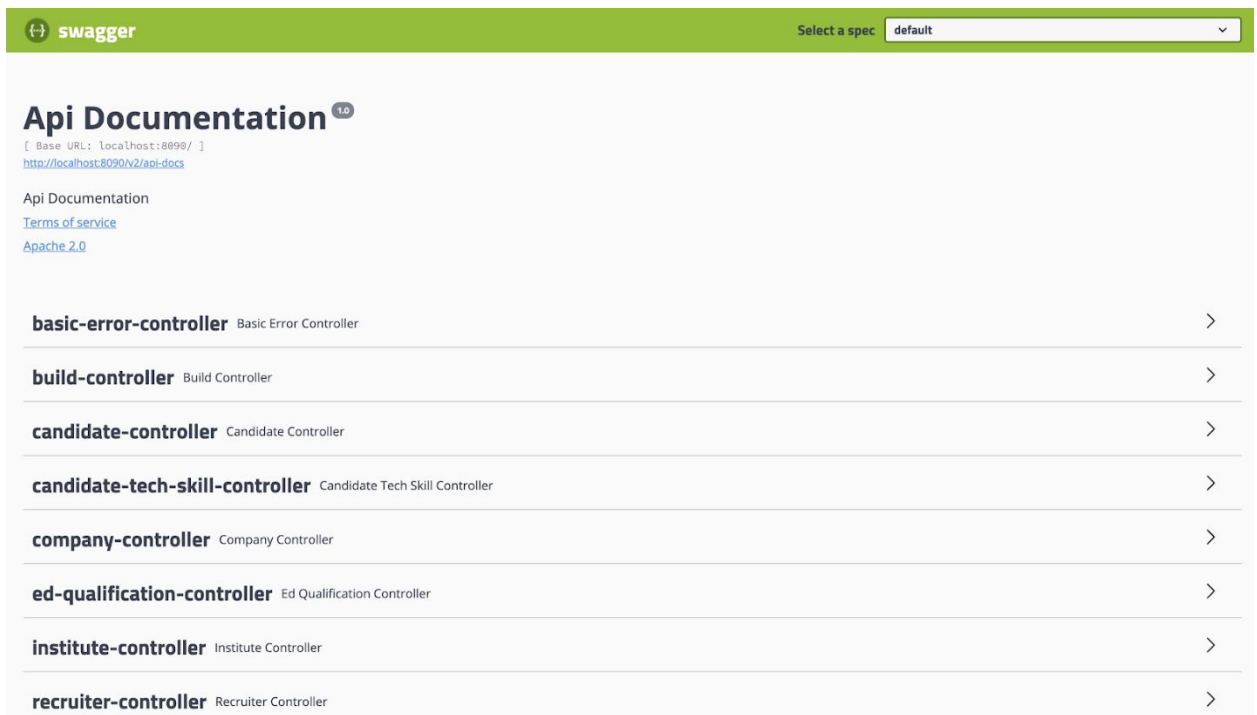


Figure 3.4: Swagger API documentation overview

company-controller Company Controller	
GET	/api/company getAll
POST	/api/company create company with default user and recruiter
PUT	/api/company update
GET	/api/company/{id} getByid
POST	/api/company/addVacancyComment addVacancyComment
POST	/api/company/applyvacancy candidateApplyVacancy
GET	/api/company/candidate-applied-vacancy/{candidateId} getCandidateAppliedVacancies
GET	/api/company/candidateappliedvacancy/detail/{appliedVacancyId} getCandidateAppliedVacancyDetail
PUT	/api/company/consumeaccesscode consumeAccessCode
POST	/api/company/createvacancy createVacancy
PUT	/api/company/generateaccesscode generateAccessCode
GET	/api/company/getaccesscodes/{id} getAccessCodes
GET	/api/company/getrecruitervacancies/{recruiterId} getRecruiterVacancies

Figure 3.5: Company Controller API documentation overview

company-controller Company Controller

GET /api/company getAll

Parameters Cancel

No parameters

Execute Clear

Responses Response content type \*/\*

Curl

```
curl -X GET "http://localhost:8090/api/company" -H "accept: */*"
```

Request URL

```
http://localhost:8090/api/company
```

Server response

Figure 3.6: Get companies GET REST service request

Request URL  
<http://localhost:8090/api/company>

Server response

Code	Details
200	<p>Response body</p> <pre> {   "web": "www.test.com",   "contact": "012345",   "email": null }, {   "id": 2,   "name": "Mutants International",   "description": "Mutants International is a startup company catering mainly ERP solutions for both local and international market",   "address": "Bernards Business Park, Kobbuwa",   "code": "mutants",   "maxAccessCodeCount": 10,   "usedAccessCodeCount": 7,   "web": "www.mutants.lk",   "contact": "0112886756",   "email": "info@mutants.lk" }, {   "id": 3,   "name": "Inova IT Systems",   "description": "Inova is a global IT &amp; Consulting company providing solution-oriented approach and custom software development to its clients across multiple business domains.",   "address": "5 Sulaiman Terrace, Colombo 00500",   "code": "inova",   "maxAccessCodeCount": 50,   "usedAccessCodeCount": 0,   "web": "https://www.inovaitsys.com/",   "contact": "+94 112 593 823",   "email": null } </pre> <p>Response headers</p> <pre> connection: keep-alive content-type: application/json date: Sat, 20 Jun 2020 20:29:24 GMT keep-alive: timeout=60 transfer-encoding: chunked vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers </pre> <p>Responses</p>

Figure 3.7: Get companies GET REST service response

POST </api/company/createvacancy> createVacancy

Parameters Cancel

Name	Description
<b>request</b> <span style="color: red;">*</span> <small>required</small> (body)	request Example Value   Model <pre> {   "companyId": 0,   "description": "string",   "edSkills": [     "string"   ],   "expiryDate": "yyyy-MM-dd",   "id": 0,   "recruiterId": 0,   "responsibilities": [     "string"   ],   "techSkills": [     "string"   ],   "title": "string" } </pre> <p>Cancel</p> <p>Parameter content type  <input type="text" value="application/json"/></p>

Execute

Figure 3.8: Create vacancy POST Rest service

candidate-controller Candidate Controller	
GET	/api/candidate getAll
POST	/api/candidate insert
PUT	/api/candidate update
GET	/api/candidate/{id} getByid
GET	/api/candidate/findByUser/{id} getByUser

Figure 3.9: Candidate controller API overview

Request URL  
<http://localhost:8090/api/candidate>

Server response

Code	Details
200	<p>Response body</p> <pre>{   "data": [     {       "id": 1,       "address": "22/2, 2nd Lane, Thalawathugoda Road, Kotte",       "dob": "1990-04-14",       "maritalStatus": 0,       "user": {         "id": 2,         "firstMame": "sasanga",         "lastName": "candidate",         "userName": "sasanga_candidate",         "password": "password",         "lastLogin": "2020-06-09T22:43:06",         "lastLogout": null,         "failedLoginAttempts": 0,         "status": 1,         "role": 2,         "email": "candidate@testmail.com",         "contact": "01234556789",         "token": "841f8999-c854-4b83-98e1-bec0eb9df0e4"       },       "aboutMe": "Highly enthusiastic and committed candidate. I like to play guitar in the free time. I am a self made coder.",       "title": "Senior Software Engineer at Pearson Lanka"     },     {       "id": 2,       "address": null, </pre> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Sat, 20 Jun 2020 20:51:55 GMT keep-alive: timeout=60 transfer-encoding: chunked vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers</pre>

Responses

Figure 3.10: Get all candidates GET REST service

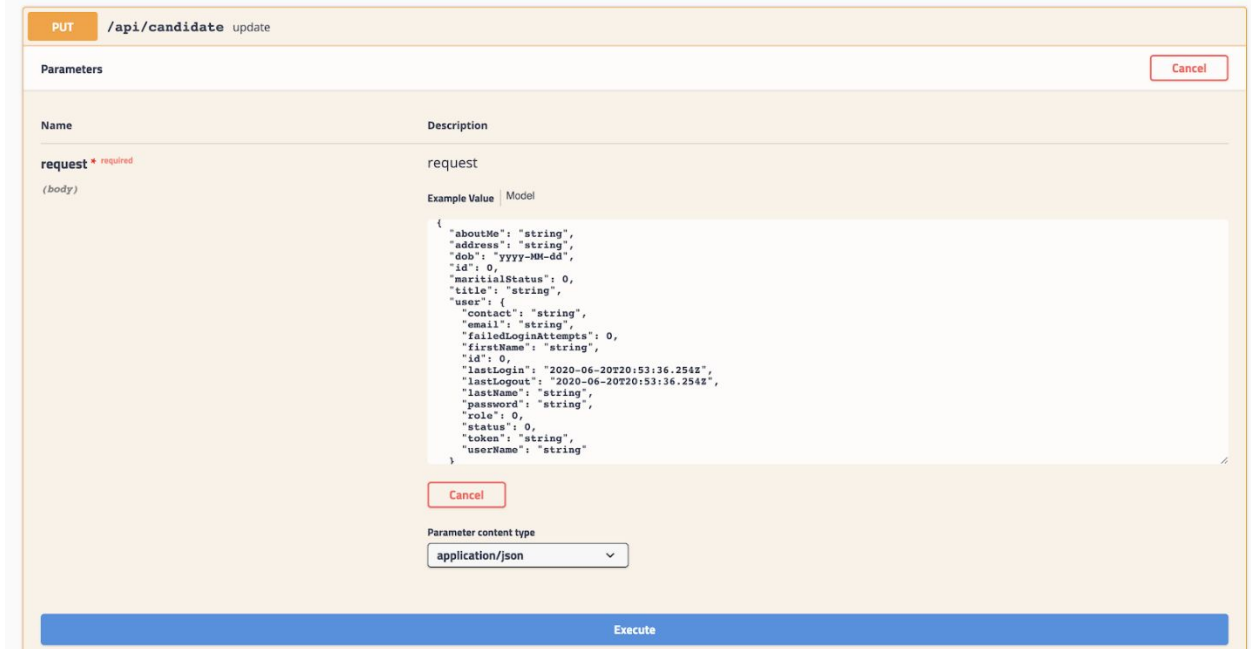


Figure 3.11: Update candidate POST REST service

## 3.2 Methodology and Implementation considerations

**Software development life cycle method:** Incremental and Iterative development methods will be used as software life cycle methods. Development will be carried in a set of sprints according to agile methodology. System will be divided into modules which contain tasks and subtasks; these will be developed, tested and delivered in respective iterations. Both front-end tasks and back-end tasks will be developed simultaneously in each iteration. To follow this methodology there will be a product backlog and the tasks for each iterations will be selected according to the priority.

**Deployment:** system will be deployed as a web-based system.

**Architecture:** the system architecture is client-server model, along with public APIs for integrating with other systems. Client will be a web-based system (dynamic web system). Server (back-end) will be APIs both public and private. Private APIs can be accessed only by the front-end provided by the system. Backend consists of data access layer, business logic layer, utility layer and service layer. Front End consists of MVC pattern. Database will be centralized for access by the backend. System implements token-based authorization for security.

