The World Islamic Sciences and Education University

جامعة العلوم الاسلامية العالمية

Faculty of Information Technology

كلية تكنولوجيا المعلومات

GRADUATION PROJECT

Title

Human Recourses Management System

Students

Haya Waleed Alsughair	3190601078	Computer Science
Mohammed Jehad Al-Ashqar	3190601027	Computer Science
Rahaf Raed Jasser Abed	3190605075	Software Engineering

Supervisor

Dr. Malek Almomani

SEMESTER I

2022/2023

Acknowledgements

The accompanying satisfaction the successful of completion of any task in this project so far would be incomplete without the sincere expression of gratitude to the group of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all the efforts with success. First and foremost, we would like to express our sincere gratitude to Allah for helping us finishing our graduation project. Second, we extend our thanks and appreciation to our project supervisor **Dr. Malek A. Almomani** who provided us with a lot of his knowledge, experience and encouragement for that our project sustained throughout real life, Also We extend our thanks to our families with their support and continuous encouragement and we would like to thank all our close friends. We the members of the project would like to thank our university for allowing us to be in this university and to thank the faculty members for the knowledge they gave us to reach the end of our journey with the bachelor's degree. Last but not least, deepest thanks go to everyone who has helped me in various ways for support and encouraging me in making this thesis real. God bless you all.

Abstract

Context: Recently, most of manual systems are forced to move into electronic one. Many reasons to that, such as green technology and for more performance. Therefore, it is required to build a system that helps faculty members to obtain in order to perform the internal and external human resource activities. **Objective:** The main objective of this study is: “To develop software that allows HR to access the information of employees & to help recruiters Tracking in an orderly and smooth manner”. **Methods:** The system was created by following the scrum approach for managing the system development process. It required six sprints’ cycles to complete the ForEach HR system. **Outcomes:** To ensure the system behaves, the system was tested in two-level. First Level, unit, integration, and system testing where every piece of software that can be compiled is tested. Second Level, user acceptance testing was conducted by using a questionnaire approach to validate and verify the system. **Conclusion:** The system provides great benefit to faculty members of the Companies in terms of facilitating the process of Hiring and Attending process the ForEach HR in an easy way.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

HRMS	:	Human Recourses Management Systems
HR	:	Human Recourses
CRUD	:	Create, Read, Update, and Delete
HTML	:	Hyper Text Markup Language
CSS	:	Cascading Style Sheet
JS	:	Java Script
ERD	:	Entity Relationship Diagram
DFD	:	Data Flow Diagram
UML	:	Unified Modeling Languages
STM	:	Software Testing Methodology
UT	:	Unit Testing
VS	:	Visual Studio

CHAPTER 1

INTRODUCTION

1.1 Overview

In Jordan Companies We are searching for a platform that handles all the process of HR. In easy, practical, and efficiency way so the employees can access to see their profiles
and can change their information early in a practical method.

HR Management Systems (HRMS), an HR system is a business application designed to manage the complete employee lifecycle from recruitment and onboarding through core HR administration to employee engagement and performance management.
Is a software application used to store employee information and support various human resource functions, such as attendance, recruiting, and board. HR requires tools to help them hire, and manage employees the success of an organization depends on it. .HR project management is a smarter way of working that enables HR teams to carry out their tasks and plans in an organized, efficient manner.

Web Technology refers to the Various tools and technique that are utilized in the process of communication between different types of devices over the internet, Also the way of Computers/ devices communicates with each other using markup languages. Web browser is used to access web pages. Web browsers can be defined as programs that display text, data, pictures, animation, and video. Hyperlinked resources on the World Wide Web can be accessed using software interfaces provided by web Browsers.

HR mangers in Jordan Companies miss some feature in all the Human Recourses management systems, so we are handling to integrate all the main basic features in one HR system, so the HR can use a single platform without being scattered with different managements Styles.

1.2 Problem Statement

At HR systems There is difficulty of having resources in one place in the HR system promotes different advantages over other systems, so some HR systems have some problems in tracking jobs, applying attendance employees and Managing their Task and viewing employees in terms of actual time of work. Therefore, we seek to develop a system that includes all these features in one system. Work simplification makes it easier for companies to organize their work into a single framework. According to previous discussed difficulties we will handle all the main functionalities of all Human Recourses management system.

1.3 Project objectives

1.3.1. Main Objectives

The main objective of this study is: “To develop software that allows HR to access the information of employees & to help recruiters Tracking in an orderly and smooth manner”.

1.3.2. Sub objectives

1. To understand the current situation of the project scope and problem statement.
2. To develop software that allows HRs to access the information of employees & to help recruiters in Tracing the jobs
3. To validate a simple system for helping HRs

1.4 Research strategy (Framework)

A Research Strategy is a step-by-step plan of action that gives direction to your thoughts and efforts, enabling you to conduct research systematically and on schedule to produce quality results and detailed reporting [1]. Human Resources management system (HRMS) has been developed on the Scrum Framework, Scrum is an agile development methodology used in the development of Software based on an iterative and incremental processes. Scrum is adaptable, fast, flexible, and effective agile framework that is designed to deliver value to the customer throughout the development of the project. The primary objective of Scrum is to satisfy the customer's need through an environment of transparency in communication, collective responsibility, and continuous progress. The development starts from a general idea of what needs to be built, elaborating a list of characteristics ordered by priority (product backlog) that the owner of the product wants to obtain. Agile Scum Teams work together to deliver value at the end of every sprint in an incremental form. So, deployment is basically a process to take your code and put it into satellite which is connected. Scrum is precisely an evolution of Agile Management; scrum methodology is based on a set of very defined practices and roles that must be involved during the software development process. It is a flexible methodology, scrum is executed in temporary blocks that are short and periodic, called sprint, which usually range from 2 for 4 weeks which is the term for feedback and reflection.

Scrum approach has been developed for managing the systems development process, main idea of scrum is that systems development involves several environmental and technical variables that are likely to change during the process. This makes the development process unpredictable and complex, requiring flexibility of the system development process for it to be able to respond changes, Scrums helps to improve the existing engineering practices in an organization, for it involves frequent management activities aiming at consistently identifying any deficiencies or impediments in the development process as well as the practices that are used.

There are three phases in Scrum. The first is an outline planning phase where you establish the general objectives for the project and design the software architecture. This is followed by a series of sprint cycles, where each cycle develops an increment of the system. Finally, the project closure phase wraps up the project, completes required documentation such as system help frames and

user manuals, and assesses the lessons learned from the project. The innovative feature of Scrum is its central phase, namely the sprint cycles. A Scrum sprint is a planning unit in which the work to be done is assessed, features are selected for development, and the software is implemented. At the end of a sprint, the completed functionality is delivered to stakeholders. As shown in Figure 1.1

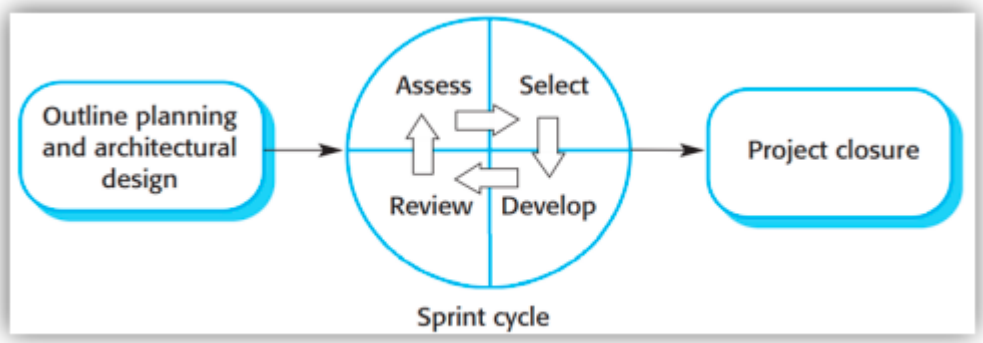
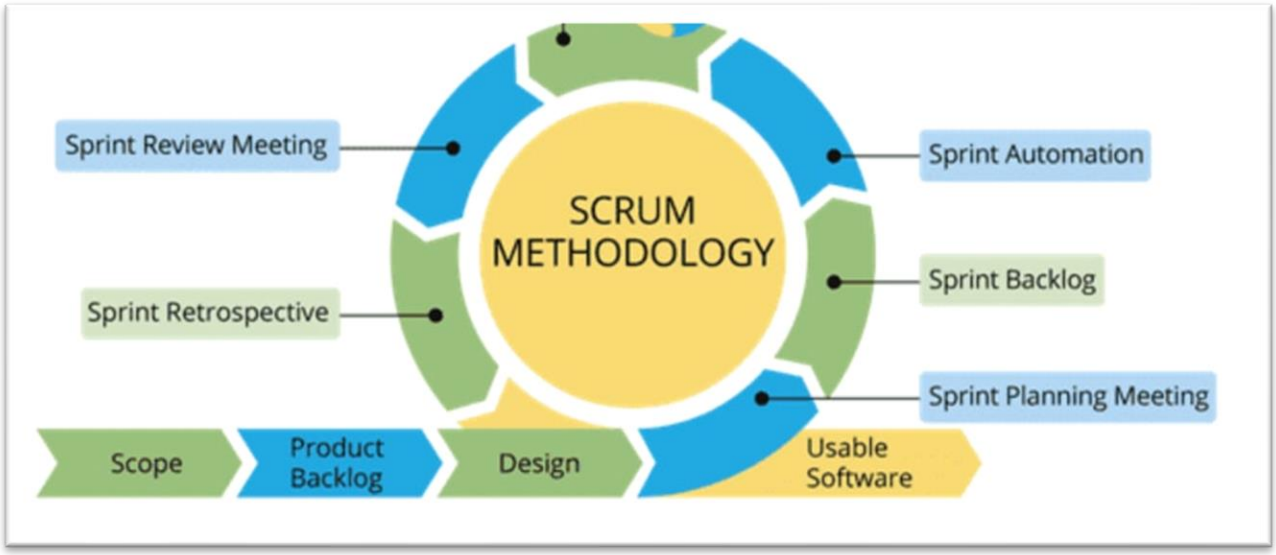


Figure 1.1: Sprint Cycle

Scrum methodology is used mainly for software development, but other sectors are also taking the advantages of its benefits in implementing this methodology in their organization model such as sales, marketing, and HR Teams. **Benefits of Scrum Methodology:** *Easy Scalable *Compliance of expectations *Flexible to change *Time to market reduction *Higher software quality *Timely prediction *Reduction of risk.

The system is developed in Sprints. Sprints are iterative cycles where the functionality is developed or enhanced to produce new increments. Each Sprint includes the traditional phases of software development: requirement, analysis, design, evolution and delivery phases. The sprint is planned for one week to one month. Also, there may be more than one team building the increment. Scrum is characterized by a sprint, an iteration of between one and four weeks. Sprints are time-boxed in that they are of a fixed duration and the output of a sprint is what work the team can accomplish during the sprint. The delivery date for the sprint does not move out. This means that sometimes a sprint can finish early, and sometimes a sprint will finish with less. Functionality than was proposed. A sprint always delivers a usable product as shown in Figure 1.2.

Figure 1.2:Scrum Development Process



The Primary objective of Scrum is to satisfy the customer's need through an environment of transparency in communication, collective responsibility and continues progress. The development starts from a general idea of what needs to be built, elaborating a list of characteristics ordered by priority (product backlog) that the owner of the product wants to obtain. Starting with product backlog creation which consist of prioritized user stories, moving to sprint planning and sprint backlog creation, also working on the sprint and discusses it during the scrum meeting. Testing and Product demonstration and after the retrospective and the next sprint planning.

1.5 Scope (boundary)

The project Scope is the total amount of work that needs to be done to complete a project.
It's also one of the Triple constraints of project Management. Defining a Scope is part of the project planning process that helps project managers determine what is the project goals, deliverables, tasks, costs, and deadlines.

- ✓ **Project Goals &Objectives:** The project requirements or Acceptance Criteria
- ✓ **Project Deliverables:** The outcomes of the Projects tasks.
- ✓ **Project Exclusion & constraints:** As a project Manager you should explain what can't be done and why.
- ✓ **Project Assumption:** Some initial assumptions that the project management team has before executing the work.
- ✓ **Project Milestone:** These mark important moments in your project life cycle, as the end of the phase.
- ✓ **Scope Baseline:** Your Original Scope as you planned it. The scope baseline allows you to compare actual results against what it's in your scope statement.

In the bellow template it represents all the required information's for the project boundaries and all the iteration as shown in Table 1.1:

Table 1.1: Project Scope Statement

Project ID	123598-85	**Date**	16/10/2022
Touchpoint	Website	**Finish Date**	19/1/2023
Prepared by	Team		
Project Name	Human Recourses System (For Each HR).		

