

Web Based Recruitment Management System

**T I K Vithanage
2021**



Web Based Recruitment Management System

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

**T I K Vithanage
University of Colombo School of Computing**



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: T I K Vithanage

Registration Number: 2017/MIT/083

Index Number: 17550838



Signature:

Date: 28/11/2021

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

Ms. T.I.K Vithanage

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

Certified by:

Supervisor Name: W.V. Welgama

Signature:

Date:

Abstract

Recruitment is an important factor in present companies. This consumes much time, cost and resources to handle this tedious task. Most of the companies use third party generic soft wares to handle these. Unfortunately, these are unable to customize according to the customer needs. Then it is needed to purchase multiple soft wares separately to complete the full process of recruitment. Otherwise they have to use third-party services to assist the process. This might be a risk for the company. Because they have to provide their company details, job role details, salary details and some of the standards of the company that they follow. Any how this occupy number of days to finalize even a one task of whole process. Then it is impossible to complete all the tasks which are expected from the company such as progress reporting, evaluation reporting and other related report generating tasks.

After considering identified issues, this computerized solution was proposed against existing manual process. But it should be accessible to external candidates. Because of that, this Web-Based Recruitment Management System was proposed. This system contains all steps of the recruiting process. Including job postings, candidate handling, candidate evaluation, and recruitment-related reporting. This system has adapted MVC (Model View Controller) architecture and object-oriented techniques. Then UML (Unified Modeling Language) was used to analyze and design the system. For the implementation Typescript and JavaScript was used. Visual Studio Code has been used as the development IDE. Both MySQL and Type ORM have been used to manage the data storing and retrieving and aspects. Chart.js has been used to generate reports.

Finally, the system was effectively developed and successfully implemented. According to the user evaluation feedback, this proposed web-based recruitment management system fulfills and satisfies most of the requirements of the organizations.

0.2 Acknowledgment

In order to achieve something, we have to contribute our own efforts and strength. But as an individual, it is hard to achieve. Because most of the areas which have been considered are beyond the subject areas that we are already engaged in. Sometimes they are beyond our existing knowledge areas. Because of that, this effort would have been in vain without the help and support of these people. They have provided maximum support and effort to make this effort successful.

In my initial consideration, I would like to convey my sincere gratitude to the University of Colombo School of Computing for providing us a great opportunity to improve our knowledge, values, and skills in the IT industry. And also to all staff members who helped individuals like me to successfully step on to this stage with their guidance, suggestions, and support from the beginning of this master's degree program.

Then I would like to express my most gratitude to my supervisor, Mr. W.V Welgama, Senior Lecturer, University of Colombo School of Computing for being available there to give guidance, feedback, advice, and support for me whenever needed. Without his encouragement, it is unable to drive this project into this level of success.

Finally, I would like to convey my heartfelt gratitude to my parents and family members, lecturers, and friends who were with me during the project period by providing advice, guidance, and valuable support to make this effort successful.

Contents

0.1	Declaration	i
0.2	Acknowledgment	i
0.3	List of Acronyms	1
1	Introduction	2
1.1	Motivation	2
1.2	Aims and Objectives	3
1.3	Scope	3
1.4	Structure of the thesis	4
2	Background	6
2.1	Requirement Analysis	8
2.1.1	Methods and techniques used for requirement analysis	8
2.1.2	Data flow diagram normal recruitment management system	9
2.2	Review of Similar Systems	11
2.2.1	Goodfirms	11
2.2.2	Simplicant	11
2.2.3	Recruity	12
2.2.4	Mighty Recruiter	13
2.2.5	Zoho Recruit	13
2.2.6	Comparison of Similar Systems	15
2.2.7	Comparing the Current System with proposed system	16
2.2.8	Software Development Methodology Agile Model	18
3	Design	20
3.1	Introduction	20
3.1.1	Comparison of alternative design strategies	21
3.2	System Design	25
3.2.1	Use Case Diagram for Recruitment Management System	25

3.2.2	ER Diagram for Recruitment Management System . . .	30
3.2.3	Class Diagram for Recruitment Management System . . .	32
3.2.4	Sequence Diagram for Recruitment Management System . . .	33
3.2.5	Data modeling diagrams	35
3.2.6	Data Flow Diagram for Recruitment Management Sys- tem	38
3.2.7	Process Diagram for Recruitment Management System . . .	41
3.2.8	Activity Diagram for Recruitment Management System . . .	43
3.2.9	User interface design	46
3.2.10	Report View	48
4	Implementation	50
4.1	Considering the implementation environment	51
4.2	Justification for the choice of implementation platform	52
4.2.1	Acknowledgment of any reused existing codes / APIs	57
5	Testing And Evaluation	59
5.1	Test Plan and Test Cases	59
5.1.1	Sample Test Cases	60
5.2	Requirement Testing	60
5.3	Software Testing Approaches	61
5.3.1	Unit Testing	62
5.3.2	Integration Testing	62
5.3.3	System Testing	62
5.3.4	UI Testing	63
5.4	Proof of testing of work / Results of work	63
5.4.1	API Testing	65
5.5	Evaluation	67
6	Conclusion	69
6.1	Critical evaluation of the project	69
6.2	Lesson Learnt	69
6.3	Future Works	70
Appendix A	System Manual	74
A.1	Step 1	74
A.2	Step 2	75
A.3	Step 3	75
Appendix B	Design Documentation	76
B.1	Database Design	76

Appendix C User Manual	79
C.1 Logging to the system	79
C.2 Maintain and modify data in profiles	81
C.2.1 Candidate Module	81
C.3 Common buttons and validations used	85
C.3.1 Common buttons used	85
C.4 Error validation for system	85
C.4.1 Common field validations	85
C.4.2 Common message validation for system	86
Appendix D Management Reports	88
D.1 Report Types	88
D.2 Job posting against departments report	89
D.3 Evaluation detail report	89
Appendix E Use Case Narratives	90
E.1 Use Case Narrative for System Login	90
E.1.1	90
E.2 Use Case Narrative for Candidate Profile Creation	91
E.3 Use Case Narratives for Job posting	91
E.3.1	91
Appendix F Test Cases and Evaluation Results	92
F.1 common test cases	92
F.1.1 Navigation related testing	92
F.1.2 Common Functions testing	93
F.2 System Evaluation	93
F.2.1 Questionnaire used	93
F.2.2 User feedback results	93
F.2.3 User feedback results - graph	94
Appendix G Code Listing	95
G.1 Model Layer implementation	95
G.2 View implementation	97
6.3 Glossary	99

List of Figures

2.1	Existing system basic functions	7
2.2	Data flow of normal company	9
2.3	Data flow of normal scenario	10
2.4	Goodfirms	11
2.5	Simplicant	12
2.6	Recruity Recruitment Management System	13
2.7	Mighty Recruiter- Recritment Management System	13
2.8	Zoho Recruit - Recruitment Management System	14
2.9	Comparison of Similar Systems	15
2.10	Agile	18
3.1	Overall Use Case Diagram for RM System	24
3.2	DetailedProposeSystem	26
3.3	Proposed-Admin	27
3.4	Proposed-DeptManager	28
3.5	ProposedHRManagerr	29
3.6	ER Diagram	31
3.7	ClassDiagram	32
3.8	Sequence Diagram for Recruitment Management System	34
3.9	ERD diagram - Part 1	36
3.10	ERD diagram - Part 2	37
3.11	First level DFD for Recruitment Management System	38
3.12	second level of DFD	40
3.13	Candidate Onboarding Process	41
3.14	RMS Process Diagram	42
3.15	RM-Activity Diagram	44
3.16	LoginActivityDiagram	45
3.17	Dashboard UI	47
3.18	User privileges	47
3.19	Candidate UI	48

3.20	Common report view	49
5.1	Requirement verification	61
5.2	UI based testing	63
5.3	System error testing - test cases examples	64
5.4	System login function - test cases examples	64
5.5	Postman Application	65
5.6	example01 API	66
5.7	example02 API	66
5.8	example03 API	67
5.9	example04API	67
B.1	Candidate Profile	76
B.2	Job Category	77
B.3	Job Vacancy	77
B.4	User	78
B.5	Role	78
C.1	URL	79
C.2	Login credentials	80
C.3	System dashboard	80
C.4	System dashboard to select options	81
C.5	Space for personsl details	82
C.6	Space for qualifications	82
C.7	Space for review information list	83
C.8	Space for review information personal view	83
C.9	Space for update information	84
C.10	Commonly used Edit and delete button	85
C.11	Commonly used Add data and Form Reset	85
C.12	Common field validation	86
C.13	Common field validation for success	86
C.14	Common message for delete confirmation	87
C.15	System validation for successful function	87
C.16	System validation for failed function	87
D.1	Report List View	88
D.2	Job posting against departments	89
D.3	Job posting against departments	89
F.1	User evaluation graph	94

List of Tables

4.1	Hardware Requirements	51
5.1	Sample Test Case Example 01	60
A.1	Software Verification for the system	74
A.2	Hardware Verification for the system	75
E.1	Use Case Narrative for System Login	90
E.2	Use Case Narrative for Candidate Profile Creation	91
E.3	Use Case Narratives for Job posting	91
F.1	Common Test Result 02	92
F.2	Common Function Test Result 01	93
F.3	Sample Questionnaire	93
F.4	Sample Questionnaire	94

0.3 List of Acronyms

MVC Model View Controller.	22
ERD Entity Relationship Diagram.	30
GUI Graphical User Interface.	46
NPM Node Package Manager.	53
UI User Interface.	46
UML Unified Modeling Language	23
CSS Cascading Style Sheet).	54
RMS Recruitment Management System	30
DFD Data Flow Diagram	38

Chapter 1

Introduction

This project will be developed a web-based Recruitment Management System for companies which are conducting bulk interviews for their recruitment. When it considers the recruitment process it needs more attention. Because most of the companies have branches, sub-branches, and other small centers. But their recruitment process finalization is done through the main branch. Then it needs to interconnect all these activities through the main branch. Then all the recruitment activities are done through the HR department of the main branch or the Head Office. Among all the activities done through the HR department handling interviews and evaluating their performance is a tedious task. Once they have to interview more than 100 candidates. And it takes time to evaluate them. Because of that, candidates have to face number of interviews and they have to face number of interviews. Sometimes talented candidates may lost because of this routine.

1.1 Motivation

Companies need to recruit employees to continue their day-to-day works. Day by day technology changes. Then the needed talents also get changed. Because of that, it is necessary to identify the most appropriate candidate as an employee among most candidates. It needs good selection methods. The problem is, in most cases, there is a large number of applicants. It may be hundreds, thousands, or more than small numbers. Evaluating these applications one by one is time-consuming. Apart from that, it may avoid the most appropriate candidates as the application limit exceeds when selecting them manually. Human errors may occur. As same it is possible to cheat on

selection as lack of transparency. In most cases, some may avoid applying as they have difficulties in normal posting methods. But if it is possible to log in to some system and continue, it may get more popular and helpful.

1.2 Aims and Objectives

This application is trying to simplify the candidates' handling process of a manual recruitment management system. Through this application, the HR department can efficiently handle the candidates and it is easy to track the performances of the candidates. This application will achieve the following objectives in order to improve the state of the current operations.

- This project's main objective is to simplify the recruitment handling process of the HR Department.
- Effectively and efficiently handle the candidate details and other related documents of the candidates.
- Provide various analysis reports regarding candidates and their performances according to the feedback given.
- Get the support from all departments to evaluate the candidates and their performances.
- Keep track of each and every candidate and help to improve their abilities and identify the weaknesses and encourage them for further improvements.
- Identify the outstanding individuals.

1.3 Scope

Maintains a system for recruitment process is more helpful to handle recruitment process efficiently. To achieve this process it has identified these steps as scope of the project.

- Provide facility to register for the recruitment management system
- Candidate application insertion with personal details and qualifications
- Facilitate to review provided information

- Approve applied candidates
- Evaluate applied candidates
- Make job posting
- Add Departments and Interviewers or the supervisors
- Generating recruitment related reports
- Manipulate and retrieve Candidates, Employees, Interviewers and system users details

1.4 Structure of the thesis

Dissertation is the document which contains overall information of the project in chapter wise. This dissertation contains six main chapters followed by reference and appendix.

- Chapter 01: Introduction This chapter provides an overview of the content to motivate reader. Introduction to project, scope and objectives included here.
- Chapter 02: Background This chapter is written to provide an idea about background of implementing the project. Other than that, chapter provides a review of similar systems, tools and technologies which used in the system implementation.
- Chapter 03: Analysis and Design All information and requirement gathering techniques, functional and non-functional requirements which need to specify software requirement specification and analysis will be included in this chapter. Design chapter includes methodologies, tools and techniques used to design the system. Design selection is based on sound justifications and user interface, database and modules of system will be designed.
- Chapter 04: Implementation This chapter describes how specifications were gathered (in analysis and design phases) are converted into executable program. Furthermore chapter incorporates with Implementing technologies and concepts.

- Chapter 05: Testing and Evaluation This chapter describes how the system was tested by using various testing methods. It also reported errors and how to overcome those errors and how to modify the system. And it implies the security options used and its functionalities.
- Chapter 06: Conclusion This chapter include the all summarize details of the projects and also include findings and lessons learned during the project with further improvements of the project.
- References All the referred books, URL references and other materials will be included in this section.
- Appendix This include further details and further proofs and supplementary readings for the related chapters.

Chapter 2

Background

Nowadays most of the HR Departments in companies using manual methods of evaluating candidates. Still, they are using Microsoft Excel documents based calculations to evaluate candidates. This method is time-consuming. And does not contain accuracy for considerable points of acceptance and transparency. Because of that, they need to have a modified web-based system to manage their recruitment process. As I suggested this will help to handle candidates efficiently and effectively until they complete their recruitment process. Human Resource Managers are responsible for positioning the right candidate at the right place. It involves assigning specific tasks and responsibilities to an individual. Because giving proper placement is important to both employee and the organization. Proper placement helps to improve employee morale. Then placement ends the human resource recruitment function. Because of that Recruitment Management Software helps an organization with human resource planning, specifying the positions to be filled and the skills required for these positions. Helps to identify the list of unfilled positions. Helps to identify the specific tasks, duties, and requirements of these positions. Online recruiting has become one of the primary methods employed by the human resources department to identify potential candidates for available positions within an organization. It represents the first contact between the prospective employee and the organization of interest. It is the process of looking out the right people for the right position at the right time. This automated system enables prospective employees to apply for jobs online. Which saves costs associated with printed applications and helps to keep the hiring process much more organized. And also it helps to streamline the recruitment process by identifying the posting of jobs, and

automatically sorting and filtering applications to find the best prospects. Apart from that it will help to keep the transparency of the recruitment process. Because of these it is necessary to have such an automated system to make HR recruitment process much easier. Existing recruitment management systems which are available online are more general. They have only the common features such as posting advertisements, applying and make responds. [17] [18] But in this system it is unique because it has its own features. Such as initial stage selection, organization related job posting and only most appropriate candidates can be selected. And this will provide ability to evaluate the talent of selected candidate. Because of this, this will be a unique system from other systems.

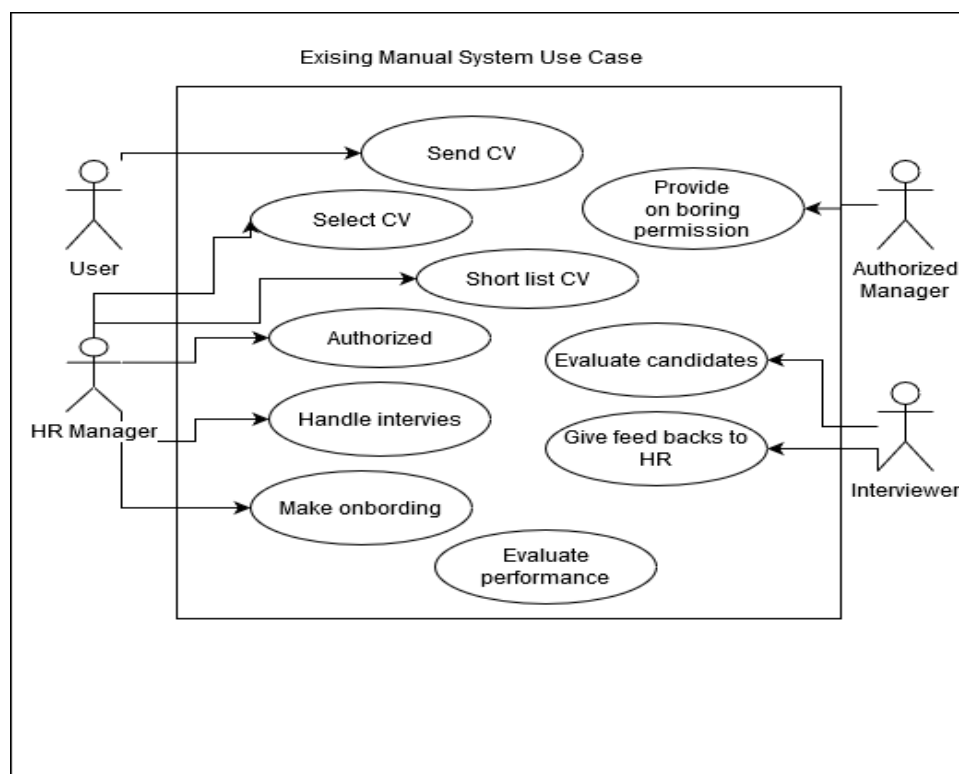


Figure 2.1: Existing system basic functions

2.1 Requirement Analysis

2.1.1 Methods and techniques used for requirement analysis

- Make interviews and discussions with related people in the organization.
- Use some historical figures and documents related to the recruitment management systems in some organizations.
- Reading and analyzing documentations related to the existing systems publicly available.
- Referring to the existing systems.

2.1.2 Data flow diagram normal recruitment management system

Data Flow Diagram for existing manual system

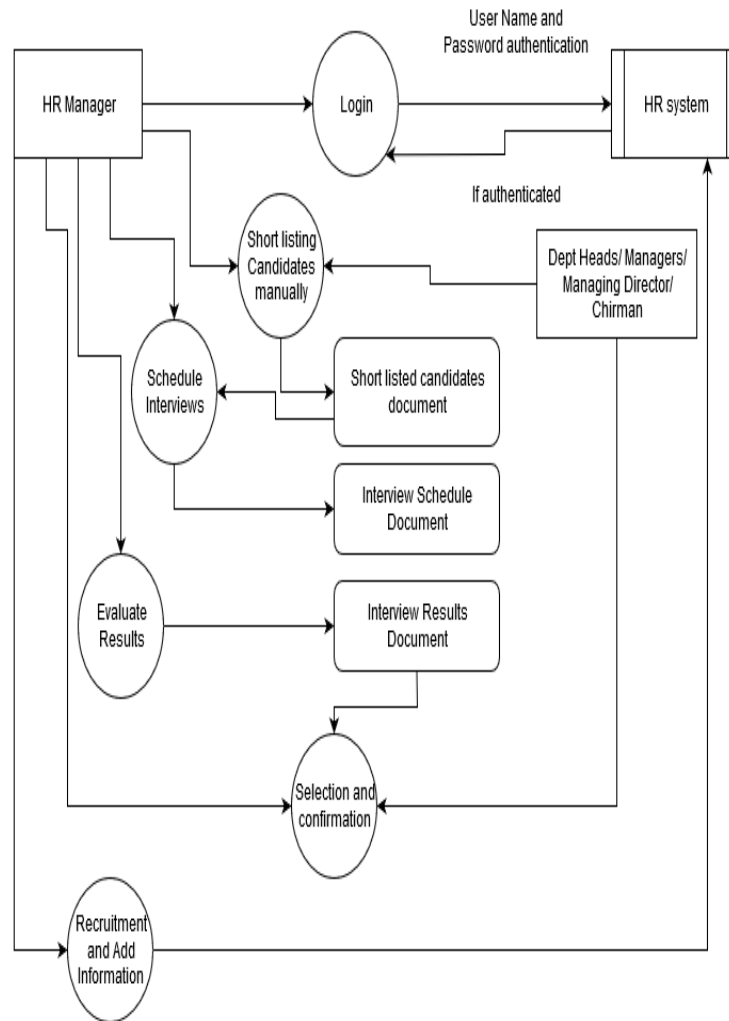


Figure 2.2: Data flow of normal company

Data Flow Diagram for existing manual system

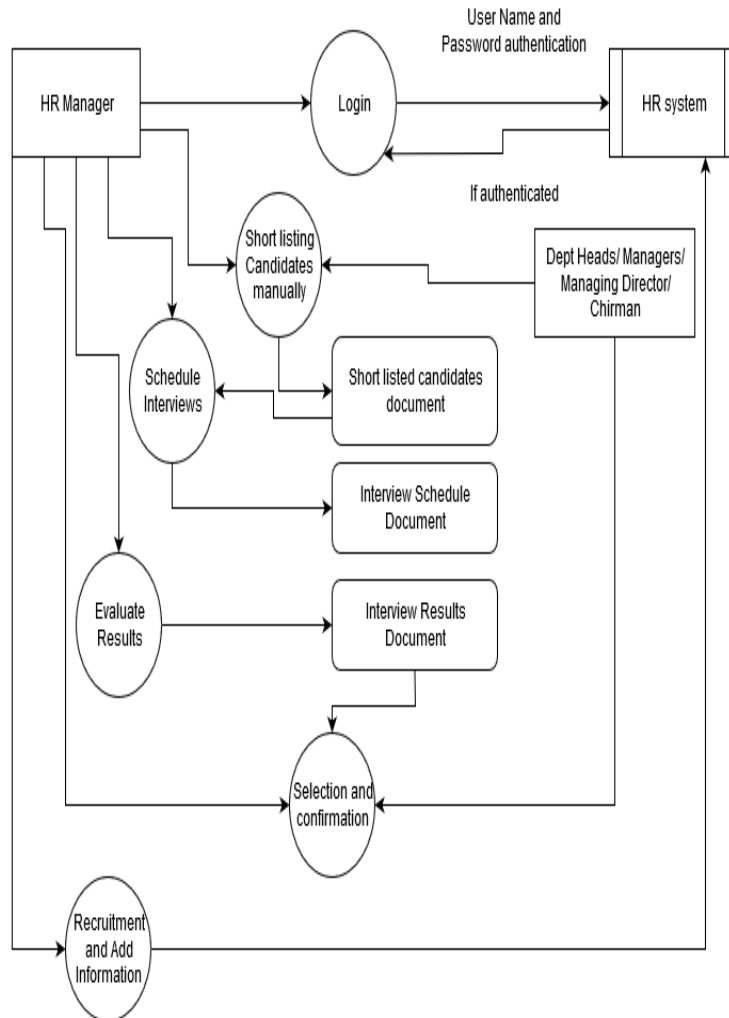


Figure 2.3: Data flow of normal scenario

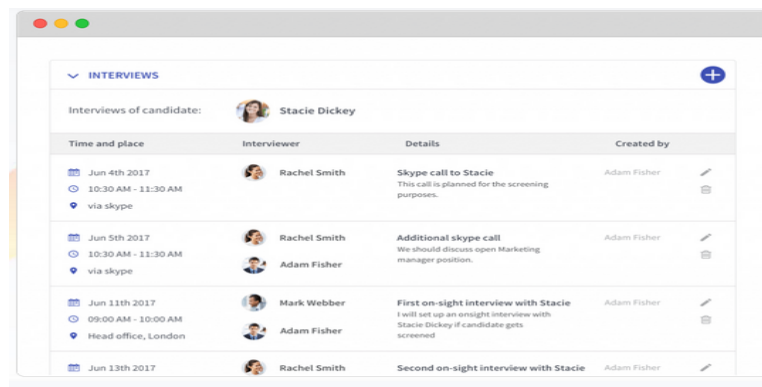
2.2 Review of Similar Systems

Day by day technology getting improved. But observing existing system it is possible to obtain more experience on how our proposed system should be developed and how required features should be presented to the user.