### 3.3 Work breakdown and project plan

<input type="radio"/>	Preparing project proposal	■ T	01/07/2019	09/07/2019	7 days
<input type="radio"/>	Requirement Gathering	■ T	08/07/2019	08/08/2019	24 days
<input type="radio"/>	Database Development	■ T	09/08/2019	01/01/2020	104 days
<input type="radio"/>	Requirement Gathering Phase II	■ T	01/10/2019	30/11/2019	44 days
<input type="radio"/>	Back-end Development	■ T	03/09/2019	01/06/2020	195 days
<input type="radio"/>	Front End Development	■ T	20/09/2019	01/06/2020	182 days
<input type="radio"/>	Testing	■ T	15/09/2019	30/06/2020	207 days
<input type="radio"/>	System Deployment	■ T	24/06/2020	30/06/2020	5 days

Figure 3.12 Work breakdown

## Recruitment Management System

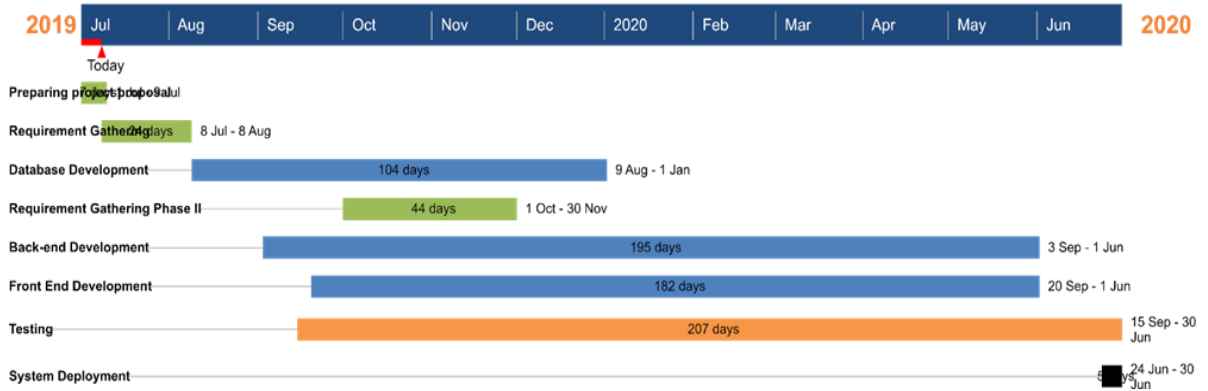


Figure 3.13: Project plan

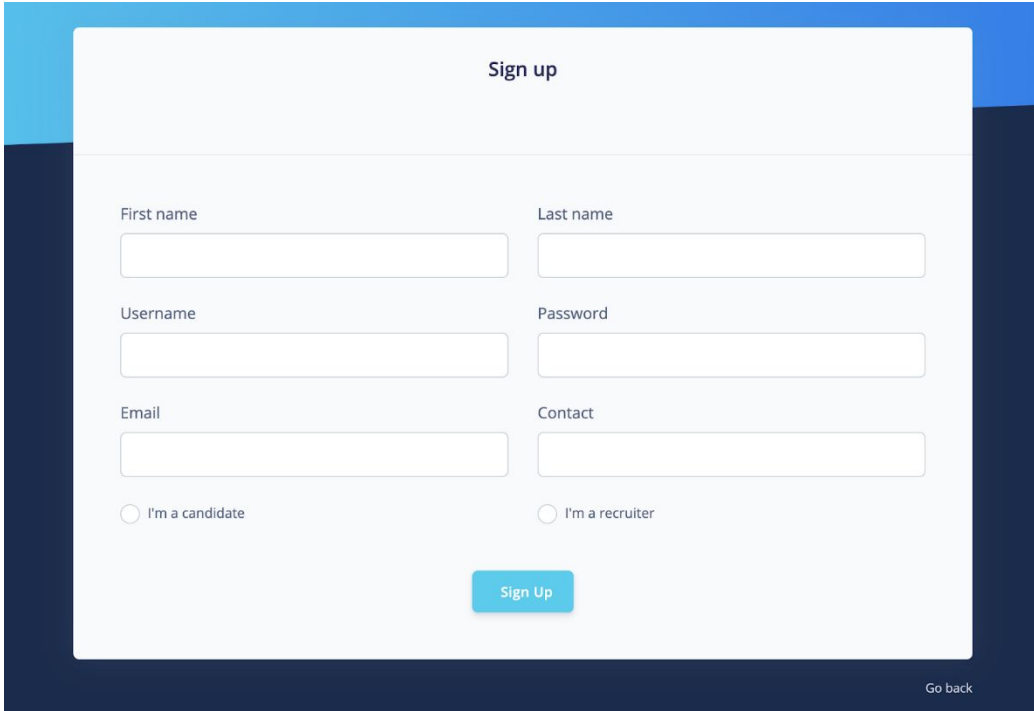
## 3.4 Functional Requirements

### 3.4.1 User Stories

#### 3.4.1.1 User Registration

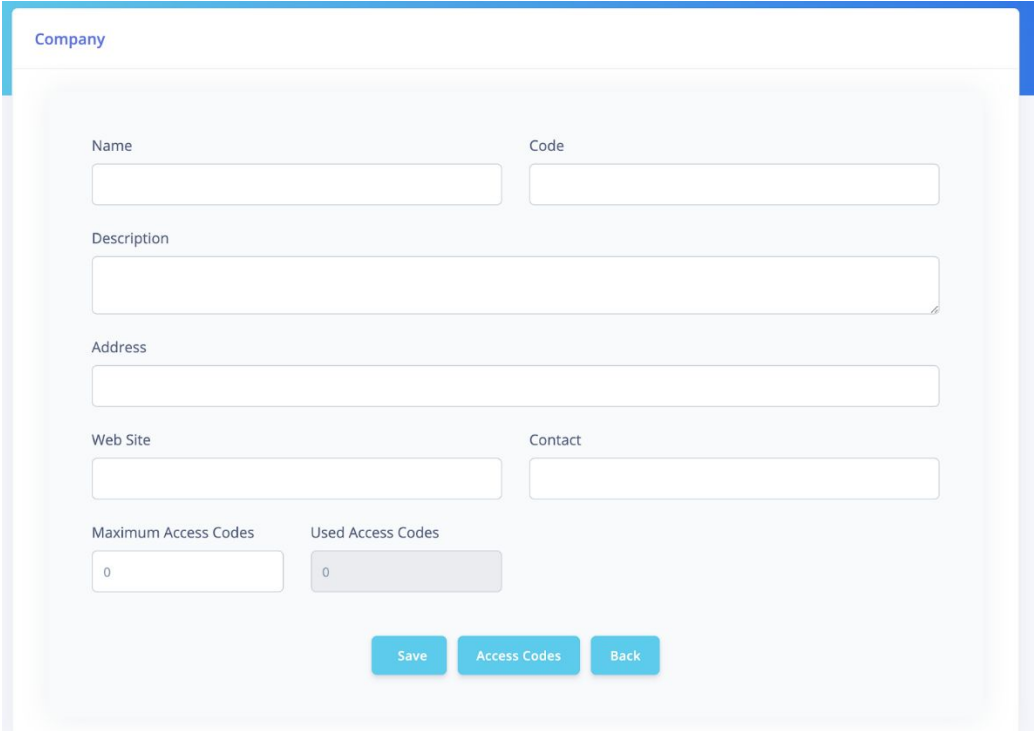
Actor	User Story	Acceptance Criteria
Candidate	As a candidate, I want to register in the system, so that I can successfully become a user.	A user must be able to enter first name, last name, username, password, email contact number to the appropriate text field and must be able to select the user role as a candidate or a recruiter. All these are mandatory fields.  Username must be unique in the system. If it duplicates an error message will be displayed.
Recruiter	As a recruiter, I will be able to register as a recruiter, so that I can successfully become a user of the system.	As a result of successful registration, the user will be redirected to the login page.
Admin	As an admin, I will be able to register a company, so that a company can become registered to the system as a company user.	An admin user must be able to enter company name, code, description, address, web site, contact and maximum access code count via text input fields. Name, Code, maximum access code counts are mandatory fields.  Code must be unique to the company. If it duplicates an error message will be displayed.  As a result of successful registration, the company will be displayed in the list of companies.

Table 3.1: user stories for user registration



The image shows a user registration page titled "Sign up". It features a light blue header with the title. Below the header, there are six input fields arranged in two columns: "First name", "Last name", "Username", "Password", "Email", and "Contact". At the bottom of the form, there are two radio buttons: "I'm a candidate" and "I'm a recruiter". A blue "Sign Up" button is centered below the radio buttons. In the bottom right corner of the page, there is a "Go back" link.

Figure 3.14: User registration page



The image shows a company registration page titled "Company". It features a light blue header with the title. Below the header, there are several input fields: "Name", "Code", "Description", "Address", "Web Site", and "Contact". At the bottom of the form, there are two input fields for "Maximum Access Codes" and "Used Access Codes", both containing the value "0". Three blue buttons are centered at the bottom: "Save", "Access Codes", and "Back".

Figure 3.15: Company registration page



## Validations

- In the user registration page first name, last name, username, password, email and contact fields are all mandatory fields. Therefore when a user clicks on the sign up button without filling any of the above mandatory fields, an error message should be displayed on top of the page with the content of the particular field cannot be empty.
- Username must be a unique field. If the username duplicates then the error message should be displayed with the content of username already exists.
- In the company registration page name, code and maximum access codes files are mandatory fields. If a user clicks on the save button while keeping any of the above fields empty an error message should be displayed with the content of the particular field cannot be blank.
- Company code input field is a unique field therefore if it duplicates an error message should be displayed with the content of company code already exists.

### 3.4.1.2 User Login

Actor	User Story	Acceptance Criteria
Candidate	As a user,	A user must be able to enter username and password via text fields. All input fields are mandatory.  Error messages will be displayed for incorrect username or password.
Recruiter	I will be able to enter my credentials in the login page,	
Admin	So that I can successfully login to the system.	

Table 3.2: user stories for user login

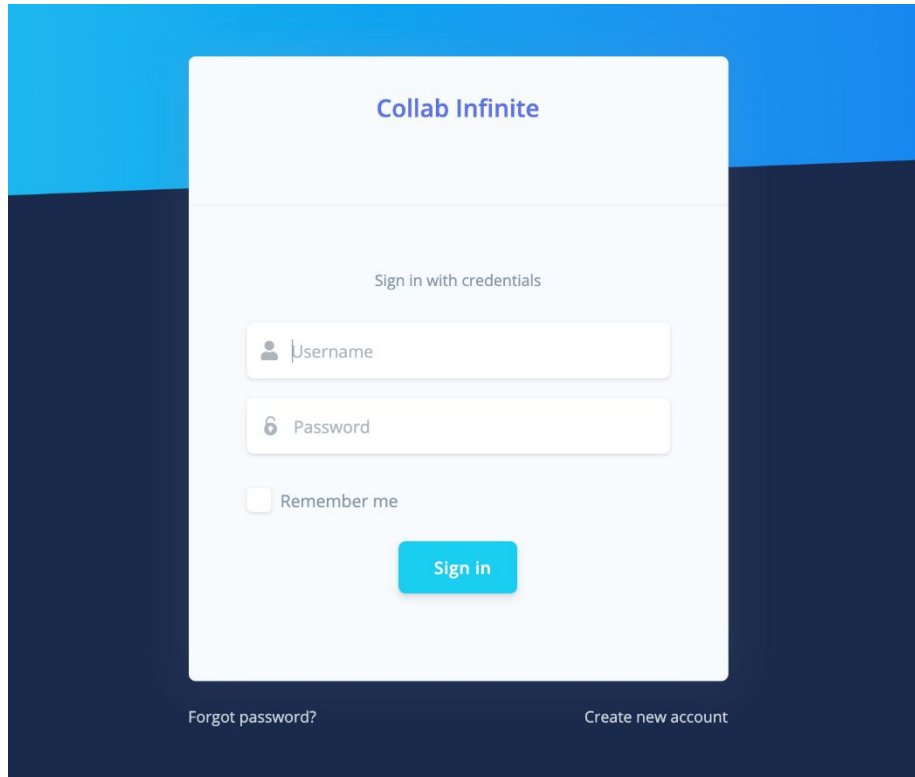


Figure 3.16: Login page

## Validations

- Username and password fields both are mandatory fields. Therefore, if a user clicks on the sign in button while keeping any of the above fields empty an error message should be displayed with the content of the particular field name cannot be empty.
- If a user entered an incorrect username an error message should be displayed with the content of username is incorrect.
- If a user entered an incorrect password an error message should be displayed with the content of password is incorrect.

### 3.4.1.3 Candidate Profile Management

<b>Actor</b>	<b>User Story</b>	<b>Acceptance Criteria</b>
Candidate	As a candidate,  I will be able to click Basic Info button,  So that my basic information will be displayed.	When a candidate user clicks Basic Info button in the profile management page, relevant basic information of the candidate must be displayed below the button.
Candidate	As a candidate,  I will be able to click Technical Skills button,  So that list of my technical skills will be displayed.	When a candidate user clicks the Technical Skills button in the profile management page, relevant technical skills of the candidate must be displayed below the button.
Candidate	As a candidate,  I will be able to click Educational Qualifications button,  So that list of my technical skills will be displayed.	When a candidate user clicks the Educational Qualifications button in the profile management page, relevant educational qualifications of the candidate must be displayed below the button.
Candidate	As a candidate,  I will be able to click Work History button,  So that list of my technical skills will be displayed.	When a candidate user clicks the Work History button in the profile management page, relevant work history of the candidate must be displayed below the button.