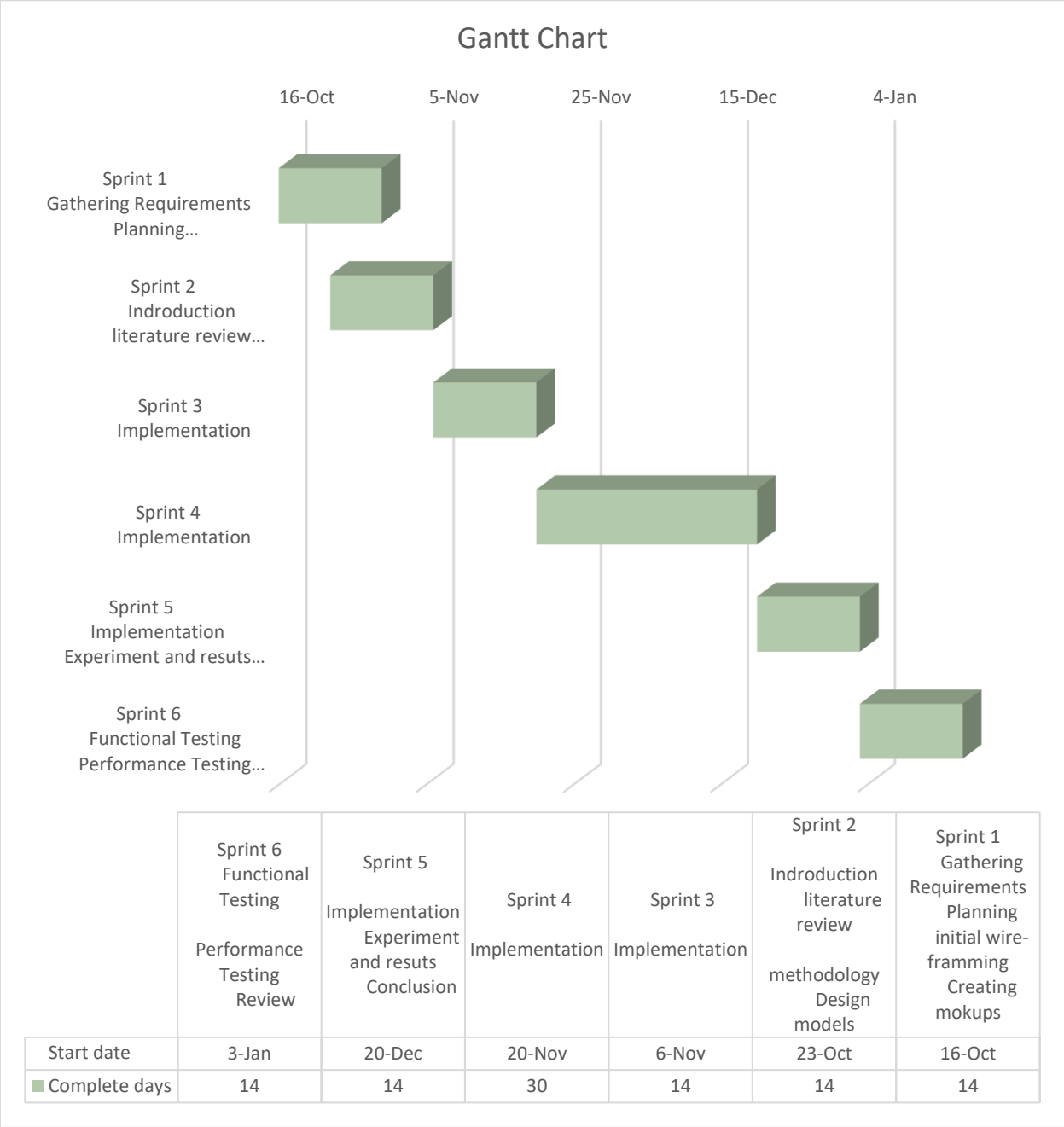
Project Objectives	The main objective of this study is: "To develop software that allows HR to access the information of employees & to help recruiters Tracking in an orderly and smooth manner". ➢ To understand the current situation of the project scope and problem statement. ➢ To develop software that allows HRs to access the information of employees & to help recruiters in Tracing the jobs ➢ To validate a simple system for helping HRs
Product Scope Definition	Personnel aspect-This is concerned with manpower planning, recruitment, and development , Industrial Relations: Since it is a highly sensitive area, it needs careful interactions with labor or employee unions, addressing their grievances and settling the disputes effectively in order to maintain peace and harmony in the organization. The main aim is to safeguarding the interest of employees by securing the highest level of understanding to the extent that does not leave a negative impact on organization. It is about establishing, growing and promoting industrial democracy to safeguard the interests of both employees and management.
Project Requirements	**1. Identifying stakeholder requirements** Document Management, Applicant Tracking and Recruiting, Employee Onboarding and Administration, performance management, Time and Attendance management, Reporting and dashboard, Task List / To Do and managing employees. **2. Documenting stakeholder needs and expectations** Every Step will be considered with all results and expectations using documented. **3. Managing requirements throughout the project** We are using Agile Scrum; the managing process is with sprints.

Project Boundaries	➢ **In Scope**: Dashboard, My Profile, Attendance, Recruitment, Onboard, Retrieving Emails, To-Do, and Customize Skills. ➢ **Out of Scope:** Biometric Authentication, Workflow, Payroll, Evaluation, and Training.
Project Deliverables	Checking all the requirements within the scope of the project with identifying the progress of work: you can check the documents, 1-HR can manage recruiters in efficient way 2- HR can manage departments. 3- HR can check the Attendance list. 4- HR manage all his tasks / see performance analytics
Product Acceptance Criteria	**Given** the HR navigate to the system and **when** the applicants fill the form, **Then** the system will redirect the HR to the recruitment process to recruits the applications within the steps and the applicants become employees / and users on the system to start, so all users of the system can take all the benefits of a system.
Project Constraints	Finishing the project: in 19/1/2023. Daily meeting within 15min to discuss what is going on.

Project Assumptions	➢ You'll have access to all the resources you need to complete the project, both human and material. ➢ Project team members will have the resources they need to complete their individual tasks on time, from specialized equipment and software to electricity during working hours. ➢ Personnel costs will not change during the project cycle. ➢ Other material and resource costs will remain consistent throughout the project. ➢ The overall cost of day-to-day operations will not increase. ➢ All equipment will be in working condition through the project cycle. ➢ The scope of the project will not change throughout the life cycle.
Initial Project Organization	The members of the project team, as well as stakeholders, are identified. The organization of the project is also documented.
Schedule Milestones	The customer, Project Sponsor, or performing organization can identify milestones and can place imposed dates on those schedule milestones. These dates can be addressed as schedule constraints. ***On Chapter 1 Gantt chart***.
Fund Limitation	The limitation is to deliver a system with all the project scopes within the required date.
Approval Requirements	Every step is documented and approved from all sides.

1.5 Gant chart

A Gantt chart is a project management tool assisting in the planning and scheduling of projects of all sizes, Project management timelines and tasks are converted into a horizontal bar chart, showing start and end dates, as well as scheduling and deadlines, including how much of the task is completed per stage as shown in figure 1.3:



	Sprint 6 Functional Testing Performance Testing Review	Sprint 5 Implementation Experiment and resuts Conclusion	Sprint 4 Implementation	Sprint 3 Implementation	Sprint 2 Indroduction literature review methodology Design models	Sprint 1 Gathering Requirements Planning initial wire-framming Creating mokups
Start date	3-Jan	20-Dec	20-Nov	6-Nov	23-Oct	16-Oct
Complete days	14	14	30	14	14	14

Figure 1.3:Gantt Chart

1.7 Project outline

Chapter 1: A summary of the problem to be solved, the objectives of the project to be built and the action plan with a clarification of its steps

Chapter 2: Comparing the system with other existing system

Chapter 3: Feasibility study and methodology used in the project, functional and non-functional requirements.

Chapter 4: Show how the system works by a set of diagrams in the easiest and simplest way and show how to use the system.

Chapter 5: In this chapter, we will talk about implementation and evaluation. With an explanation of each implementation and how it was implemented, also an overview experiments and results of the project testing methodology.

Chapter 6: A summary of the project and its future work that can be applied.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This is chapter contains literature and information on the occupational stress and what the relationship with organizational commitment, which this research seeks to improve. We introduce the related work. The related work section may also be called a literature review. The point of the section is to highlight different works that have similar functionalities, and to discuss the current scenario for our project. This chapter presents an overview of the importance of the HR properties, and the characteristics of the current systems, this chapter also presents the critical success features for the development system to fill the research gap. Furthermore, this site is proposed to save time and effort in helping Companies to manage all tracking requitement process in HR/Manger Dashboard, and helping employee to view his data smoothly. We support the idea that all these processes should be based on one platform, the last section of this chapter explain the result of the gap analysis.

2.2 Related Work

The relevant work section is essential for most research articles. It is also one of the most important sections to install while the paper is out for review. A comparative study is an act of comparing two or more things to discover something about one or all of the things being compared. The comparative study helps to define the organizational structure of the subjects as well as give the differential points between the subject's matter [2]. My project has been compared to some similar websites such as the following:

- ZenHR:

  ZenHR is a human resources solutions group that is committed to providing a variety of cloud-based products to help you streamline and simplify your recruitment process from "Acquire" to "Retire". Our Mission is to facilitate the hiring process from acquire to retire through our cycle of Cloud-based products. Our Vision is to become the leading HR Solutions company, offering top-notch HR Services globally.

- Monday:

  Monday is a cloud-based platform that allows users to create their own applications and project management software, is the Employee management system that tracks your talent pipeline and engages employees.

2.2.1 Functionalities of ForEachHR System

- Dashboards:

  A dashboard is *a visual display of all of your data,* dashboard is a type of graphical user interface which often provides at-a-glance views of key performance indicators relevant to a particular objective or business process. For example, an HR and Manger have a different screen in the dashboard not same as the Employee and the Team Leader of the Teams.

- Applicant Forms:

  An application form is an official document that employers want their job candidates to fill in while applying for a job. The employer would ask a series of questions that candidates must answer.

- Authentication process:
  - Home page:

    A home page is the primary web page that a visitor will view when they navigate to a website via a search engine, and it may also function as a landing page to attract visitors. In some cases, the home page is a site directory, particularly when a website has multiple home pages. Good home Page design is usually a high priority for a website.
  - Login Page:

    The login page allows a user to gain access to an application by entering their username and password or by authenticating.
  - Password-based:

    Passwords can be in the form of a string of letters, numbers, or special characters. To protect yourself you need to create strong passwords that include a combination of all possible options.

- Token-based:

  Token-based authentication technologies enable users to enter their credentials once and receive a unique encrypted string of random characters in exchange. You can then use the token to access protected systems instead of entering your credentials all over again. The digital token proves that you already have access permission. Use cases of token-based authentication include RESTful APIs that are used by multiple frameworks and clients.

- Recruitment Process (Recruitment Tracking):

  Recruitment is the overall process of identifying, sourcing, screening, shortlisting, and interviewing candidates for jobs within an organization. Recruitment also is the processes involved in choosing individuals for roles. For example, an applicant filled the form, a series of steps will be performed to join a work.

  - Hiring Process: The HR is responsible for hiring new employees. The HR will review applications from the applicants and select the best candidate for the job. The HR will then contact the selected applicant and arrange an interview. After the interview, the HR will make a discission on whether to hire the applicant or not. If the applicant is hired, the HR will the inform the employee and arrange for the necessary paperwork to be completed.
  - Onboarding: Is the last Step after finishing the hiring process, an applicant will receive an approval email for the job within all required details.

- Profiles:

  personal details mean all information about You from which Your identity is apparent or can reasonably be ascertained; and includes Your name, address, contact telephone number and other details.

- Attendance:

  Attendance is the concept of people, individually or as a group, appearing at a location for a previously scheduled event. Measuring attendance is a significant concern for many organizations, which can use such information to gauge the effectiveness of their efforts and to plan for future efforts. is the presence of your employees at their designated worksite during the required hours, showing up for and fully working their scheduled shift for the

day. For example, an attendance system should be located in same location precented in the employee information, an employee can make the attending process and leaving process in efficient manner.

- Attending: Is the process where an employee can check-in and checkout for his work time.

- To-Task List:

  Task list is a document that details or lists out the duties that employees of a particular organization or project must complete. the primary contact for the project, the professionals working on it and status update, essential if you're going to beat work overload. For example, this section will help employees to manage their tasks in their daily work with hiring them in priority levels, as a HR can see on his task list all his interviews within one way to organize all his work within a day.

  - CRUD Task: Is the process of create, read, update and delete for a specific task: as required in the website

- Customize Hiring:

  customize the hiring and recruitment stages for your job openings from start to finish. You can create your own unique filters throughout your hiring pipeline for a more streamlined shortlisting, interviewing, and hiring process, after a job applicant fill out all the required answers, the skills of the applicant will be customized to build, fit, or alter according to individual specifications. For example, an HR can reorganize the hired people in their skills as the chosen position.

- Managing Process:

  Employee management is an approach to supporting your employees to reach their peak performance and meet your company's goals. Employee management is a comprehensive process that encompasses practically all aspects of human resources. For example, an HR can see a hierarchy tree with all employee in an organization.

In the table below, we will present a group of systems and the services provided through their websites as Shown in Table 2.1: Related Work

Table 2.1: Related Work

Functions	ZenHR	Monday	ForEachHR
Authentication Process	✓	✓	✓
Hiring process	✓	✗	✓
Onboarding	✓	✗	✓
Attending	✓	✓	✓
CRUD Tasks	✗	✓	✓
Customize Hiring	✓	✗	✓
Managing Process	✗	✗	✓
Create Applicant	✓	✗	✓

(✓) Available (✗) Not available

From the literature review, we can indicate that there are no studies that provide all the required HR systems, unlike the ForEachHR System which provides all the services presented in the table within one site, from the previous table we can indicate that the web is valid.

2.3 Summary

After reviewing the application of similar application and taking the strengths of each application, we relied on establishing our current application and trying to create a better or equal application

CHAPTER 3

METHODOLOGY

3.1 Overview

This chapter includes defining the feasibility study, the tools we work with, the collected requirements (functional and non-functional), and the methodological process by which the project was implemented.

3.2 Feasibility Study

The main goal of the feasibility study is to assess the economic viability of the proposed business, the outcome of the feasibility study will indicate whether or not to proceed with the proposed venture. If the results of the feasibility study are positive, then the cooperative can proceed to develop a business plan, If the results show that the project is not a sound business idea, then the project should not be pursued [3]. Although it is difficult to accept a feasibility study that shows these results, it is much better to find this out sooner rather than later. The Feasibility Study is a critical document that defines the initial system concepts, objectives, requirements, and alternatives. The study also forms the framework for the system development project and establishes a baseline for further studies. Moreover, it is the initial study of the proposed plan or project. Before the start of a project and the investment of funds and time, the project manager shall analyze to assess the efficiency of implementing the project to ascertain the capacity. Besides, capabilities, funds, and manpower are required for the project and Requirements collected through the questionnaire.

3.2.1 Technical Feasibility study:

A technical feasibility study helps organizations determine whether they have the technical resources to convert the idea into a fully functional and profitable working system. The study reports whether there exists correct required resources and technologies which will be used for project development. In addition, look at the slight components of how you will convey an item or organization as Shown in Table 3.1: Technical Feasibility Study.

Table 3.1: Technical Feasibility Study

Items	Estimated cost (EC)
Training	FREE
Software and Hardware	50JD
Computer equipment	50JD
Software development	100JD
Having a device connected to the Internet	50JD
Courses	50JD
Cost Analysis	200JD
Design and Prototyping.	100JD
Other Expenses	80JD
Total Costs	680JD

3.2.2. Operational Feasibility study:

An operational feasibility study evaluates whether or not your organization is able to complete this project. This includes staffing requirements, organizational structure, and any applicable legal requirements. Belong to part of solving issues with the support of another proposed system. It helps in exploiting the open doors and satisfies the condition of classifying the time between improvements of the project as shown Table 3.2: Operational Feasibility Study.

Table 3.2: Operational Fisability Study

Items	Percentage (%)
Performance	85%
Information	80%
Services	90%
Control	75%
Efficiency	85%
Economy	65%

3.2.3. Economic Feasibility study:

This is one of the most important factors for the feasibility of the project. This factor is the sum of all project costs, and it is a measurement of the required efforts that the organization executing the project must make. This allows acknowledging if there is devaluation in costs as well as an increase in benefits. Likewise allows acknowledging if the cost-effective data administrations to the business as shown in Table 3.3: Economic Feasibility Study.

Table 3.3: Economic Feasibility Study

Items	**Estimated cost (EC)**
Dashboards	Free
Front-End	Free
Back-End	Free
Total Cost	Free JD

3.3 Risk Analysis

Risk analysis involves examining how project outcomes and objectives might change due to the impact of the risk event. Once the risks are identified, they are analyzed to identify the qualitative and quantitative impact of the risk on the project so that appropriate steps can be taken to mitigate them. The Scrum framework is the most widely used among agile methodologies in software project management It provides a set of good practices aimed at fast delivery value to the customers and can simplify the work process, reduce development time, and enhance organizational transparency The aim of this study is to understand how risk management is performed in projects that use the Scrum framework, with the following objectives as shown in Table 3.4: Risk Analysis.