2.2.1 Goodfirms

In this also contain number of open source most appropriate systems according to the requirements. But it is unable to find all the requirements in one system. It is needed to get more than one system. And then it is needed to inter connect them to fulfill our requirements. On the other hand the technologies used are not matching with the existing technologies that the business contains. Also it is unable to customize the system according to the priorities defined by the business.

In this system: it contains simple dashboard and the very common features of adding CV and viewing different job opportunities. But it does not contains much details on selection process or recruitment process. It just consider on the collecting CVs. But in our suggested system, it just collect the vacancy related information form the candidate and it helps to improve the system efficiency and avoid unnecessary complications. [17]



The screenshot displays a web application interface for managing interviews. At the top, there is a header with a dropdown menu labeled 'INTERVIEWS' and a plus sign icon. Below the header, the candidate's name 'Stacie Dickey' is shown with a profile picture. The main content is a table with the following columns: 'Time and place', 'Interviewer', 'Details', and 'Created by'. The table contains four rows of interview data.

Time and place	Interviewer	Details	Created by
Jun 4th 2017 10:30 AM - 11:30 AM via skype	Rachel Smith	Skype call to Stacie This call is planned for the screening purposes.	Adam Fisher
Jun 5th 2017 10:30 AM - 11:30 AM via skype	Rachel Smith Adam Fisher	Additional skype call We should discuss open Marketing manager position.	Adam Fisher
Jun 11th 2017 09:00 AM - 10:00 AM Head office, London	Mark Webber Adam Fisher	First on-sight interview with Stacie I will set up an on-sight interview with Stacie Dickey if candidate gets screened	Adam Fisher
Jun 13th 2017	Rachel Smith	Second on-sight interview with Stacie	Adam Fisher

Figure 2.4: Goodfirms

2.2.2 Simplicant

This is one of the most appropriate systems that can be identified for the requirements identified. These contain number of features that is needed.

But it is a commercial version and it allows limited number of candidates for one time. And it is unable to customize features according to the needs of the business.

In this system most of the features have been added. But it does not allow facilities to evaluate number of candidates in a little amount of time. As same it does not provide any support for recruiting and interview management activities. They work as third party people for on-boarding process. This provide pity much easier but it does not provide much effective way to solve these at once. [17]

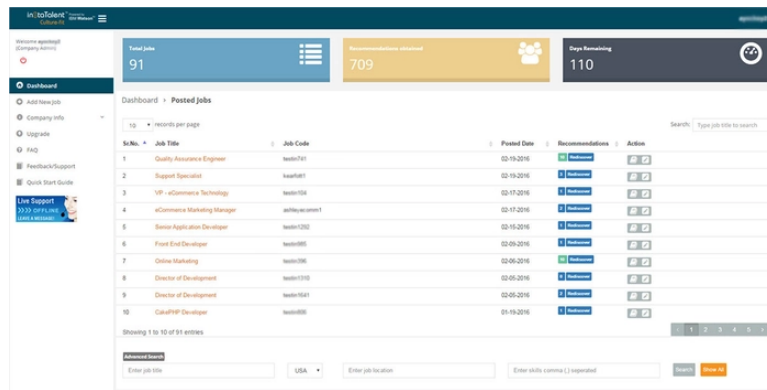


Figure 2.5: Simpicant

2.2.3 Recruity

Recruity is Recruitment management system which provides the facilities to post vacancies, providing incoming responses, and the posting of jobs on the web and social media. This contains optimization process. Then this provides synchronization of the data with smartphones and tablets.

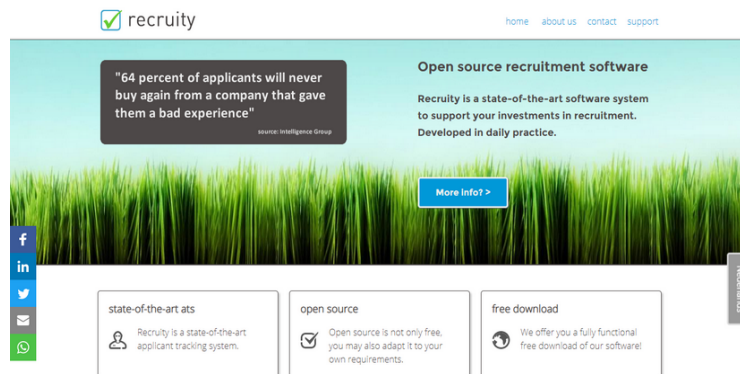


Figure 2.6: Recruity Recruitment Management System

2.2.4 Mighty Recruiter

Mighty Recruiter is your modern-day recruitment solution that covers the entire candidate recruitment, matching, hiring, and management process. And the best part is that it can be used by both small- and large-sized companies.

In addition, this free recruitment software posts jobs for reputable job boards. And both free and paid. It ensures that the job requirement reaches to the masses and the company is able to find the right fit. [26]

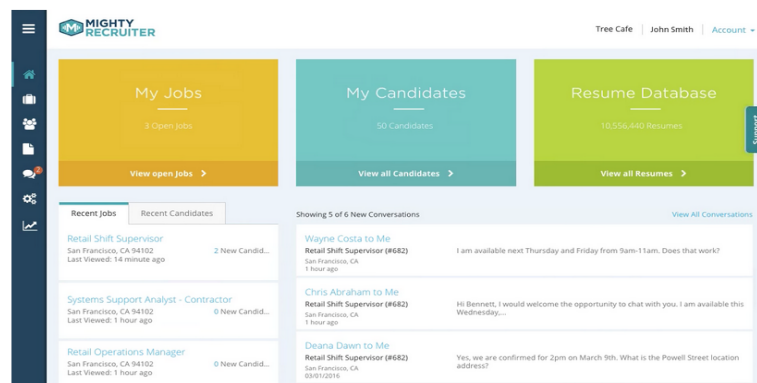


Figure 2.7: Mighty Recruiter- Recruitment Management System

2.2.5 Zoho Recruit

One of the most popular free recruitment software. Zoho Recruit has a lot of features that make sure that all your recruitment processes are streamlined

and you don't miss a good hire. They make it possible by offering a user-friendly interface combines with robust functionalities.

In fact, the software is equipped with security options so as to protect your data from prying eyes. You can even get insights into the candidate interview status or job opening status with their advanced analytic feature. [26]

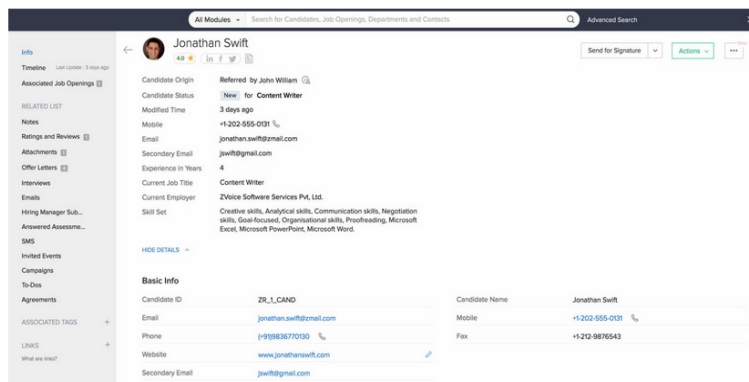


Figure 2.8: Zoho Recruit - Recruitment Management System

By considering these systems, it is identified these common issues :[22]

1. User friendliness and effective use

When considering and analyzing these systems, it is identified that they are more complicated for the people who does not have much IT knowledge. Because of that it is identified that the suggested system should be more user friendly. In this scenario, interfaces should contain less data to input from user end and most data should be provided from the system. Then it is decided that more accuracy could be achieved and user may not get panic.

2. Interview and recruitment processing

In most cases there it does not contain a one system to handle vacancies, accept CVs and applicants, evaluate details and manage interviews. Because of that it is suggested to build up a system to achieve all these issues.

3. Selection

In most cases, selection should be done in manually. Because of that it is needed to print CVs and identify the qualifications one by one. Then

it consumes cost, time and it may miss the correct candidates. Because of that it is decided to develop this system with special selecting method.

4. Effectiveness

When it contains different systems to manage HR, interviews and recruiting. But in most cases they are not interconnected. Because of that it may lose some important data. And it will lose the transparency. Because of that it is suggested to have special method to interconnect each and every related data.

2.2.6 Comparison of Similar Systems

Functions and Features	Simplicant	Goodfirms	Recruity	Mighty Recruiter	Zoho Recruit
Candidate Application Tracking	Yes	Yes	Yes	Yes	Yes
Background Verification	Yes	Yes	Yes	Yes	Yes
Recruitment/ Interview Management	No	No	Yes	Yes	Yes
Selection/ Hiring	No	No	Yes	Yes	Yes
Job posting	Yes	Yes	Yes	Very slow	Yes
Ontime evaluation	No	No	No	No	No
Ability process large companies	No	No	No	Yes	No

Figure 2.9: Comparison of Similar Systems

2.2.7 Comparing the Current System with proposed system

Current system contains paper based manual system. It contains Excel based calculations and evaluations. In proposed system it is decided to implement automated system to calculate and evaluate the candidate at the same time. This will help to reduce cost and time of both inspectors and candidates. Then the candidate related job posting, job applying and selection related all functions are interconnected through this system. Because of that it helps to achieve goals, efficient and effectively. For this implementation it is needed to identify the new requirements for the proposed system. There it was followed, Observation, Document Analysis and Interviews to collect relevant information.[21]

Functional Requirements

- **Manage Job Posting**
This helps to store large amount of items with their information and easily search, edit and delete related jobs.
- **Manage Interview**
- **Manage Users**
This will facilitate storing and managing user details with personal details including their job related details.
- **Manage Departments**
This helps to manage individual department and its job requirements.
- **Manage Evaluations**
All individual evaluations and details related to each candidate will be stored and managed.
- **Manage Interviews**
Since interviews have to be conducted and its all details will be stored and retrieved.
- **Manage Candidates**
Main characters of this system are candidates. Their details will be stored and managed.

Non Functional Requirements

Non functional requirements describe the features which should have to improve the quality of the system. Non-functional requirements are difficult to work with the functional ones because they depend on each user's perspective and preferences. But, the userfriendliness of the system depends on how well these nonfunctional requirements are met. Following is the list of nonfunctional requirements aim to offer in the proposed system.

- **Security**

Since system will store sensitive details such as personal employee details, there are security measures in place such as user access management, privileges control and unauthorized accesdes prevention.

- **Usability**

System userinterface contain easy to navigate menus, dashboard and a clear way to indicate the current location user is in. This maximizes the usability of the system allowing user to switch between modules seamlessly.

- **Accuracy** Proposed system has mechanisms in place to handle transactions with minimum error probability allowing users to perform their tasks and retrieve information with high accuracy.

- **User-Friendliness** Userinterface of the system is very simple and easy to navigate allowing even a new user to learn to perform tasks very quickly.

- **Efficiency and consistency** System has very fast response rates and accurate database management mechanisms preventing issues such as data inconsistencies and slow data retrieval.

2.2.8 Software Development Methodology Agile Model

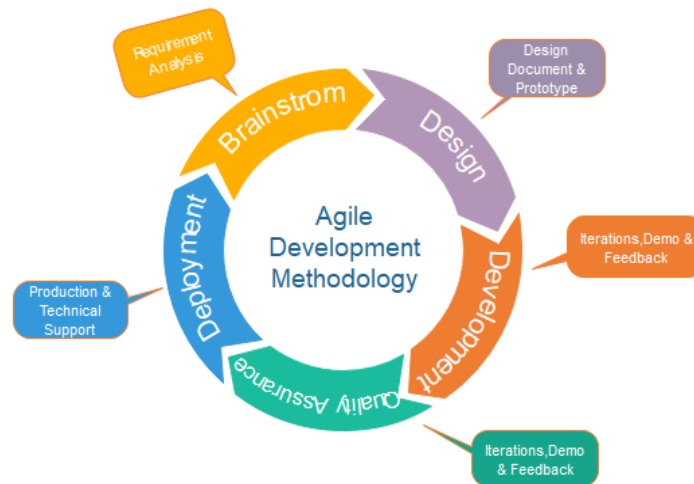


Fig. Agile Model

Figure 2.10: Agile

Agile process model refers to an approach based on iterative development. This allows to break task into smaller iterations. These are called system backlogs and each task contains its initial time frame. Both scope and the duration of the each iteration should be clearly defined. These small tasks probably contains one to four weeks of duration[2].

Phases of Agile Model

- **Requirements analysis** In this initial phase it is needed to define the requirements. Then it is needed to explain the business opportunities and identify the duration of the system completion. Again it helps to identify the effort needed for.
- **Design** This phase includes the design of requirements in a way where majority could understand. There it could be used UML diagrams or User flow diagrams.
- **Development** After defining the requirements, development phase begins. The aim of this phase is to provide a working product.

- **Testing / Quality assurance** In this testing phase, development will be compared with the user requirements and identify the quality of the system. Then it will consider whether to continue with the same task or come up with a new task.
- **Deployment** After testing, if it is issue free, then it will be deployed in user environment.
- **Feedback** After releasing to the user environment, it will consider about the user feedback and identify the success of the development.

Justification for the choice of the development life cycle

After considering the existing systems and needed requirements of this proposed system, an agile development approach was selected as development method. The most prominent reason for this is because the requirements are unclear and it could change. Apart from that modules contains number of sub modules and then modules contain complexity. Then it helps to make changes.

This framework provides a flexible environment for developing the system with continuous user change requests. Then it will be a more flexible and efficient way to develop.

If it consider about the waterfall methodology, it needs to capture all the requirements as early as possible. Then all the captured data should be clear. Then it should define all requirements until the project completed. Other wise it is unable to fulfill the clients expectations. In this scenario it is used Agile methodology as it is needed to develop modules and units separately when ever they are needed. Then it provides the ability to overcome the difficulties face while developing the system.

[20]

Chapter 3

Design

3.1 Introduction

After collecting the requirements, it is essential to design the system to achieve success in the final system. Requirement gathering is the most challenging part of a project. And it is able to confirm and finalize the system needs through this phase. Here, it is possible to consider different types of design methods to identify and finalize the structure of the system. This will be the core of the system that is going to complete step by step. In this chapter it is discussed the system architecture by using different types of UML diagrams. Each and all diagrams contains its own value of expression. These design architectures will give a clear image and view of the system that is going to develop. This is important because no one have an idea apart from the BA's, Business owners and the Architectures of the system. As a result of these introductions will help to simplify the meshed requirements. Apart from that, it helps to simplify the verification and validation of the system. This will helps both customer and people who involve in the development process. Finally it will help all the parties involve in the development process to overcome most of the problem which are to be occurred and which will be occurred in the future.

After performing the analysis phase, it is considered and identified the functional and non-functional requirements. But it is necessary to have a blueprint for further coding and implementations. In this chapter will discuss the chosen architectural design and user interface design strategies of the system

will also be explained in detail.

For the better output of the system, it should be documented, measurable, testable, traceable, related to identified needs of functionalities, and defined to a level of detail sufficient for system design. [2]

First the requirements are collected. For this it is using various methods. After that it will be classified into two categories. They are Functional Requirements and Non Functional Requirements. Then it will identified up to what degree they could implement the system.

3.1.1 Comparison of alternative design strategies

Maintain new system for some processes along side existing file based system

With this strategy, the new system could help to improve the job posting, candidate evaluation, and selection. But this will result in the efficiency and effectiveness of the recruitment process together with HR activities. As same this will help to save time and keep trustworthy in between both parties.

Use and existing free recruitment management system software

It is common to know that, there are free soft wares and systems that could be used. But the problem is they are unable to customize according to the need of each and every person who is using. In general, it could be defined as they are unable to customize. They are generic and could be used for more general purposes.

Justification for the selected design strategy

This proposed system is being developed as a web-based system because the users of this system are outsiders. And according to the current situations, it should be able to access it, anyone, from anywhere. The most appropriate way is accessible via the internet. Because of that, it is decided to build this as a web-based system. In this scenario, candidates are the target users. And they should be able to access this any time anywhere without having

any disabilities. Because of that, the web-based approach is ideal for this scenario.[21]

Among all it is provide number of benefits from this web-based solution:

- Reduce cost and time use for selection and evaluation processes
- Access from any time, any where
- Platform independent nature of webbased solutions (Since only real requirements are a web browser and internet access).

Architectural design of the system

The proposed system is being implemented using the Model View Controller. (MVC) MVC architecture. This pattern isolates the representation of information from user interactions. This is a widely used Object-Oriented design pattern for developing web-based and desktop applications.

Since this model provides isolation between different components based on their usage, it is very suitable for the development of this solution. Furthermore, the nature of this model allows developers to focus on important parts of the system without getting distracted by unnecessary details. This model consists of the following three major components.

- Model
This represents the shape of the data and business logic. It maintains the data of the application. Models are being used as objects to retrieve and store information to and from a database
- View
This is a user interface. View display data with the help models to the user and also enables them to modify the data
- Controller
This handles user requests and logical operations. Typically, the user interacts with a View, which inturn raises appropriate URL request, this request will be handled by a controller

MVC modal is popular due to its powerful nature of separating data and its representations. Simply put, the same data can be shown to two different

users in two different ways with the help of this modal.

When modeling system operations and business processes, UML Diagrams were used. Unified Modeling Language (**UML**) UML is a set of modeling diagrams that allows looking at a system in multiple angles. Graphical symbols provided by UML allows to modal the proposed system in a way that can improve the understand-ability between designers and developers, sometimes end users.

Proposed system UML

Use Case Diagram for propose sysetem

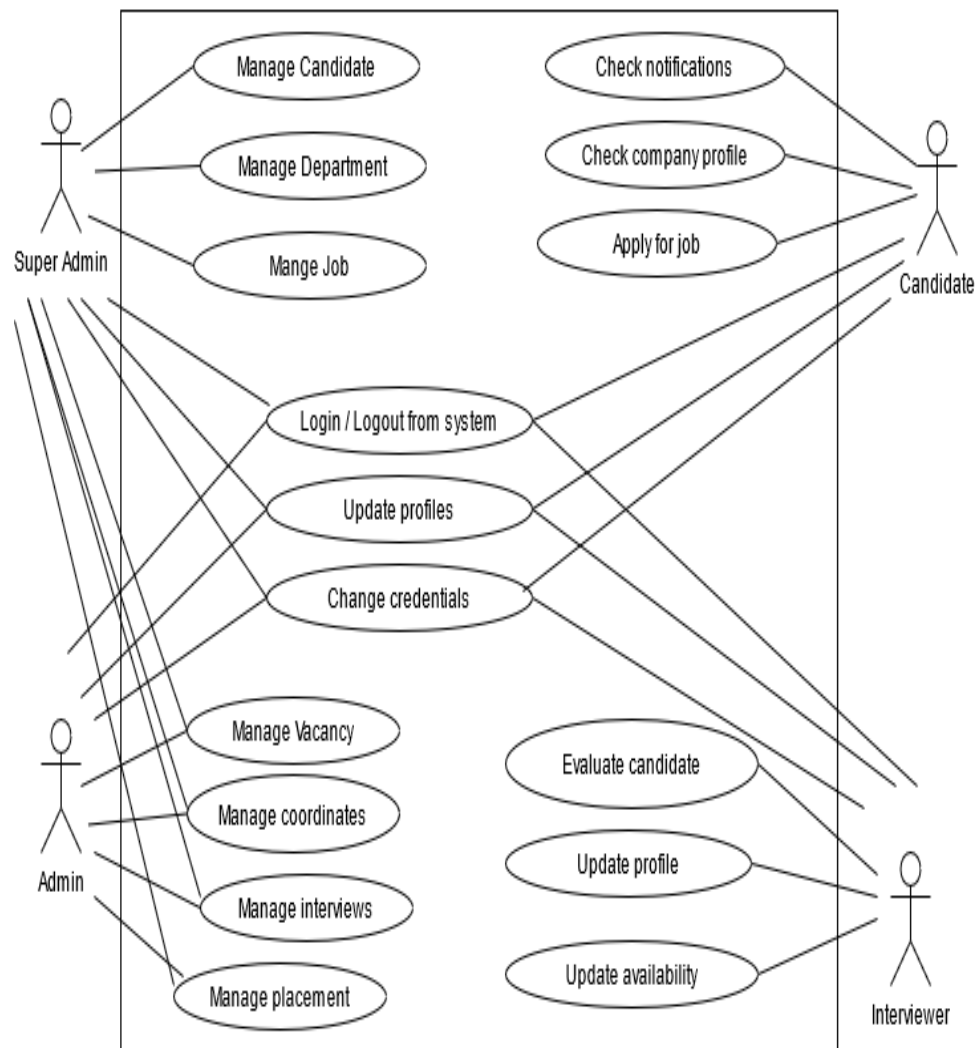


Figure 3.1: Overall Use Case Diagram for RM System

3.2 System Design

3.2.1 Use Case Diagram for Recruitment Management System

Use Case Diagram is a graphical indication of the interaction among the elements of RMS. It represents the methodology used in system analysis to identify, clarify and organize system. The main actors of Recruitment Management System. The main actors of RMS are: Super Admin, System user, Candidate and coordinators: who perform the different type of use cases such as Manage placement, Manage candidates, Manage interviews, and manage users etc. Manage users and full Recruitment Management System operations. Major elements of the UML use case diagram of RMS are indicate in the diagram as below.