Candidate	<p>As a candidate,</p> <p>I will be able to edit my basic information,</p> <p>So that my basic information will be updated in the system.</p>	<p>When a candidate clicks the edit button in the basic info page, a modal window will be popped up displaying relevant input fields to edit the candidate's basic information.</p>
Candidate	<p>As a candidate,</p> <p>I will be able to add or edit my technical skills,</p> <p>So that my technical skills will be updated in the system.</p>	<p>When a candidate clicks the add button in the technical skills page, a modal window will be popped up displaying relevant input fields to add a new technical skill.</p> <p>When a candidate clicks any edit button in the list of technical skills, a modal window will be popped up to facilitate the edit of a relevant technical skill.</p> <p>When a candidate clicks any delete button in the list of technical skills, a modal window will be popped up to facilitate the deletion of a relevant technical skill.</p>
Candidate	<p>As a candidate,</p> <p>I will be able to add or edit my educational qualifications,</p> <p>So that my educational qualifications will be updated in the system.</p>	<p>When a candidate clicks the add button in the educational qualifications page, a modal window will be popped up displaying relevant input fields to add a new educational qualification.</p> <p>When a candidate clicks any edit button in the list of educational qualifications, a modal window will be popped up to facilitate the edit of a relevant educational qualification.</p> <p>When a candidate clicks any delete button in the list of educational qualifications, a modal window will be popped up to facilitate the deletion of a relevant educational qualification.</p>
Candidate	<p>As a candidate,</p> <p>I will be able to add or edit my work history,</p>	<p>When a candidate clicks the add button in the work history page, a modal window will be popped up displaying relevant input fields to add a new work history.</p>

	<p>So that my work history will be updated in the system.</p>	<p>When a candidate clicks any edit button in the list of work history, a modal window will be popped up to facilitate the edit of a relevant work history.</p> <p>When a candidate clicks any delete button in the list of work history, a modal window will be popped up to facilitate the deletion of a relevant work history.</p>
--	---	---