- identify applied risk management practices in software development;
- conduct a case study in software projects that use Scrum to analyze the respondent's agreement with the risk management practices identified in scientific literature;
- analyze the agreement among the respondents and the standard respondent;
- rank the risk management practices according to the results.

Table 3.4: Risk Analysis

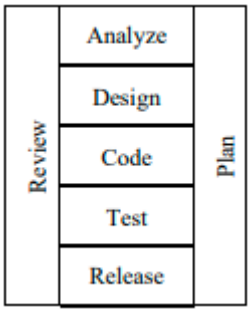
Category of Risk	Risk Event	Likelihood	Risk Level	Risk Owner	Risk Response	Trigger of Risk
Technical	Hacking Fraud, and stealing users' information's	Probable	High	Technical Team, Programmers, Security Team	Make Sure add more options for security, roles, and privacy	The system Contains of Evaluation and Users Data
	Project don't meet the functionality as expect	Possible	High	Requirement Analyst Team, Project Manger	Define Analyst requirements from target users using: 1-Questioners 2-Brain Storming 3-ProtoType for Project	A lot of Companies will use the System, and requirements specification not completed.
External	Customers uses Different Operating Systems	Probable	High	Technical Team, Programmers	Make sure the System works in different operating systems	Unknown which OS to use
	Wrong using from the Companies users	Probable	Medium	Requirement Analyst Team, Technical Team, QA Team	1-Provide Tips How to use 2-make sure all the test cases, and scenarios are executed correctly	This system will be used from different Users development Teams
Organization	Two Tasks Require the Same Resource for completion	Possible	Low	Requirement Analyst Team, Project Manager	Make sure no how tasks depend on same resources	Availability and provide Resources for all tasks
Project Management	Overhead Expected Costs and time of project	Possible	High	Requirement Analyst Team, Project Manger	Make sure requirements been executed and completed as possible	Poor Planning Requirements

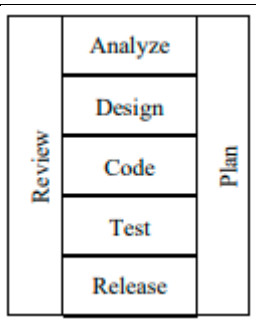
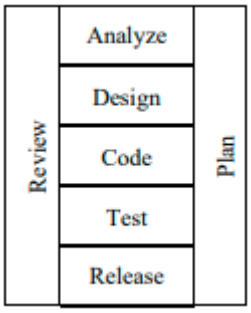
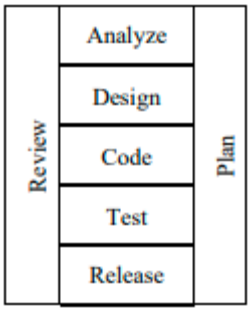
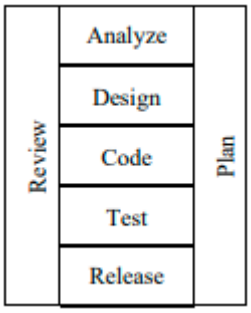
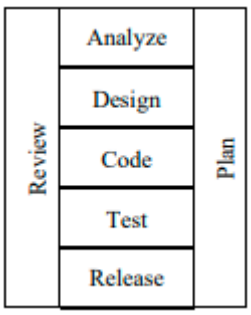
In conclusion, software development companies can rely on the ranking of risk management practices presented by this study. They show the lack of risk management in Scrum and bring opportunities to integrate traditional risk management with this framework. For the success of further research, it is suggested to carry out a grouping of the risk management practices according to artifacts, ceremonies, and roles on Scrum. The identification of the Scrum items that most contribute to risk management can also be carried out, perform the mapping, on dealing with project risks.

3.4 Methodology process

Scrum is an agile framework for developing, delivering, and sustaining complex products with an initial emphasis on software development It is designed for teams of ten or fewer members who break their work into goals that can be completed within time-boxed iterations, called sprints, no longer than one month and most commonly two weeks [4]. The Scrum Team tracks progress in 15-minute time-boxed daily meetings called daily scrums. At the end of the sprint, the team holds a sprint review to demonstrate the work done and a sprint retrospective to improve continuously. Agile product development is a sprint is a period during which specific work has to be completed and made ready for review. Each sprint begins with a planning meeting. During the meeting, the product owner and the development team agree upon exactly what work will be accomplished during the sprint. The duration of a sprint is determined by the scrum master. Below we discuss the sprint that we used in HRMS Project as shown in Table 3.5: Scrum Methodology Sprints.

Table 3.5: Scrum Methodology Sprints

# OF Sprints		Date	Duration of Sprint	Requirement item
Sprint 1		16-Oct-2022 To 29-Oct-2022	2Weeks	1-Gathering the requirements of the main functions of the system. 2-Data collections. 3-Identifing the scope of a project 4-project start date 5- scope boundaries of the work 6-constraints in resources people 7-project environment deliverables 8-budge

Sprint 2		29-Oct-2022 To 11-Nov-2022	2Weeks	1-Domain Name 2-Web Hoisting 3-Bussinss email address 4-Logo Design 5- Icons & Images 6- Text Content 7-Choosing the tools 8-Designing Models
Sprint 3		11-Nov-2022 To 24-Nov-2022	2Weeks	1-For Each HR Front-End Code
Sprint 4		24-Nov-2022 To 24-Dec-2022	4Weeks	1-For Each HR Back-End Code Designing the request page and connecting it with the Dashboard and Database
Sprint 5		24-Dec-2022 To 7-Jan-2023	2Weeks	1- Connecting the Front & Back End Code
Sprint 6		7-Jan-2023 To 19-Jan-2023	2Weeks	1-Testing Methodology's Results 2- The End Result Delivery

3.5 Requirements

Listed below are the functional requirements and non-functional requirements of the system. In the priority column, the following shorthand's are used:

M – mandatory requirements (something the system must do).

D – desirable requirements (something the system preferably should do).

O – optional requirements (something the system may do).

3.5.1 Functional Requirements

A Functional Requirement (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform as shown in Table 3.6: Functional Requirement

Table 3.6: Functional Requirement

No.	Functional	Description	Priority	Iteration
1	Hiring	The hiring process is the process of recruiting, interviewing, selecting, and onboarding new employees. It is a critical part of any organization success and in a key factor in employee retention and job satisfaction. Hiring process typically begins with a job posting, followed by the review of resumes and applications, and then selection of candidate's interview. After interviews the hiring manager will make a discission and extend an offer of employment. The Onboarding process then begins.	M	3,4,5
2	Onboarding	The Onboarding is the process of integrating new employees into an organization. It typically involves introducing new hires to the company, providing them with the necessary resources to do their job, and helping them become productive members of the team. The goal of onboarding is to ensure that new employees have the tools and knowledge they need to be successful in their new positions.		
3	Customizing Application	Customizing applicants according to skills can be done by using a variety of methods. One way is to use a skills-based assessment tool to evaluate applicants and match them to job requirements. This can be done by creating a list of skills required for the job and then assessing each applicant's skills against the list. This can be done through a variety of methods such as		

		interviews, tests, and surveys. Additionally, employers can use online job boards and social media to search for applicants with specific skills. This can help employers quickly identify and contact applicants who have the skills they need.		
4	Authentication	Authentication is the process of verifying the identity of a user or device. It is typically done by comparing a user-provided identifier (such as a username or email address) and a user-provided password or other credentials (such as a fingerprint or security token) against a database of authorized users. If the credentials match, the user is authenticated and allowed access to the system.		
5	Managing	Managing is the process of leading and directing people and resources to achieve organizational goals. It involves setting objectives, developing strategies, and implementing plans to ensure that the organization is successful. It also involves monitoring progress, making adjustments, and providing feedback to ensure that the organization is on track. Managing requires strong leadership skills, and communication skills.	M	
6	Create Application	Creating an applicant is the first step in the hiring process. It involves gathering information about the applicant, such as their contact information, work experience, education, and skills. This information is then used to determine if the applicant is a good fit for the position. The applicant should also be asked to provide references and any other relevant information that may be needed. Once the applicant has been created, they can be contacted for an interview or further evaluation.		3,4,5
7	Attending	Attendance process is the process of tracking employee attendance in the workplace. It involves recording the arrival and departure times of employees, as well as any absences or tardiness. The attendance process is important for businesses to ensure that employees are present and productive during their scheduled shifts. It also helps employers to monitor employee performance and ensure compliance with labor laws. The attendance process can be managed manually or through automated systems such as time clocks, biometric scanners, and software programs	D	
8	CRUD Tasks	"CRUD" stands for Create, Read, Update, and Delete, it's a process used to manage data. Tasks is a list of tasks that need to be completed. It is a great way to keep track of what needs to be done and prioritize tasks. To-do lists can be used for both personal and professional tasks, and can help you stay organized.		

3.5.2 Non-Functional Requirements

Table 3.7: Non-Functional Requirement

No.	Functional	Description	Priority	Iteration
1	Usability	Easy To Use – Users will be able to successfully complete a benefits calculation without needing any external instruction or help screens	M	3
2	Security	Developing safer systems in this system we have always used, login information with username, and password for each user		4
3	Availability	The user can use the application any time		4
4	Confidentiality	Savings the system saves Employees time, effort in Registering.		5
5	Reliability	the degree of consistency of a measure.		5
6	Correctness	The System do what is required and do it successfully	D	5
7	Maintainability	Good Code Quality.		5

3.5.3 Tools

Table 3.8: Tools

No.	Tools Utilized
1	Visual Studio
2	Excel for performing Gantt Chart
3	Word for Writing Documentation
4	GitHub
5	Discord & Microsoft Teams for Meetings
6	Draw.IO
7	MongoDB
8	Node mail Server
9	Google APIs Consol
10	OrgCharts Library
11	Cloudinary Storage
12	Passport encryption for passwords
13	Chart.js

3.5.4 Programming Language

Table 3.9: Programming Languages

No.	Programming Languages
1	HTML
2	CSS
3	JS
4	Bootstrap
5	Mongoose- No SQL
6	Node JS
7	Express Library Routes
8	jQuery

3.5.5. Data Collection:

Collecting data is an important part of any research project. It is the process of gathering information from a variety of sources in order to answer a research question or to test a hypothesis. Data collection can be done in a variety of ways, including surveys, interviews, observations, and experiments. It is important to choose the right method for the research project in order to ensure accurate and reliable results. However, if your research question dictates that interviews are the best method of data gathering, Interviews can be conducted in person, over the telephone, or electronically using a program such as Skype. The advantage to being face to face is the ability to see facial expressions and body language. There are a few types of interviewing styles to choose from: structured interviews, semi-structured interviews, and unstructured interviews. In Collecting our data, we will use the Semi-structured interviews are one of the most useful data collection methods for studying a wide range of information behaviors, allow for more flexibility. They involve having a set of guiding questions that will keep the interview on track.

The Following Questions was asked during an interview with an HR employee:

1- What features of the HRMS system do you find most useful?
2- How has the HRMS system improved the efficiency of your HR department?
3- How have you used the HRMS system to streamline the recruitment process?

All The following Questions have been provided on our website and they are our main Functional Requirements in the system:

Q1)

What features of the HRMS system do you find most useful?
1.Performance Management: Performance **management** <Tag 5> systems allow HR departments to track employee performance and provide feedback. ***2***.Recruiting and Onboarding: HRMS systems can streamline the **recruiting** <Tag 1> and **onboarding** <Tag 2> process by automating job postings, applicant tracking, and onboarding paperwork.***3.*** Compliance Tracking: HRMS systems can help HR departments stay compliant with labor laws and regulations by tracking employee data and providing alerts when changes need to be made***.4.*** Reporting and Analytics: HRMS systems provide reporting and analytics capabilities that allow HR departments to track employee data and make informed decisions.

Q2)

How has the HRMS system improved the efficiency of your HR department?
The HRMS system has improved the efficiency of our HR department in a number of ways. It has streamlined the **recruitment process** <Tag1>, allowing us to quickly and easily post job openings, review resumes, and **track applicants**. It has also enabled us to automate many of our HR processes, such as **onboarding**<Tag2>, payroll, and benefits administration. This has allowed us to reduce manual data entry and paperwork, freeing up our HR staff to focus on more strategic **tasks <**Tag4 **>**. Additionally, the system has enabled us to better track employee performance and **attendance <**Tag 6**>,** allowing us to identify areas of improvement and reward employees for their hard work. Finally, the system has enabled us to better manage our employee data, allowing us to quickly and easily access information when needed.

Q3)

How have you used the HRMS system to streamline the recruitment process?
I have used the HRMS system to streamline the **recruitment process** <Tag 1> by automating the process of posting job openings, tracking applications, and scheduling interviews. This has allowed us to quickly identify and contact qualified candidates, as well as to easily track the progress of each applicant throughout the recruitment process. Additionally, the HRMS system has enabled us to easily store and access candidate information, such as resumes, contact information, and interview notes, which has allowed us to make more informed hiring decisions.

CHAPTER 4

DESIGN MODELS

4.1 Overview:

In this chapter, we will illustrate the implementation of our source code in a diagrammatic design. Models are abstractions used to represent and communicate what is important, devoid of unnecessary details, and helps developers deal with the complexity of the problem being investigated or the solution being developed design and build relationships between system components by using some diagrams such as the context diagram, the use case diagram, data flow diagram, and the ER diagram. This collection of diagrams is "a symbolic representation of information" which shows the interactions, processes, and structure.