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. [15]

Overall Use Case Diagram for RM

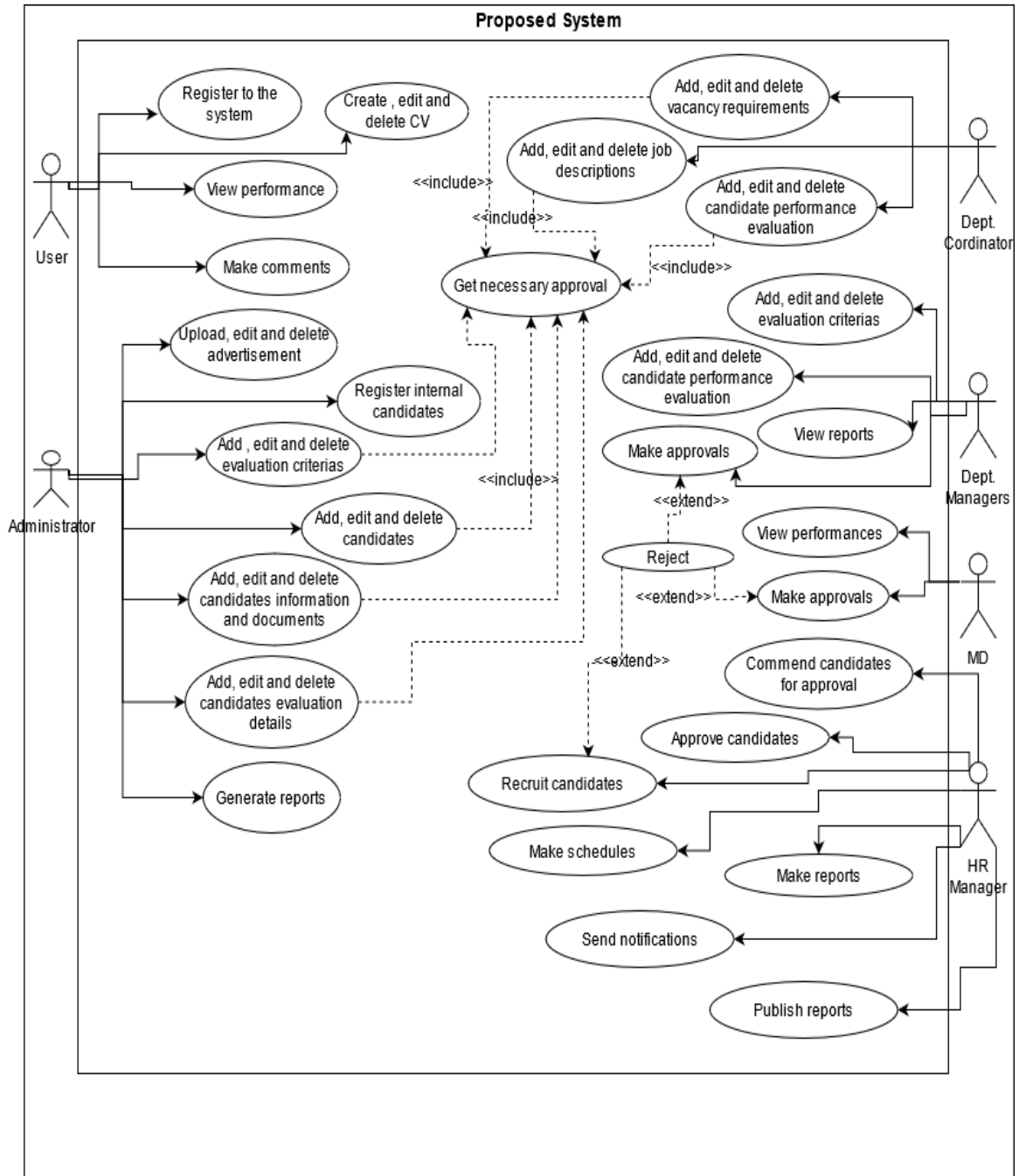


Figure 3.2: DetailedProposeSystem

Proposed system System Admin

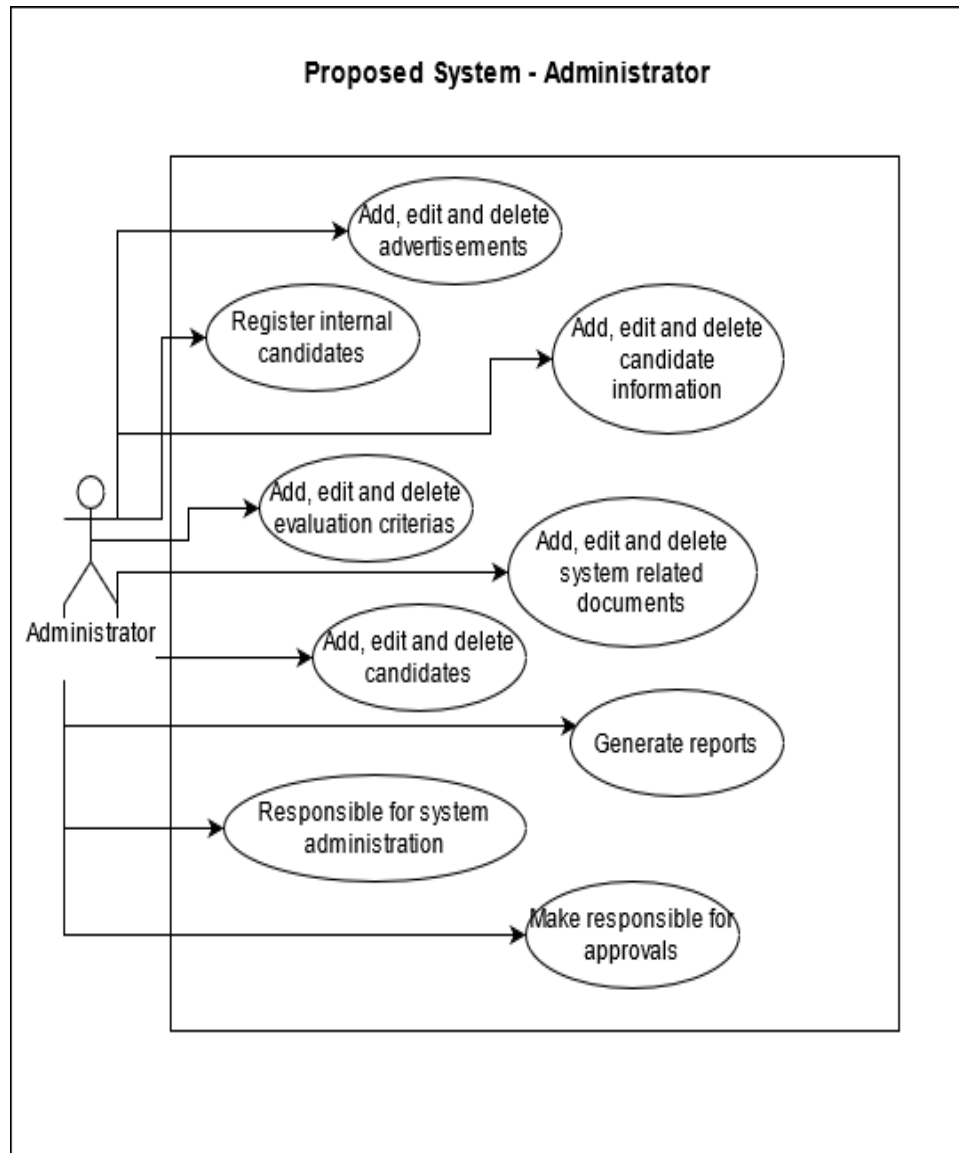


Figure 3.3: Proposed-Admin

Proposed system Department Manager

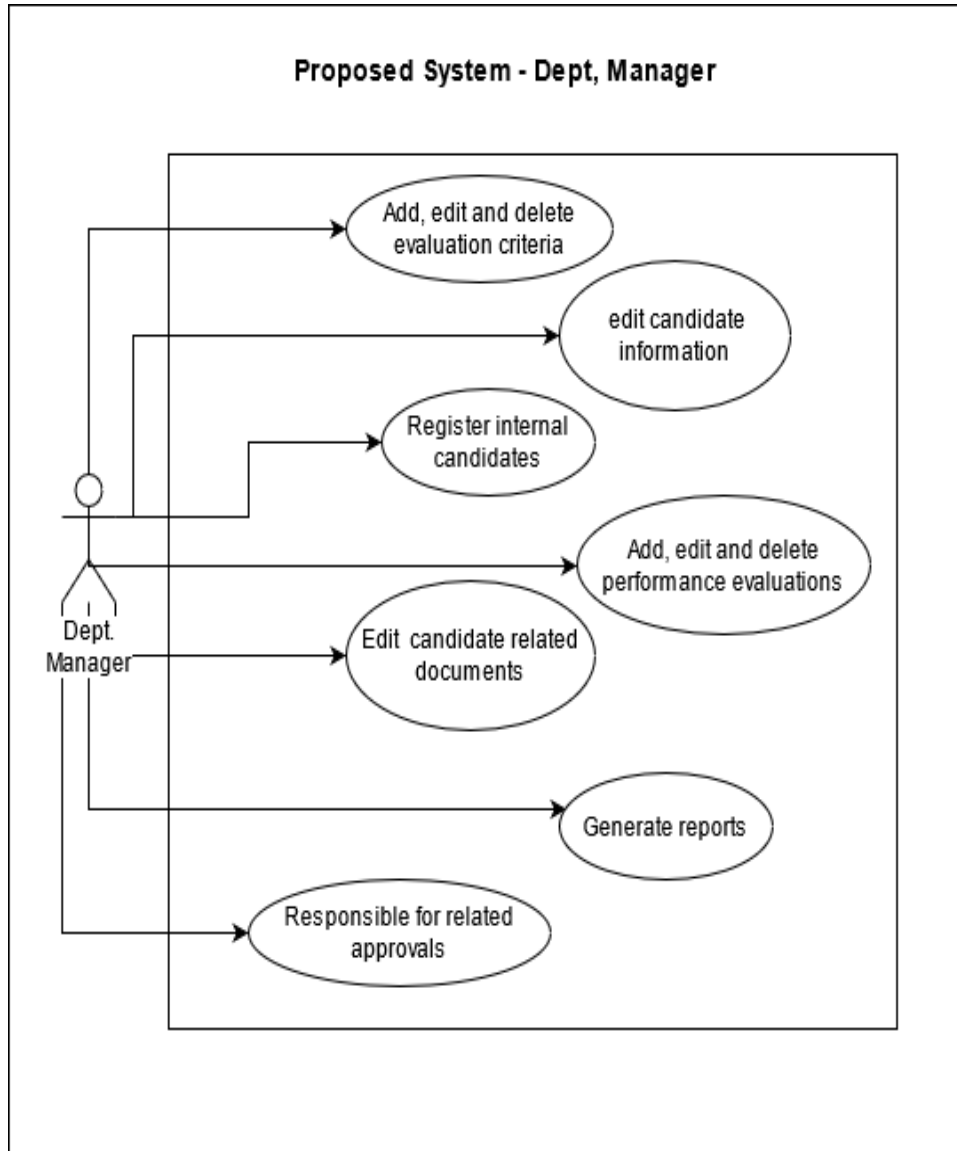


Figure 3.4: Proposed-DeptManager

Proposed system HR Manager

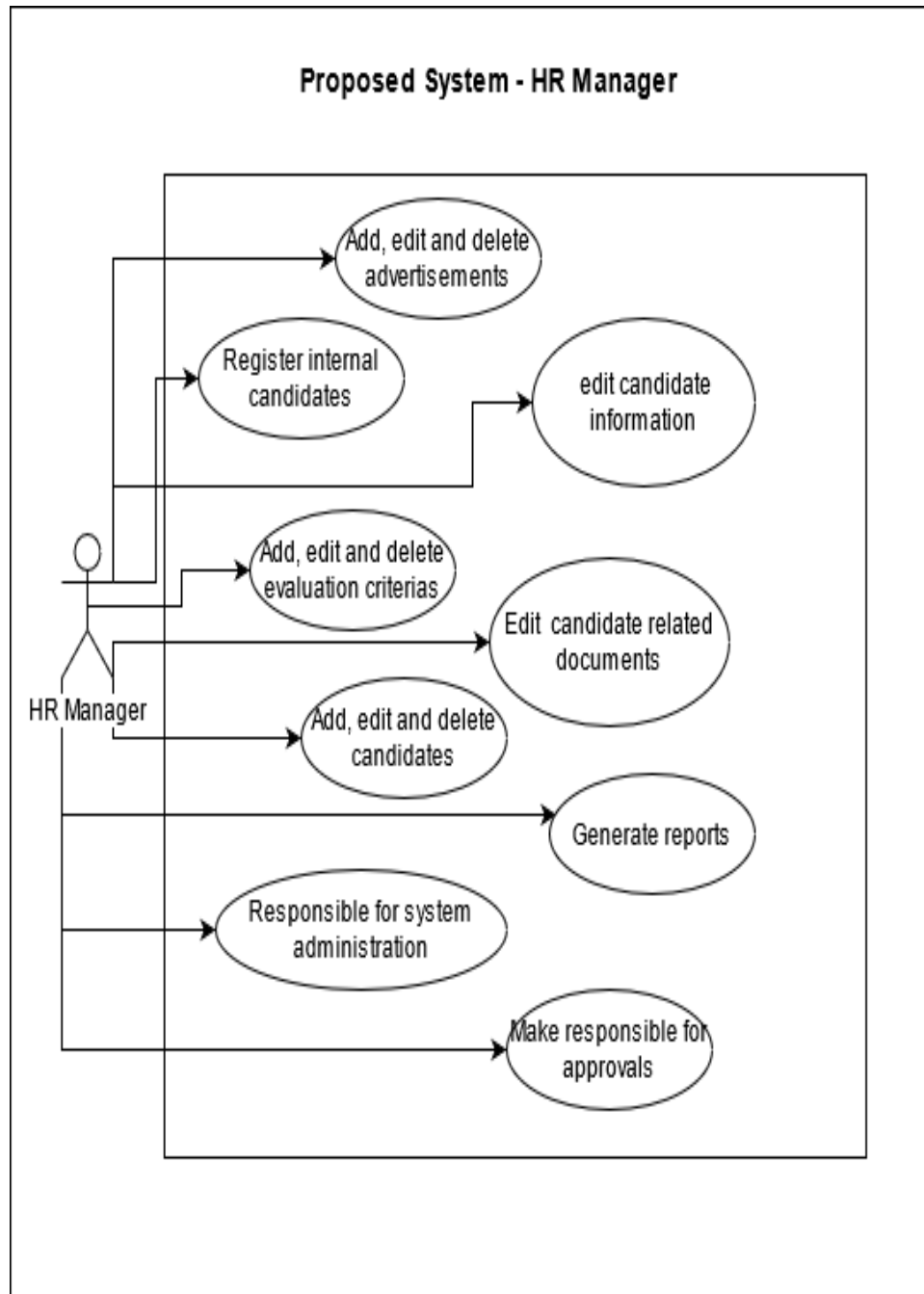


Figure 3.5: ProposedHRManagerr

3.2.2 ER Diagram for Recruitment Management System

Entity Relationship Diagram. (ERD) ERD represents the model of Recruitment Management System (RMS) RMS. The entity-relationship diagram of RMS shows all the objects of database tables with its relationships. It is used to structure data and to define the relationships between structured data groups of RMS functionalities. The main entities of the Recruitment Management System are Department, Candidate, Jobs, Placement and interviews.

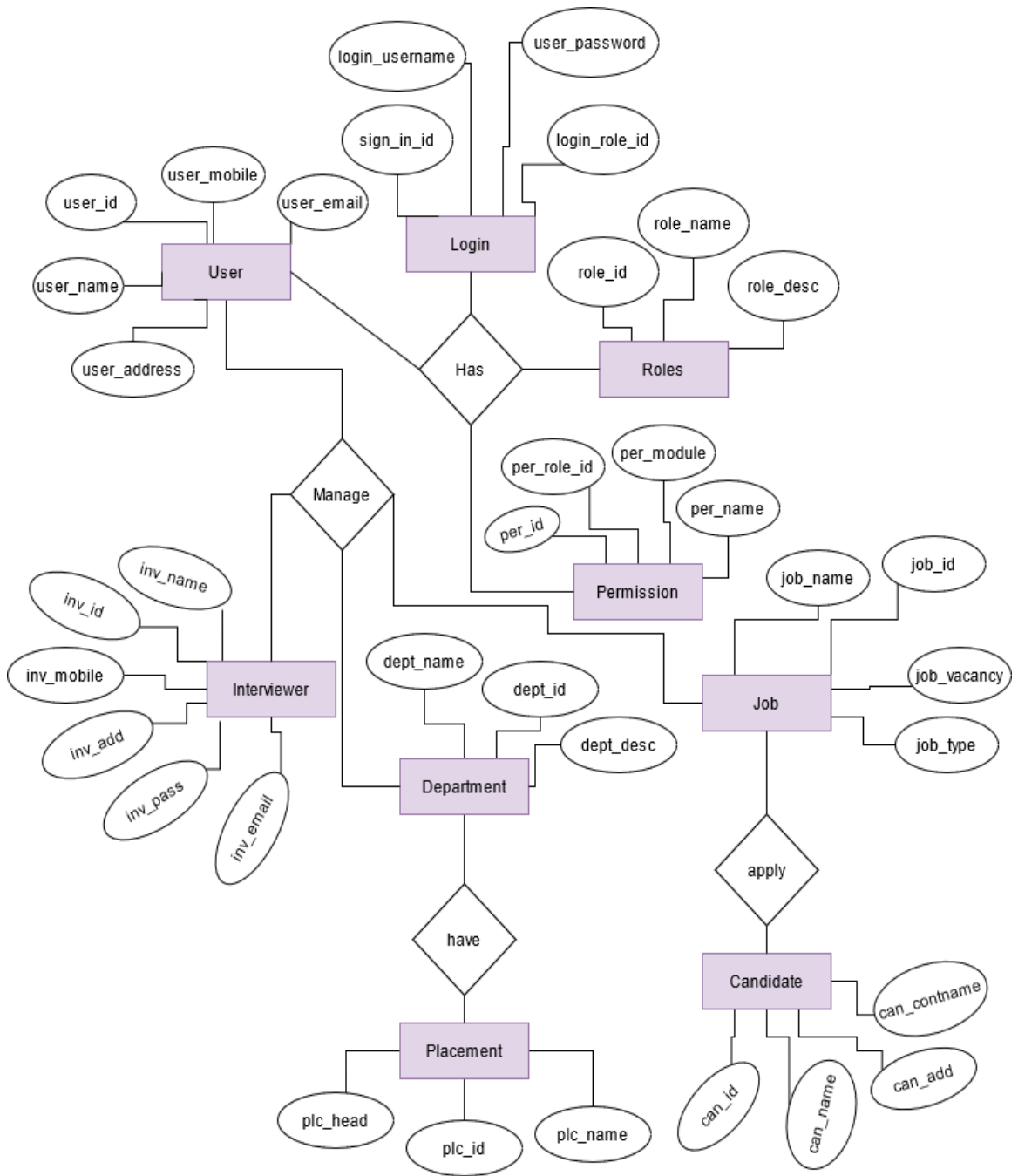


Figure 3.6: ER Diagram

3.2.3 Class Diagram for Recruitment Management System

Class Diagram for Recruitment Manage System

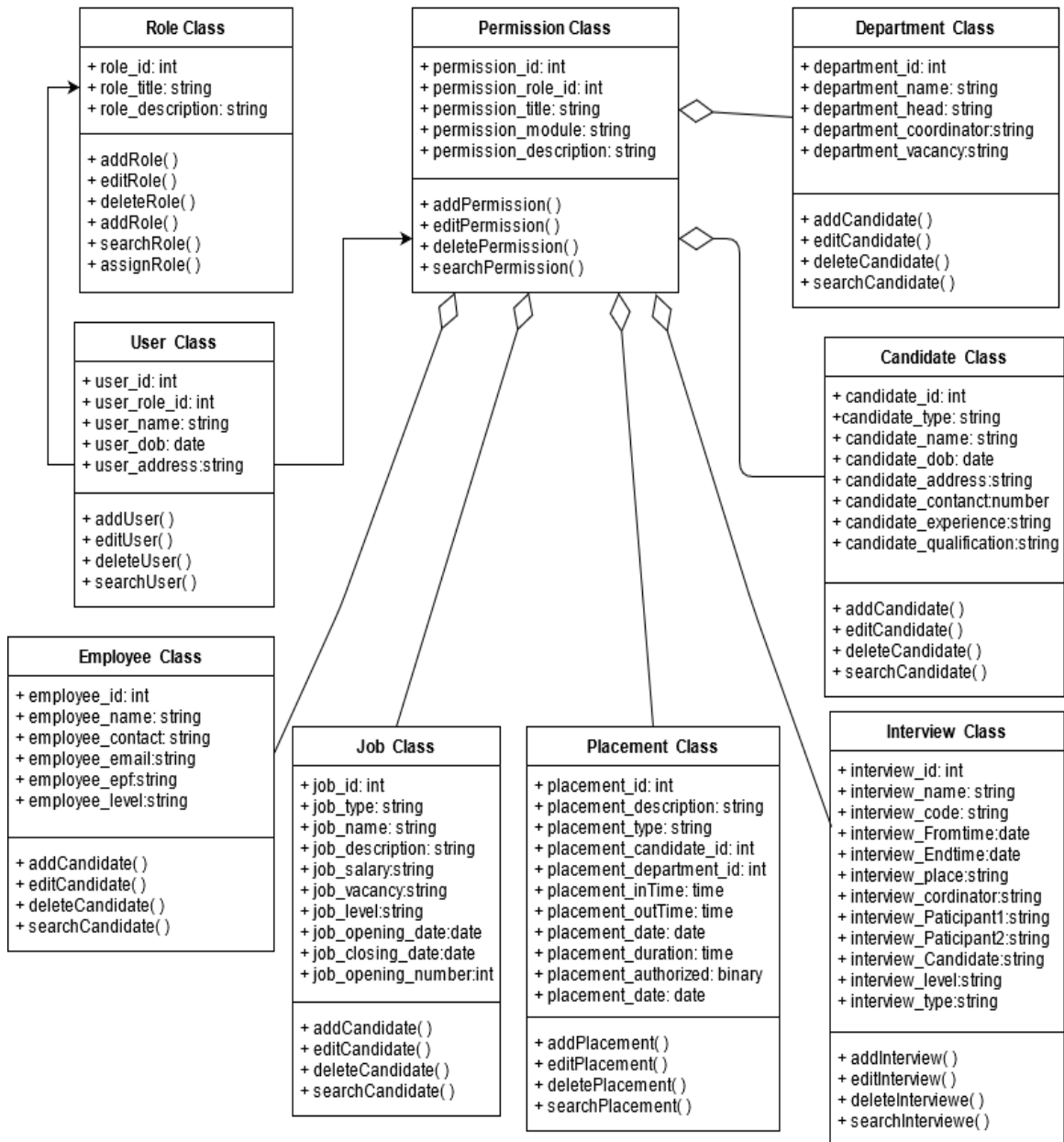


Figure 3.7: ClassDiagram

Recruitment Management System Class Diagram describes the system related classes. It contains the details about the structure of the system databases. It provides the details of classes, attributes and behaviors of the system. Then it provides the relationship between each object. As in diagram it provides Department class, Candidate class, Job class, placement class and interview class. In each class it contains separate two parts in the diagram. First one for its objects and the second one is for its behaviors. Objects describes the parameters which are using to communicate with each object and behaviors. Behaviors focus on the methods that are going to use.

3.2.4 Sequence Diagram for Recruitment Management System

This sequence diagram shows the interaction between the objects of interviews, placement, candidate, job, department. The instance of class objects involved in this UML sequence Diagram as follows:

- Interviews object
- Placement object
- Candidate object
- Job object
- Role object

In the login process it has it own sequence. This describes where admin powers of login in their account using their credentials. After login user can manage all the operations on candidate, interviews, placement, department and job. All the pages such as placement, department, placement and job are secure and user can access these page after login. The various objects in the department, candidate, interview, placement and job page interact over the course of the sequence and user will not be able to access this page without verifying their identity. In this diagram it mainly focus on the interaction between the objects of interviews, placement, candidate, job, department. The instance of the class objects involved in this UML sequence diagram mentioned above.