Table 3.3: User stories for candidate profile management

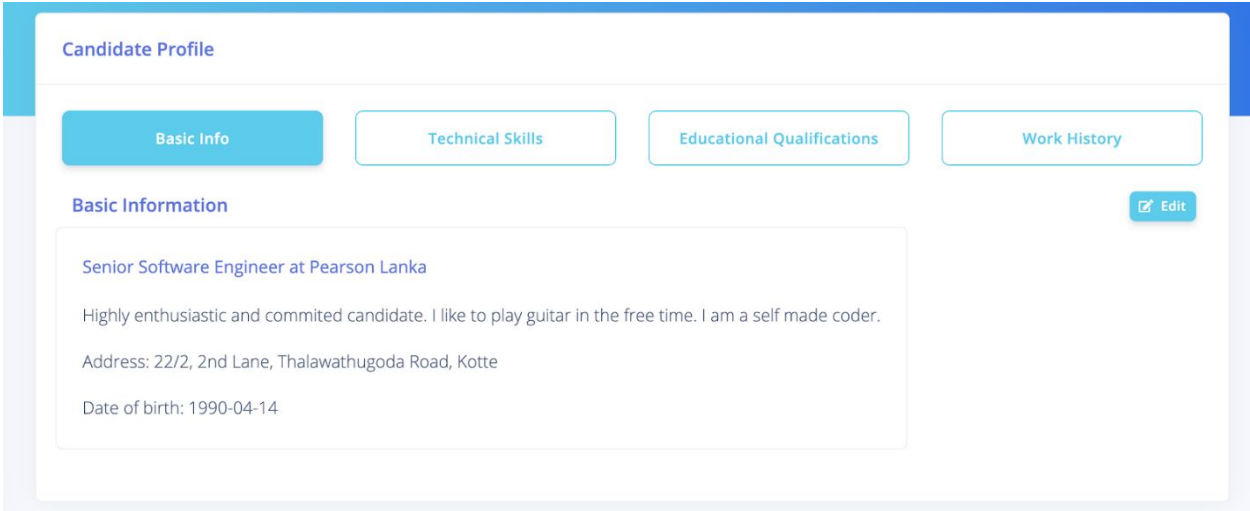


Figure 3.17: Basic information tab

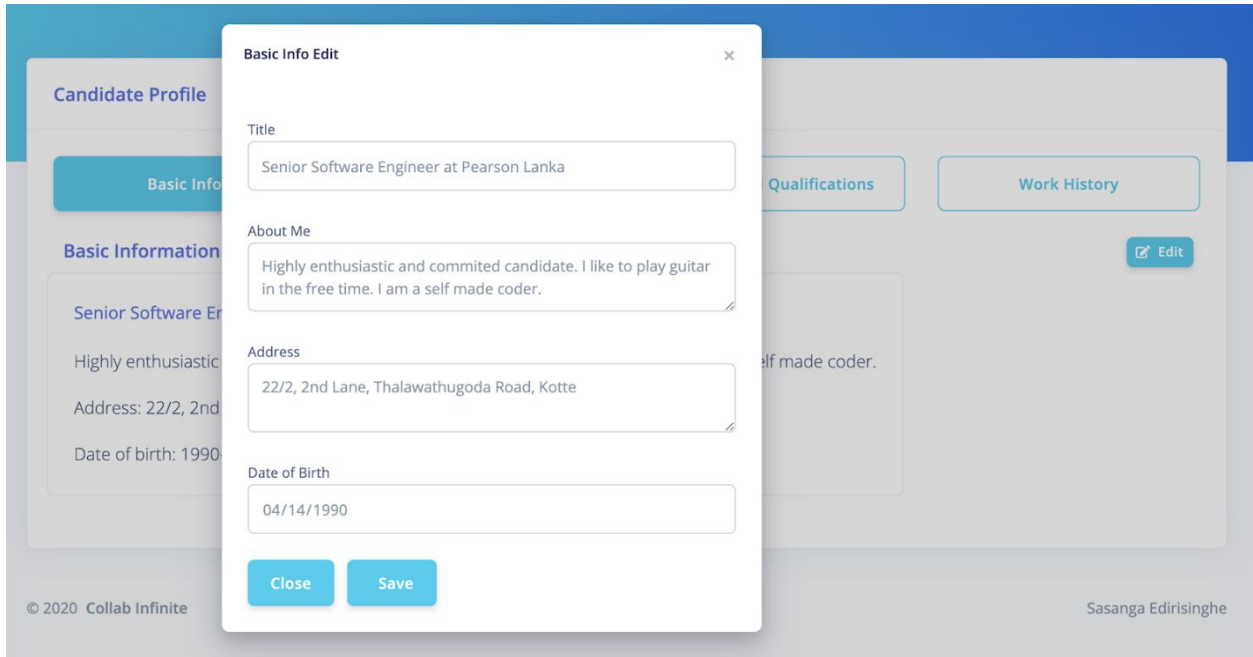


Figure 3.18: Edit basic info modal window

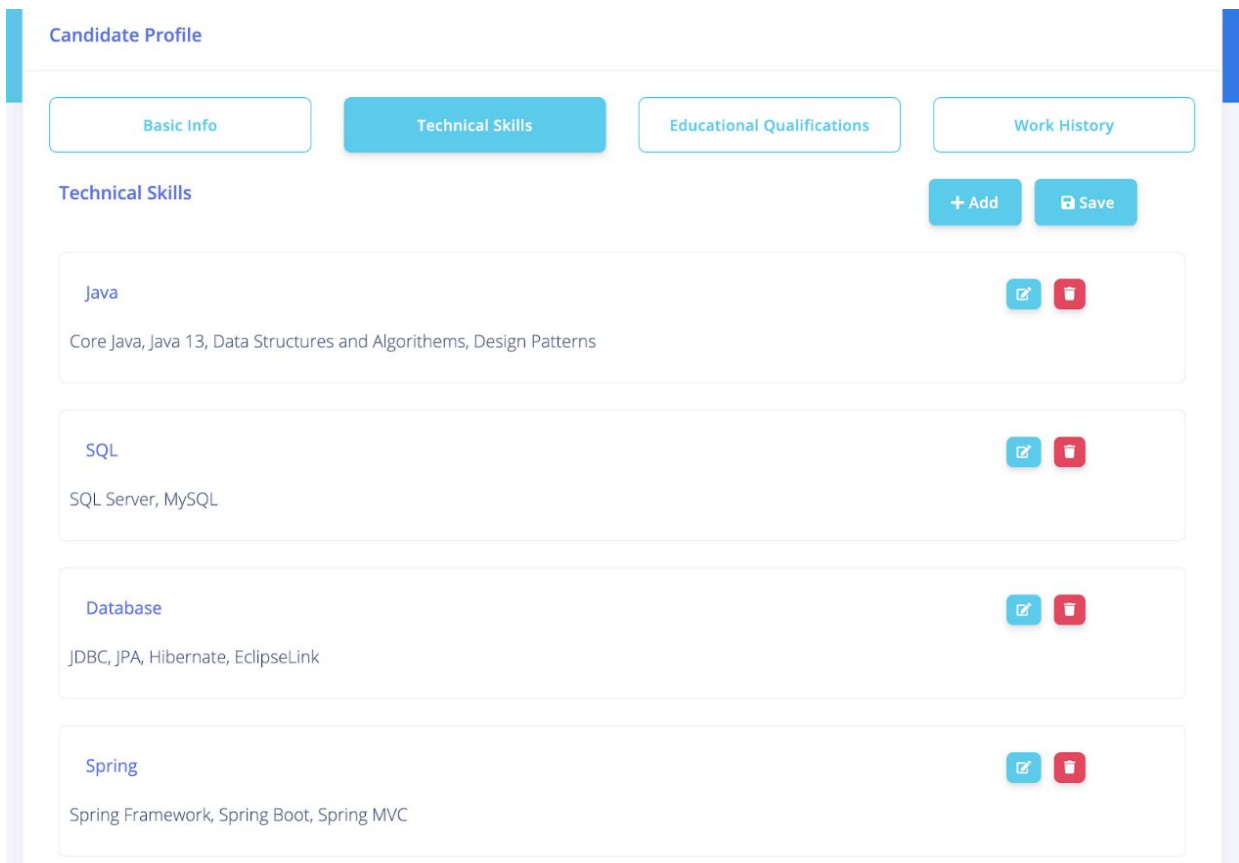


Figure 3.19: Technical skills tab

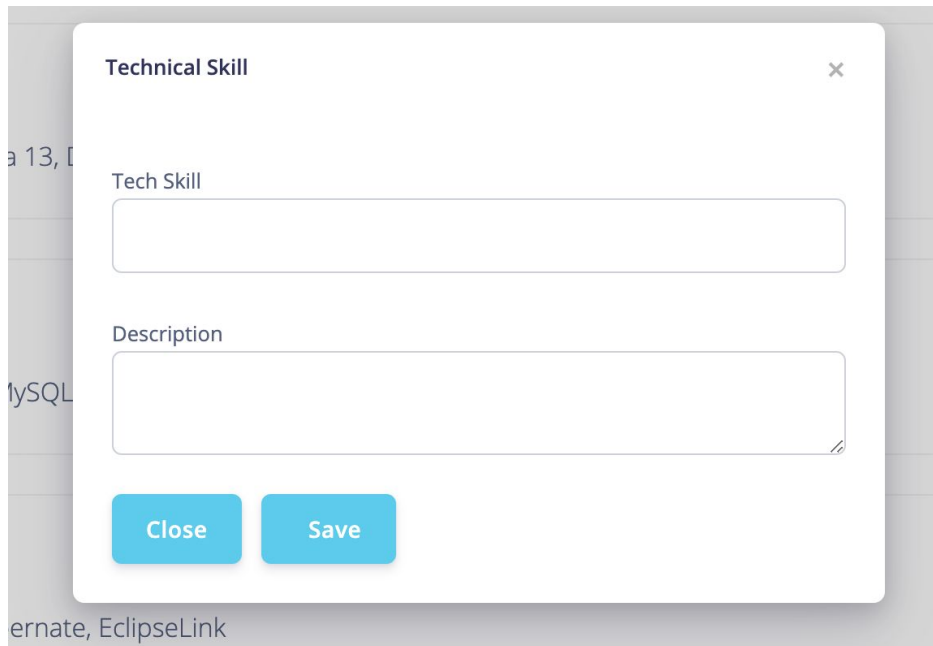


Figure 3.20: New technical skill modal window

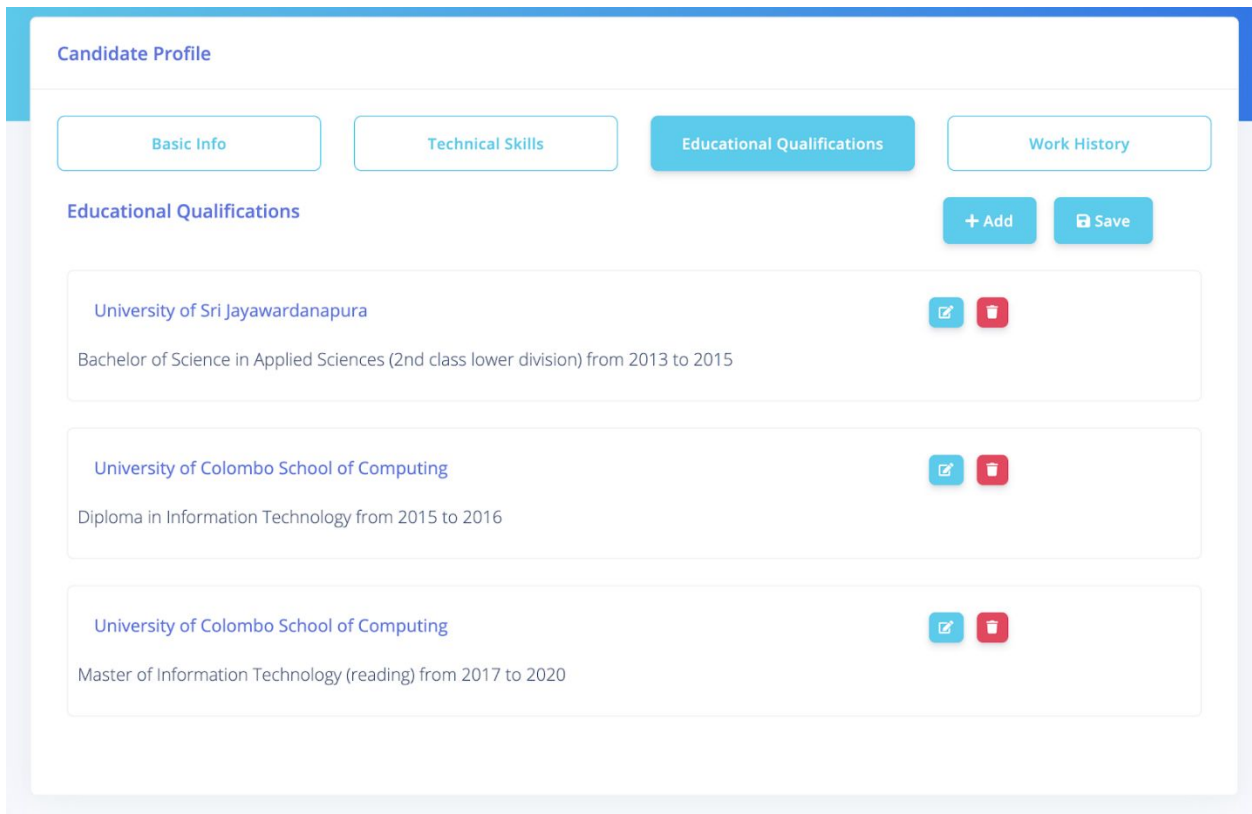
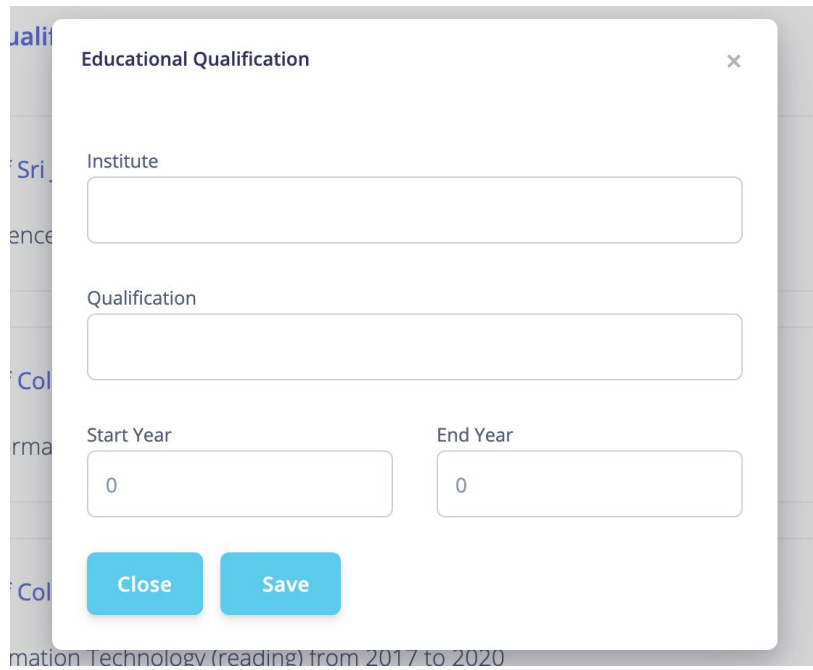


Figure 3.21: Educational qualifications tab

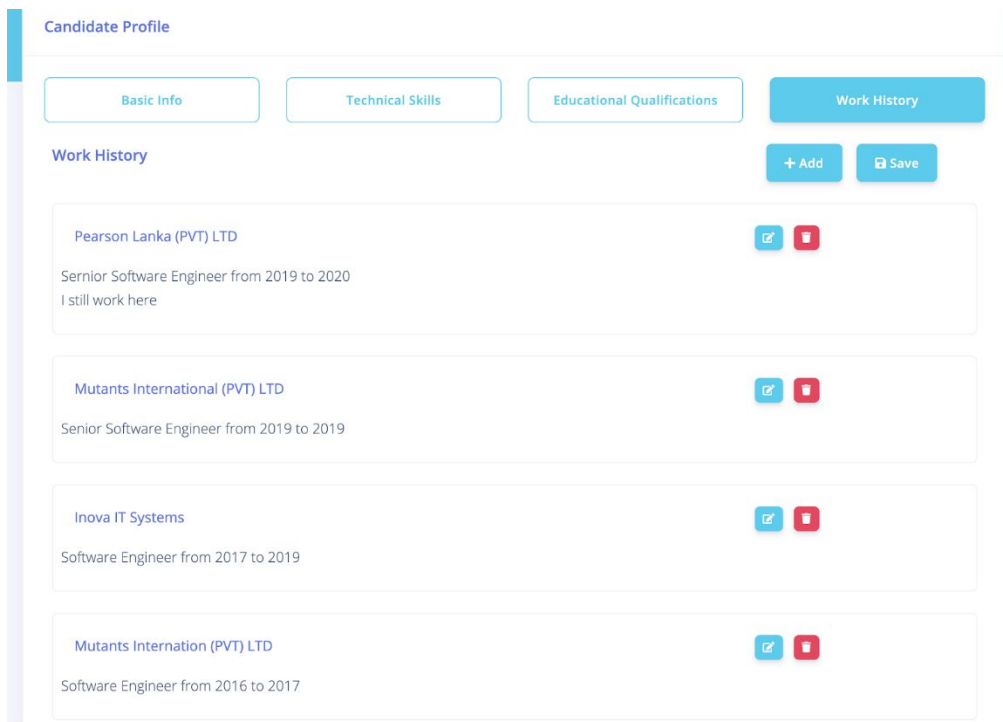


The image shows a modal window titled "Educational Qualification" with a close button (X) in the top right corner. The form contains the following fields:

- Institute:** A text input field.
- Qualification:** A text input field.
- Start Year:** A text input field containing the number "0".
- End Year:** A text input field containing the number "0".

At the bottom of the modal, there are two buttons: "Close" and "Save".

Figure 3.22: New educational qualification modal window



The image shows the "Candidate Profile" page with a "Work History" tab selected. The page has four tabs: "Basic Info", "Technical Skills", "Educational Qualifications", and "Work History". The "Work History" tab is active and contains the following content:

- Buttons:** "+ Add" and "Save" (with a save icon).
- Work History Entries:**
  - Pearson Lanka (PVT) LTD:** Senior Software Engineer from 2019 to 2020. I still work here. Includes edit and delete icons.
  - Mutants International (PVT) LTD:** Senior Software Engineer from 2019 to 2019. Includes edit and delete icons.
  - Inova IT Systems:** Software Engineer from 2017 to 2019. Includes edit and delete icons.
  - Mutants Internation (PVT) LTD:** Software Engineer from 2016 to 2017. Includes edit and delete icons.

Figure 3.23: Work history tab



The image shows a modal window titled "Work History" with a close button (X) in the top right corner. The form contains the following fields and controls:

- Company:** A single-line text input field.
- Designation:** A single-line text input field.
- Start Year:** A single-line text input field containing the number "0".
- End Year:** A single-line text input field containing the number "0".
- I'm currently working here:** A radio button.
- Description:** A multi-line text area with a small icon in the bottom right corner.
- Buttons:** Two blue buttons labeled "Close" and "Save" are positioned at the bottom of the modal.

Figure 3.24: New work history modal window

## Validations

- When a user clicks on the candidate profile menu item; basic info, technical skills, educational qualifications and work history tab buttons should be displayed.
- When a user clicks on the basic info tab button, the basic info tab should be displayed which consists of basic information of the candidate.
- When a user clicks on the edit button on the basic info page a popup window should be displayed with the mandatory fields of title, address, about me and date of birth. If a user clicks the save button while keeping any of the above fields empty, an error message should be displayed with the content of the particular field cannot be empty.
- When a user clicks the technical skills tab button list of the candidate's technical skills should be displayed. Each item should be edited and deleted with the respective buttons in the list. By clicking the edit button or add button a popup window should be displayed. In that only the tech skill field is mandatory, if a user tries to skip it and save an error message should be displayed by saying tech skill cannot be empty.
- When a user clicks the educational qualifications tab button list of the candidate's educational qualifications should be displayed. Each item should be edited and deleted with the respective buttons in the list. By clicking the edit button or add button a popup

window should be displayed. In that popup window institute, qualification, start year and end year all are mandatory fields. If a user clicks on the save button while keeping any of these fields empty, an error message should be displayed by saying the particular field cannot be empty.

- When a user clicks the work history tab button list of the candidate's work history should be displayed. Each item should be edited and deleted with the respective buttons in the list. By clicking the edit button or add button a popup window should be displayed. In that popup window company, designation, start year and end year are mandatory fields. If a user clicks on the save button while keeping any of these fields empty, an error message should be displayed by saying the particular field cannot be empty.

### 3.4.1.4 Vacancy Management

Actor	User Story	Acceptance Criteria
Recruiter	As a recruiter,  I will be able to click vacancies menu item,  so that list of vacancies posted by me will be displayed.	When a recruiter clicks the vacancies menu item, a list of vacancies posted by the recruiter will be displayed in the vacancies page.  List must be sorted by the descending values of date and time posted.
Recruiter	As a recruiter,  I will be able to add a vacancy,  so that vacancy will be posted to the system.	When a recruiter clicks the create new button, redirects to vacancy create page. A recruiter can add basic information of the vacancy which are title, description and expiry date. And also, can add any number of technical skills, educational qualifications and roles and responsibilities.
Recruiter	As a recruiter,  I will be able to update a vacancy,	When a recruiter clicks the edit button in the list of vacancies, redirects to vacancy create page. A recruiter can update basic information of the vacancy which are title, description and expiry date. And also, can add and update any number of technical skills, educational qualifications and roles and responsibilities.

	so that vacancy will be updated in the system.	
Candidate	As a candidate,  I will be able to click jobs menu item,  so that available job vacancies will be displayed in the job vacancies page	When a candidate clicks jobs menu item, it redirects to the job vacancies page and displays available job vacancies.  Job vacancies must be sorted by descending date and time values posted.
Candidate	As a candidate,  I will be able to click view more link in any job vacancy,  so that I will be able to see the detailed job vacancy.	When a candidate clicks any view more link from the list of job vacancies, it will redirect to the detailed job vacancy page.
Candidate	As a candidate,  I will be able to click the apply button in the job vacancy,  so that I can apply for the job vacancy.	When a candidate clicks the apply button in the detailed job vacancy page, he will be able to apply for the job vacancy.