4.2 Context diagram:

The context diagram is used to establish the context and boundaries of the system to be modeled: which things are inside and outside of the system being modeled, and what is the relationship of the system with these external entities [6]. The entire software system is shown as a single high-level process, with its relationship to external entities as shown in Figure 4.1: Context Diagram

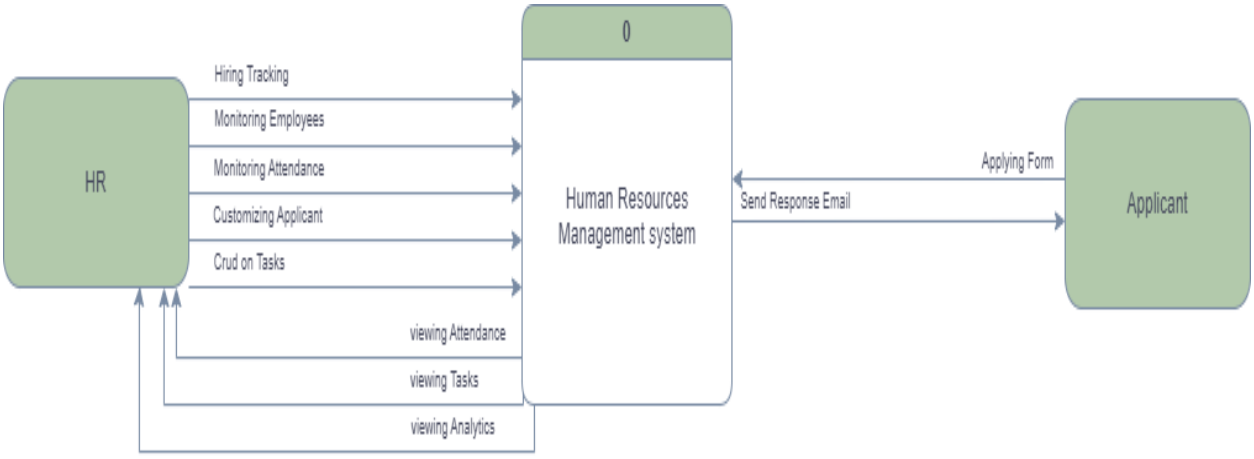


Figure 4.1: Context Diagram

4.3 Data flow Diagram:

Here we went into more detail than a context diagram. In a level 1 data flow diagram, the single process node from the context diagram is broken down into subprocesses. We can think of a level 1 DFD as an “exploded view” of the context diagram as shown in Figure 4.2: Data Flow Diagram.

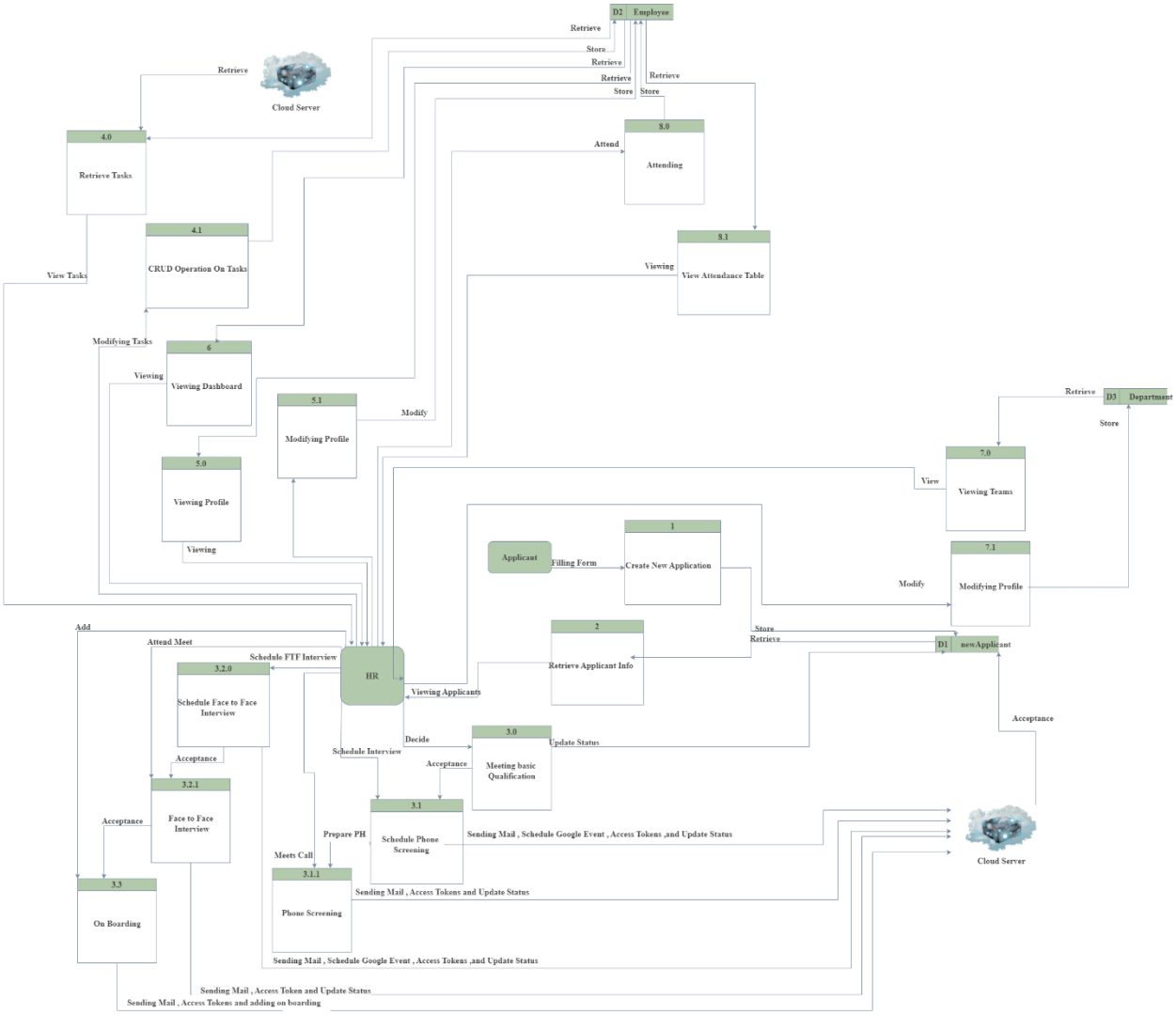


Figure 4.2: Dataflow Diagram

4.4 Use Case Diagram:

Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities. They also help identify any internal or external factors that may influence the system and should be taken into consideration [5]. They provide a good high-level analysis from outside the system. Use case diagrams specify how the system interacts with actors without worrying about the details of how that functionality is implemented as shown in Figure 4.3: Use Case Diagram

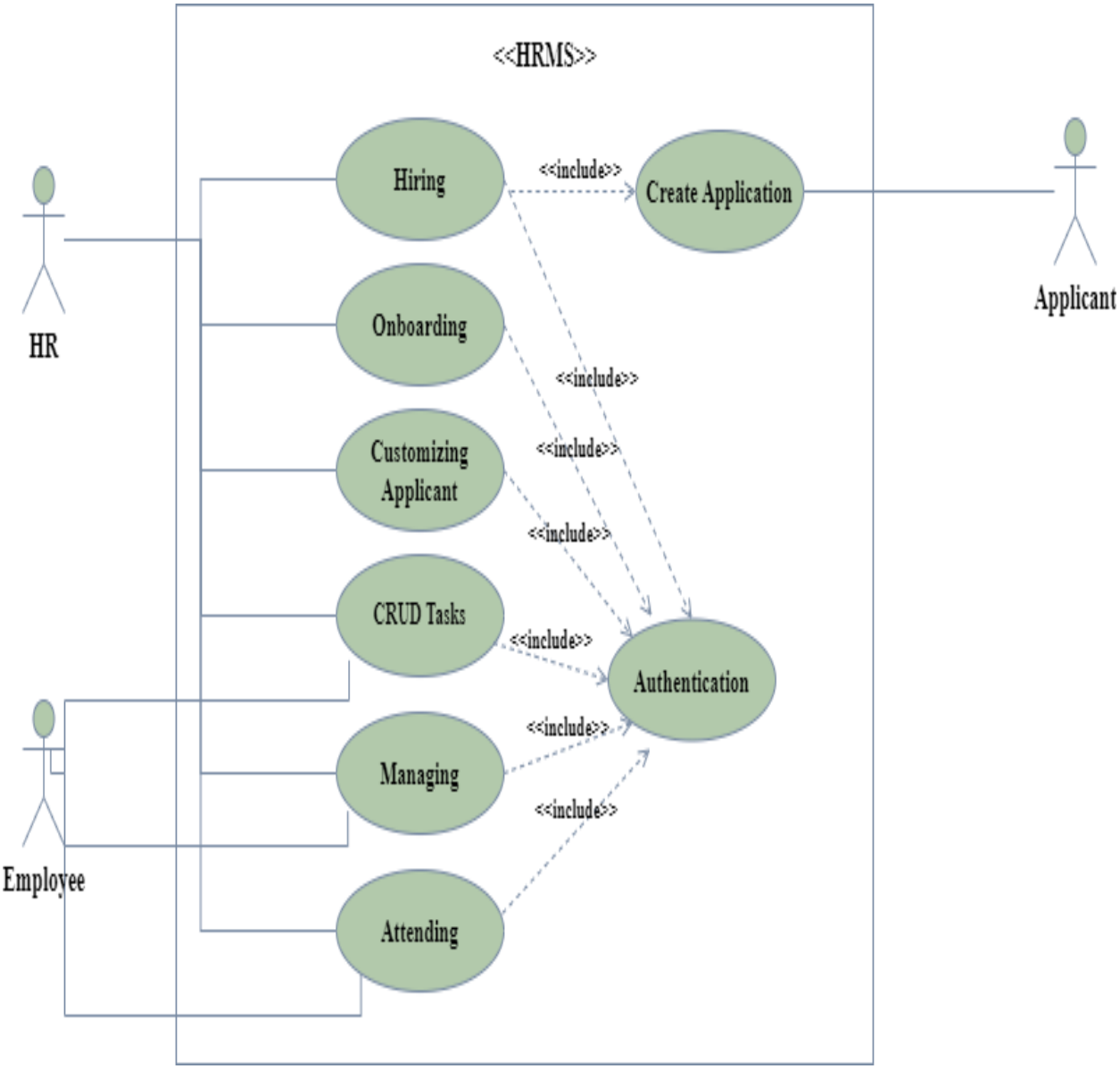
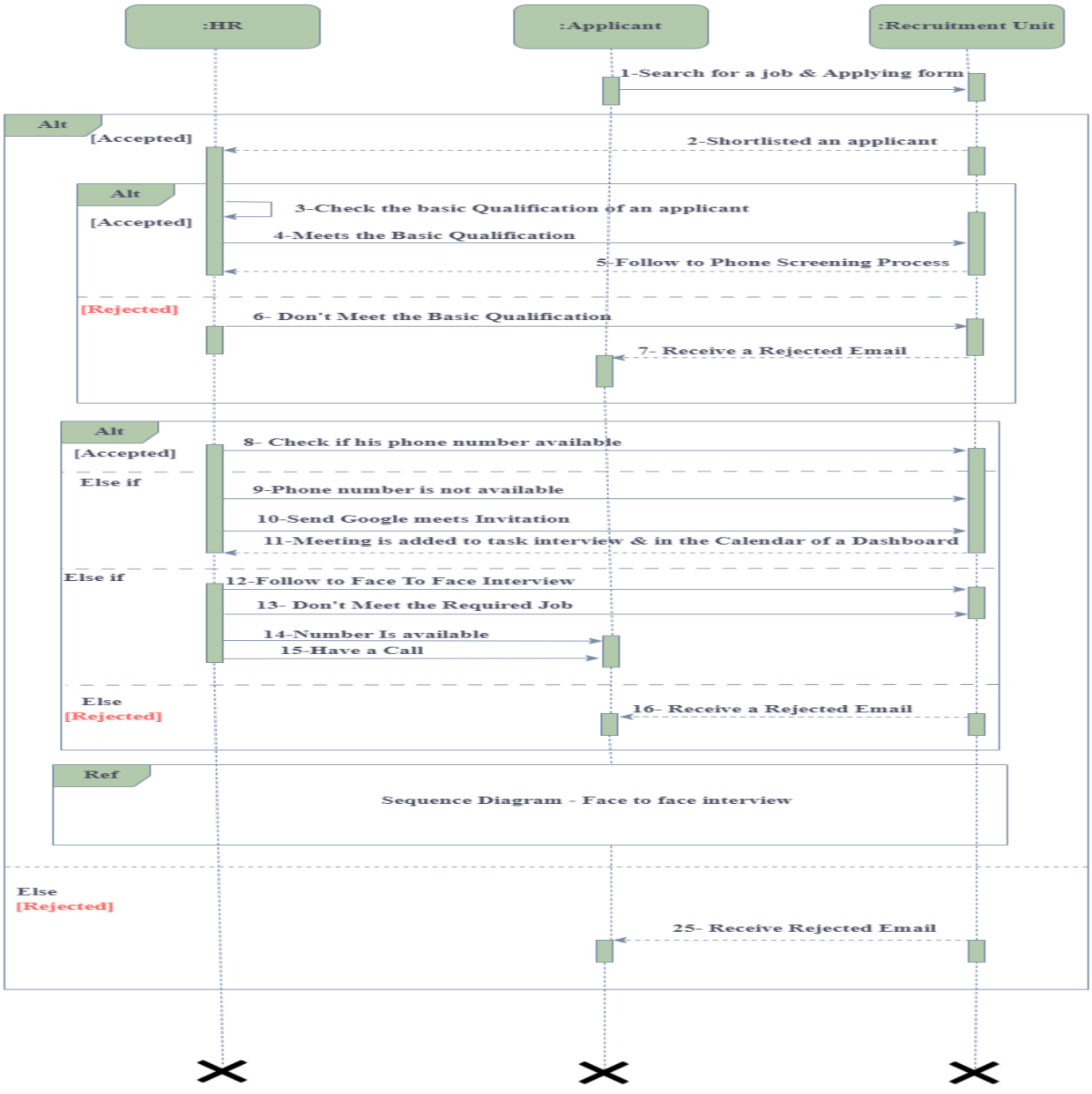


Figure 4.3: Use Case Diagram

4.5 Sequence diagram:

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction as shown in Figure 4.4: Sequence Diagram this Diagram Represent the Recruitment Process.