Sequence diagram of the Recruitment Management System

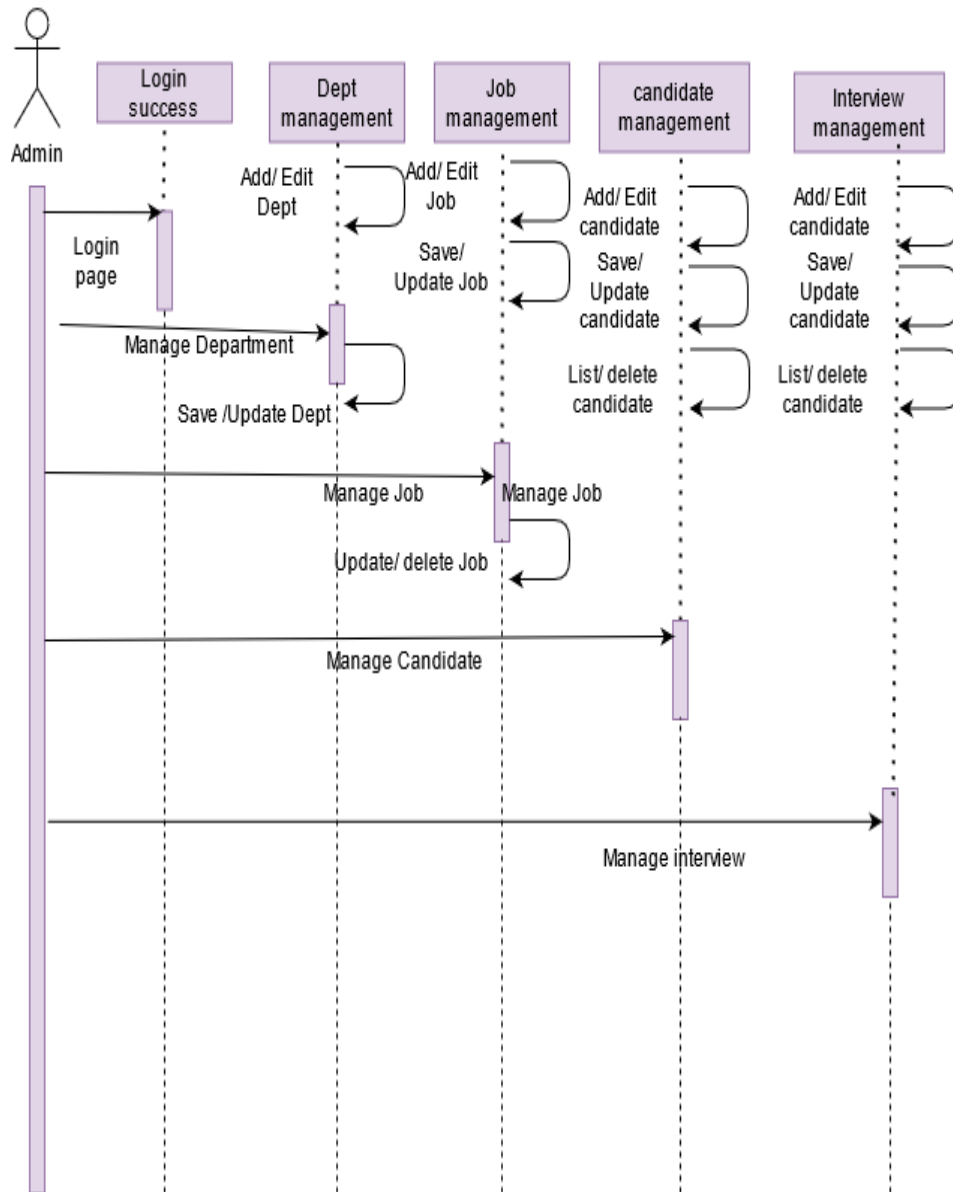


Figure 3.8: Sequence Diagram for Recruitment Management System

3.2.5 Data modeling diagrams

Entity Relationship Diagram, also known as **ERD**, ER Diagram or ER model, is a type of structural diagram for use in database design. An ERD contains different symbols and connectors that visualize two important information: The major entities within the system scope and the interrelationships among these entities. [15]

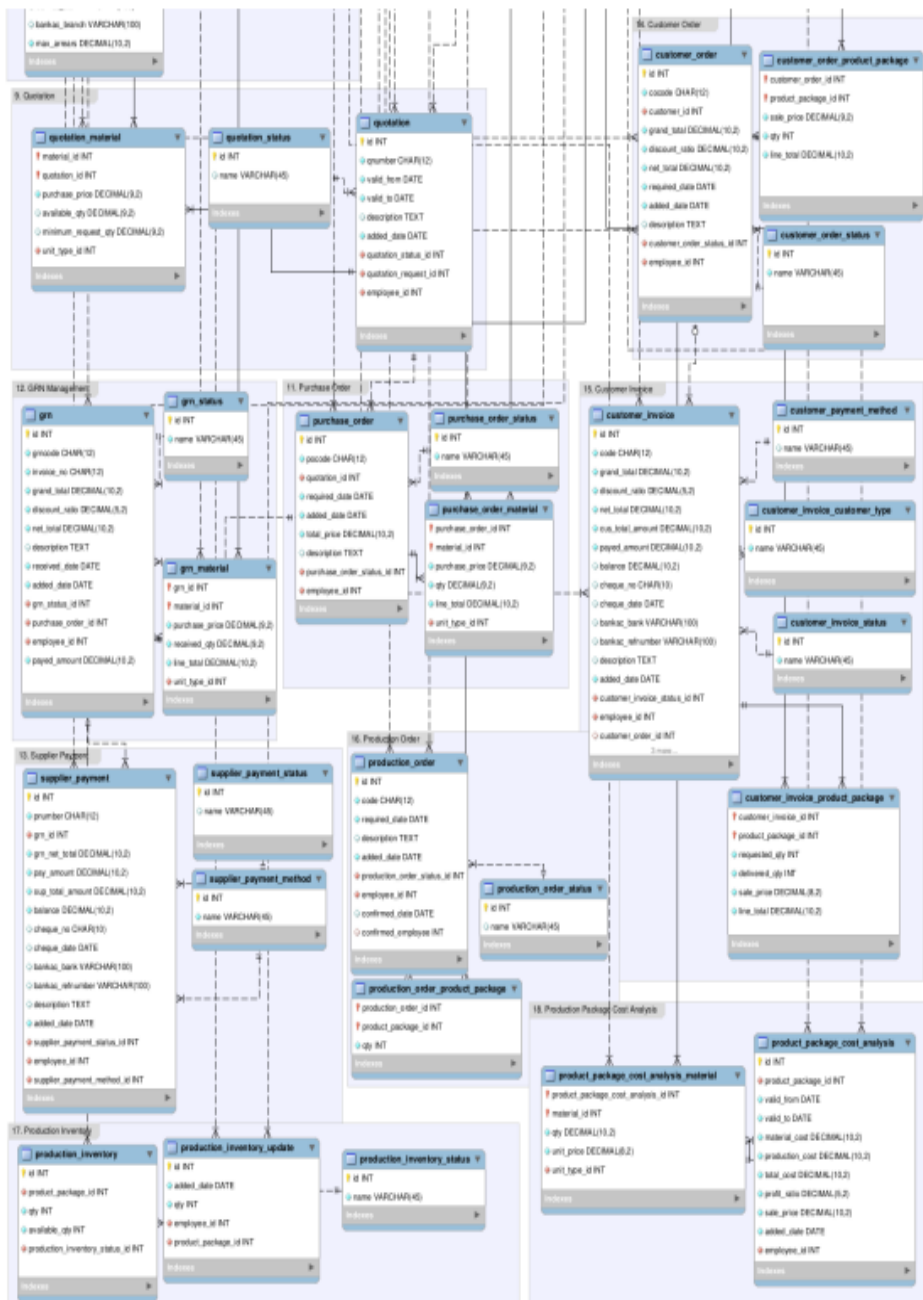


Figure 3.9: ERD diagram - Part 1

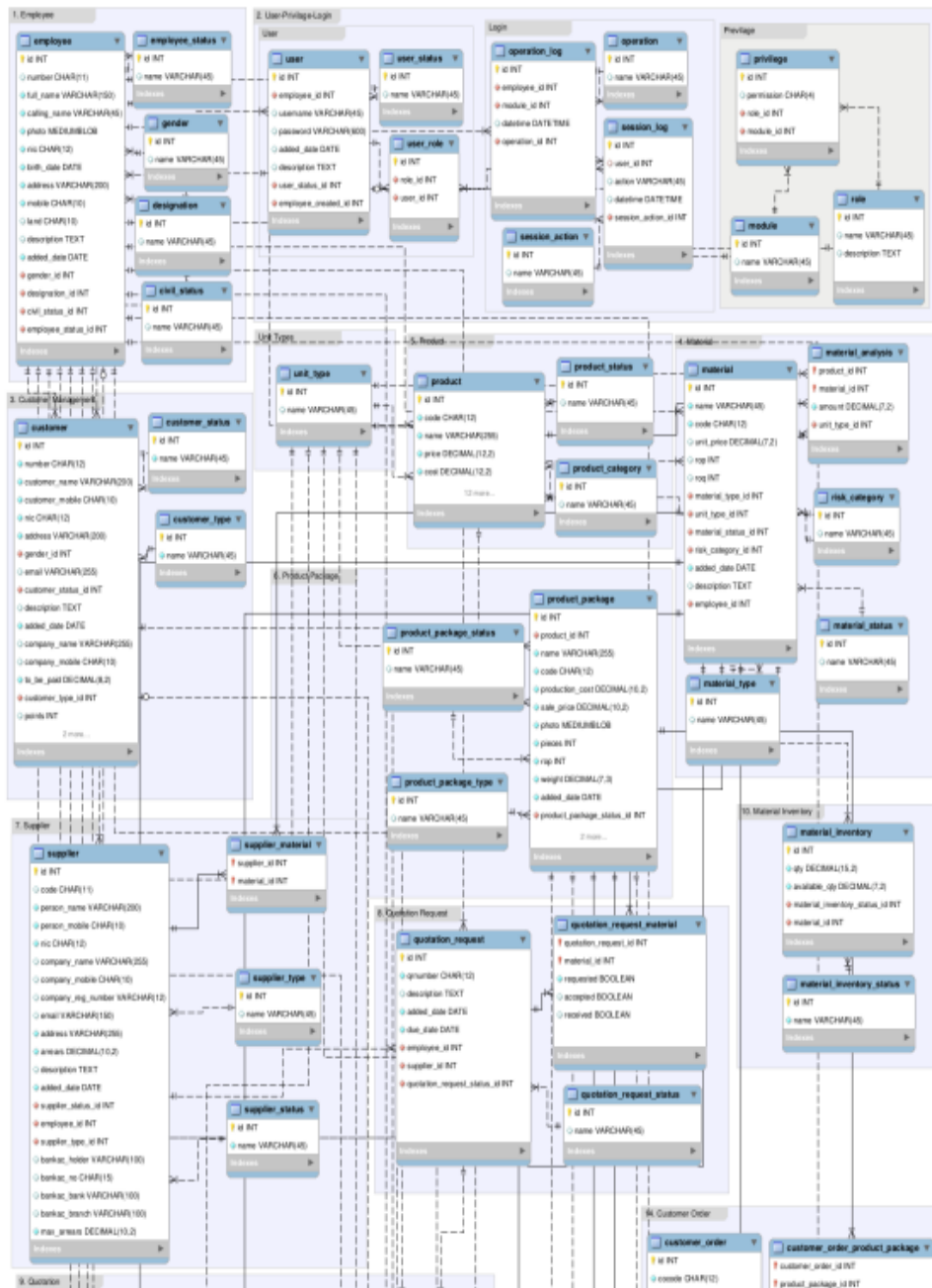


Figure 3.10: ERD diagram - Part 2

3.2.6 Data Flow Diagram for Recruitment Management System

Recruitment Management System Data Flow Diagram (DFD) DFD is often used as a preliminary step to create an overview of the Recruitment without going into great detail, which can later be elaborated. It normally consists of overall application data flow and processes. It contains all of the user flow and their entities such all the flow of candidate, interview management, job post and different type of search. All the diagrams have been used to visualize of data processing and structured design of the Recruitment process and working flow.

DFD for level1

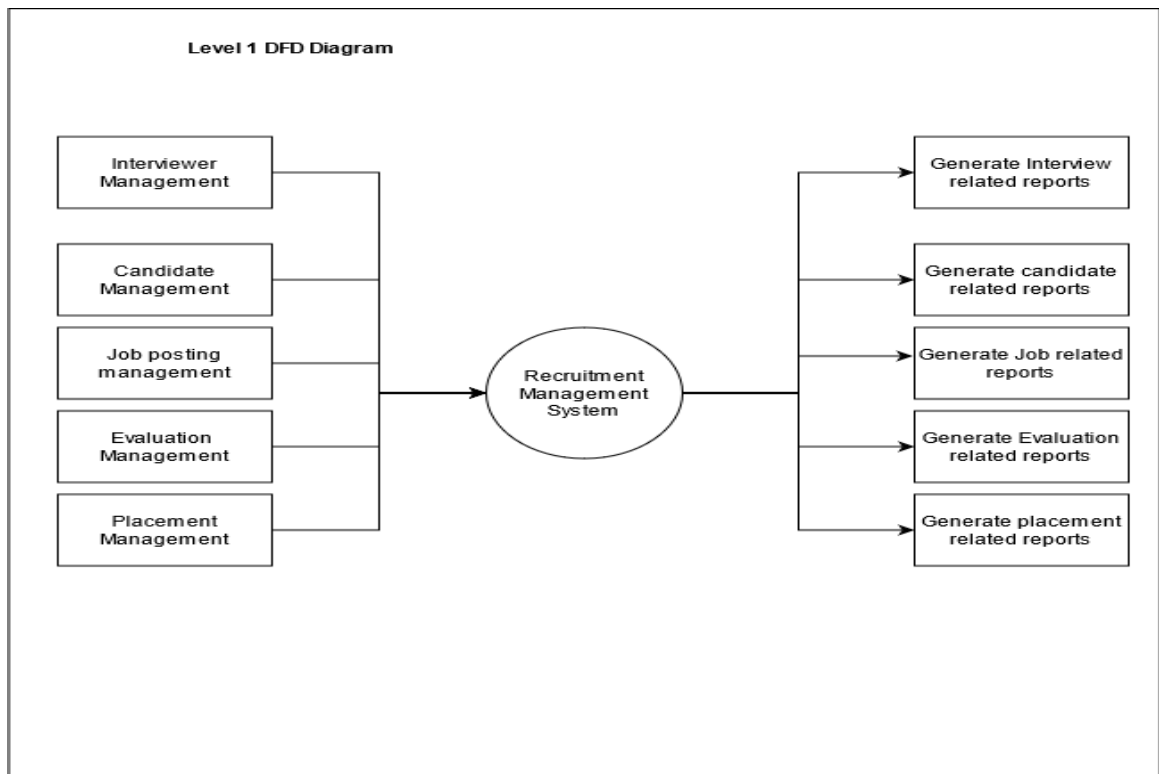


Figure 3.11: First level DFD for Recruitment Management System

First Level DFD of RMS shows how the system is divided into subsystems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the Recruitment Management System as whole. It also identifies internal data

stores of Login, registration, search, interview and post jobs. DFD Level provides a more detailed breakout of pieces of the 1st level DFD.

- Processing Candidate records and generate report of all Employee.
- Processing Candidate registration records and generate report of all candidate registration.
- Processing Post job records and generate report of all Post job
- Processing interview records and generate report of all interview
- Processing Search job records and generate report of all search Job
- Processing Login records and generate report of all Login

DFD for level2

DFD Level 2 then goes one step deeper into parts of Level 1 of Recruitment. It may require more functionalities of RMS to reach the necessary level of detail about the Recruitment functioning. First Level DFD (1st Level) of RMS shows how the system is divided into details of Login, Search job, interview, post job and candidate.

- Admin logs in to the system and manage all the functionalities of RMS.
- Admin can add, edit, delete and view the records of Candidate, Post Job, Search Job and Login
- Admin can manage all the details of candidate registration, interview and evaluation
- Admin can also generate reports of candidate, candidate registration, Post job, interview and search job
- Admin can search the details of candidate registration, search job and evaluation
- Admin can apply different level of filters on report of Candidate, interview and search job
- Admin can tracks the detailed information of candidate registration, post job, interview and search job

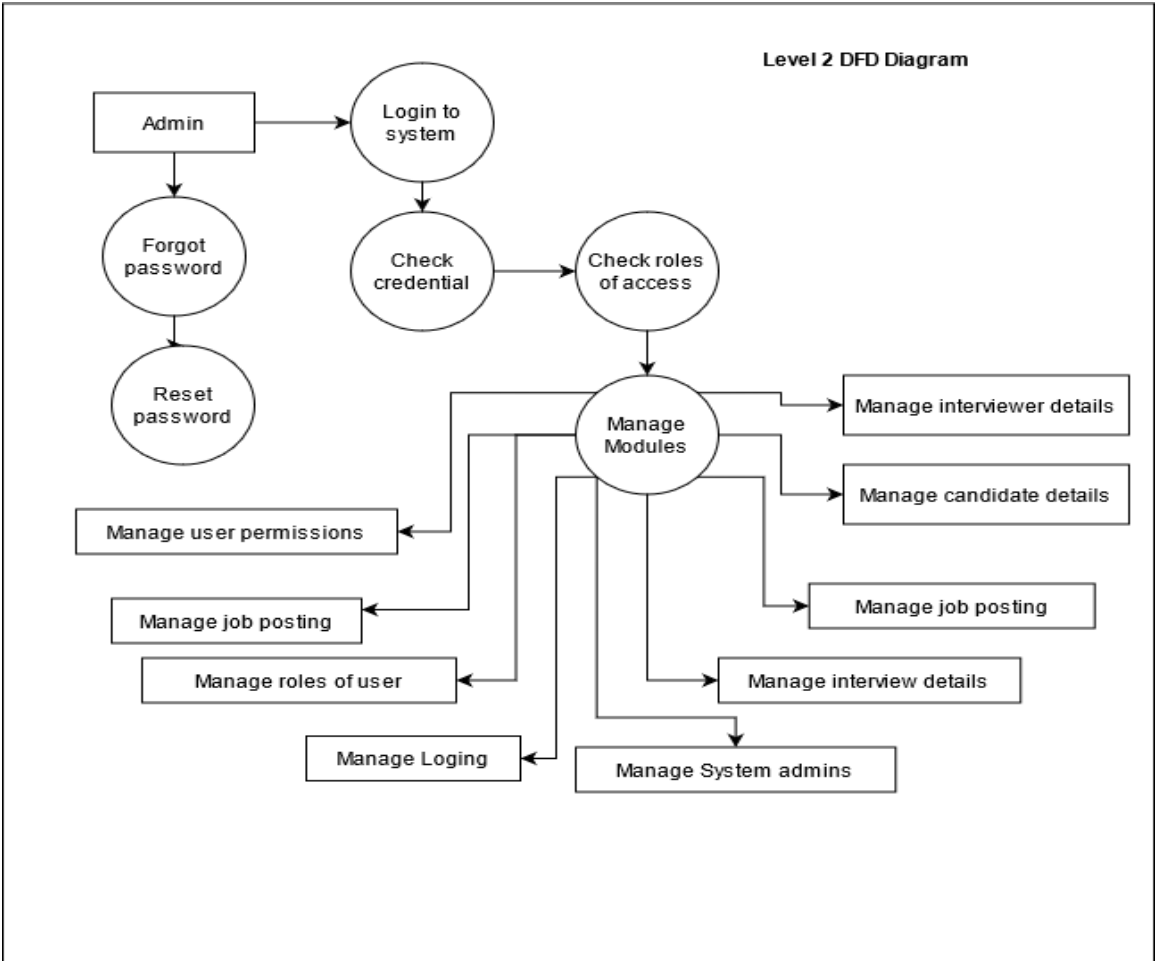


Figure 3.12: second level of DFD

3.2.7 Process Diagram for Recruitment Management System

Candidate on Boarding Process RM Process Diagram

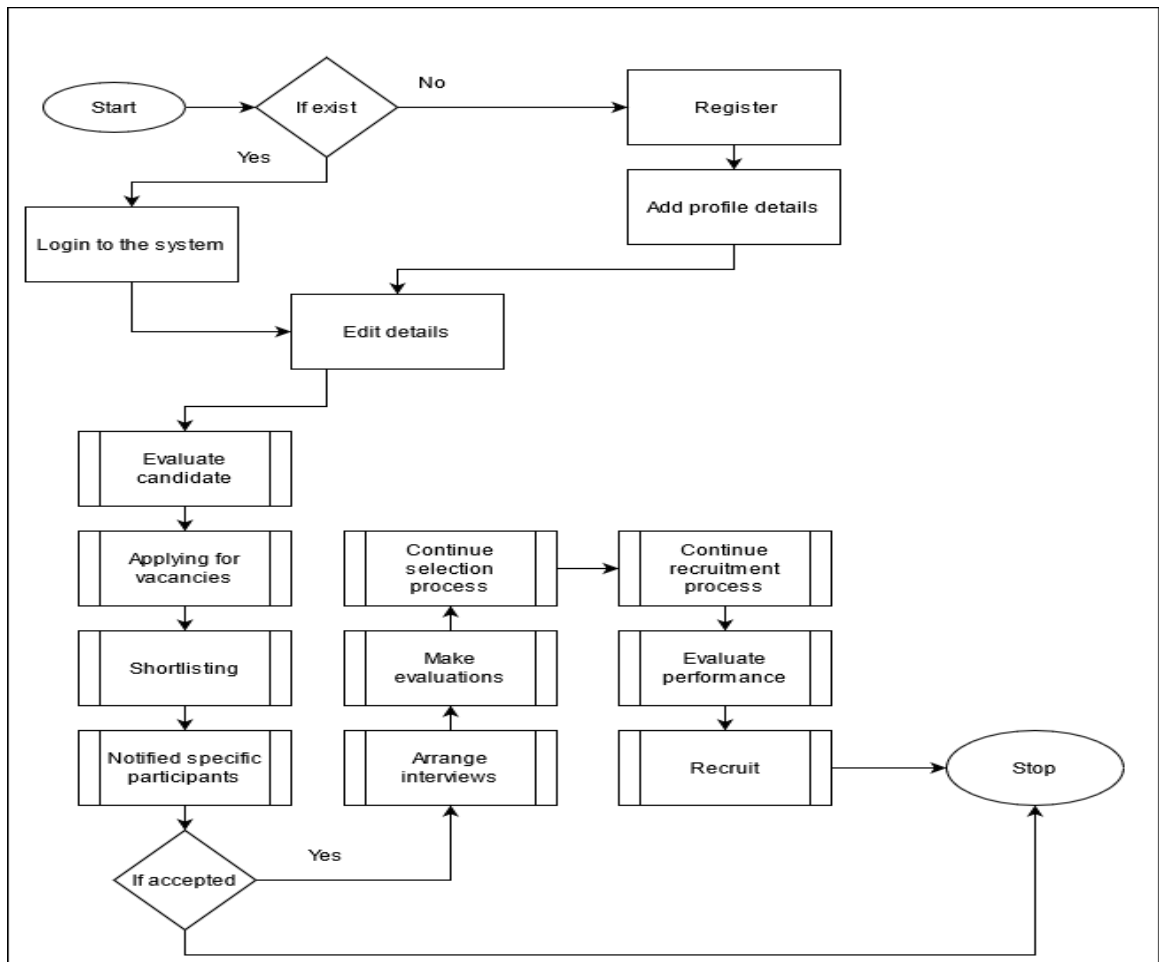


Figure 3.13: Candidate Onboarding Process

RMPProcessDiagram

Overall Process Diagram

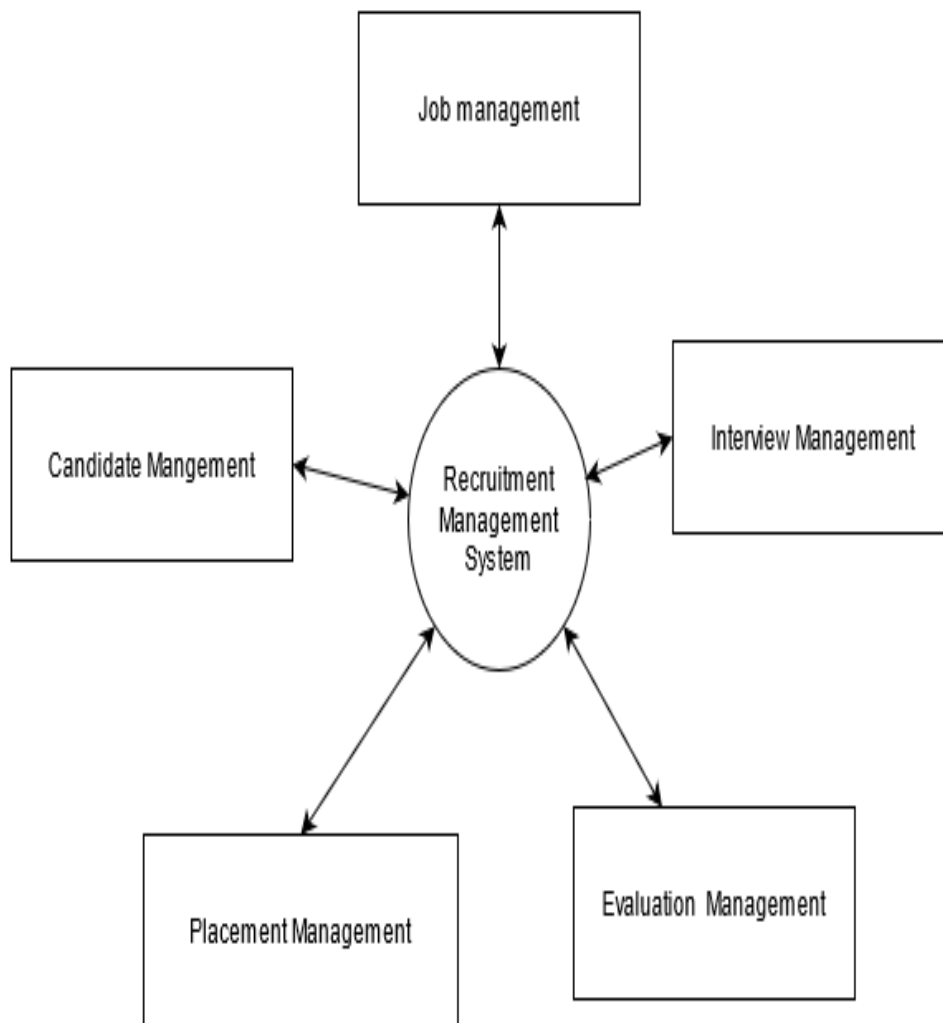


Figure 3.14: RMS Process Diagram

3.2.8 Activity Diagram for Recruitment Management System

This is the Activity UML diagram for Recruitment Management System which shows the flow between the activity of Admin, User, Candidate, Department, Placement, Interview and Job.