Table 3.4: User stories for vacancy management

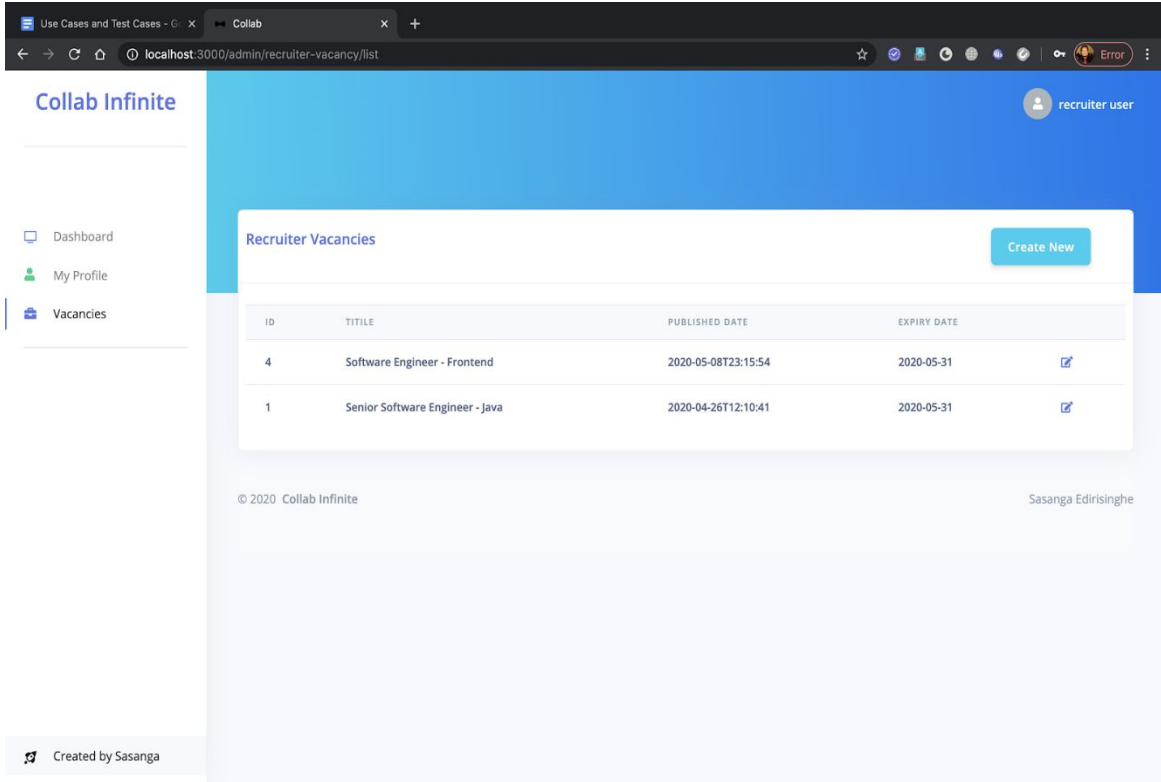


Figure 3.26: Recruiter vacancies page

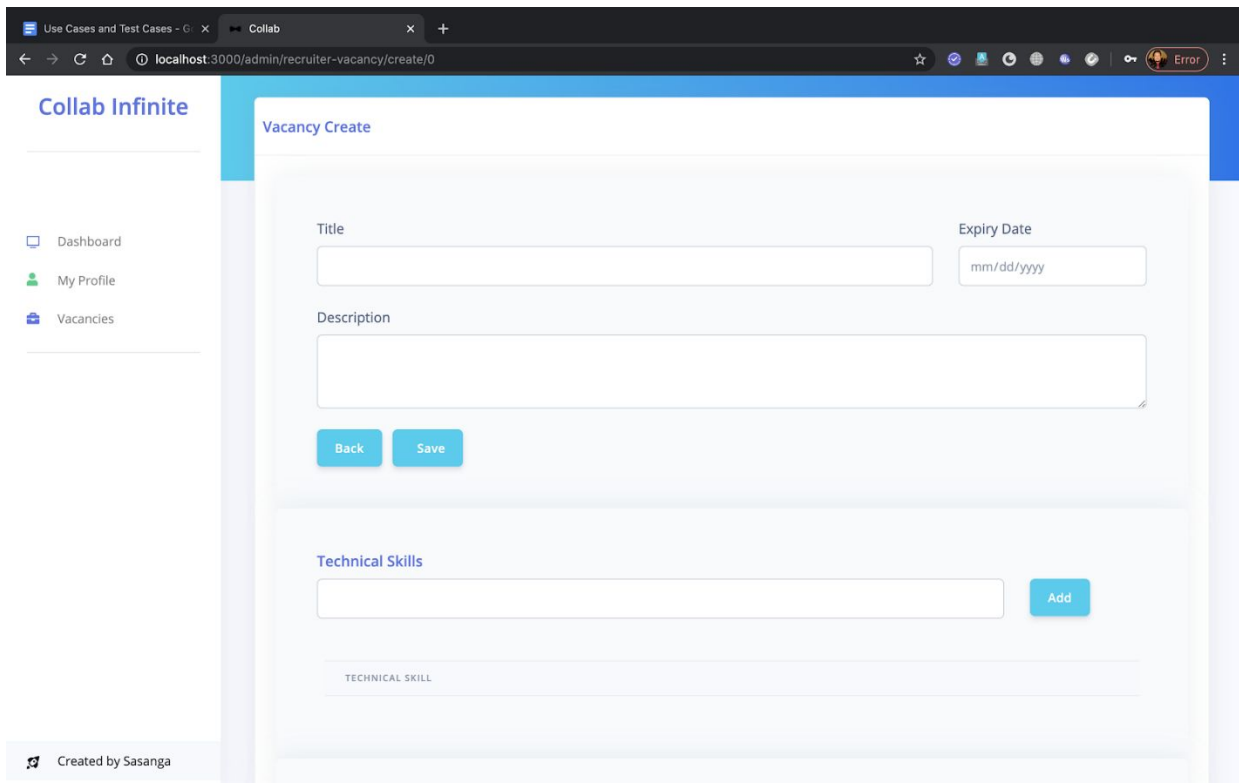


Figure 3.27: New vacancy creation page

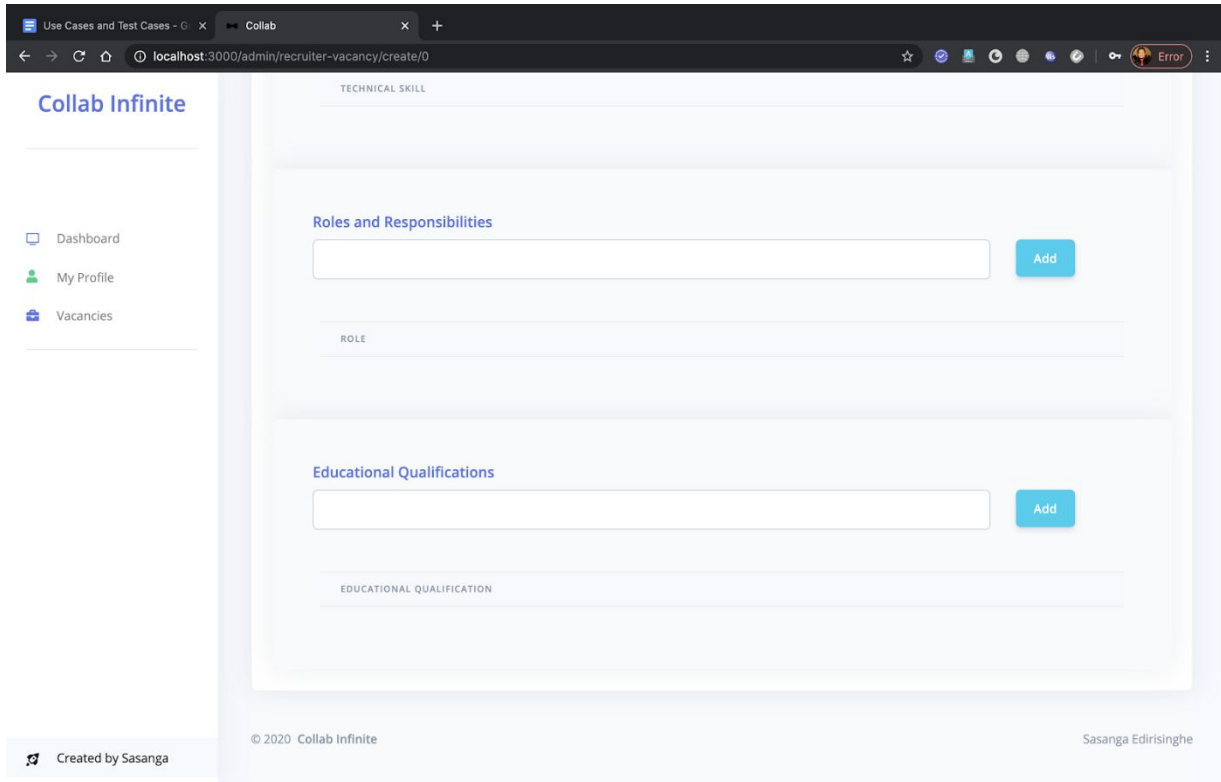


Figure 3.28 New vacancy creation add roles and responsibilities

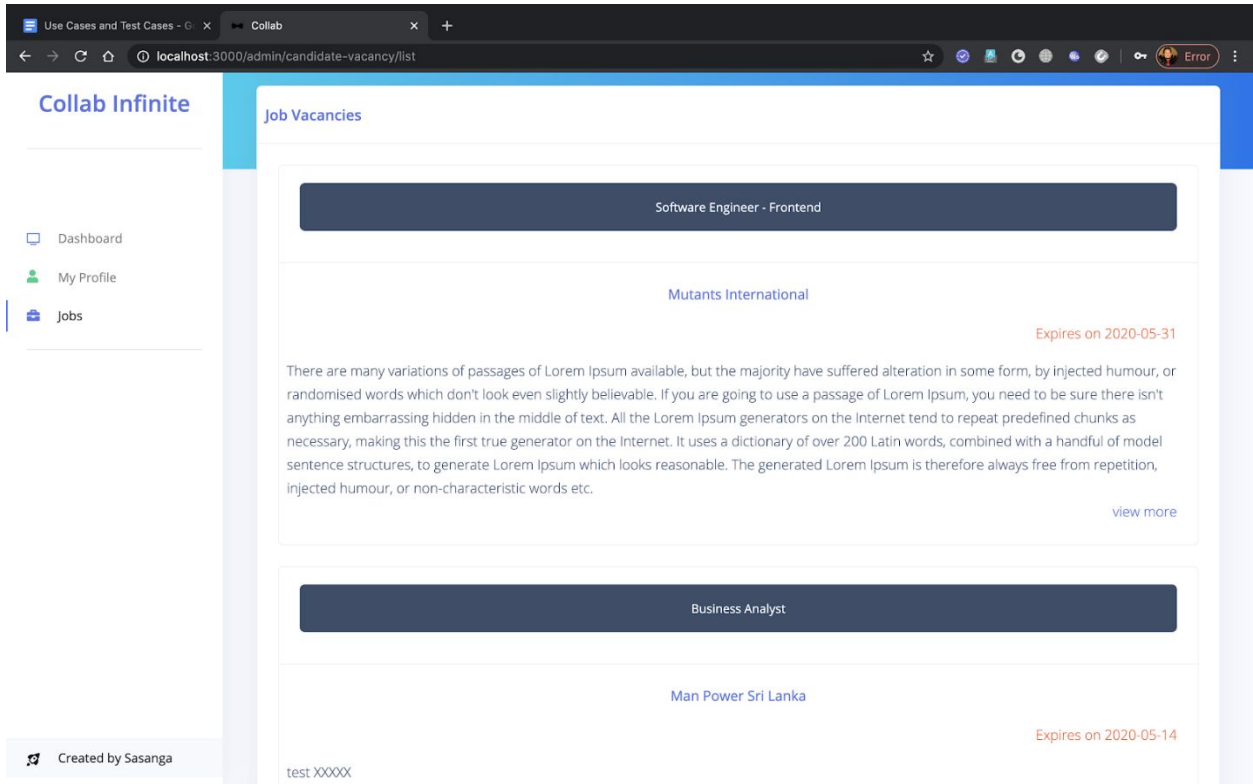


Figure 3.29: Candidate available vacancies

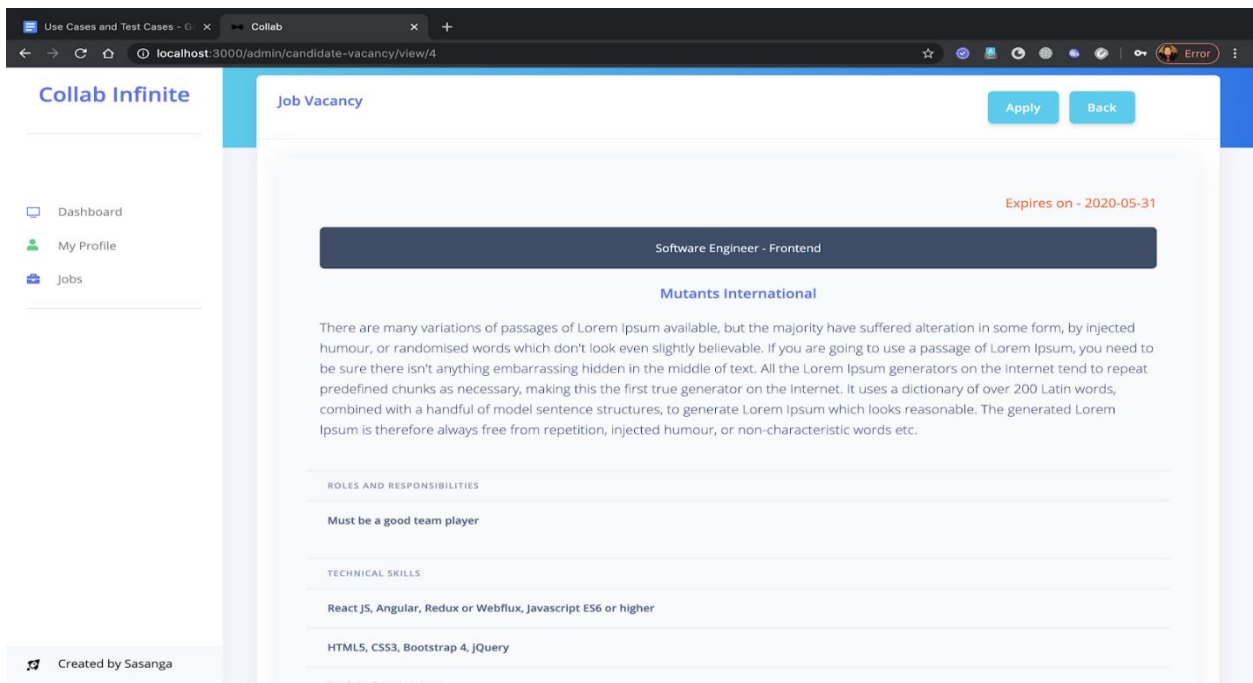


Figure 3.30: Candidate detailed vacancy page

## Validations

- When a recruiter clicks on the Vacancies menu item, a list of vacancies created by the particular recruiter should be displayed.
- When a recruiter clicks on the create new button it should be navigated to the create vacancy page. Title, expiry date are mandatory fields. Recruiters should also be able to separate lists of technical skills, roles and responsibilities, educational qualifications to the particular vacancy.
- When a candidate clicks on the jobs menu item it should be redirected to the candidate available vacancies page. It should display a list of available vacancies. When a candidate clicks on the view more button, it should be navigated to the detailed description of the vacancy.

### 3.4.1.5 Candidate Applied Vacancies

Actor	User Story	Acceptance Criteria
Candidate	As a candidate, I will be able to click applied jobs menu item, so that list of my applied vacancies will be displayed.	When a candidate clicks the applied jobs menu item, a list of my applied vacancies will be displayed in the applied vacancies page.  List must be sorted by the descending values of date and time applied.
Candidate	As a candidate, I will be able to click the view icon, so that it will redirect to the particular applied vacancy details page.	When a candidate clicks the view icon of the applied vacancies page it will be redirected to the detailed applied vacancy page.

Table 3.5: Candidate applied vacancies user stories

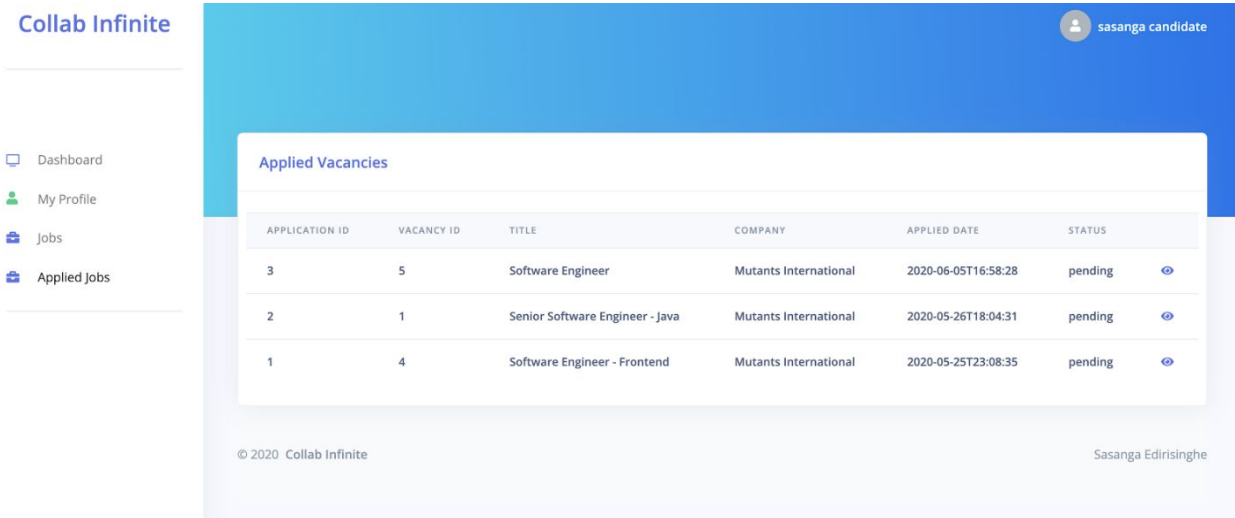


Figure 3.31: Candidate applied vacancies

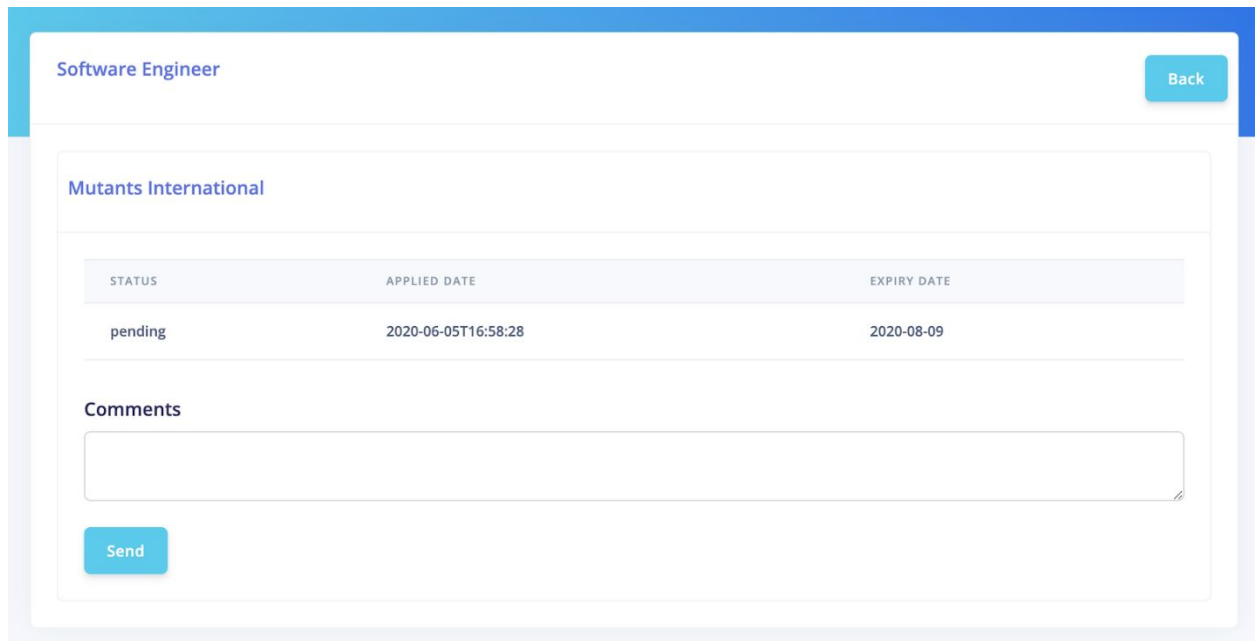


Figure 3.32: Candidate applied vacancy detail

## Validations

- In the detailed candidate applied vacancy page, the comment field is a mandatory field. If the candidate clicks the send button while keeping the comment text area empty, an error message should be displayed by saying the comment cannot be empty.



## Chapter 4. Evaluation

Throughout this chapter a critical evaluation on the proposed system will be given and a gap analysis between the project objectives and outcome was clearly stated. All the obstacles, distractions and limitations faced during the time of project development has been mentioned. Lessons learned upon the failures were identified in order to ensure the further improvements for the given solution.