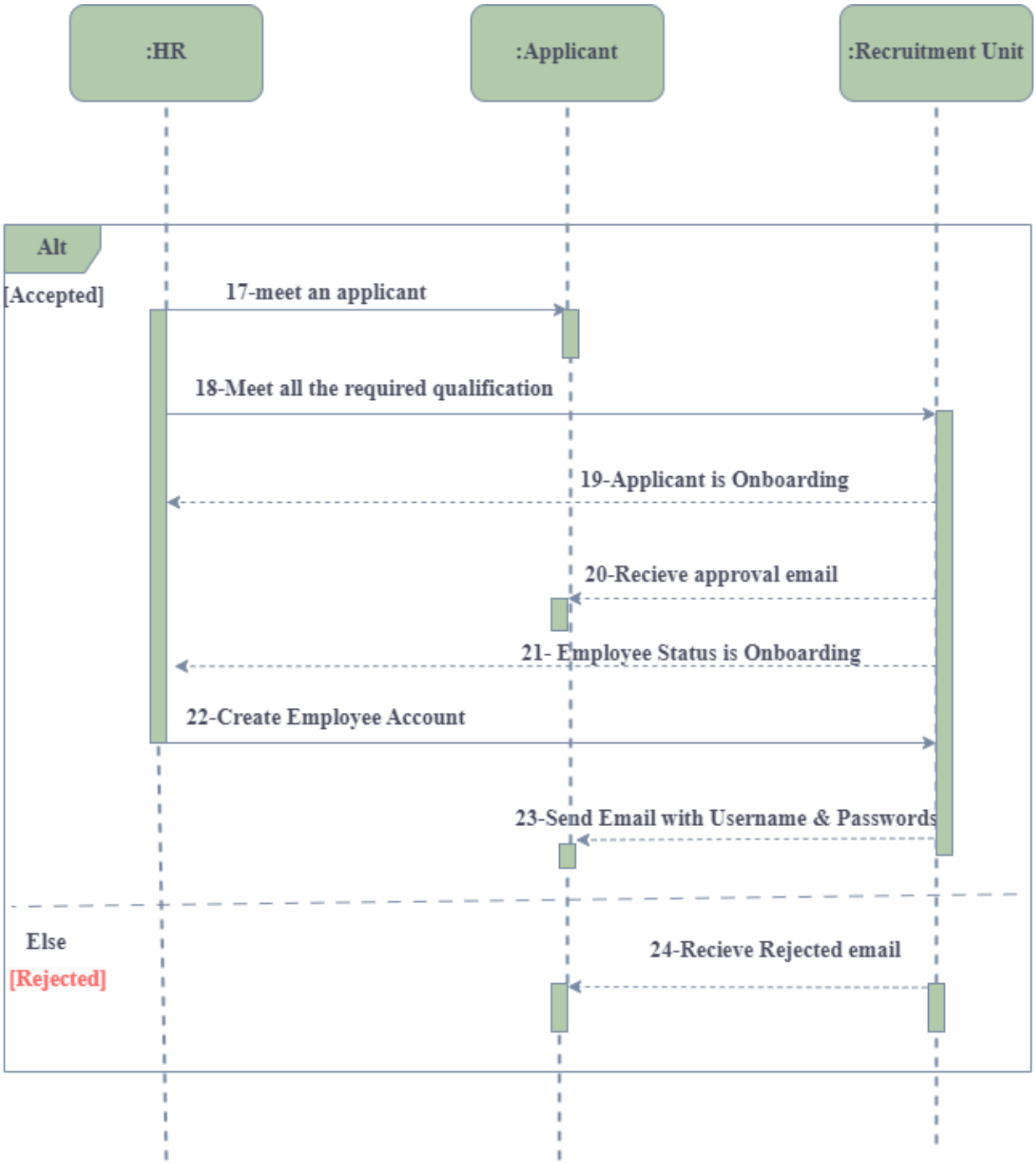


Figure 4.4: Sequence Diagram

4.6 Use case specification:

Figure 4.5: Use Case specification "Hiring"

Table 4.1: Use Case specification "Hiring"

*ID:*1	Hiring
***Description**:* Hiring process refers to the process of finding, selecting and hiring new employees to a company. This process has three key segments: planning, recruitment, and employee selection, the five distinct phases during the hiring process that recruiters can assist hiring managers with: opening the requisition, screening the applicants, interviewing the candidates, selecting the best, and making the offer.	
***Actors**:* HR	
Pre-conditions: 1- Already registered on HR account 2- All applicants appear in shortlisted table of all applications.	
Post-conditions: 1- The successful candidate has been offered and accepted the job. 2- The new hire has been provided with an orientation to the company and job role.	
Basic Flow: 1. Job Posting: The employer posts the job opening on job boards, websites, and other outlets. 2. Resume Screening: The employer reviews resume and selects candidates for further consideration. 3. Interviews: The employer conducts interviews with the selected candidates to assess their qualifications and fit for the job. 4. Background Checks: The employer conducts background checks on the selected candidates to verify their credentials and work history. 5. Reference Checks: The employer contacts the references provided by the candidates to get additional information about their qualifications and work history. 6. Job Offer: The employer makes a job offer to the selected candidate. 7. Acceptance: The candidate accepts the job offer and signs the employment contract. 8. Onboarding: The employer provides the new hire with orientation and training to help them become acclimated to the job and the company.	
Alternative Flow: [A-1] verify that user couldn't use google console api's with an invalid google app credentials. [A-2] Meeting scheduling client side validation to be throw works hours.	

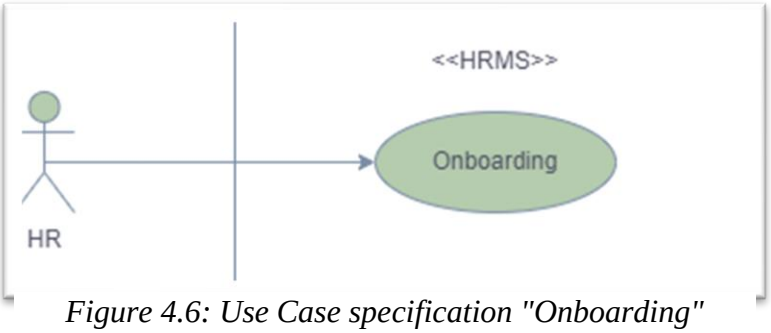


Figure 4.6: Use Case specification "Onboarding"

Table 4.2: Use Case specification "Onboarding"

ID:2	Onboarding
***Description*:** Onboarding is the process of introducing and orienting new employees to an organization, helping them become productive and successful members of the team	
***Actors**:* HR	
Pre-conditions: 1- An applicant must finish the Hiring Process	
Post-conditions: 1- Applicant now is on the onboarding status to become an employee.	
Basic Flow: 1- Send out an email to the new hire with a link to a pre-onboarding survey. 2- Introduce the new hire to the team and provide an overview of the onboarding process. 3- Provide an orientation to the company, its culture, and its policies 4- Follow up with the new hire to ensure they are settling in and have all the resources they need. 5- Send out a welcome package to the new hire with all the necessary information and materials they need to get started.	
Alternative Flow: NONE	

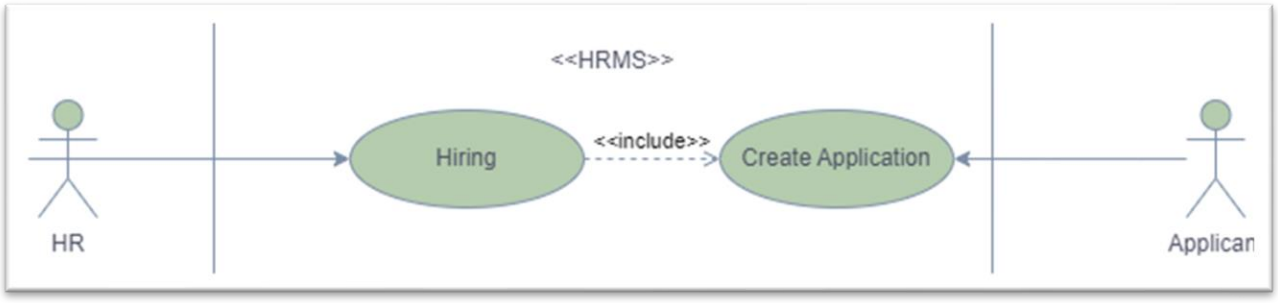


Figure 4.7: Use Case Specification " Create Applicant"

Table 4.3: Use case Specification "Create Applicant"

ID:3	Create Applicant
***Description*:** allows users to quickly and easily create job applications. It provides a streamlined process for creating and submitting applications, allowing users to quickly and easily create professional-looking applications that stand out from the competition	
***Actors*:** HR, Applicants	
Pre-conditions: 1- An Applicant filled the form	
Post-conditions: 1- Applicant is now in the shortlisted applications 2- On the Hiring Process	
Basic Flow: 1- Applicant fill all the required fields. 2- Applicant submit the form. 3- Receive email of submission.	
Alternative Flow: [A-1]: Verify that the Registration form contains Username, First Name, Last Name, Password, Confirm Password, Email Id, Phone number, Date of birth, Gender, Address, and a Pdf CV.	

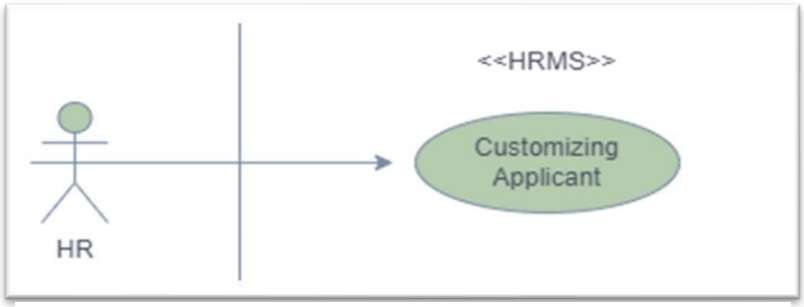


Figure 4.8: Use Case Specification "Customize Applicant

Table 4.4: Use case Specification "Customize Applicant"

ID:4	Customizing Applicant
***Description*:** used to manage job applications, skills, and other recruitment-related data. It provides tools to streamline the recruitment process, such as automated job postings, applicant tracking, and candidate screening.	
***Actors*:** HR	
Pre-conditions: 1- An Applicant filled the form 2- Applicant is shortlisted 3- HR can choose according to the skills	
Post-conditions: 1- HR can start the hiring within the required skills	
Basic Flow: 1- An applicant fills all the required fields 2- Applicant is shortlisted in the hiring table 3- HR can choose applicant according to the required skills 4- Applicant is now on the hiring process	
Alternative Flow: NONE	

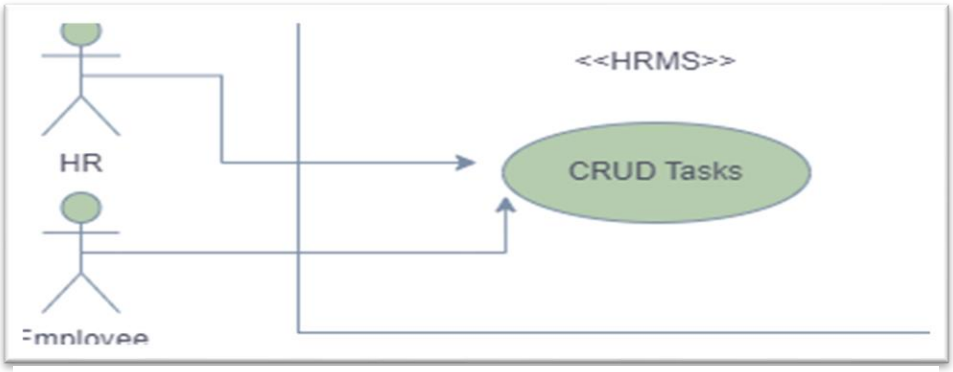


Figure 4.9: Use Case Specification " CRUD Task"

Table 4.5: Use Case Specification " CRUD Task"

ID:5	CRUD Task
***Description*:** A task management tool that helps you keep track of tasks that need to be completed, allowing you to prioritize and plan your day.	
***Actors*:** HR, Employee	
Pre-conditions: 1- Register as an Employee and HR	
Post-conditions: 1- Create a to-do list 2- Prioritize tasks 3- Break down tasks into smaller steps 4- Set deadlines 5- Track progress 6- Celebrate accomplishment	
Basic Flow: 1- Click on the to-do 2- An HR account can see all the interviews 3- HR and Employee can add, update, and delete task.	
Alternative Flow: NONE	

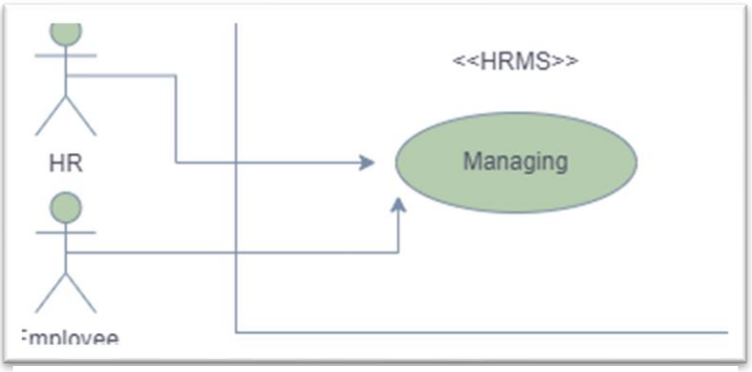


Figure 4.10: Use Case Specification "Managing"

Table 4.6: Use Case Specification "Managing"

ID:6	Managing
***Description*:** Managing involves planning and organizing resources, leading and motivating teams, and monitoring and evaluating performance to achieve desired outcomes.	
***Actors*:** HR, Employee	
Pre-conditions: 1- HR & Employee should be logged-in	
Post-conditions: 1- Ensuring that all employees are aware of their job responsibilities and expectations. 2- Ensuring that all employees are provided with the necessary resources and support to perform their job duties.	
Basic Flow: 1- After a user logged-in successfully. 2- User can see a hierarchy tree of all employees within one organization.	
Alternative Flow: [A-1]: Redirect to set team leader page when the team leader is not exist.	

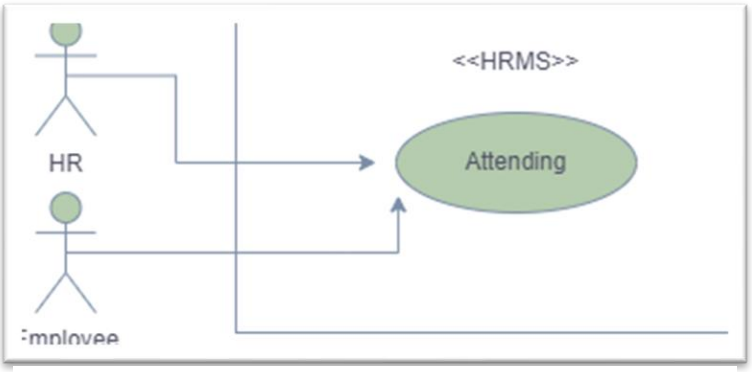


Figure 4.11: Use Case Specification "Attending"

Table 4.7: Use Case Specification "Attending"

ID:7	Attending
Description: involves arriving at the designated location on time and participating in the event or activity as required.	
Actors: HR, Employee	
Pre-conditions: 1- HR & Employee should be logged-in	
Post-conditions: 1- After HR & Employee are logged-in 2- HR & Employee can do the attending	
Basic Flow: 1. Arrive at the Work Hours: Make sure to arrive at the specific on time. 2. Check-in: Check in with the work hours. 3.Check-out: Check out after 8 hours of work 4.Check Attending Table	
Alternative Flow: [A-1]: verify that the Location information is available otherwise Location information is unavailable. [A-2]: Verify that the request to get user location timed out.	