- Admin user can search Candidate, view description of a selected candidate, add candidate, update candidate and delete candidate.
- It shows the activity flow of editing, adding and updating of placement
- User will be able to search and generate report of interviews, department and job.
- All objects such as (Candidate, Placement, Job) are interconnected.
- It shows the full description and flow of candidate, department, job, interviews and placement.

RM Activity Diagram

Initial stage activity diagram of the Recruitment Management System

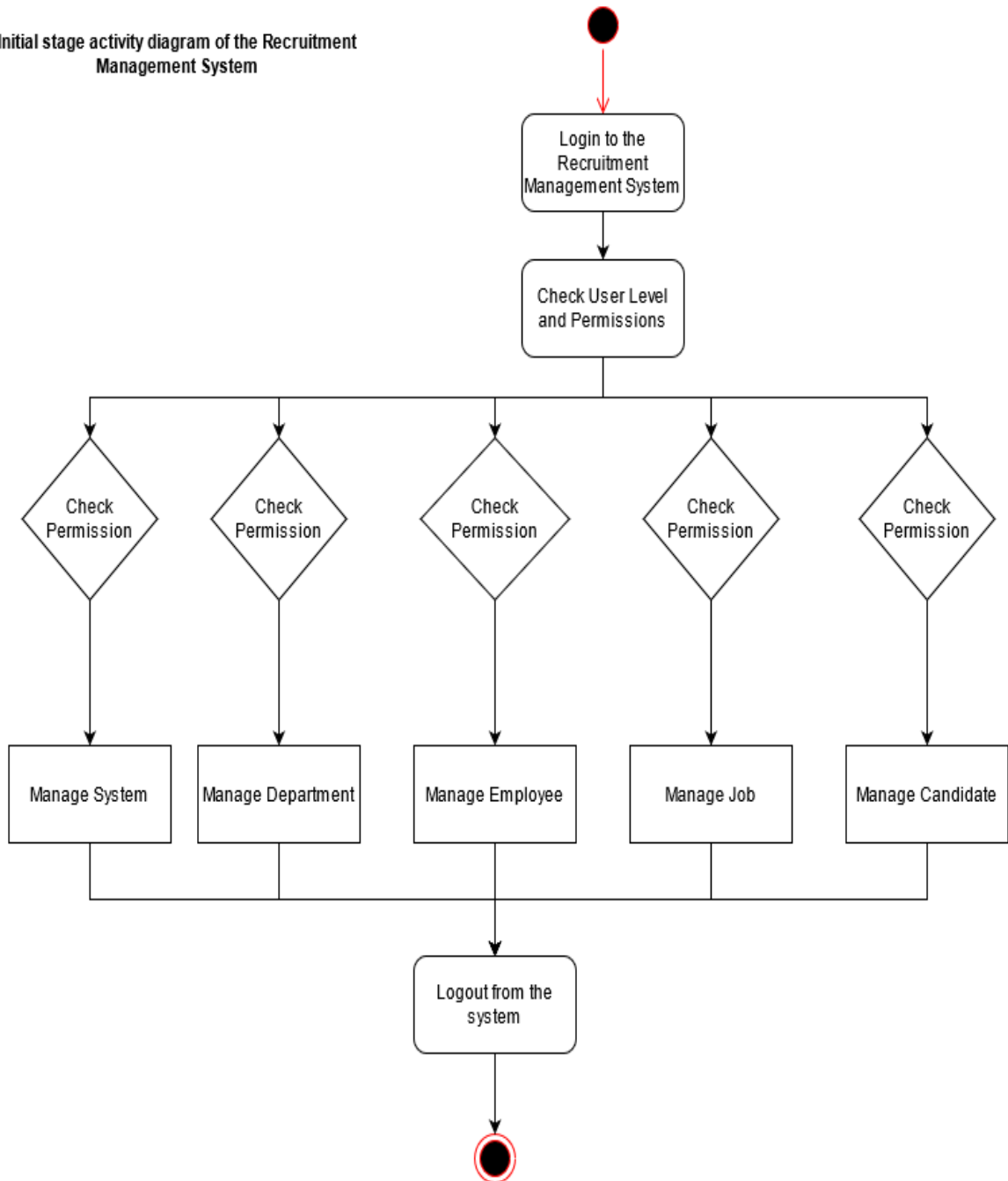


Figure 3.15: RM-Activity Diagram

Login Activity Diagram

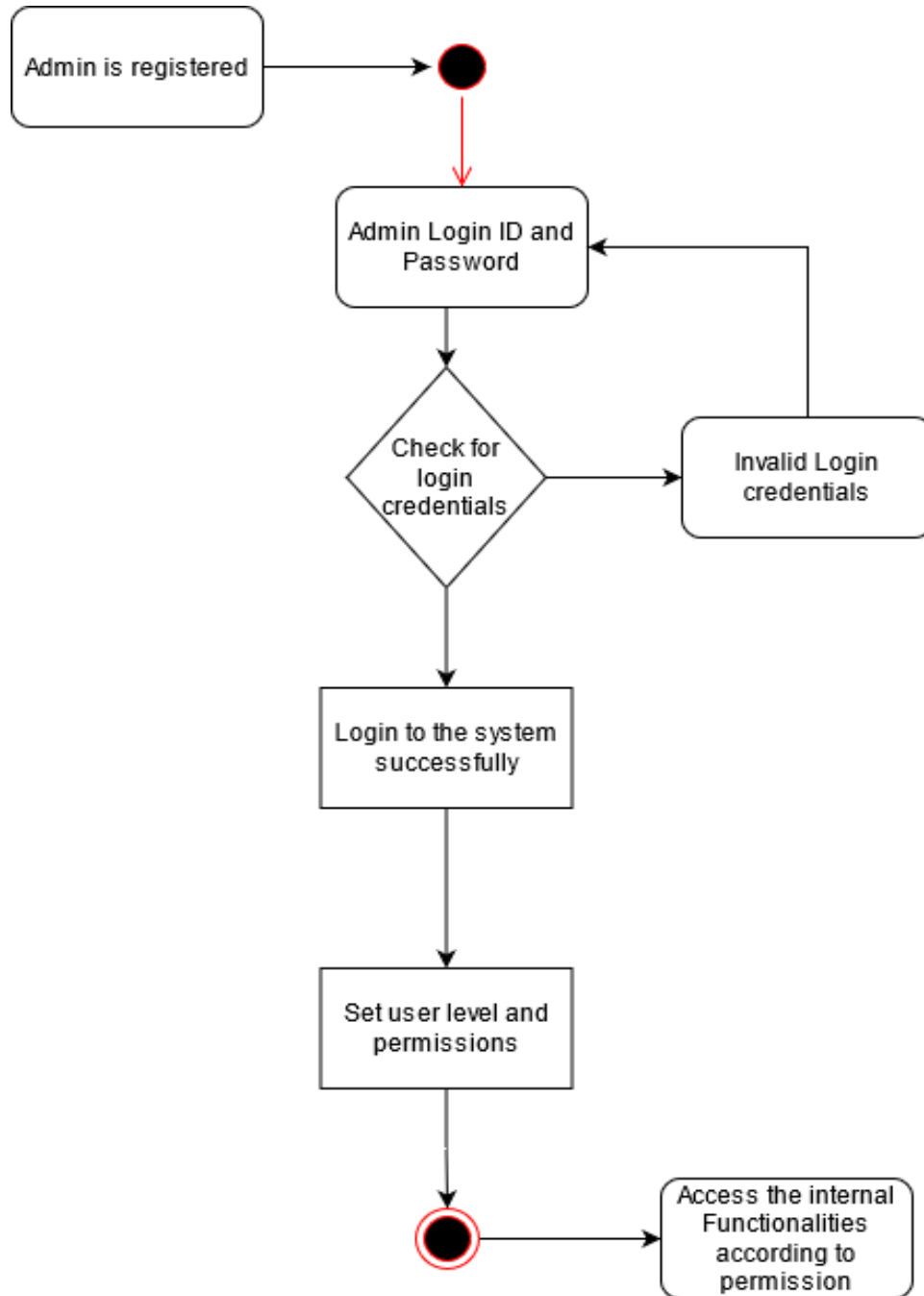


Figure 3.16: LoginActivityDiagram

3.2.9 User interface design

User Interface, or UI can be simply described as the view users see and perform actions on. This is one of the most critical aspects of the system since the complexity and user friendliness of the UI can impact the success of the overall project. UI design can also be defined as the process of designing the way in which system users can access system functionality, and the way that information produced by the system is displayed.

A good User Interface. (UI) UI provides a “userfriendly” experience, allowing the user to interact with the software or hardware in a natural and easy way. Nearly all software programs have a Graphical User Interface, or Graphical User Interface. (GUI) GUI. This means the program includes graphical controls, which the user can select using a mouse or keyboard. A typical GUI of a software program includes a menu bar, toolbar, windows, buttons, and other controls.

Since the users of this system are with intermediate computer skills, the system should allow easy navigation with consistency between separate views. It should also provide simplicity and descriptive feedback messages users with minimal technical knowledge can easily understand.

Dashboard UI

User dashboard is the main interface which user directed initially to access the system. It will provides all necessary links in a user friendly way. Dashboard options will be restricted according to the login users privileges.

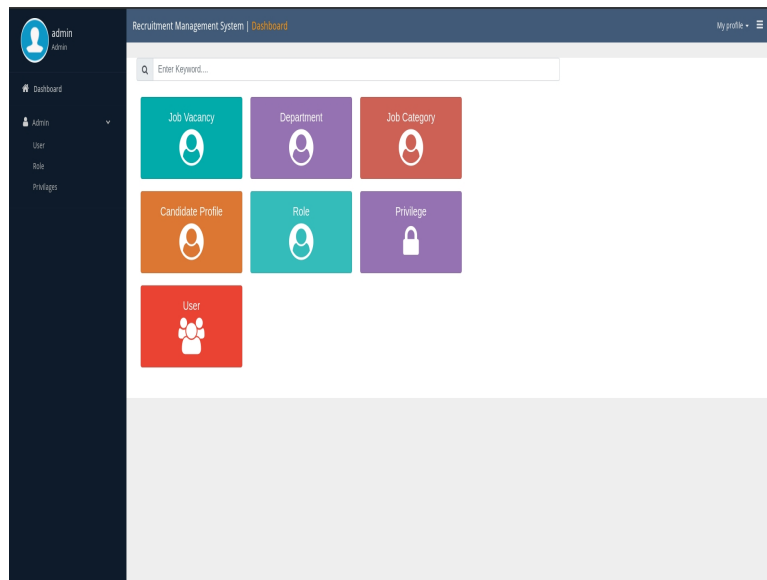


Figure 3.17: Dashboard UI

User Privilege View

This is the admin allowed place where user privileges will be provided. According to the privileges user actions and the access will be restricted.

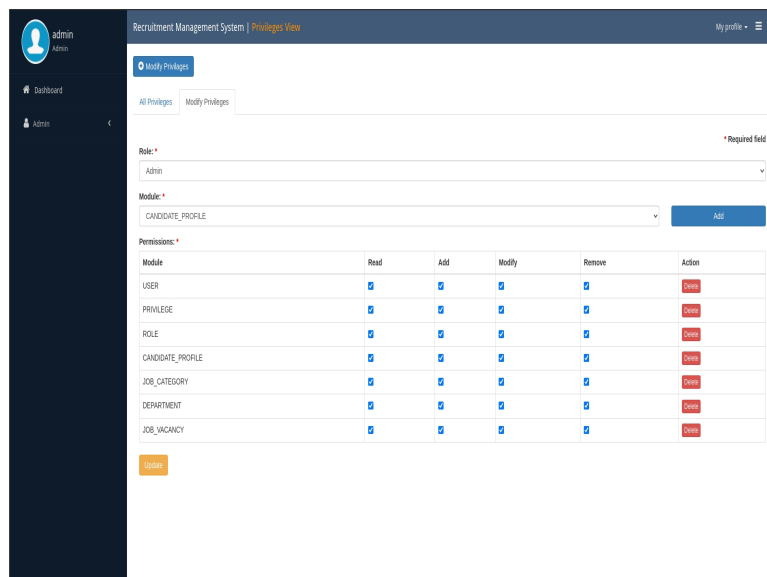
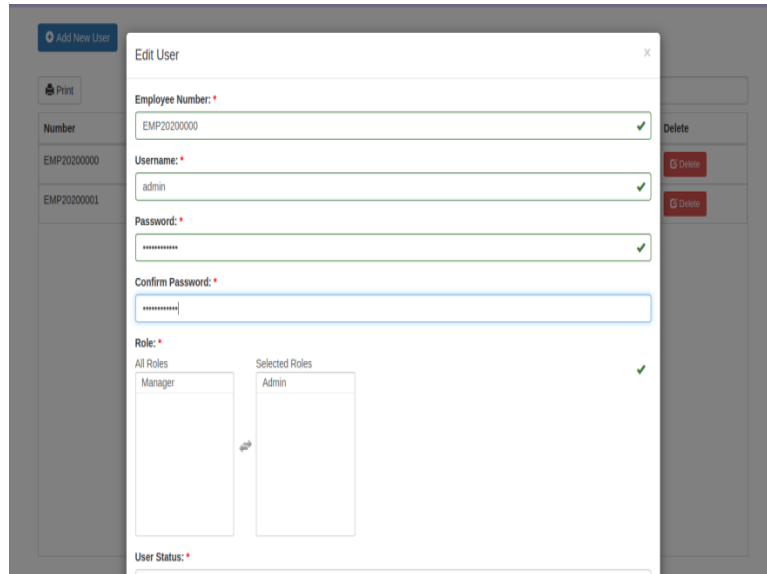


Figure 3.18: User privileges

Data insert form

This system contains different data inserting forms in each module. Below is the candidate related data input and update form. It consists with both client-side JavaScript validation and server-side validation to confirm the right data is fed into the database.



The screenshot shows a modal window titled "Edit User" with a close button (X) in the top right corner. The form contains the following fields and elements:

- Employee Number:** A text input field containing "EMP20200000" with a green checkmark on the right.
- Username:** A text input field containing "admin" with a green checkmark on the right.
- Password:** A password input field with masked characters "*****" and a green checkmark on the right.
- Confirm Password:** A password input field with masked characters "*****" and a green checkmark on the right.
- Role:** A section with two columns: "All Roles" containing "Manager" and "Selected Roles" containing "Admin". A green checkmark is on the right. An arrow points from "All Roles" to "Selected Roles".
- User Status:** A text input field at the bottom, currently empty.

The background shows a sidebar with "Add New User" and "Print" buttons, and a table with columns "Number", "EMP20200000", and "EMP20200001". To the right of the table are "Delete" buttons.

Figure 3.19: Candidate UI

3.2.10 Report View

Mainly this system contains reports to provide different details for specific user. Below is a common report view for super user.

Reports

All Reports > Candidate Quality by Referrer

Candidate Quality by Referrer

Filters and more (3 jobs)

Credited To	Application Review	Phone Interview	Face to Face	Hiring Manager Review	Offer	Hires	Quality ⓘ
Total	19	2	12	5	6	3	37%
Emma Hand	1	0	1	0	0	0	25%
Jovie Lanier	1	0	1	1	1	1	100%
Margot Dean	3	2	2	0	1	1	50%
No referrer	14	0	8	4	4	1	30%

Figure 3.20: Common report view

Further descriptions related to the interfaces and reports are added as Appendix C. Reporting related interfaces and details are added in Appendix D.

Chapter 4

Implementation

This chapter will explain how the implementation activities were carried out during the system development. In this phase, designed technical solutions in the design phase are transformed into an executable system that satisfies the client requirements collected in the analysis phase. This chapter will discuss about the implementation process, required hardware, software, design patterns, tools, languages, and frameworks used in the system.

Requirement gathering and analysis is a major task for any kind of project. Mainly, requirement gathering is done by analyzing existing systems and referring to the documentation provided on the internet. There it is focused on the challenges in the recruitment process and how to attract interviewers. Then it was identified that paper-based existing system does not meet the needs of the modern world recruitment related requirements, and it took much time to proceed when it increases the number of interviewers. As same, sometimes it was unable to identify the correct person for the right place. Due to some paper-based issues or attending issues. Because of these, it was decided to work on these identified barriers. [21] [22]

After the requirement gathering, it was identified the normal existing system deployment feasibility needs. Initially, the system was designed according to these feasibility requirements. [23]

4.1 Considering the implementation environment

After taking this business organization's nature, functional and nonfunctional requirements into account, hardware, software, and other resources were selected for the implementation process in order to ensure high performance, maintainability, and technological feasibility. Furthermore, userfriendliness and overall experience were taken into consideration while making these choices.

Hardware requirements

Server Computer :	
Processor:	3 GHz or more
RAM :	2 GB or more
Hard Disk:	8 GB or more
Client Computer:	
Processor:	1.5 GHz or more
RAM :	1 GB or more
Hard Disk:	4 GB or more ⁵
Software	
Operating System:	Windows 7 upwards
Web browser:	Mozilla Firefox / Chrome

Table 4.1: Hardware Requirements

Software requirements

- Windows 10 / Ubuntu 20 LTS
- Visual Studio Code
- MySQL Server - 8.0
- ExpressJs 4.17
- TypeORM
- Bootstrap latest version 3/5
- JQuery 3.5
- Microsoft Project 2016

- Visual paradigm
- Node.js 12.0
- NPM (Package Manager)
- Google Chrome

4.2 Justification for the choice of implementation platform

After taking many factors into account, various tools and resources were selected for the implementation platform. Following are the major tools that were selected with a brief description of why they were selected and some technical aspects of those tools

Visual Studio Code

Due to the fact that this project is mainly developed in TypeScript language, an editor that supports TypeScript features and excellent integration with the language was a must. Therefore, after considering multiple text editors and IDEs such as Eclipse and Sublime Text, Visual Studio Code was selected as the main development tool for this project because of its excellent features and developerfriendly workflow. Furthermore, since it is an open-source project, additional licensing fees were not necessary in order to use it and that helped to minimize the overall implementation cost.

Visual Studio Code has been the first crossplatform development tool in the Microsoft Visual Studio family that runs on Windows, Linux, and macOS. It is free, open source, and it is definitely a code eccentric tool, which makes it easier to edit code files and folderbased project systems as well as writing crossplatform web and mobile applications over the most popular platforms, such as Node.js with integrated support for a huge number of languages and rich editing features such as IntelliSense, finding symbol references, quickly reaching a type definition, and much more. [3]

TypeScript Language

Since both the backend and frontend of this project were developed using Javascript, TypeScript was selected as the language for the majority of the development. This was due to the fact that it is a superset of Javascript with additional features such as type checking and decorators. Not only that, But also since TypeScript is strongly typed, it was perfect for finding bugs and writing good code.

From a technical standpoint, TypeScript generates JavaScript. Instead of requiring a completely new run-time environment, TypeScript generated JavaScript can reuse all of the existing JavaScript tools, frameworks, and wealth of libraries that are available for JavaScript. The TypeScript language and compiler brings the development of JavaScript closer to a more traditional object oriented experience.[4]

Node.js

The server side of this project was developed using TypeScript, which basically results and Javascript code. But in order to run Javascript outside the browser, something called a “runtime” is required. There are two major run-time for Javascript called “Node.js” and “Deno”. For this, Node.js was selected due to its high popularity and massive community support. Simply Node.js is a run-time for writing server side JavaScript applications. It is built on top of the V8 JavaScript run-time and uses an eventdriven, nonblocking I/O model that makes it perfect for dataintensive, realtime applications.

Node is often used to build back end services that communicate with client-side applications. These applications get and send data through a back end service called an API. The API serves as an interface between different programs, so they are able to talk to each other[5] .

NPM Node Package Manager. (NPM)

In order to reduce complexity, development cost and improve stability, various third party libraries were selected to use throughout the project. These libraries should be well organized and maintained to ensure issues like version mismatches and security vulnerabilities. NPM is the Node.js package manager that can provide all these facilities. It was selected as the package manager due to the fact that it comes bundled with the

Node.js installer itself and it works seamlessly with Node projects.