### 4.1 System feature evaluation

For each feature the user story has been evaluated in order to simplify the business scenario and the respective acceptance criteria was stated to define the technical perspective of it. All the test cases have been provided as evidence to prove the successful evaluation of the system.

#### 4.1.1 User Registration

Test Case #	Test Case	Expected Result	Actual Result	Pass / Fail
01	User register with correct data for input fields	Redirect to login page.	As expected	Pass
02	Click the sign-up button with empty mandatory fields.	Display error message	As expected	Pass
03	Click the sign-up button with an existing user name.	Display error message	As expected	Pass
04	Click the sign-up button with the backend disconnected.	Display error message	As expected	Pass
05	Click the save button in company registration with entering correct values to input fields.	Redirect to company list page	As expected	Pass
06	Click the save button in company registration with keeping mandatory fields empty.	Display error message	As expected	Pass

07	Click the save button in company registration with entering existing company code.	Display error message	As expected	Pass
----	--	-----------------------	-------------	------

Table 4.1: test cases for user registration

The image shows a 'Sign up' form with the following fields and elements:

- First name**: Input field with a vertical cursor.
- Last name**: Input field.
- Username**: Input field with a tooltip error message: "Please fill in this field." (indicated by a yellow warning icon).
- Password**: Input field.
- Email**: Input field.
- Contact**: Input field.
- Radio buttons**: "I'm a candidate" and "I'm a recruiter".
- Sign Up**: A blue button.
- Go back**: A small link at the bottom right.

Figure 4.3: Test case 02 - User sign up

The screenshot shows a 'Sign up' form with a red error banner at the top that reads 'username already exists'. The form fields are as follows:

First name	duplicate	Last name	user
Username	sasanga_candidate	Password	*****
Email	madhumaledirisinghe@gmail.com	Contact	0770406395

At the bottom, there are two radio buttons: 'I'm a candidate' (selected) and 'I'm a recruiter'. A blue 'Sign Up' button is centered at the bottom.

Figure 4.4: Test case 03 - Username already exists

The screenshot shows a 'Sign up' form with a red error banner at the top that reads 'network error'. The form fields are as follows:

First name	duplicate	Last name	user
Username	sasanga_candidate2	Password	*****
Email	madhumaledirisinghe@gmail.com	Contact	0770406395

At the bottom, there are two radio buttons: 'I'm a candidate' (selected) and 'I'm a recruiter'. A blue 'Sign Up' button is centered at the bottom. A 'Go back' link is visible in the bottom right corner.