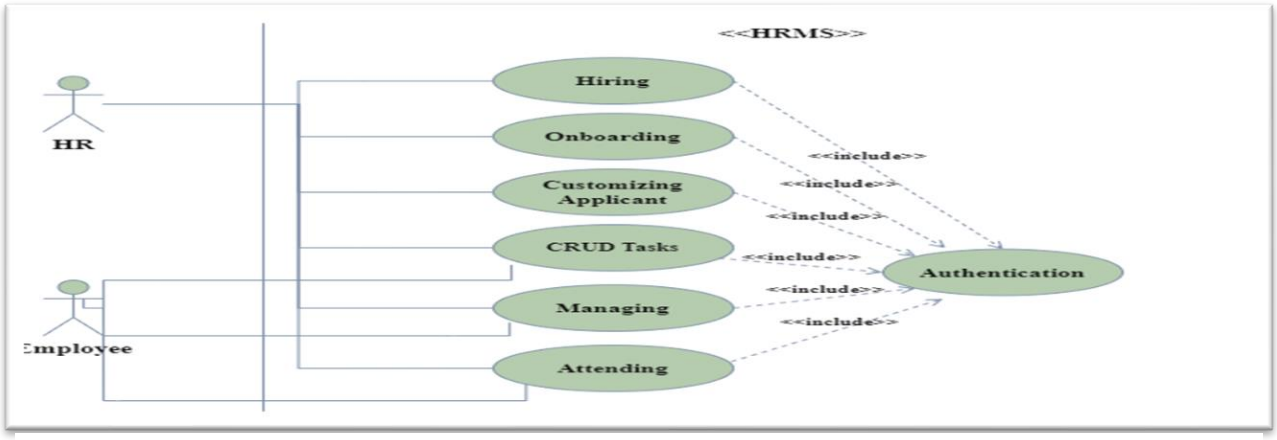


Figure 4.12: Use Case Specification "Authentication"

Table 4.8: Use Case Specification "Authentication"

ID:8	Authentication
***Description*:** Authentication is the process of verifying the identity of a user or system, typically by requiring a username and password. It is used to ensure that only authorized users can access a system or service.	
***Actors*:** HR, Employee	
Pre-conditions: 1- The user must have a valid username and password. 2- The user must have access to the authentication system. 3- The authentication system must be secure and reliable. 4- The user must provide valid credentials to the authentication system. 5- The authentication system must be able to verify the user's identity.	
Post-conditions: 1- User has been successfully authenticated and is now able to access the system. 2- The user's identity has been verified and the user is now able to access the resources they are authorized to access. 3- The authentication process has been completed and the user is now able to use the system	
Basic Flow: 1- User enters their credentials (username and password) into the login form. 2- The credentials are sent to the authentication server. 3- The authentication server verifies the credentials against the user database. 4- If the credentials are valid, the authentication server sends an authentication token to the user. 5- The user sends the authentication token to the application server. 6- The application server verifies the authentication token with the authentication server. 7- If the authentication token is valid, the application server grants access to the user	

Alternative Flow:
[A-1]: verify that user couldn't user google console api's with an invalid credentials

4.7 Activity Diagram:

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram as shown Figure 4.13: Activity Diagram for Attendance Process.

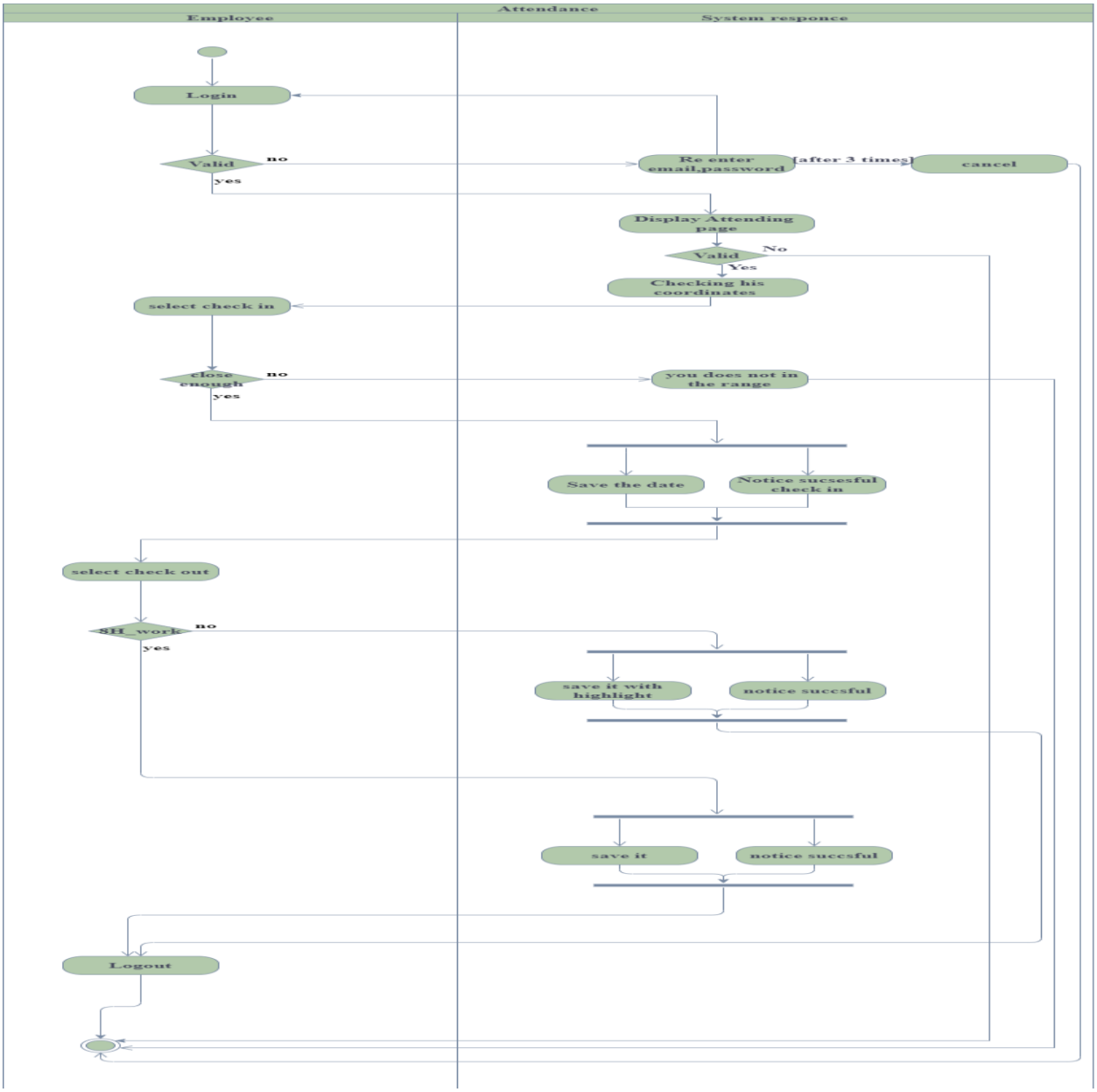


Figure 4.13: Activity Diagram

4.8 ER Diagram:

An Entity Relationship Diagram is a diagram that represents relationships among entities in a database, describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities as shown in Figure 4.14:ER Diagram.

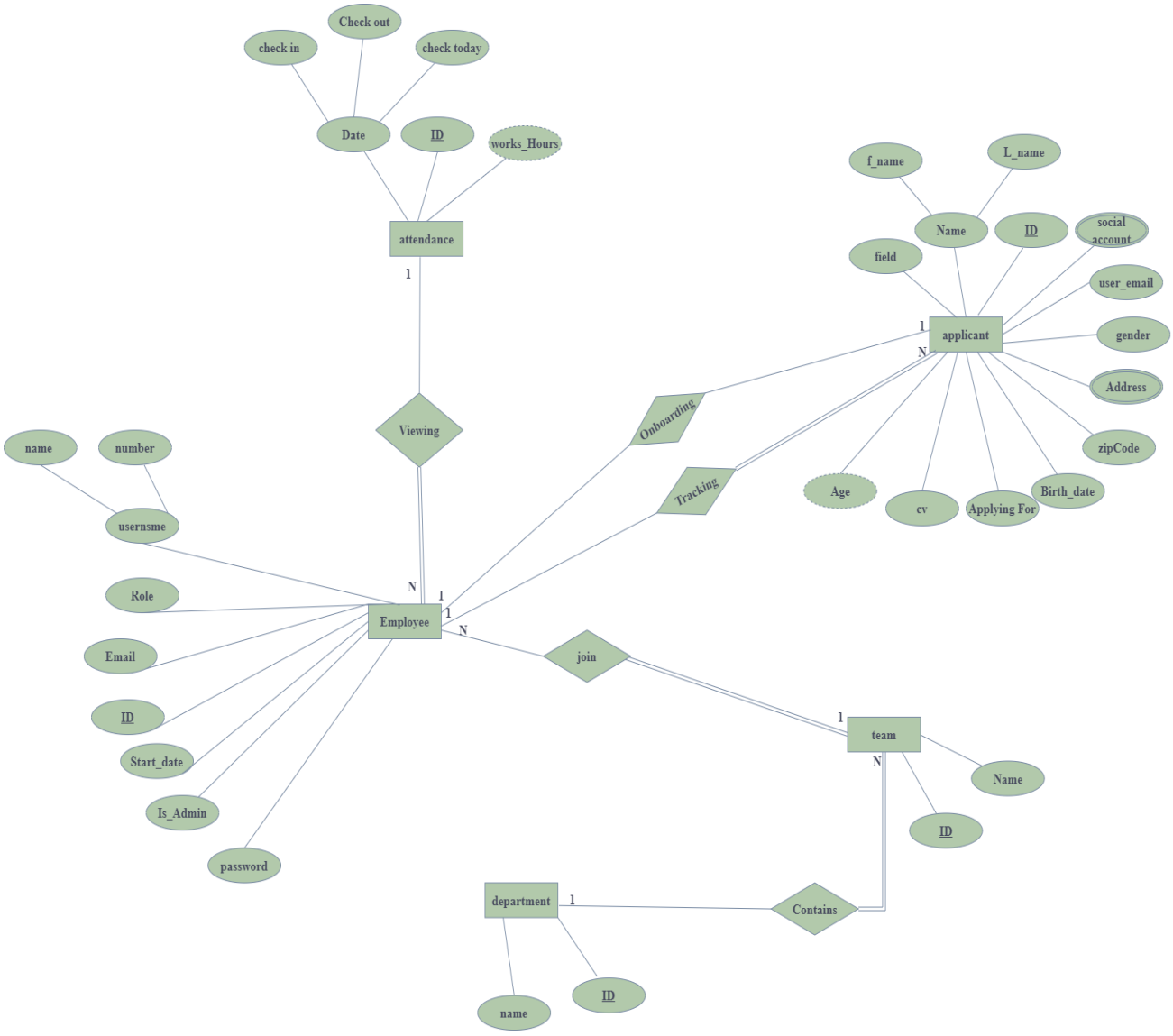


Figure 4.14:ER Diagram

4.9 Relational Model:

The relational model in DBMS is an abstract model used to organize and manage the data stored in a database. It stores data in two-dimensional inter-related tables, also known as relations in which each row represents an entity and each column represents the properties of the entity as Shown Figure 4.15: Relational Model.

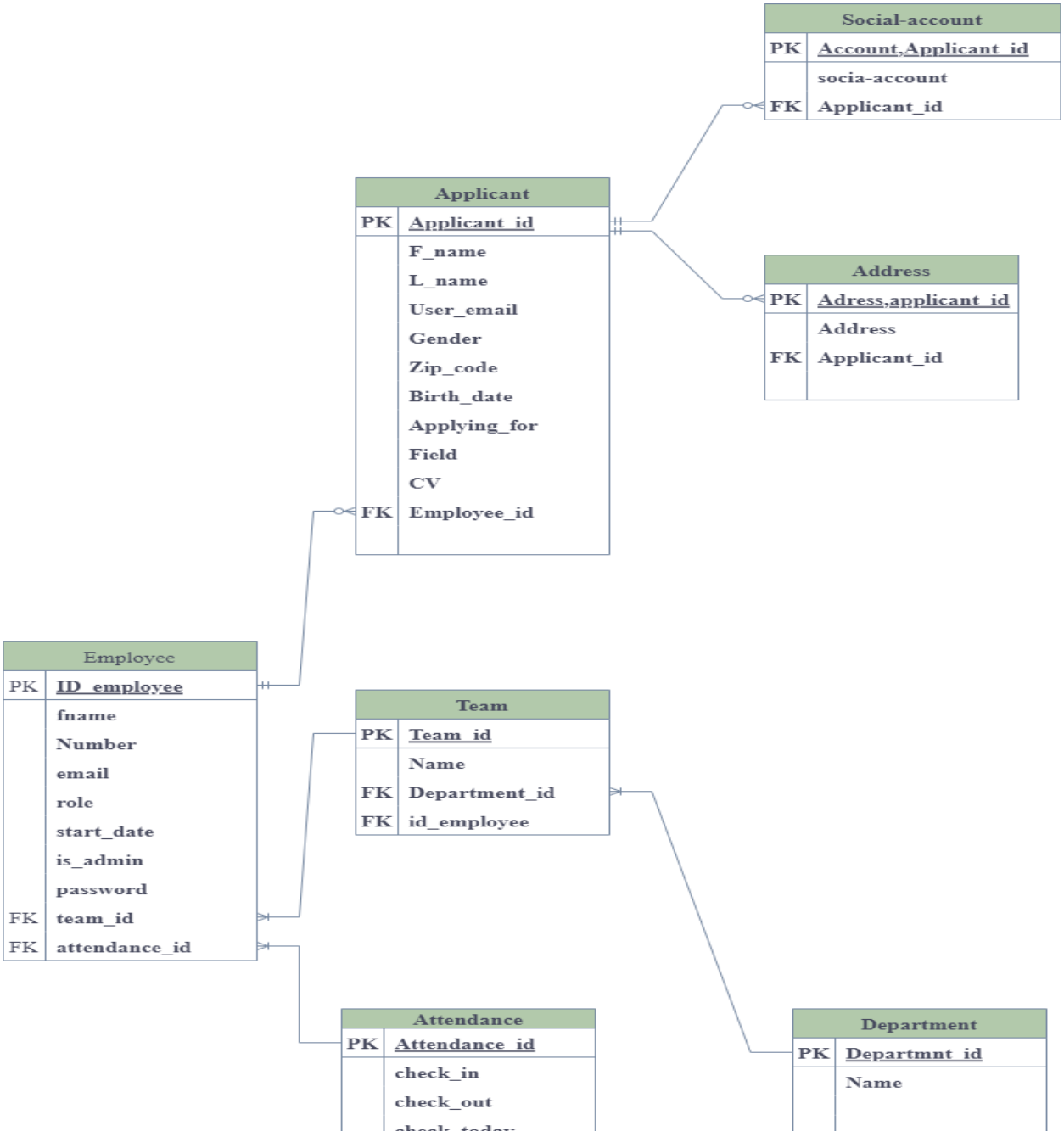


Figure 4.15: Relational Diagram

4.10 Summary:

A model summary is automatically created when running a regression modeling or a classification modeling. The model summary displays the name of the model, the model type, and the model formula. The purpose of this study was to examine the relationship between the impact of the work.

CHAPTER 5

EXPERIMENTS AND RESULTS

5.1 Overview

In this chapter we will talk about how we test the system with the intent to find whether it satisfies the specified requirement or not. The system has been tested and ensured that each unit performs the required task and is consistent with the rest of the units of the system by using software testing methods and using the system in an environment that matches the environment for which it was created. we made several tests to reach the highest accuracy, assurance, and safety to ensure that the system runs properly.