In more technical terms, NPM, Node Package Manager, is two things:

- An online repository for the publishing of open source Node.js projects
- It is a commandline utility for interacting with the said repository that aids in package installation, version management, and dependency management [5] .

MySQL 8.0

Since database design of this system was done according to the relational database design, multiple choices were available to use as the Relational Database Management System such as PostgreSQL and Microsoft SQL Server. From those, MySQL was selected for this project due to its suitability for the client's requirements and open source nature.

It must be said that MySQL is easy to use and its operation is very fast. MySQL requires at first a general knowledge of SQL to work effectively with it. MySQL does not require much more knowledge, but a little knowledge of common relational database management system (RDBMS) is helpful [6]

ExpressJs - Web Server

or things such as routing and handling requests while checking for authentication, a framework named "Express.js" was selected. Express.js is a web application framework for Node.js. It provides various features that make web application development fast and easy which otherwise takes more time using only Node.js.

In more technical terms, Express.js is based on the Node.js middle ware module called connect which in turn uses http module. So, any middleware which is based on connect will also work with Express.js. [8]

Bootstrap 3

Coding Cascading Style Sheet). (CSS) CSS manually for each and every module of the system is a time consuming tedious task. But there are a number of previously written, industry standard CSS libraries with reusable

code and components. From those, a library called “Bootstrap 3” was selected as the main library for designing UI components for modules in this system.

Also, Bootstrap was used to avoid writing lengthy code and responsiveness of each UI was well maintained due to the fact that it is intended to create responsive websites [13].

jQuery

Most components in the clientside of this application was programmed using Javascript. Since writing plain Javascript can lead to lengthy code, JQuery was selected to ease up the process. Furthermore, jQuery was needed as a dependency for Bootstrap 3 library.

“Created by John Resig, jQuery is an opensource project with a dedicated core team of topnotch JavaScript developers. It provides a wide range of features, an easy to learn syntax, and robust cross platform compatibility in a single compact file. And over a hundred plugins have been developed to extend jQuery’s functionality. Then it makes an essential tool for nearly every clientside scripting occasion. [9]

Google Chrome

Here Chrome is selected as the primary web browser as it is necessary to access the client side. Google Chrome is the most popular browser with better support for modern Javascript features. And then it was selected as the primary browser for this project. Further it should be mentioned that the DevTools in Chrome was used when debugging and testing the clientside of the system. Then Google Chrome is the primary browser, since this web-based application complies with W3C standards, it can also be used in other browsers such as Mozilla Firefox, Opera, and Apple Safari.

Windows 10 OS

Node.js and JavaScript development and deployment can be done in all kind of platforms, both Windows 10 and Ubuntu 20 was used throughout the development process. Also, Ubuntu was selected as the most suitable operating system for the server side aspect of the project. Following are brief descrip-

tions of both of these operating systems.

Microsoft released Windows 10 in July 2015 as a followup to Windows 8. The company has said it will update Windows 10 continuously, rather than release a new, fullfledged operating system as a successor. Windows 10 features builtin capabilities that allow corporate IT departments to use mobile device management (MDM) software to secure and control devices running the operating system[14]

My SQL Workbench 8.0

Due to the complexity of the system, a tool which can be used to model relational databases easily using ERD was required. Since MySQL workbench can be used in the design phase as a diagramming tool and in the implementation phase as a development tool to forward engineer and generate code from design diagrams, it was selected as the best choice for modeling the database.

According to MySQL documentation and guides, MySQL Workbench is a graphical tool for working with MySQL servers and databases. MySQL Workbench supports MySQL server versions 5.6 and higher. It is also compatible with older MySQL server 5.x versions, except in certain situations (like displaying the process list) due to changed system tables. It does not support MySQL server versions 4.x[7]

TypeORM

When working with relational databases of large scale, manually running SQL queries and managing them is a tedious task which can lead to long development times and unstable systems. To simply this process, there are tools known as “ORMs”. These are basically used to interact with databases in a more abstract manner. For Node.js, there are various ORMs available. From those, TypeORM was selected for this project due to its compatibility with TypeScript, great documentation and large community.

“ORM is a type of tool that maps entities with database tables. ORM provides simplified development process by automating object to table and table to object version. Once it is written the data model in one place, it becomes

easier to update, maintain, and reuse the code”.[\[12\]](#)

Since, the model is weakly bound to the rest of the application, it can be easily changed without any hard dependency with other part of the application and can be easily used anywhere inside the application. TypeORM is very flexible, abstracts the RDBMS system away from the application and allows us to benefits from the use of OOP concepts [\[12\]](#)

4.2.1 Acknowledgment of any reused existing codes / APIs

Following are the reused codes and API details used. This is taken as the opportunity to acknowledge them. Details as follows:

Custom JavaScript table library

Combining parts from previous projects with necessary changes, a new library was created to display and handle tabular data for this project. This was custommade to meet client requirements while facilitating features such as handling large amounts of records.

Chart.js library

Chart.js is a collection of prewritten JavaScript classes that can be used to generate various charts and visualize them using HTML5 Canvas. It is capable of displaying interactive, detailed charts for a given data set. For that reason, this library was used for the report generation functionality of this system.

jQuery library

jQuery is not a language, but it is a well written JavaScript code. As quoted on official jQuery website, “it is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development”.

Bootstrap framework

Bootstrap is a free and opensource frontend framework for designing websites and web applications. It contains HTML and CSS based design templates

for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

Chapter 5

Testing And Evaluation

This chapter describes the testing and evaluation of the Web Based Recruitment Management System which is developed. Testing covers the Requirement Testing, Software Testing and the Usability Testing. In the first scenario, requirement testing refers to the testing which confirms the identified requirements are satisfied through the developed functions. Then Software testing certifies the functions and the appearances of the developed system. Finally, the usability testing provides the performance and user experiencing issues and its quality. The system is then evaluated from a software point of view. Below sections will provide the details descriptions of the used testing types and its results.

When it consider about testing types, it contains number of testing methodologies. Which are deal with requirement-related testing, back-end validation testing, services testing, front end UI testing, and System testing. Testing could be identified as an iterative process throughout the development process. Because Bugs, issues, and mismatches are addressed and corrected. These corrected functions retest. Using agile methodology means testing is essential and takes place in every action. That means that the development and testing are parallel and are closely connected.

5.1 Test Plan and Test Cases

Testing begins with the implementation, code is reviewed while developing stage for testing. Test plan included all phases of testing and also used as a guide for the overall testing process. Before the system implementation, the

test plan was designed. A test plan includes: test objectives, schedule and logistics, test strategies and especially test cases.

Test cases were created according to the designed test plan. That contains data, procedure, and expected result and represents which use to system or part of the system run. To reduce complexity of the testing process test cases were designed for each module independently. The following tables specify some test cases.

Manual testing method and procedures used for testing rather than automation tools and technologies.

5.1.1 Sample Test Cases

Table 5.1: Sample Test Case Example 01

Test Case Id	01		
Test Component	Dashboard		
Module	Dashboard		
Test Case	Dashbord links		
Expected output	Highlighted fields when cursor provided and when click, link to the expected page		
No	Test Case	Actual output	Status
01	Navbar arrow expand function	List should be expand	Result
0 2	Navbar arrow close function	List should be closed	Result
0 3	User Link	Link should be directed to the User Portal	Result
0 4	Privilege Link	Link should be directed to the User Privelege Portal	Result
0 5	Role Link	Link should be directed to the Role Portal	Result
0 6	Candidate Link	Link should be directed to the Candidate Portal	Result
0 7	Job Category Link	Link should be directed to the Jon Category Portal	Result
0 8	Department Link	Link should be directed to the Department Portal	Result
0 9	Job Vacancy Link	Link should be directed to the Job Vacancy Portal	Result
10	Candidate Profile Review Link	Link should be directed to the Candidate Profile Review Portal	Result

5.2 Requirement Testing

These are some of the highest level testings. It is a type of testing which considers that the prioritized goals of the project achieved. Here the software is tested against its original specifications. Then it creates a list of each scenario identified during the requirement gathering process. Then it will be listed and categorized into modules, tasks, and functionalities. It is tested through every functionality. Then it helps to identify the related issues, bugs, and missing functionalities while developing the system. Then retesting takes place until the function and the requirement definition meet together.

In this scenario, it has been used all the test cases as the document for requirement testing. It contains all the UI requirements and the functional

requirements as well. Then using it as a check list, this requirement test has been done. Below is the one example of check lists.

#	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	Project Name	Recruitment Management System										
3	Module name	Applying										
4	Written by											
5	Written date											
6	Executed by											
7	Executed date											
8												
9	#	Test case title or ID	Test case description	Test steps	Pre Condition	Test Data	Post Condition	Expected Results	Actual Result	Pass/ Fail	Test Date	Comments
10	1	Vacancies drop down check	Verify that the related details are shown in the dropdowns	Navigate to form data apply Details	Navigates to User details page.					pass		
11				Click on drop downs	Navigates to User details page.					pass		
12				Select option	Navigates to User details page.					pass		
13				Navigate after selection	Navigates to User details page.					pass		
14												
15	2	Loading drop down check	Verify that the related details are shown in the dropdowns	Navigate to form data apply Details	Navigates to User details page.					pass		
16				Click on drop downs	Navigates to User details page.					pass		
17				Select option	Navigates to User details page.					pass		
18				Navigate after selection	Navigates to User details page.					pass		
19												
20	3	Name field	Verify that the inserting fields are working properly.	Navigate to form data User Details	Navigates to User details page.					pass		
21				Click on empty field	Navigates to User details page.					pass		
22				Field should be editable	Navigates to User details page.					pass		
23					Navigates to User details page.					pass		
24												

Figure 5.1: Requirement verification

5.3 Software Testing Approaches

Software testing, two main areas. They are verification testing and validation testing. [2] Verification involves the testing and confirmation according to the specification. But the validation involves testing the system according to the user view along with user expectations. Static and Dynamic testing techniques play a great role here. Static testing includes inspection and analysis, while Dynamic contains the practical functional and physical testing of the system. The general testing phase contains five stages. They are Unit testing, Module testing, Subsystem testing the System testing. Unit testing contains the testing of each function. Module testing contains the testing

of the collection of units or functions. Most of the time this is a one-page or portal which contains inter depend on functionalities. Then subsystem contains a collection of interdependent collection of modules. Finally, system testing is testing the full system as a whole.

5.3.1 Unit Testing

Unit testing is a software testing method used to test individual smallest unit of source code. This could be a method, class or a module. In this system, individual modules were tested as units considering associated control data, usage procedures, and operating procedures. Then it helps to make sure that they are functioning according to the level of expectation. This allowed early detection of problems that could have caused issues later in the development and helped to improve the quality of the overall development process.

The smallest testable part of a software is selected unit test part and, it usually consists of few inputs and a single output. While coding or just after coding is finished, it is possible to make a unit test.

5.3.2 Integration Testing

After doing unit test, individual units will be interconnected and test as an integrated group. This helps to identify the interoperability of connected modules. Then it helps identify that those modules are running together without providing issues. In most cases, it will help to identify the calculation issues could be occurred while merging units together.

5.3.3 System Testing

This is the final testing done after the development process finished. Here the entire system is considered as a whole. Testing is conducted by considering the context of functional requirement specification or the system requirement specification. This could be considered as black-box testing. Along with this software is evaluated from the users' point of view with the help of required documents. Expanding of this test will be from software requirement to hardware specifications evaluation. performance issues and minor mismatches between actual user requirements and system provided functionalities. Further it helped to make sure the system is performing well under the minimum required software and hardware specifications.

5.3.4 UI Testing

UI Testing or GUI Testing is a mechanism meant to test the aspects of any software that a user will come into contact. Usually means testing the visual elements to verify that they are functioning according to requirements – in terms of functionality and performance.

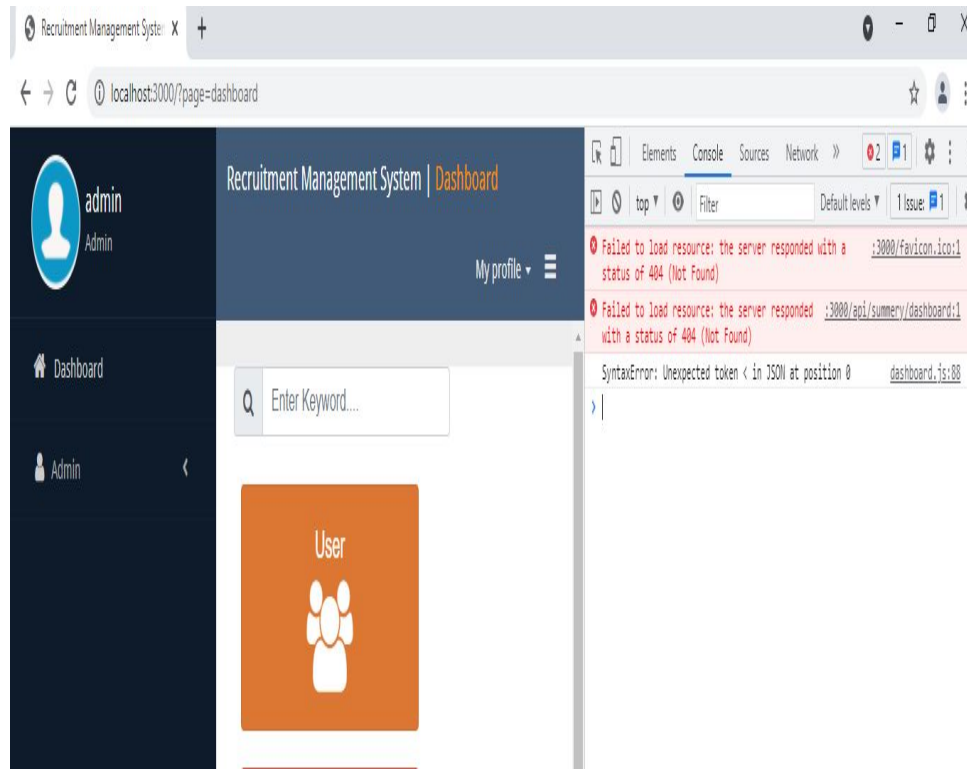


Figure 5.2: UI based testing

5.4 Proof of testing of work / Results of work

Below, provided test cases are created by considering the Requirement specification document to cover most of the important functionalities of the system. According to the inter operability of the system, each scenario contains its own priority.


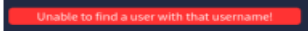
Test No	Description	Steps to Test	Expected Result	Pass / Fail
1	Validate user input details.	<ol style="list-style-type: none"> 1. Set account status to enabled. 2. Enter correct username. 3. Enter correct password. 4. Press login button. 	Successfully login to the system.	Pass
2	Validate user input details.	<ol style="list-style-type: none"> 1. Set account status to enabled. 2. Enter correct username. 3. Enter incorrect password. 4. Press login button. 	Display error message "Password you provided is wrong". 	Pass
3	Validate user input details.	<ol style="list-style-type: none"> 1. Set account status to enabled. 2. Enter incorrect username. 3. Enter correct password. 4. Press login button. 	Display error message "Unable to find a user with that username". 	Pass

Figure 5.3: System error testing - test cases examples

Test No	Description	Steps to Test	Expected Result	Pass / Fail
1	Logout of the system.	<ol style="list-style-type: none"> 1. Open the drop-down menu in the top right corner. 2. Press "logout" option. 	Logout from the system and redirect to login page	Pass
2	Check logout state change.	<ol style="list-style-type: none"> 1. Logout from the system. 2. Refresh the webpage. 3. Press the "back button" of the browser. 	Redirect back to the login page.	Pass

Figure 5.4: System login function - test cases examples

5.4.1 API Testing

API testing refers to Application Programming Interface related testing. Main focus of this testing is to meet the functional expectations, reliability, performance and security.

In this project Postman software is used for API testing. This helps test all the logic based service testing.

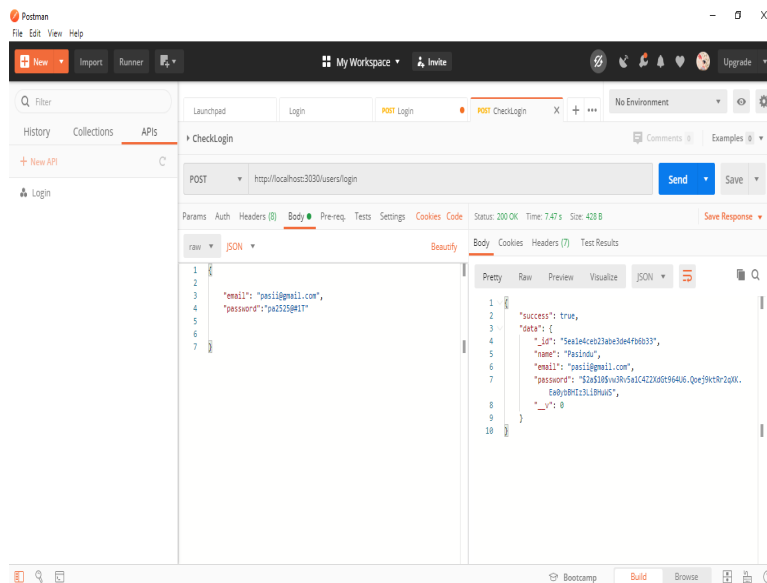


Figure 5.5: Postman Application

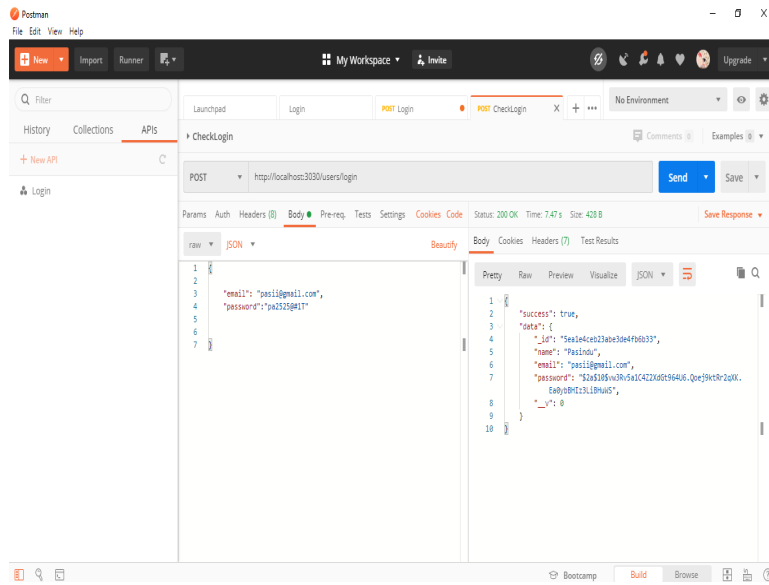


Figure 5.6: example01 API

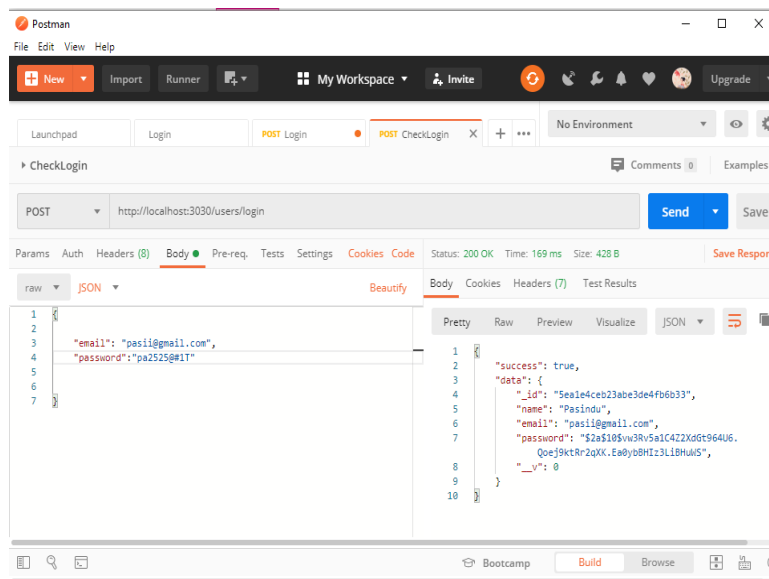


Figure 5.7: example02 API

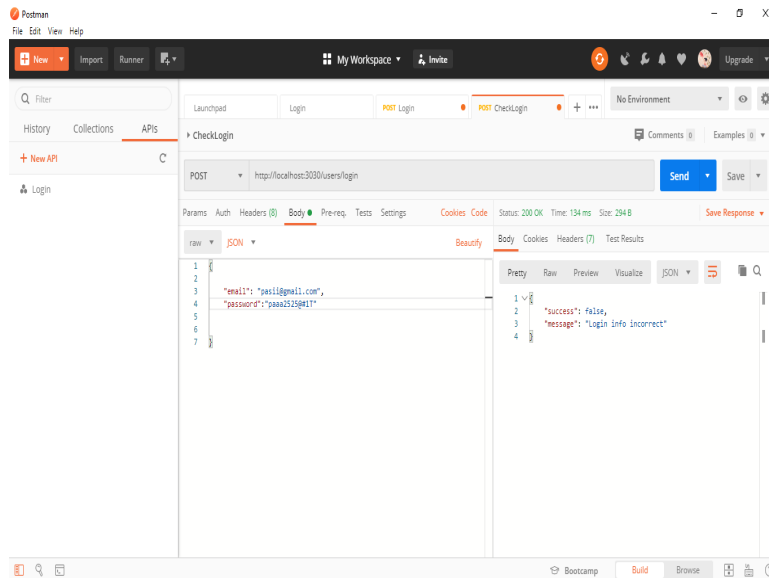


Figure 5.8: example03 API

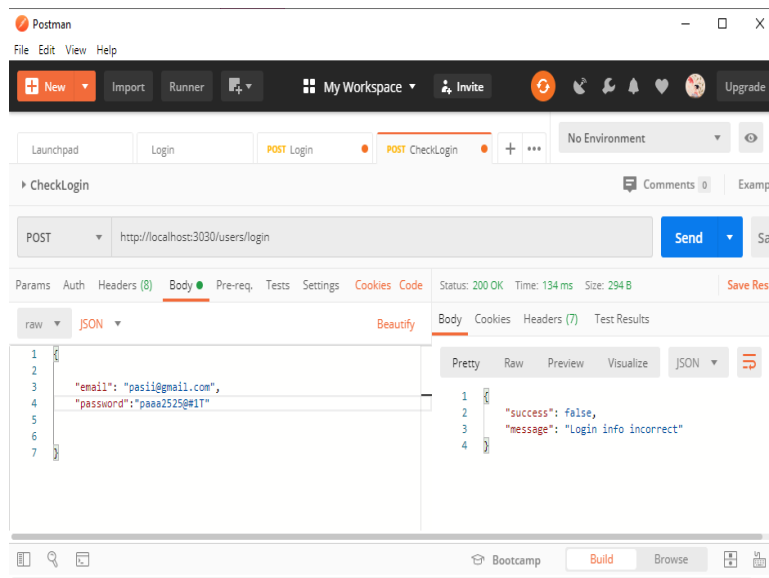


Figure 5.9: example04API

5.5 Evaluation

After completion of the project implementations, it is needed to conduct user evaluation session. For this scenario, system has given for some random

users and get their comments while they are using the system. The questionnaire used, has been appended to appendix F. According to the users, they have provided below comments and they have been corrected as change requests:

- Login form should be small and should appeared easily Due to some issue, login does not appeared as developed. Then it was corrected.
- Users request for a date indication in the system to easily identify the days It has been added to the system as a change request.
- Name spaces expanding was requested.
- According to the request some qualifications has been added.
- Some creative ideas were provided and they have been kept as future developments.