Figure 4.5: Test case 04 - Disconnected backend

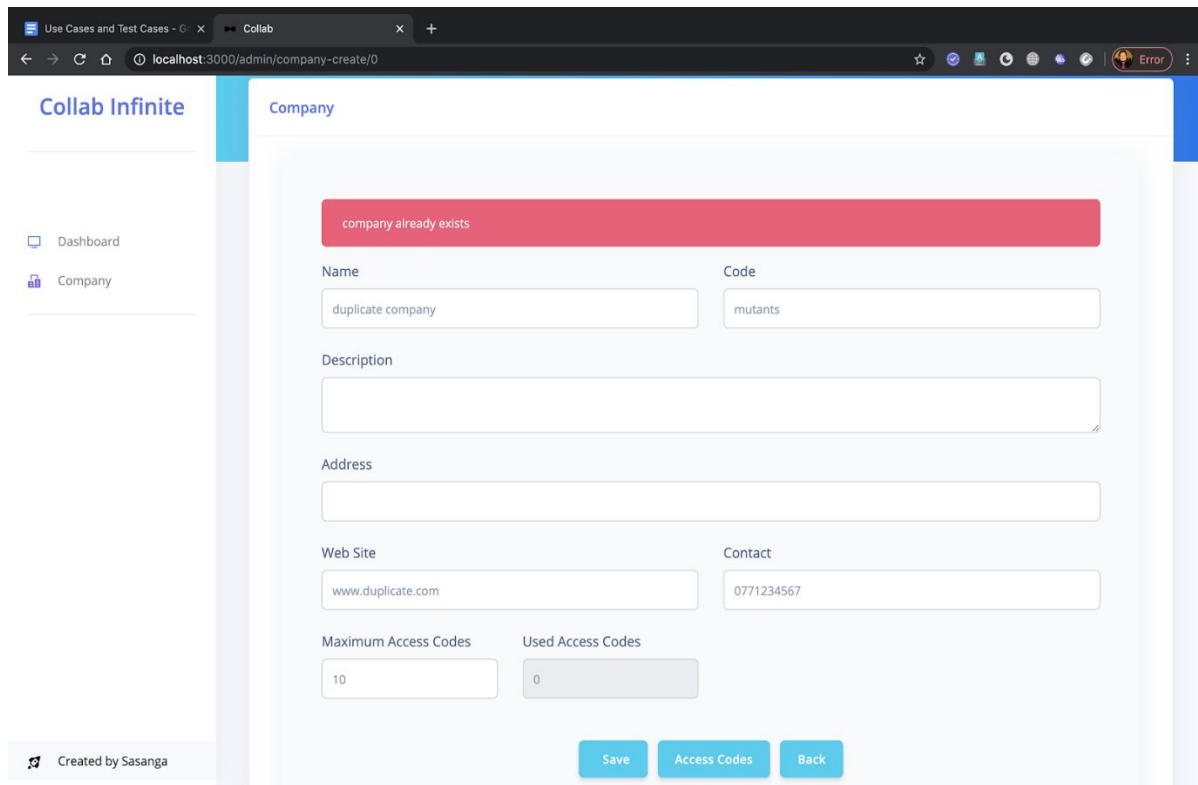


Figure 4.6: Test case 07 - Existing company code

### 4.1.2 User Login

Test Case #	Test Case	Expected Result	Actual Result	Pass / Fail
08	User login with correct username and correct password	Successful login	As expected	Pass
09	User login with empty username and/or empty password	Display error message	As expected	Pass
10	User login with invalid username	Display error message	As expected	Pass
11	User login with invalid password	Display error message	As expected	Pass

Table 4.2: Test cases for user login

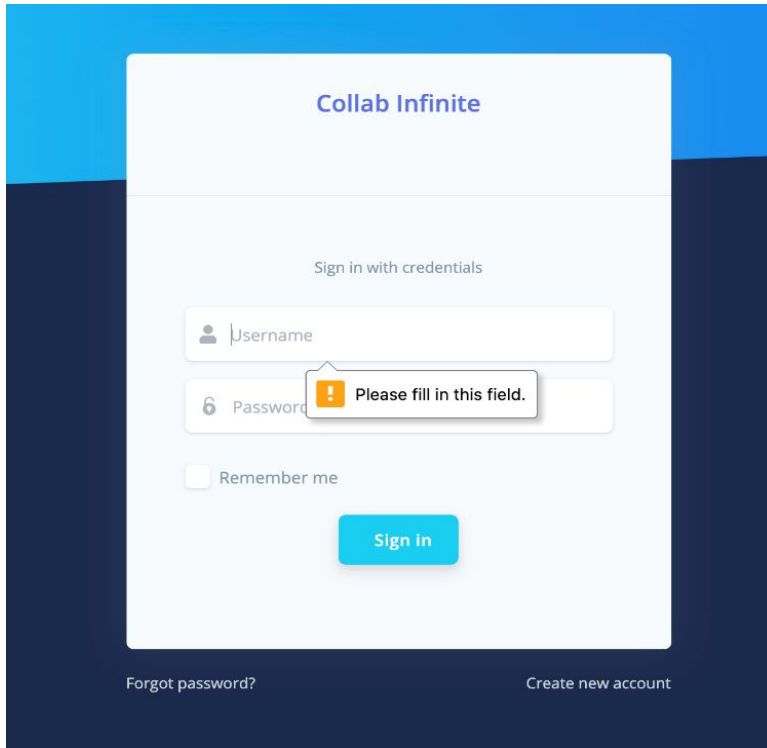


Figure 4.8: Test case 09 - Empty username

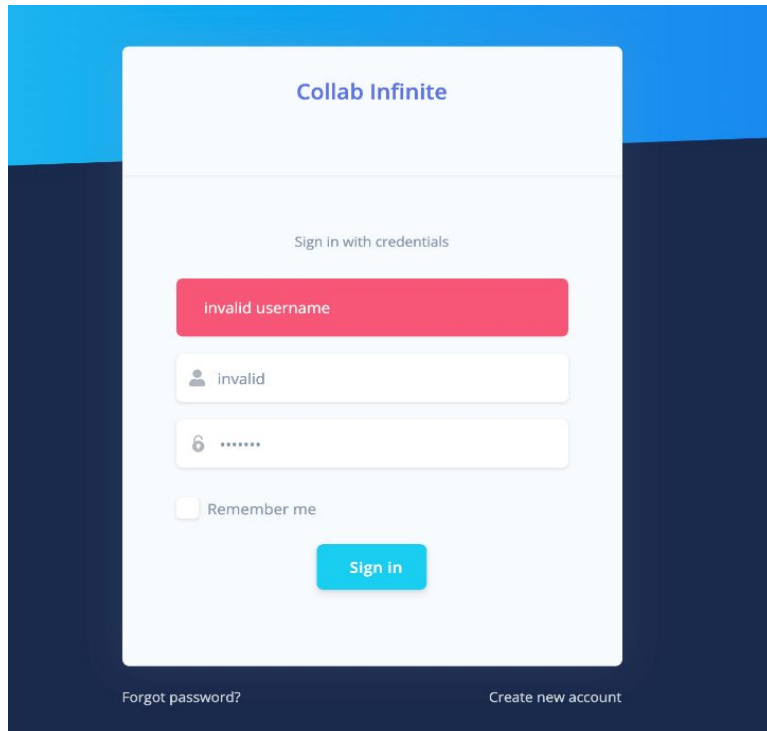


Figure 4.9: Test case 10 - Invalid username

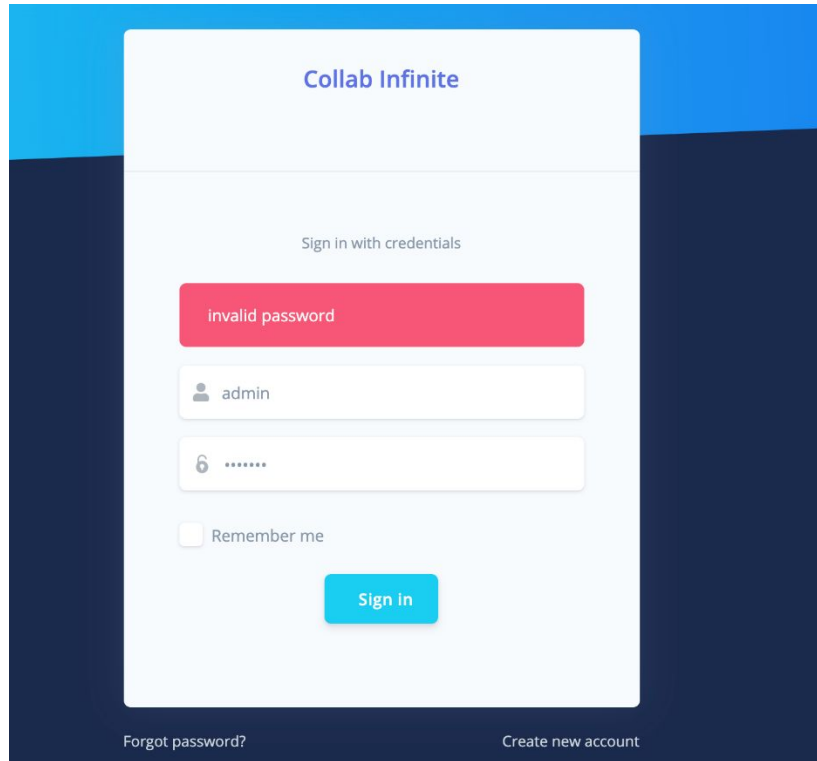


Figure 4.10: Test case 11 - Invalid password

### 4.1.3 Candidate Profile Management

Test Case #	Test Case	Expected Result	Actual Result	Pass / Fail
12	Click basic info button	Display basic info page	As expected	Pass
13	Click technical skills button	Display technical skills page	As expected	Pass
14	Click educational qualifications button	Display educational qualifications page	As expected	Pass
15	Click work history button	Display work history page	As expected	Pass

16	Click edit button in the basic info page	Popup a modal window with relevant input fields	As expected	Pass
17	Click save button in the basic info edit modal window with necessary input values	Update basic info	As Expected	Pass
18	Click save button in the basic info edit modal window keeping mandatory input fields empty	Display error message	As expected	Pass
19	Click add button in the technical skills page	Popup a modal with relevant empty input fields	As expected	Pass
20	Click save button in the popup modal for add technical skills page with keeping mandatory fields empty	Display error message	As expected	Pass
21	Click save button in the popup modal for add technical skills page with necessary input values	Add new technical skill	As expected	Pass
22	Click any edit button from the list of technical skills.	Popup a modal which populates relevant values in text fields.	As expected	Pass
23	Modify input fields in the edit modal window and click the save button.	Updated values in the relevant technical skill.	As expected	Pass
24	Click the save button in edit modal without updating the values.	List of technical skills remains the same.	As expected	Pass
25	Click any delete button in the list of technical skills.	Popup a confirm dialog for the deletion.	As expected	Pass
26	Click the yes button in the delete confirm window.	Delete the relevant technical skill.	As expected	Pass
27	Click add button in the educational qualifications page	Popup a modal with relevant empty input fields	As expected	Pass

28	Click save button in the popup modal for add educational qualifications page with keeping mandatory fields empty	Display error message	As expected	Pass
29	Click save button in the popup modal for add educational qualifications page with necessary input values	Add new technical skill	As expected	Pass
30	Click any edit button from the list of educational qualifications.	Popup a modal which populates relevant values in text fields.	As expected	Pass
31	Modify input fields in the edit modal window and click the save button.	Updated values in the relevant educational qualification.	As expected	Pass
32	Click the save button in edit modal without updating the values.	List of educational qualifications remains the same.	As expected	Pass
33	Click any delete button in the list of educational qualifications.	Popup a confirm dialog for the deletion.	As expected	Pass
34	Click the yes button in the delete the confirm window.	Delete the relevant educational qualifications.	As expected	Pass
35	Click add button in the work history page	Popup a modal with relevant empty input fields	As expected	Pass
36	Click save button in the popup modal for add work history page with keeping mandatory fields empty	Display error message	As expected	Pass
37	Click save button in the popup modal for add work history page with necessary input values	Add new work history	As expected	Pass
38	Click any edit button from the list of work history.	Popup a modal which populates relevant values in text fields.	As expected	Pass