5.2 Testing methodologies

Software testing methodologies (STM) are the various strategies or approaches used to test an application to ensure it behaves and looks as expected and it is a process of running for the system to find errors [7]. These encompass everything from front to back-end testing (STM) is defined as strategies and testing types used to certify that the Application under Test meets client expectations. Test Methodologies include functional and non-functional testing as shown in Figure 5.1: Software Testing Methodology

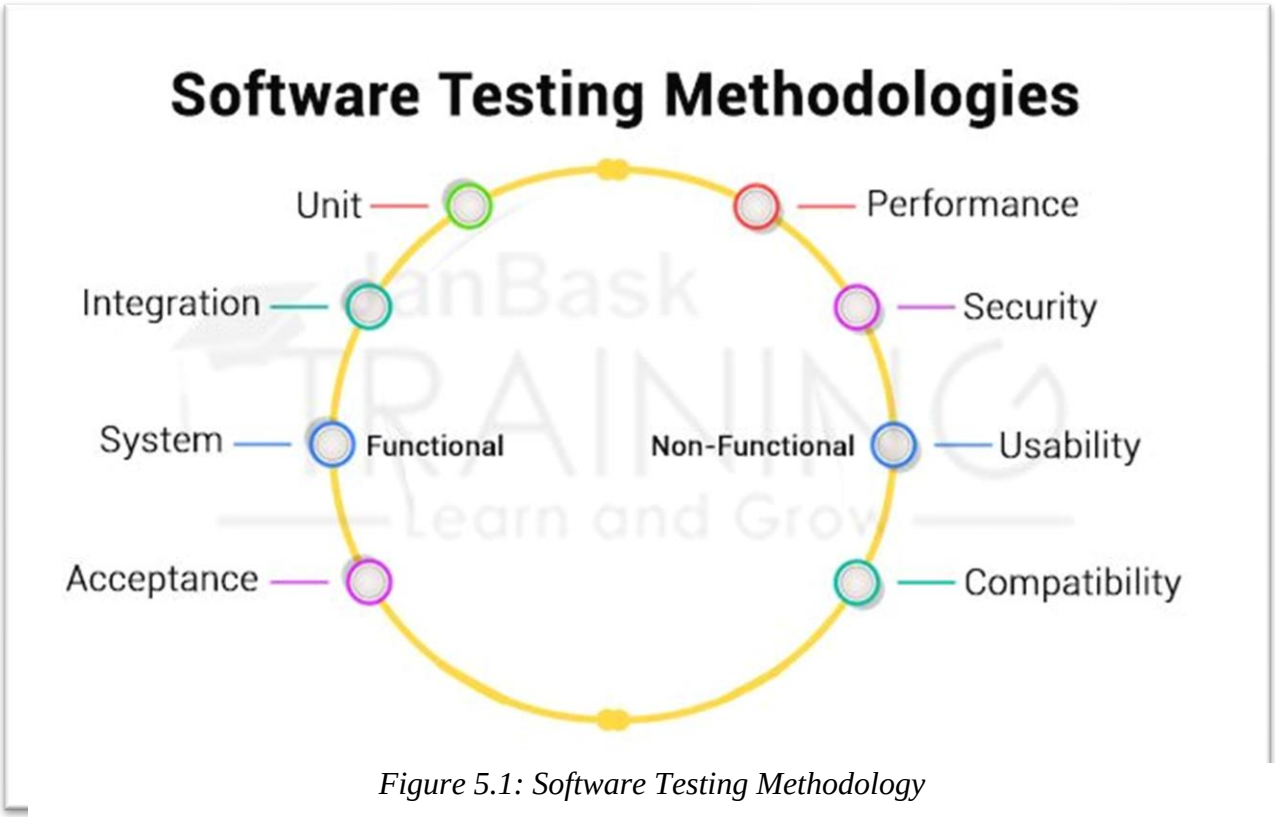


Figure 5.1: Software Testing Methodology

Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system, and also it adds value to the system by complying with the requirements used as Shown in Figure 5.2: Testing Methodology.

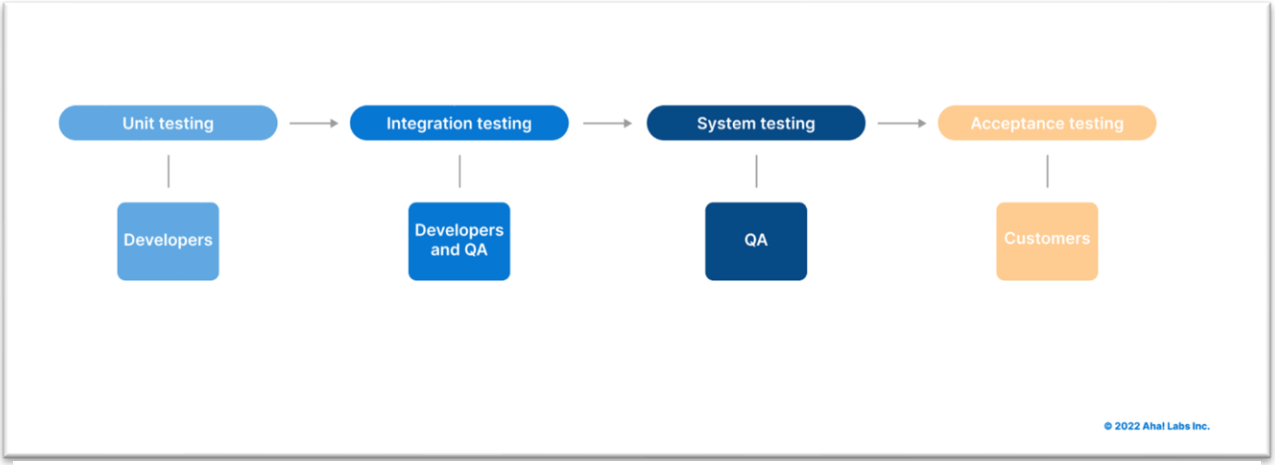


Figure 5.2: Testing Methodology

5.2.1 Unit Testing Results

A Unit Testing (UT) is type of software testing where individual units or components of a software are tested, the smallest testable piece of software that can be compiled, assembled, linked, and loaded. A unit is usually the work of one programmer and consists of several hundred or fewer lines of code. The purpose is to validate that each unit of the software code performs as expected as Shown in Figure 5.3: Unit Testing Life Cycle.

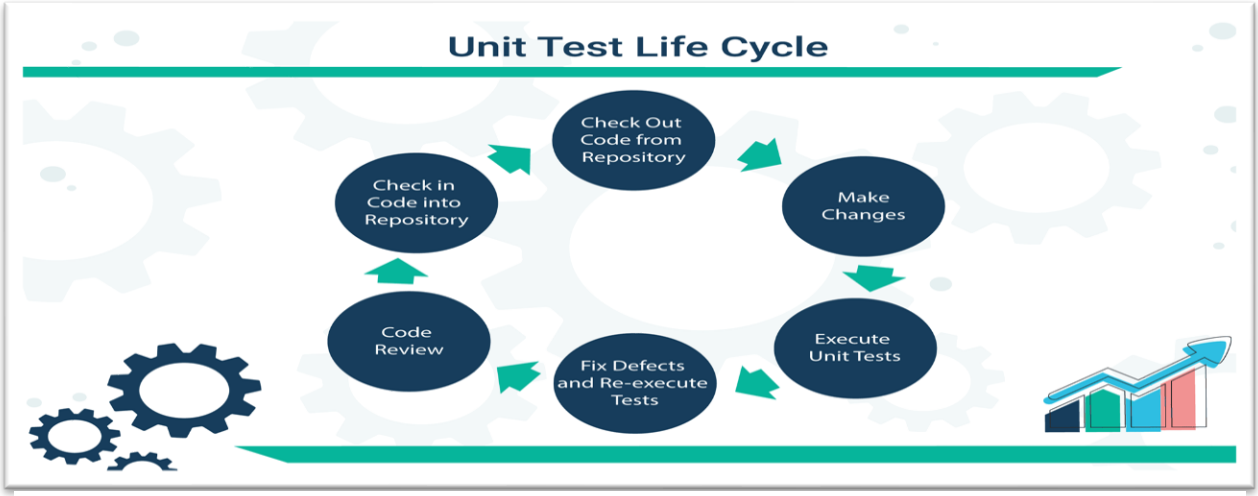


Figure 5.3: Unit Testing Life Cycle

Table 5.1: Testing"1"

Steps:

1.Verify if a user will not be able to login with an invalid username or invalid password.
2. Verify if a user cannot login with a valid username and an invalid password.
3. Verify if the data in password field is displayed as ****** sign.
4. Verify that the user couldn't login with the old password.
5. Verify that the time taken to log in with a valid username and password is a Realtime (Performance testing).
6. Verify that there is a popup messages for each mandatory field.

Pic:

Figure 5.4: Test"1"

Code:

```
<form action="/login" method="post">
  <img src="/stylesheet/id-card(1).png" />
  <h2 class="title">Login</h2>
  <% if(error.length>0){ %>
    <div class="alert alert-danger" role="alert" style="max-width: 400px;">
      <%= error[0] %>
      <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close" style="font-size: 12px;" ></button>
    </div>
  <% } %>
  <div class="input-div one">
    <div class="i">
      <i class="fas fa-user"></i>
    </div>
    <div class="div">
      <h5>Username</h5>
      <input type="text" name="username" class="input" required />
    </div>
  </div>
  <div class="input-div pass">
    <div class="i">
      <i class="fas fa-lock"></i>
    </div>
    <div class="div">
      <h5>Password</h5>
      <input type="password" name="password" class="input" required minlength="8" />
    </div>
  </div>
  <a href="/forgot">Forgot Password?</a>
  <button type="submit" class="btn">SUBMIT</button>
</form>
</div>
</div>
```

Figure 5.5: Front-End "1"

```
router.get('/login', (req, res) => {

    res.render('users/login')
})

router.post('/login', passport.authenticate(['local', 'emp-local'], { failureFlash: true,
    failureRedirect: '/login', failureMessage: 'Username or password is incorrect' }), AsyncHandler(async (req, res) => {

    const redirectUrl = req.session.returnTo || `/dashboard/${req.user._id}`
    res.redirect(redirectUrl)

}))
```

Figure 5.6: Back-End "1"

Table 5.2: Testing "2"

Steps:

1.Verify the Change password functionality by entering the valid old password and matched new - confirmed password.

2. Verify the change password functionality by leaving the old password field blank and entering valid new and confirm password.

3. Verify the password change functionality by entering the invalid old password and valid (matching) new and confirm password.

4. Verify the MAX and MIN Limit for Password.

5. Verify the canceling of current session after change the password.

Pic:

Figure 5.7: Test "2"

Code:

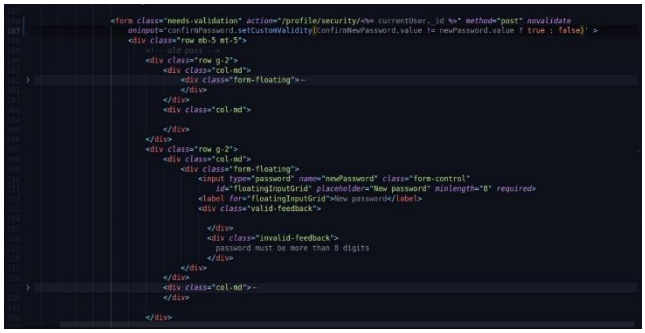


Figure 5.8: Front-End "2"

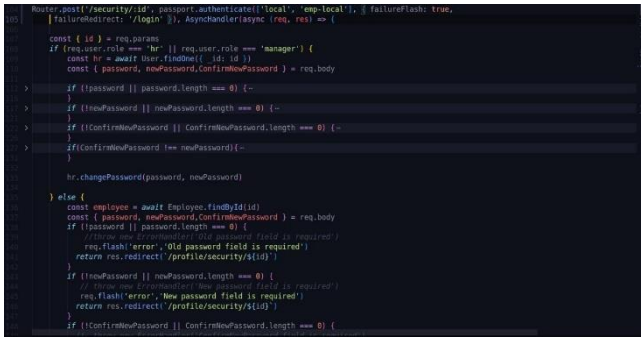


Figure 5.9: Back-End "2"

Table 5.3: Testing"3"

Steps:

1.verify that user location is in the located space out of the required space location can't check-in.

2.verify that a user can check-in within the work hours.

3.verify that a user can check-out.

4. verify that a user can view his attending table.

5.verify that a user can see his working hours, attending time, and leaving time.

6.verfiy that Browser denied the request for Geolocation.

7.verify that the Location information is available otherwise Location information is unavailable.

8.Verify that the request to get user location timed out.

Pic:

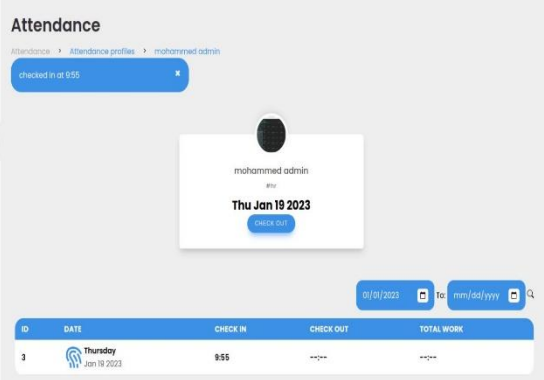


Figure 5.10: Test"3"

Code:

Figure 5.11: Front-End "3"

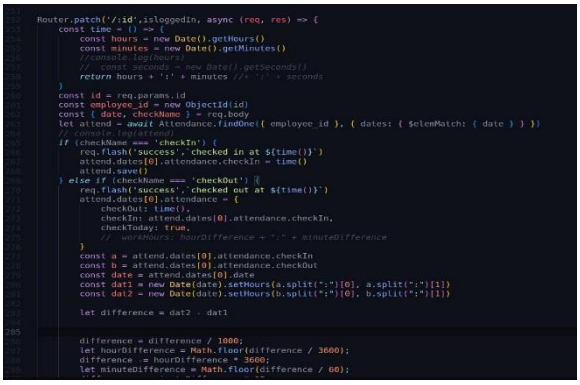


Figure 5.12: Back-End "3"

Table 5.4: Testing"4"

Steps:

1. Verify that the Registration form contains Username, First Name, Last Name, Password, Confirm Password, Email Id, Phone number, Date of birth, Gender, Address, and a Pdf CV.

2. Leave the email field blank.
3. Verify that clicking on submit button after entering all the mandatory fields, submits the data to the server & send a submitted email to the applier.

4. Verify that clicking on submit button after entering all the mandatory fields, submits the data to the server.

5. Verify that all the fields such as Username, First Name, Last Name, Password and other fields have a valid placeholder.