Chapter 6

Conclusion

6.1 Critical evaluation of the project

Due to the present situation, web based systems become more popular. Among them keeping recruitment management system online based is more important. Because it helps to three different parties to work together without having any hesitation in any condition. Then it helps to continue day to day works continuously.

Earlier and most of the companies are using normal manual methods of interviewing and evaluation. This process will take hours to provide relevant solutions. Due to this reason, neediness of automated system. By considering all these scenarios it is identified the requirements for the proposed system. Then this system was built using MVC architecture. Node.js, Javascript,HTML,CSS and SQL server has been used as technologies.

6.2 Lesson Learnt

The knowledge added throughout the project was actually valuable. Starting from the feasibility studies, to the end of development this process gave incomparable experience in many ways. This project gave a chance to test and implement most important theories and technologies learn throughout the MIT degree program. It also facilitated to learn very interesting new and updated technologies (JavaScript, jQuery, Entity frameworks) in order to improve the system performance. Moreover, special efforts were taken to learn the MVC design pattern. Furthermore working on the project encourage me

to improve technical skills as well as intellectual skills by collaborating with many individuals from collective fields. Proper time management and planning is very important aspects learn out during this project. Should carry out the work according to the schedule.

6.3 Future Works

As this is a web based project, this is storing personal details, qualifications and other important details of users. Then it may contain risks for the users. Because one can use these to gain competitive advantage against this system. By considering this issue, it has identified to improve these areas as future work:

- Add more reports that uses machine learning.

Due to increased usage of machine learning in businesses to gain competitive advantage, I plan to add reports that make use of machine learning to provide predictions for things such as customer orders and demands for the future.
- Implement OTP-token system for logging system This aims to solve security risks such as password bruteforcing by requesting users to enter a token sent to their mobile number via SMS every time they login to the system

Bibliography

- [1] Bartram *kissing frogs to find princes*. International Journal of Selection and Assessment, (2000),8(4).
- [2] *I. Sommerville, Software Engineering* Software Engineering, 8th edition,Addison-Wesley,(2008).
- [3] *Visual Studio Code Distilled Evolved Code Editing for Windows, macOS, and Linux*. A.D. Sole, Visual Studio Code Distilled Evolved Code Editing for Windows, macOS, and Linux. Berkeley, Ca Apress,(2019).
- [4] *N. Rozentals, Mastering TypeScript* Mastering TypeScript : build enterpriseready, industrial strength web applications using TypeScript and leading JavaScript frameworks.,Addison-Wesley,(2015).
- [5] *P. Teixeira, Professional Node js* P. Teixeira, Professional Node js building JavaScriptbased scalable software, Indianapolis, Ind. Wiley, Wrox, (2013)
- [6] *E. Vanier, B. Shah, and T. Malepati, Advanced MySQL 8* E. Vanier, B. Shah, and T. Malepati, Advanced MySQL 8 : discover the full potential of MySQL and ensure high performance of your database. Packt, (2019).
- [7] *M. Mclaughlin, MySQL workbench* M. Mclaughlin, MySQL workbench : data modeling and development. McgrawHill Education,(2013).
- [8] *A. Mardan, Pro Express.js*. Apress, New York, Ny A. Mardan, Pro Express.js. Apress, New York, Ny,(2014).
- [9] *J. Chaffer and K. Swedberg, Learning jQuery* J. Chaffer and K. Swedberg, Learning jQuery better interaction desgin and web development with simple JavaScript techniques. Packt, (2007).

- [10] W. Hosch, Chrome Internet browser. (2020).
Available: <https://www.britannica.com/technology/Chrome>.
- [11] *M. Mclaughlin, MySQL* M. Mclaughlin, MySQL workbench : data modeling and development, McGrawHill Education, (2013).
- [12] Typeorm introduction, www.tutorialspoint.com,(2020).
Available:<https://www.tutorialspoint.com/typeorm>
- [13] P. Chahal, What is bootstrap? TemplateToaster Blog,(Sep 2019)
Available: <https://blog.templatetoaster.com/what-is-bootstrap/>
- [14] M. Rouse, What is microsoft windows 10? SearchEnterpriseDesktop, (2017).
Available:[https : / / searchenterprisedesktop.techtarget.com/definition/Windows-10](https://searchenterprisedesktop.techtarget.com/definition/Windows-10)
- [15] UML, agile, tagaf, pmbok, bpmn guide, www.visual-paradigm.com
Available: <https://www.visual-paradigm.com/guide>
- [16] blog for software development models,
Available:<https://existek.com/blog/sdlc-models>
- [17] solutions
Available:<https://www.icims.com/solutions>
- [18] jobvite-hire,
Available:<https://www.jobvite.com/products>
- [19] RECRUITMENT MANAGEMENT SYSTEM AND EMPLOYEE PROCUREMENT,
Available:<https://www.jobvite.com/products/jobvite-hire/>
- [20] Software Development Life Cycle
Available:<https://www.productplan.com/>
- [21] Impression Management and Interview
Available:<https://www.ncbi.nlm.nih.gov/>
- [22] Challenges in Recruitment and Selection Process: An Empirical Study
Available:<https://www.mdpi.com/>
- [23] Feasibility analysis for Web project engineering

Available:<https://www.researchgate.net/publication/>

[24] MVC Architecture

Available:<https://www.guru99.com/laravel-tutorial.html>

[25] Software Engineering 9th Edition *Software Engineering 9th Edition*.
Software Engineering 9th Edition, (2009).

[26] Similar Systems

Available:<https://www.softwaresuggest.com/blog/top-free-open-source-recruitment-software/>

Appendix A

System Manual

These steps will be provide the detailed steps to setting up this proposed system. Below it provides the details of software and hard ware requirements needed for this proposed system.

A.1 Step 1

System requirements can be identified using the following table A.1 for software verification and table A.2 for hardware verification.

Table A.1: Software Verification for the system

Software	Minimum Requirement
Operating System	Windows 8.1 / Ubuntu 16.04 / Cent OS 7
Runtime Web Server	Node.js 10 upwards and Express Js 3.0 upwards
DBMS	MySQL Server 8.0
Database Manager	MySQL Workbench 8.0
Web Browser	Google Chrome 70.0 / Firefox 72.0

Table A.2: Hardware Verification for the system

Hardware	Minimum Requirement
Processor	3.00 GHz Dual Core
Memory Web Server	4GB
Hard Disk	250 GB
Printer	LaserJet Printer
Internet	Connection with 1Mbps download speed

A.2 Step 2

- Get a new CD copy of the Recruitment Management System form the main repository.
- Get the new CD copy of the MySQL Workbench and import the database form the Recruitment Management System DB repository.

A.3 Step 3

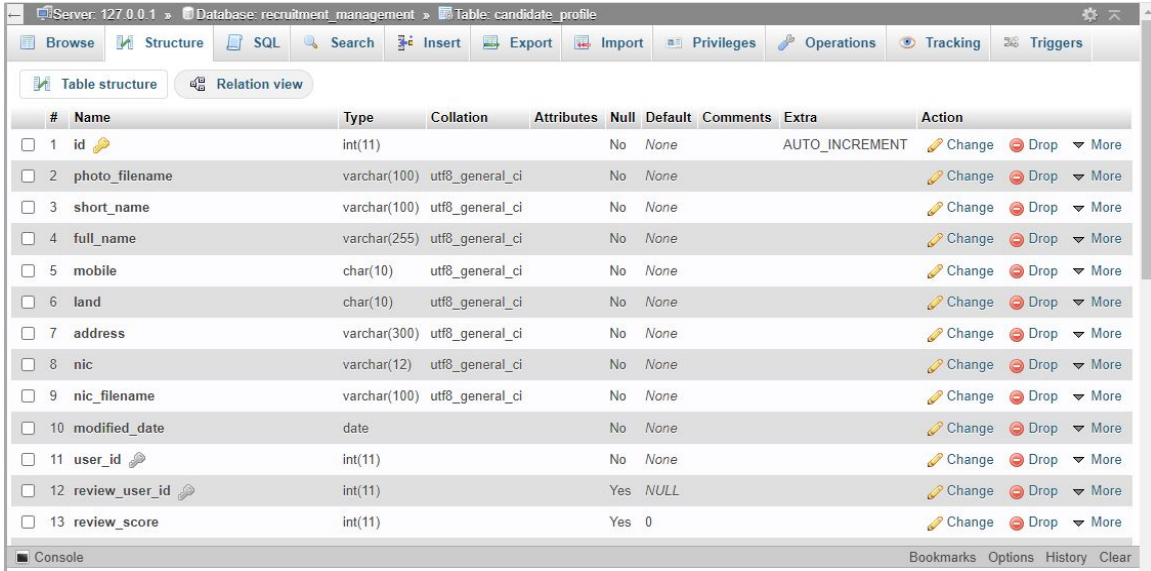
- Copy both CD contents to the selected machine and access the terminal or command prompt. Then install all he dependencies using NPM.
- Start the system by using npm start command
- Navigate to `http://localhost:<port selected>`in the web browser
- Then it is possible to access system with client interfaces.

Appendix B

Design Documentation

B.1 Database Design

Following figures show the database design of basic tables used in the system.



The screenshot shows a database management tool interface for a table named 'candidate_profile' in a 'recruitment_management' database. The table structure is displayed in a grid format with columns for #, Name, Type, Collation, Attributes, Null, Default, Comments, Extra, and Action. The table has 13 columns. The 'id' column is the primary key and is an integer with an auto-increment attribute. The 'review_user_id' and 'review_score' columns have foreign key relationships indicated by key icons.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	photo_filename	varchar(100)	utf8_general_ci		No	None			Change Drop More
3	short_name	varchar(100)	utf8_general_ci		No	None			Change Drop More
4	full_name	varchar(255)	utf8_general_ci		No	None			Change Drop More
5	mobile	char(10)	utf8_general_ci		No	None			Change Drop More
6	land	char(10)	utf8_general_ci		No	None			Change Drop More
7	address	varchar(300)	utf8_general_ci		No	None			Change Drop More
8	nic	varchar(12)	utf8_general_ci		No	None			Change Drop More
9	nic_filename	varchar(100)	utf8_general_ci		No	None			Change Drop More
10	modified_date	date			No	None			Change Drop More
11	user_id	int(11)			No	None			Change Drop More
12	review_user_id	int(11)			Yes	NULL			Change Drop More
13	review_score	int(11)			Yes	0			Change Drop More

Figure B.1: Candidate Profile

Server: 127.0.0.1 » Database: recruitment_management » Table: job_category

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 name	varchar(80)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 description	varchar(200)	utf8_general_ci		Yes	NULL			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Add to central columns Remove from central columns

Figure B.2: Job Category

Server: 127.0.0.1 » Database: recruitment_management » Table: job_vacancy

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 title	varchar(100)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 position	varchar(100)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 description	varchar(200)	utf8_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	5 location	varchar(100)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/>	6 added_date	date			No	None			Change Drop More
<input type="checkbox"/>	7 user_id	int(11)			No	None			Change Drop More
<input type="checkbox"/>	8 department_id	int(11)			No	None			Change Drop More
<input type="checkbox"/>	9 job_category_id	int(11)			No	None			Change Drop More
<input type="checkbox"/>	10 job_vacancy_status_id	int(11)			No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Add to central columns Remove from central columns

Figure B.3: Job Vacancy

Server: 127.0.0.1 » Database: recruitment_management » Table: user

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 username	varchar(45)	utf8_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	3 password	varchar(600)	utf8_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	4 photo_filename	varchar(100)	utf8_general_ci		Yes	avatar.png			Change Drop More
<input type="checkbox"/>	5 email	varchar(45)	utf8_general_ci		No	not set			Change Drop More
<input type="checkbox"/>	6 mobile	char(10)	utf8_general_ci		No	not set			Change Drop More
<input type="checkbox"/>	7 added_date	date			No	None			Change Drop More
<input type="checkbox"/>	8 user_status_id	int(11)			No	None			Change Drop More

Figure B.4: User

Server: 127.0.0.1 » Database: recruitment_management » Table: role

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 name	varchar(45)	utf8_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	3 description	text	utf8_general_ci		Yes	NULL			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext
 Add to central columns Remove from central columns

Figure B.5: Role

Appendix C

User Manual

This Recruitment Management System was developed to enhance the effectiveness of the existing recruitment process of companies. Main focus is to keep the smooth going of the process with transparency and accuracy.

This documentation provides an initial overview of the system and essential knowledge every user must have when using it. Please go through the following stepbystep guide to gain an understanding of the basics before using the system.

C.1 Logging to the system

- First enter the correct URL for system access. If it is running locally, it should be `http://localhost:3000`.

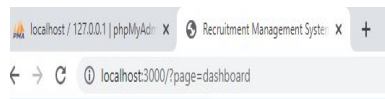


Figure C.1: URL

- Then it will directs to insert your username and the password.

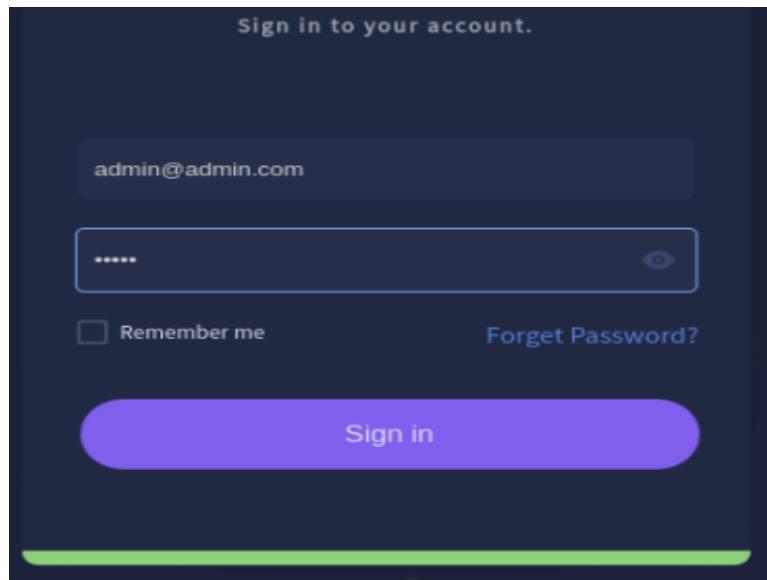


Figure C.2: Login credentials

- For the initial login it is possible to login as admin by using admin@admin.com as user name and admin as password.
- Then you will be directed to the system dashboard as below:

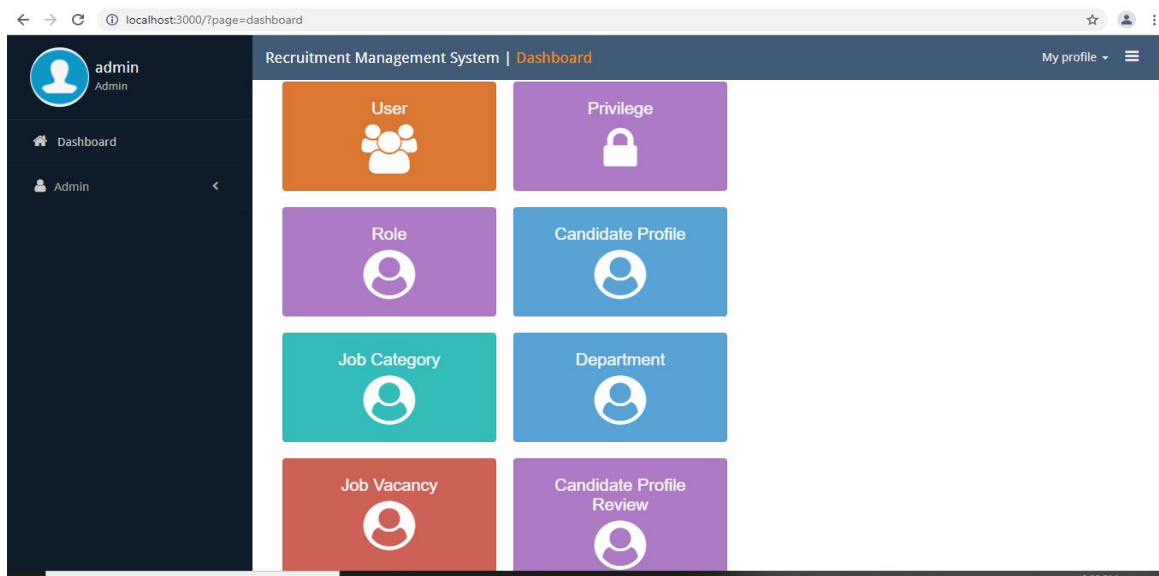


Figure C.3: System dashboard

- Then you can use each and every options to proceed. If it is needed to find something easily, you can use the search option.

This is the basic way to proceed to the system.

C.2 Maintain and modify data in profiles

C.2.1 Candidate Module

- Select Candidate Profile from the dashboard. Then it will directs to insert candidate information. There are space for personal details and qualifications.

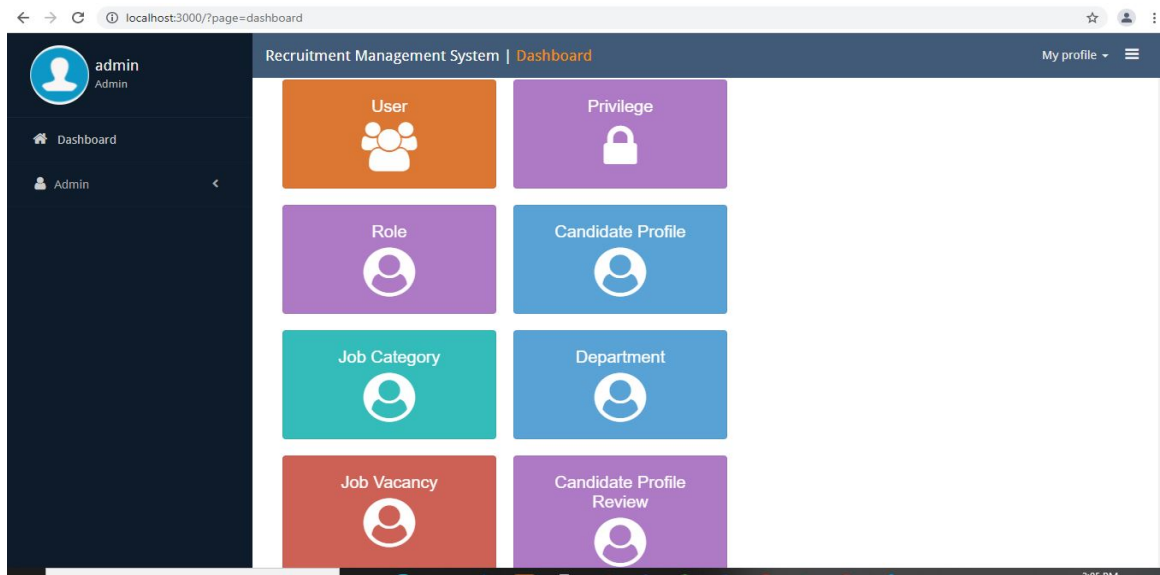


Figure C.4: System dashboard to select options

- This form will be provided to add user's personal details.

The screenshot shows a web browser window with the URL `localhost:3000/?page=candidate_profile`. The page title is "Recruitment Management System | Candidate Profile View". On the left, there is a dark sidebar with a user profile for "admin" and navigation links for "Dashboard" and "Admin". The main content area contains a form for editing a candidate's profile. The form fields are as follows:

- Photo:** A circular profile picture with a "Choose File" button and "No file chosen" text.
- Name with initials:** A text input field containing "D.K. Rathnayake".
- Name in Full:** A text input field containing "Dasun Kasange Rathnayake".
- Mobile Number:** A text input field containing "0713334321".
- Land Number:** A text input field containing "0362233214".
- Address:** A text input field containing "80/1, Colombo Rd., Rathnayake".
- NIC:** A text input field containing "980512436V".
- NIC Image / Proof:** A "Choose File" button with "No file chosen" text.

Figure C.5: Space for personal details

- This form will be provided to add user's qualifications.

The screenshot shows the same web browser window as Figure C.5, but with the form scrolled down to show the qualification and experience sections. The "NIC" and "NIC Image / Proof" fields are visible at the top. Below them are three sections:

- Education Qualifications:** A section with two rows. Each row has a "Name:" input field (containing "edu1" and "edu2" respectively) and an "Image / Proof:" input field with a "Choose File" button and "No file chosen" text. A red "x" icon is next to each "Image / Proof" field. A blue "Add Educational Qualification" button is at the bottom of this section.
- Professional Qualifications:** A section with a blue "Add Professional Qualification" button.
- Experiences:** A section with a blue "Add Experience" button.

At the bottom of the form, there is an orange "Update Profile" button.

Figure C.6: Space for qualifications

- This will provide ability to access user inserted data as a common list view.

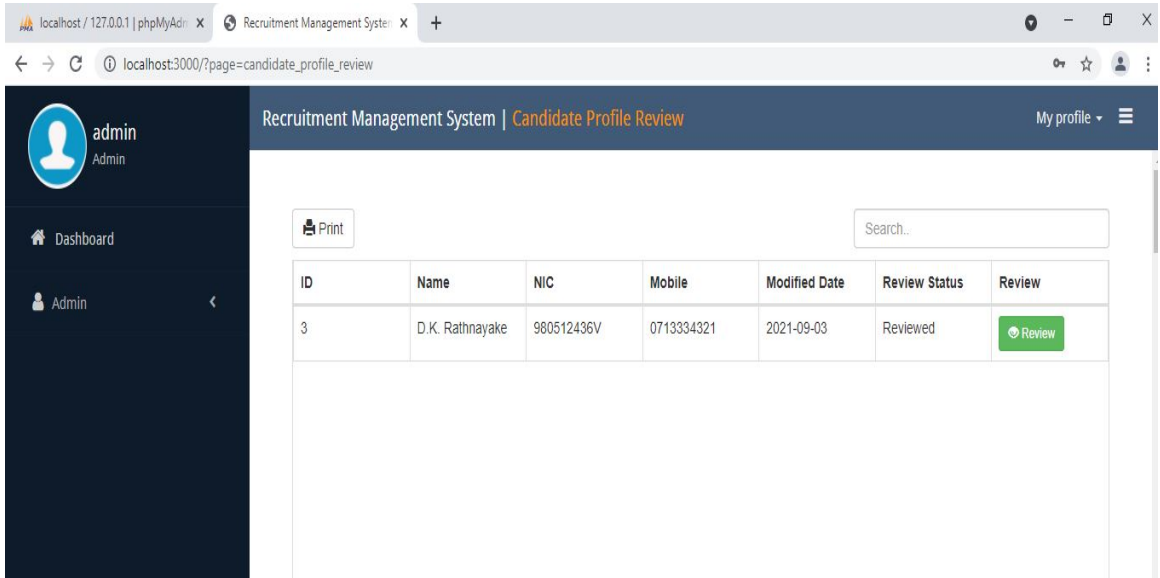


Figure C.7: Space for review information list

- This will provide ability to view and edit user inserted data.