39	Modify input fields in the edit modal window and click the save button.	Updated values in the relevant work history.	As expected	Pass
40	Click the save button in edit modal without updating the values.	List of work history remains the same.	As expected	Pass
41	Click any delete button in the list of work history.	Popup a confirm dialog for the deletion.	As expected	Pass
42	Click the yes button in the delete the confirm window.	Delete the relevant work history.	As expected	Pass

Table 4.3: Test cases for candidate profile management

#### 4.1.4 Vacancy Management

Test Case #	Test Case	Expected Result	Actual Result	Pass / Fail
43	Click vacancies menu item	Display vacancies page for the particular recruiter	As expected	Pass
44	Click create new button	Redirect to vacancy create page	As expected	Pass
45	Click the save button without filling mandatory fields	Display error message	As expected	Pass
46	Click save button with correct values to mandatory fields	Redirects to vacancy list page	As expected	Pass
47	Click the add button in the technical skills field keeping the text field empty.	Display error message	As expected	Pass
48	Click add button in the technical skills field with correct values	Add a technical skill to the list	As expected	Pass

49	Click the edit button in the technical skills list.	Display the particular technical skill in the text field.	As expected	Pass
50	Click the delete button in the technical skills list.	Deletes the particular item from the list.	As expected	Pass
51	Click the add button in the roles and responsibilities field keeping the text field empty.	Display error message	As expected	Pass
52	Click add button in the roles and responsibilities field with correct values	Add a role to the list	As expected	Pass
53	Click the edit button in the roles and responsibilities list.	Display the particular role in the text field.	As expected	Pass
54	Click the delete button in the roles and responsibilities list.	Deletes the item from the list.	As expected	Pass
55	Click the add button in the educational qualifications field keeping the text field empty.	Display error message	As expected	Pass
56	Click add button in the educational qualifications field with correct values	Add an educational qualification to the list	As expected	Pass
57	Click the edit button in the educational qualifications list.	Display the educational qualification in the text field.	As expected	Pass
58	Click the delete button in the educational qualifications list.	Deletes the item from the list.	As expected	Pass

Table 4.4: Test cases for vacancy management

Vacancy Create

Title

Expiry Date

Description

**Please fill in this field.**

Figure 4.8: Test case 45 - Keeping mandatory fields empty

Technical Skills

TECHNICAL SKILL

C#, .NET Framework	<input type="button" value="edit"/> <input type="button" value="delete"/>
ASP.NET, Web API	<input type="button" value="edit"/> <input type="button" value="delete"/>

Figure 4.9: Test case 48 - Technical skills happy path

Roles and Responsibilities

ROLE

Must be a good team player	<input type="button" value="edit"/> <input type="button" value="delete"/>
----------------------------	---

Figure 4.10: Test case 52 - Roles and responsibilities happy path

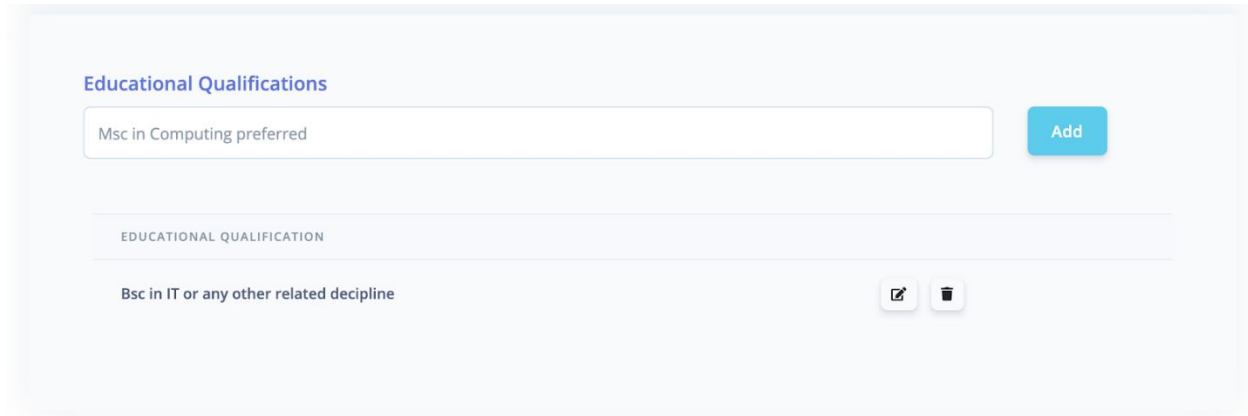


Figure 4.11: Test case 56 - educational qualifications happy path

## 4.2 Failures to achieve given objectives

Following points were identified as the failures to achieve prior stated objectives in section 2.3.

- 1.A large user base couldn't capture when testing the system features due to the prevailing situation of the world (Covid-19 pandemic).
- 2.Due to the changes in the requirements from several stakeholders the scope finalization has been delayed and because of that project implementation was failed to achieve the planned timelines.
- 3.Reporting module implementation has been limited to few reports.
- 4.With the prevailing Covid-19 pandemic situation, was unable to get the support of Devops from the client side so the system couldn't host on a remote server.
- 5.Due to the tight time schedules a user guide couldn't prepare.

## 4.3 Lessons learned during the project development

- 1.Learned new technologies – Java, Spring Boot, JPA, Hibernate, ReactJS
2. Time management in critical and unexpected situations and dealt with tight deadlines

3. Practical knowledge gained in documentation of different areas of SDLC
4. knowledge in using project management tools like Trello and Swagger and methodologies (Agile) and software development methodologies (Scrum)
5. Adapting and changing software development process according to the unexpected pandemic situation.

## **Chapter 5. Conclusion**

This chapter concludes the work and the outcomes that have been achieved so far and the actions that can be taken for further improvements of the system and a product as a whole.

### **5.1 Summary of Work**

This system was introduced in order to simplify the workload and improve the quality of process by minimizing the errors and increasing the efficiency level of recruiting and job seeking processes. In this process of making such an advanced product, several steps to ensure accuracy have been taken to make the following functions more effective and user friendly.

1. Create a resume through the system according to the qualifications which job seekers update when they get registered.
2. Job seekers will get the chance to search employers who have vacancies which go with their profiles.
3. Job applicants will get notified of the upcoming interviews and the first interview will be conducted online through the system.
4. Companies can maintain multiple recruiters under their company profile.

First the scope of the project was identified along with the boundaries and limitations. Then requirements were analyzed and priority order was created then all the analyzed and prioritized requirements were added to the product backlog. According to the priority each item is taken to respective iteration and the implementation and testing occurred.

Development of the system took place in two ways. First is to create the APIs, a model driven approach was used to create APIs. Each model was mapped to database entities. Then the frontend web application was developed. HttpServices were created to connect this frontend and the backend.

### **5.2 Improvements and future works**

#### **1. Enabling video conferencing to conduct online interviews.**

This feature will facilitate users to conduct interviews without reaching the physical location. It helps to reduce the cost and increase the efficiency and productivity of the recruitment process.

#### **2. Completing reporting module.**

Reporting module is the feature which enables recruiters and companies to generate statistics and reports on their daily activities and performance such as number of jobs posted daily, number of interviews conducted successfully etc.

### **3. Deploying the system on AWS and getting a proper domain name with search engine optimization.**

Hosting the system on AWS enables fast and smooth performance and having a good domain name helps to easily appear in the system in search results.

### **4. Facilitate real time chat service to provide online help to recruiters as well as job seekers.**

Enabling this feature will allow users to ask for help regarding the system related issues and clarifications in no time.

### **5. Starting automation testing.**

This will help to test the system against a very large number of users especially when the system gets expanded to a larger user base.

## References

- [1] Skype, "Skype," Microsoft, [Online]. Available: <https://www.skype.com/en>.
- [2] Greenhouse, "Greenhouse," Greenhouse, [Online]. Available: <https://www.greenhouse.io/>.
- [3] LessOnly, "LessOnly," LessOnly, [Online]. Available: <https://www.lessonly.com/>.
- [4] WebRecruit, "WebRecruit," WebRecruit, [Online]. Available: <https://www.webrecruit.co/>.
- [5] Monster.com, "Monster.com," Monster.com, [Online]. Available: <https://www.monster.com/>.
- [6] Jobserve.com, "Jobserve.com," Jobserve.com, [Online]. Available: <https://www.jobserve.com/lk/en/Job-Search/>.
- [7] LinkedIn, "LinkedIn," LinkedIn, [Online]. Available: <https://www.linkedin.com/>.
- [8] MySpace, "MySpace," MySpace, [Online]. Available: <https://myspace.com/>.
- [9] Facebook, "Facebook," Facebook, [Online]. Available: <https://www.facebook.com/>.
- [10] Twitter, "Twitter," Twitter, [Online]. Available: <https://twitter.com/>.
- [11] Google+, "Google+," Google+, [Online]. Available: <https://aboutme.google.com/u/0/?referer=gplus>.
- [12] Indeed.com, "Indeed.com," Indeed.com, [Online]. Available: <https://www.indeed.com/>.
- [13] CareerBuilder, "CareerBuilder," CareerBuilder, [Online]. Available: <https://www.careerbuilder.com/>.
- [14] Google, "Google," Google, [Online]. Available: <https://www.google.com/>.
- [15] HackerRank, "HackerRank," HackerRank, [Online]. Available: <https://www.hackerrank.com/>.
- [16] Pymetrics, "Pymetrics," Pymetrics, [Online]. Available: <https://www.pymetrics.ai/>.
- [17] S. M. Group, "Self Management Group," Self Management Group, [Online]. Available: <https://www.selfmgmt.com/>.
- [18] Bullhorn, "Bullhorn," Bullhorn, [Online]. Available: <https://www.bullhorn.com/eu/>.
- [19] SAP, "SAP," SAP, [Online]. Available: <https://www.sap.com/index.html>.
- [20] ICIMS, "ICIMS," ICIMS, [Online]. Available: <https://www.icims.com/>.
- [21] SmartRecruiters, "SmartRecruiters," SmartRecruiters, [Online]. Available: <https://www.smartrecruiters.com/>.
- [22] Taleo, "Taleo," Taleo, [Online]. Available: <https://tbe.taleo.net/login/index.html>.
- [23] Smashfly, "Smashfly," Smashfly, [Online]. Available: <https://www.smashfly.com/>.



- [24] Yello, "Yello," Yello, [Online]. Available: <https://yello.co/>.
- [25] ConveyIQ, "ConveyIQ," [Online]. Available: <https://www.conveyiq.com/>.
- [26] SparkHire, "SparkHire," SparkHire, [Online]. Available: <https://www.sparkhire.com/>.
- [27] Wepow, "Wepow," Wepow, [Online]. Available: <https://www.wepow.com/en/>.
- [28] Skype, "Skype," Skype, [Online]. Available: <https://www.skype.com/en/>.
- [29] Jibe, "Jibe," Jibe, [Online]. Available:  
<https://www.icims.com/talent-acquisition-software/career-site-solutions/>.
- [30] C. IQ, "Convey IQ," Convey IQ, [Online]. Available: <https://www.conveyiq.com/>.

## Appendix A. System Documentation

### Project Setup

In this section the guidelines for setting up and running the project on a remote server is explained.

#### Required tools and software

- Java 8 or higher version
- MySQL Server
- Maven
- NodeJs
- Nginx server
- Git

Download and install above software tools from their respective sites.

#### Setting up API project

- First download the source code repository from GitHub.
- Configure the MySQL credentials in application.properties file.
- Run **mvn clean install** command
- This command will package the source code and create an executable jar file.
- Go to the target folder as the jar file is created in tha.
- Run **java -jar collabApi-0.0.1-SNAPSHOT.jar** command
- Now the API project is running.
- To make sure it works type <http://localhost:8090/swagger-ui.html> in the browser

#### Setting up Web project

- Download the source code repository from GitHub.
- Run **npm run build** command.
- To test the web application works fine run **npm start** command.
- Type localhost:3000 in the browser.
- If it works then the web application will be displayed in the browser.
- To handle more traffic and improve response time add this to Nginx server.

## Appendix B. Test Results

In this section the results of the test cases are summarized below.

Number of test cases executed	58
Number of successful test cases	58
Number of failed test cases	0
Success ration	100.00 %
Failure ration	0.00%