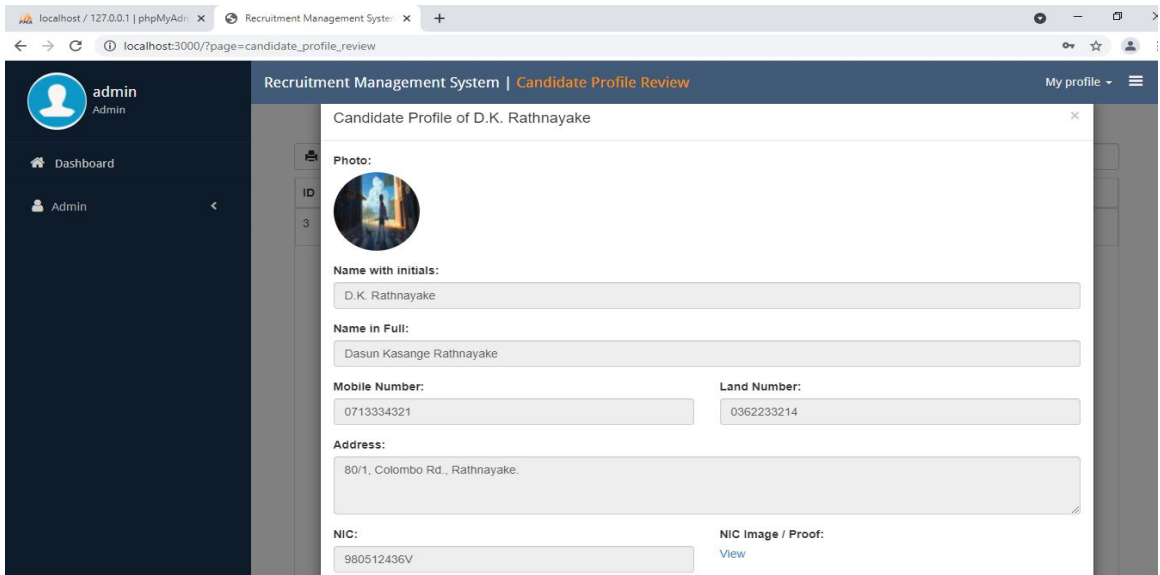


Figure C.8: Space for review information personal view

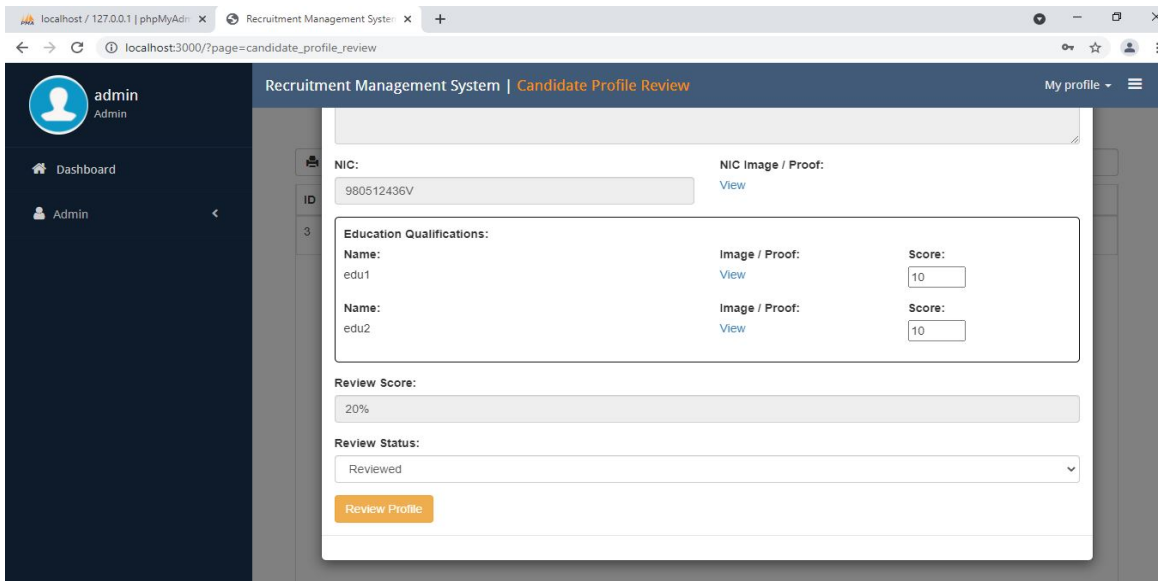


Figure C.9: Space for update information

C.3 Common buttons and validations used

In this system it provides common buttons and validation messages. This common theme has been used to make the system more user friendly.

C.3.1 Common buttons used

- Basic buttons used for Edit and Delete:



Figure C.10: Commonly used Edit and delete button

- Basic buttons used for Add and Reset:



Figure C.11: Commonly used Add data and Form Reset

C.4 Error validation for system

In this system it has used basic error handling messages. This has been used to guide user easily. It will help to identify the issue of inserting data, when a person is using the system.

C.4.1 Common field validations

- Fields will provide red cross symbol when user insert a incorrect value to the fields.

Username: *


Password: *

Figure C.12: Common field validation

- System will provide green tik mark for successful field completion.

Username: *

Password: *

Figure C.13: Common field validation for success

C.4.2 Common message validation for system

- Common message for delete validation.

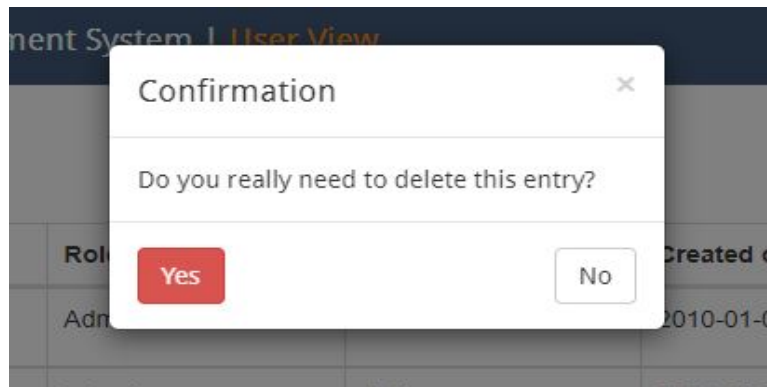


Figure C.14: Common message for delete confirmation

- Common message for function success.



Figure C.15: System validation for successful function

- Common message for common field validation.

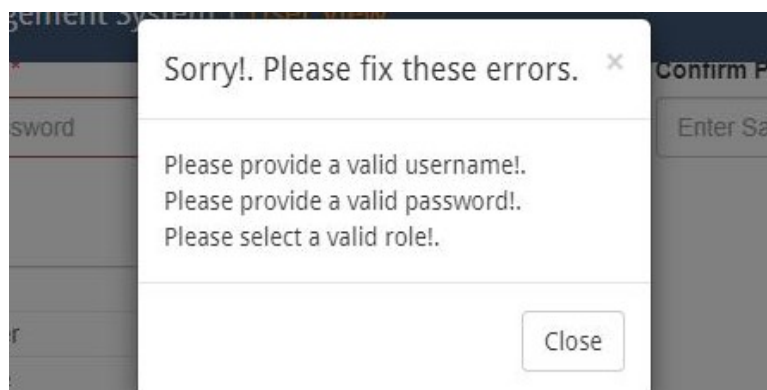


Figure C.16: System validation for failed function

Appendix D

Management Reports

This Recruitment Management system contains features. Reports also one of them. This was included because it helps for management decision making. Reports helps to predict and guess future decisions by considering their overall upgrading and degrading.

D.1 Report Types

Applications Report The complete list of your applicants and their respective profiles.	Download CSV
Jobs Report Each job's candidate flow, status and other essentials.	Download CSV
Positions Report Detailed information about your jobs and the positions within them.	Download CSV
Job Fields Report Active Job Fields for each job.	Download CSV
Hiring Team Report Members of each of your Hiring Teams by role.	Download CSV
Interviews Report Detail information about interview schedules, interviewers and their feedback.	Download CSV

Figure D.1: Report List View

D.2 Job posting against departments report

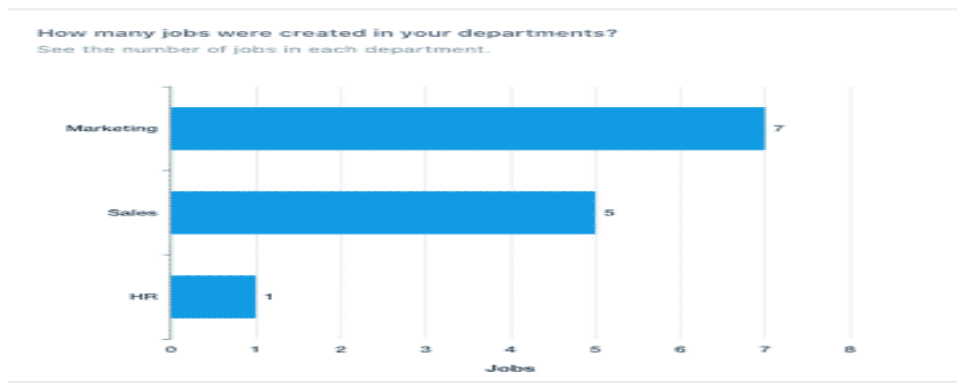


Figure D.2: Job posting against departments

D.3 Evaluation detail report

Reports

All Reports > Candidate Quality by Referrer

Candidate Quality by Referrer

Filters and more (3 jobs)

Credited To	Application Review	Phone Interview	Face to Face	Hiring Manager Review	Offer	Hires	Quality
Total	19	2	12	5	6	3	37%
Emma Hand	1	0	1	0	0	0	25%
Jovie Lanier	1	0	1	1	1	1	100%
Margot Dean	3	2	2	0	1	1	50%
No referrer	14	0	8	4	4	1	30%

Figure D.3: Job posting against departments

Appendix E

Use Case Narratives

In Design chapter it has been provided all the related design diagrams. There were main Use Case diagrams. Below are some of the basic example use case narratives.

E.1 Use Case Narrative for System Login

E.1.1

Use Case	Login to the system
Actor	All Actors
Description	User Login to the system
Preconditions	1. User account must be present. 2. User name and password should be correct
Post-Conditions	1.Appropriate dashboard is displayed to the user.
Flow of events	1. User enter user name and password 2. System validate and check user details. 3. User get redirected to the dashboard.
Alternatives	1. Error message getting displayed due to invalid credentials.

Table E.1: Use Case Narrative for System Login

Use Case	Add new candidate
Actor	Candidate
Description	User Login to the system
Preconditions	1. User account must be present. 2. User name and password should be correct
Post-Conditions	1. User should be added to the database.
Flow of events	1. User enter user name and password 2. System validate and check user details.. 3. User get redirected to the dashboard. 4. User should navigate to the Candidate profile page 5. User should be able to insert personal details 6. User should be able to insert qualifications 7. User details inserted to the database tables
Alternatives	1. Error message getting displayed due to invalid credentials.

Table E.2: Use Case Narrative for Candidate Profile Creation

E.2 Use Case Narrative for Candidate Profile Creation

E.3 Use Case Narratives for Job posting

E.3.1

Use Case	Add new job post
Actor	Admin
Description	Create new job posting
Preconditions	1. User should be logged in 2. User should have access job posting portal 3. Job posting portal should be open 4. Before posting relevance details should be provided
Post-Conditions	1. Posted job should be added to the system and visible to the users.
Flow of events	1. Select add post option 2. Select post 3. Click submit
Alternatives	1. User gets a permission error due to lack of privileges

Table E.3: Use Case Narratives for Job posting

Appendix F

Test Cases and Evaluation Results

For a system, it is more important to reduce errors that could be occurred. In this situation, testing plays a great role. In the testing and evaluation chapter, it has been discussed the test plan and the tests were used to account. Below are examples for implementing test cases and evaluations.

F.1 common test cases

F.1.1 Navigation related testing

Test Case Id	01		
Test Component	Dashboard		
Module	Dashboard		
Test Case	Dashbord links		
Expected output	Highligheted fields when cursor provided and when click, link to the expected page		
No	Test Case	Actual output	Status
01	Navbar arrow expand function	List should be expand	Pass
0 2	Navbar arrow close function	List should be closed	Pass
0 3	User Link	Link should be directed to the User Portal	Pass
0 4	Privilege Link	Link should be directed to the User Privelege Portal	Pass
0 5	Role Link	Link should be directed to the Role Portal	Pass
0 6	Candidate Link	Link should be directed to the Candidate Portal	Pass
0 7	Job Category Link	Link should be directed to the Jon Category Portal	Pass
0 8	Department Link	Link should be directed to the Department Portal	Pass
0 9	Job Vacancy Link	Link should be directed to the Job Vacancy Portal	Pass
10	Candidate Profile Review Link	Link should be directed to the Candidate Profile Review Portal	Pass

Table F.1: Common Test Result 02

Test Case Id	02		
Test Component	Common Functions		
Module	Common Functions		
Test Case	Common Functions		
Expected output	System should satisfy the customer required functions successfully		
No	Test Case	Actual output	Status
01	Login time out	Idle for 15 minutes logout automatically	Pass
0 2	Log out	Click logout button. Logout user and direct to the login page.	Pass
0 3	Test with browser Access	load system in different browsers eg. Firefox, Chrome, Opera,Edge	Pass
0 4	DB connection	Give inaccurate database details such as db name, db password and port	Pass

Table F.2: Common Function Test Result 01

F.1.2 Common Functions testing

F.2 System Evaluation

System have been provided for 10 different users. Then their feed-backs were taken using this questionnaire. Below are the results of user evaluation.

F.2.1 Questionnaire used

Below shows the sample questionnaire used for system evaluation.

Recruitment Management System - User Evaluation Form					
Name:			Date:		Designation:
I appreciate your kind attention provided for providing details for evaluating this system					
Please provide "X" for relevant selection					
No:	Question	Strongly agree	Agree	Disagree	Strongly Disagree
01	Interfaces are simple and understandable				
0 2	System display information more familiar				
0 3	Easily navigate through system				
0 4	Functions are more reliable				
0 5	System provide sufficient information				
0 6	Degree of Information provided in Reports.				
0 7	Ability to maintain data, to keep it up-to date.				
0 8	Ease of entering/ handling form fields, and handle manipulations.				
Comments					

Table F.3: Sample Questionnaire

F.2.2 User feedback results

Then above mentioned form have been provided for selected 10 users. After that below information have been identified.

Recruitment Management System - User Evaluation					
System Feature	Results				Success percentage
	4	3	2	1	
Interfaces are simple and understandable	7	3			92.5
System display information more familiar	8	2			95
Easily navigate through system	9	1			97.5
Functions are more reliable	7	3			92.5
System provide sufficient information	6	2	2		85
Degree of Information provided in Reports.	6	3	1		87.5
Ability to maintain data, to keep it up-to date.	7	3			92.5
Ease of entering/ handling form fields, and handle manipulations.	8	2			95
Comments					

Table F.4: Sample Questionnaire

F.2.3 User feedback results - graph

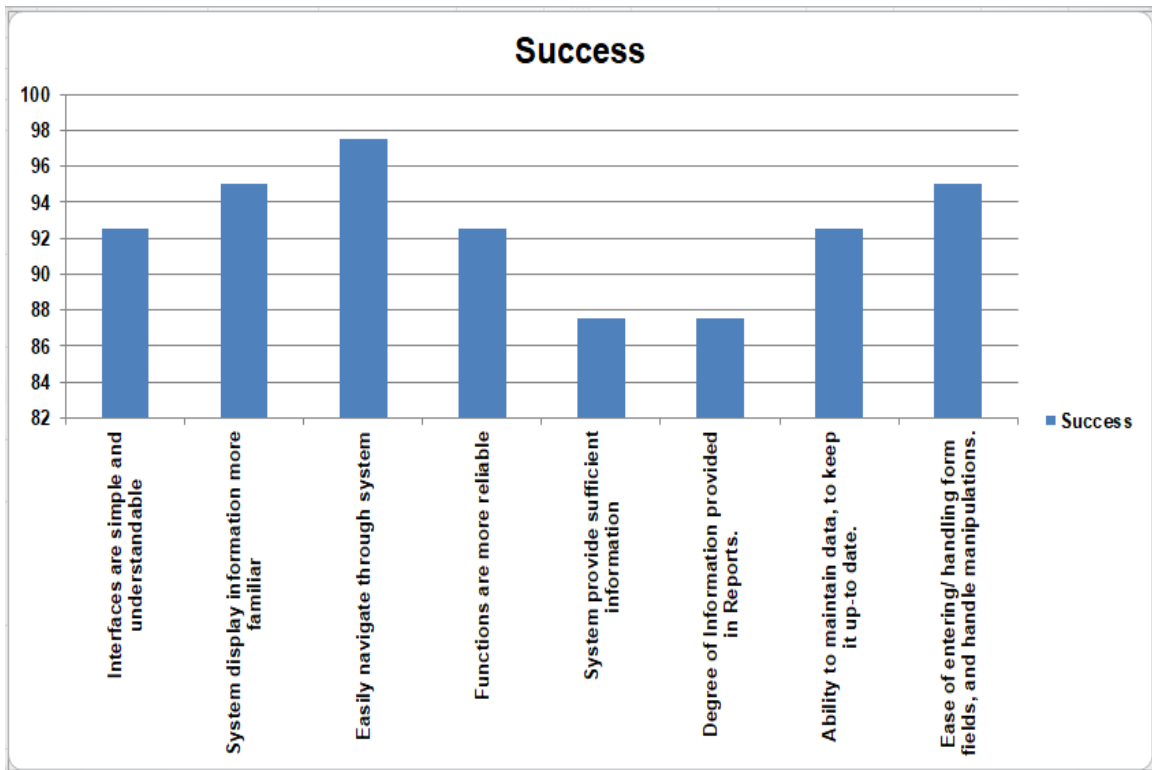


Figure F.1: User evaluation graph

Appendix G

Code Listing

MVC design pattern alongside **OOP!** (**OOP!**) was used when coding the system. Using that method, it was possible to keep a clear separation between display logic, business logic, and database interactions. Following this structure, the proposed system was developed in three layers called Model Layer, View Layer, and Control Layer. First and foremost, Model layers or simply Models were used to interact with data. Throughout the project, this layer was used to govern the procedures to access data objects and perform any kind of operations on them. This layer is independent of the other two layers.

After that, View Layer was implemented with the help of methods in models to collect data with the purpose of presenting them. It should also be mentioned that this layer reflects the changes happening in the model layer. This means a state change in models causes a state change in views that depends on that particular model. Finally, Control Layer was used to sit inbetween Models and Views to facilitate a twoway communication mechanism. Simply put, this is the layer that uses methods defined in models to query required data for views. This layer also contains logic that controls how each and every data flow happens and how to respond to unexpected events.

G.1 Model Layer implementation

A graphical tooled called "MySQL Workbench" was used to design this layer. Also, as the RDBMS MySQL Server was selected. After that, TypeORM was used to map database tables with models (model classes) in order to em-

phasize relationships such as onetoone, onetomany and manytomany. Furthermore, TypeORM provided features for performing CRUD operations as a part of the mapping process and it helped to manipulate the database in an abstract manner. Below are the two example codes for data insert and database table loading operation.

```

1
2 const showNewEntryModal = () => {
3   mainForm.reset();
4   $("#mainForm #id").val("ID will be displayed after adding."
5     );
6   // set date of adding
7   $("#mainForm #addedDate").val(new Date().today());
8   const username = mainWindow.tempData.profile.username;
9   $("#mainForm #createdUser").val(username);
10  $("#modalMainFormTitle").text("Add New Job Vacancy");
11  $("#modalMainForm").modal("show");};
12 // load main table
13 const dataBuilderFunction = (responseData) => {
14   // parse response data and return in data table friendly
15   // format
16   return responseData.map((entry) => {
17     return {
18       ID: entry.id,
19       Name: entry.title,
20       Position: entry.position,
21       Department: entry.department.name,
22       Category: entry.jobCategory.name,
23       Status: entry.jobVacancyStatus.name,
24       View: '<button class="btn btn-success btn-sm" onclick
25         ="showEditEntryModal(\'${entry.id}\', true)"><i
26           class="glyphicon glyphicon-eye-open" aria-hidden="
27             true"></i> View</button>',
28       Edit: '<button class="btn btn-warning btn-sm" onclick
29         ="showEditEntryModal(\'${entry.id}\')"><i class="
30           glyphicon glyphicon-edit" aria-hidden="true"></i>
31         Edit</button>',
32       Delete: `${
33         entry.jobVacancyStatus.name == "Deleted"? '<button style
34           ="display:none">Delete</button>': '<button class="btn
35             btn-danger btn-sm" onclick="deleteEntry(\'${entry.id}\')
36           "><i class="glyphicon glyphicon-edit" aria-hidden="
37             true"></i> Delete</button>'
38       }`,
39     };
40   });
41 };

```

Listing G.1: Model layer implementation

G.2 View implementation

View Layer consists of interfaces, users can interact with. In a sense, this can be described as a visual representation of logic and models in the backend of the system. This layer provides facilities for endusers to use the system and carry out various tasks through user interfaces such as different views, forms, and tables. Below is the common button class use for the development.

```
1 <div id="fmButtons">
2   <button type="button" class="btnFmAdd btn btn-success">Add
3     </button>
4   <button type="button" class="btnFmUpdate btn btn-warning">
5     Update</button>
6   <button type="button" class="btnFmReset btn btn-primary"
7     style="float: right; margin-left 10px">Reset <button>
8   <button type="button" class="btnFmDelete btn btn-danger"
9     style="float: right"> Delete</button> </div>
```

Listing G.2: View layer implementation

Controller implementation

The layer that sits between the data layer and the interface layer is the control layer. The control layer handles user requests, validate those, and determine what the user is trying to do. Then, it interacts with the data layer if some data need to be obtained and then send a response back to the user. This response will determine the next view user should see.

The sequencing of calls to Models, and/or the sequencing of views and required input from the user defines the application's workflow. Since the control layer acts as a bridge between user and data, the workflow of the application is defined in the control layer of the application. Below is the commonly used reusable code snippets for controller files.

```
1 private static async getOne({ id }) {
2   const entry: any = await getRepository(JobVacancy)
3     .findOne({
4       where: { id: id },
5       relations: ["user"],
6     })
7   .catch((e) => {
8     console.log(e.code, e);
9     throw {
10    }
```

```
11     status: false,  
12     type: "server",  
13     msg: "Server Error!. Please check logs.",  
14 };  
15 });
```

Listing G.3: Controller layer implementation

6.3 Glossary

Sample Virtual Machine (svm)

Actual or anticipated questions which may be posed to an information retrieval system. (Functional requirements)

Quality features of the system. Which are unable to measure directly (Non-functional requirements)

A system involving data processing which does not make use of stored program computing equipment. (Manual system)

CRUD or C.R.U.D (Create, Retrieve, Update and Delete) operations are the basic operations in any database. (CRUD operations)

Method of implementing a process. (Process modelling)

A webbased solution is a way of implementing a solution to a computing issue or business issue through doing some work on the net. (Webbased system)

A special data processing system, or part of a data processing system, which aids in the storage, manipulation, reporting, management, and control of data. (Database management system)