Pic:

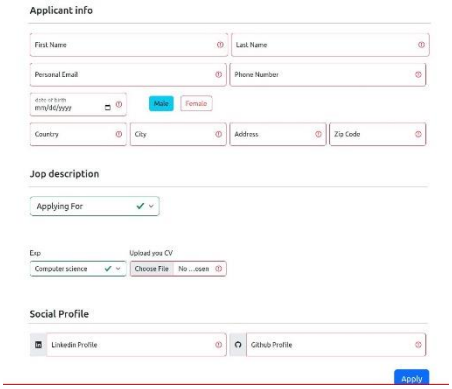


Figure 5.13: Test "4"

Code:

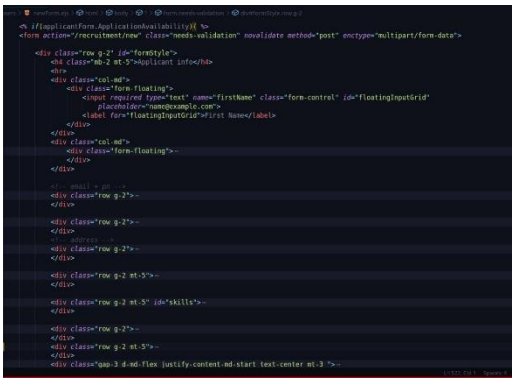
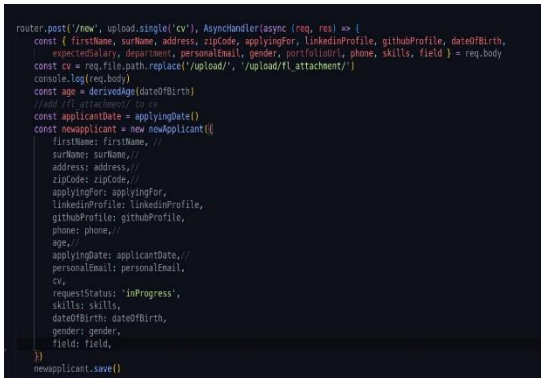


Figure 5.14: Front-End "4"



```
router.post('/new', upload.single('cv'), AsyncHandler(async (req, res) => {
    const { firstName, surName, address, zipCode, applyingFor, linkedinProfile, githubProfile, dateOfBirth,
    expectedSalary, department, personalEmail, gender, portfolioUrl, phone, skills, field } = req.body
    const cv = req.file.path.replace('/upload/', '/upload/fl_attachment/')
    console.log(req.body)
    const age = derivedAge(dateOfBirth)
    //add fl_attachment/ to cv
    const applicantDate = applyingDate()
    const newapplicant = new newApplicant({
        firstName: firstName, //
        surName: surName,//
        address: address,//
        zipCode: zipCode,//
        applyingFor: applyingFor,
        linkedinProfile: linkedinProfile,
        githubProfile: githubProfile,
        phone: phone,//
        age,//
        applyingDate: applicantDate,//
        personalEmail: personalEmail,
        cv,
        requestStatus: 'inProgress',
        skills: skills,
        dateOfBirth: dateOfBirth,
        gender: gender,
        field: field,
    })
    newapplicant.save()
```

Figure 5.15: Back-End "4"

Table 5.5: Testing"5"

Steps:

1.verify that user couldn’t use google api’s with an invalid refresh tokens.

2.verfiy that user couldn’t send request with timed out refresh tokens

3.verify that user couldn’t user google console api’s with an invalid google app credentials

Pic:

Figure 5.16: Test "5"

Code:

Figure 5.17: Front-End "5"

Figure 5.18: Back-End "5"

5.2.2 Integration Testing Results

After each unit is thoroughly tested it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated. These are then tested as a group through integration testing to ensure whole segments of an application behave as expected as Shown in Figure 5.9: Integration Testing.

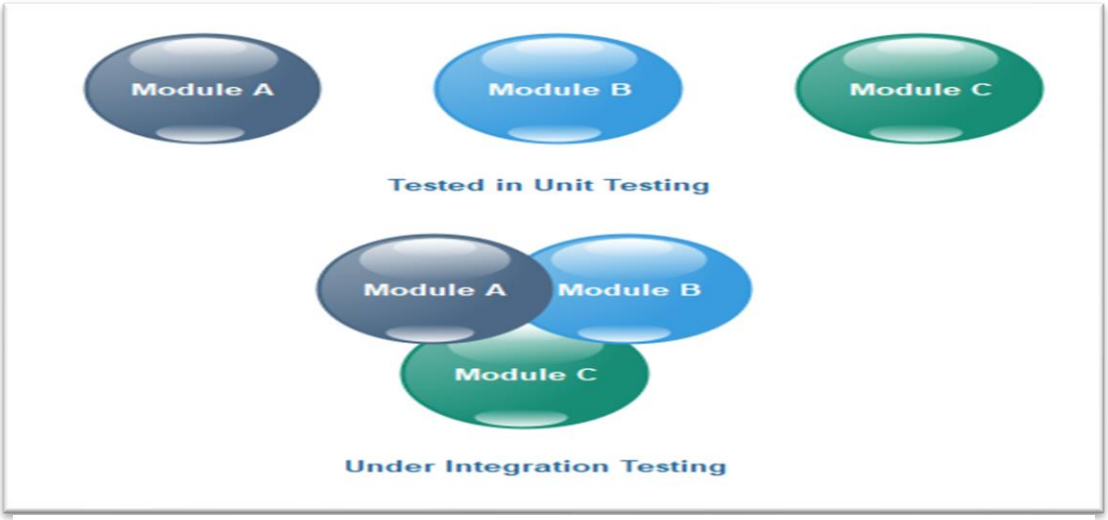


Figure 5.19: Integration Testing

Table 5.6: Integration Testing in Sprints

Sprints	Sprint 3	Sprint4
Errors	• Some mistakes while designing the main page as the navigation bar, when scrolling down the contents of the page like: images, paragraphs, and Tasks appeared above each other's	• The login is done when the most of the fields are empty or have been filled in incorrectly.
Actions	• We solved the issue by changing the properties of the of the nav bar from the CSS style sheet code, and added a button to scroll up to access the nav bar section easily and also change the height and width.	• All the fields are required and the form is not submitted unless all fields are filled out correctly.

5.2.3 System Testing Results

After testing the whole system, it is found that the system works correctly. Every test of a single component or the integrated system achieved the expected output Scenarios that have been tested. And the user can do the following without errors:

- You have been logged in and a new account has been created correctly.
- The system will not accept any form without filling in all the required fields.
- The HR is able to Hiring or Reject employees.
- The HR is able to Make Attending Process.
- The HR is able to Make Attending Process and view employee attendance.
- The Employee is able to Make Attending Process
- The Employee is able to Mange his To-Do List.
- The HR is able to Mange his To-Do List.
- The Applicant is able to View the Jobs there is place to join.
- The Applicant is able to Submit the form & receive a submission email.
- The HR is able to view the new applicant in the recruitment Screen.
- The HR can Follow up the Hiring process as required.

These are the scenarios and units that have been tested:

1.Path of Applicant: Home page =>Search for a job => Applying form => Receive submission email.

2.Path of HR: Home Page =>Login-in =>Dashboard Screen => Attending => recruitment process for hiring employee’s => To-do List =>Logout.

3.Path of Employee: Login-in =>Dashboard Screen => Attending => To-do List =>Logout.

5.2.4 Acceptance System Results

User Acceptance Testing (UAT) is one of the last stages of the software development life cycle. It is performed after the software has been thoroughly tested. A questionnaire was conducted, and a group of HRs were asked to use the site and then answer some questions, including.

Q1) How is the Site Woks?

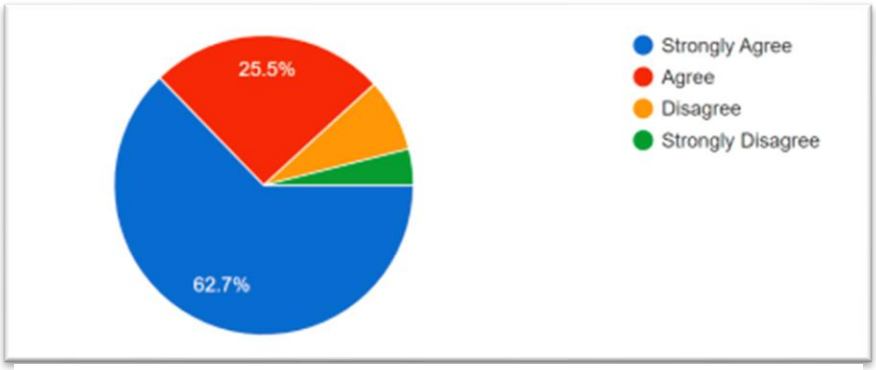


Figure 5.20: Q1 of Questioners

Q2) Do you Find that our System design is User friendly?

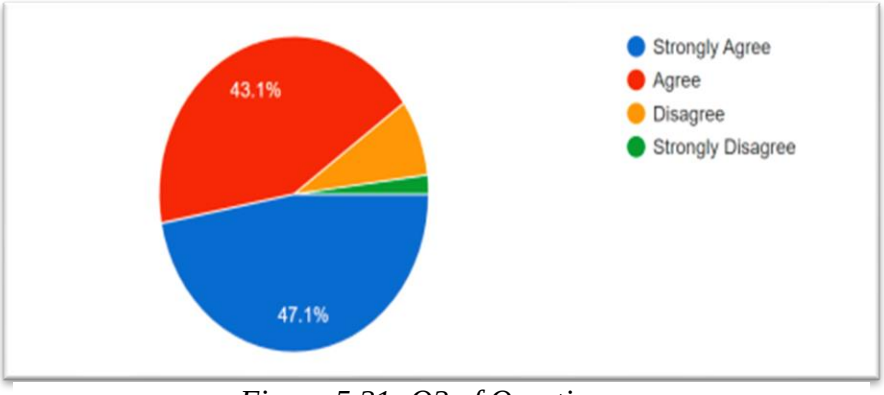


Figure 5.21: Q2 of Questioners

Q3) How the system response to all HR's Process?

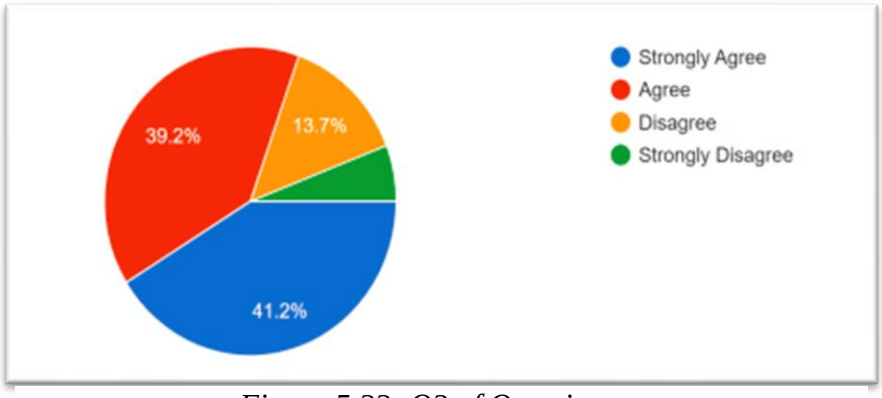


Figure 5.22: Q3 of Questioners

Q4) Do you Find our website save time & effort for HRs in the Recruitment process?

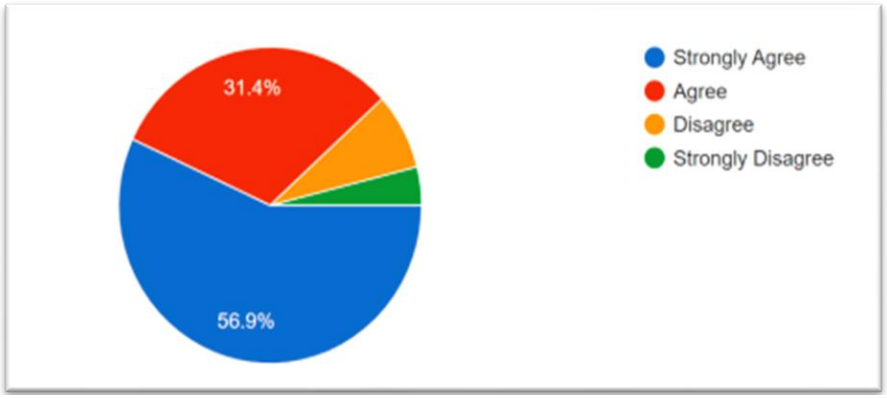


Figure 5.23: Q4 of Questioners

Q5) Did the system perform all tasks??

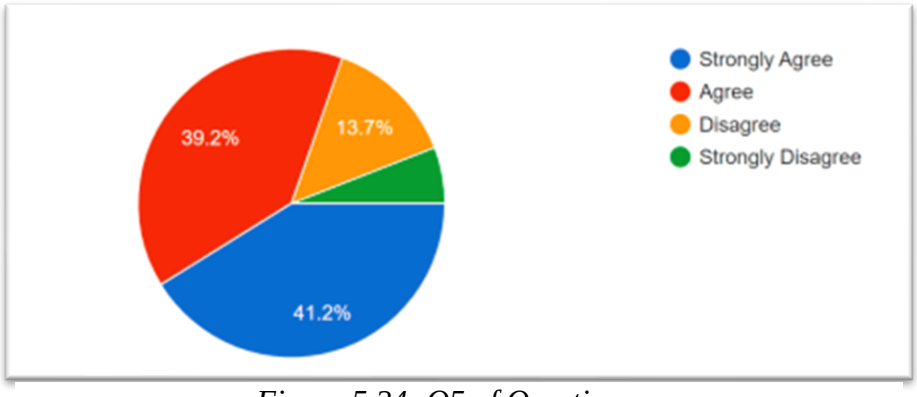


Figure 5.24: Q5 of Questioners

5.3 Discussion and evaluation

After finishing the system testing and the needed adjustments, the processes were being completely performed without errors noticed and they are listed as follows:

- Valid access to the site from HR, Applicant, Team Leader, and Manger Employee.
- The possibility of Filling out the Form by the Applicant to follow up the hiring process in the system, where HR's can manage the hiring applications in efficiently manner with a received emails while the applicant is rejected will receive an email there is a status where an HR can see the status to know in which state. HR and Employee can make the attending process as required in the managed time of the working hours of the company, also can view his attending in a managing table.
- A hierarchy of managing employee's is shown for HR's and Employee in which they can know their team meats in the company, also there is welcome aboard table in the dashboards so they can contact easily together.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Overview

This project will provide a great benefit to the HRs will save time and effort for searching new employees for a company, and any applicant who needs a work will be easy to find through our website. The main idea of the project is to have a HR system available to everyone who needs it. This project will provide a great benefit to the users of the system and the HR and Employees in general in terms of recruiting the process of searching for an applicant. This project was built using (VS) and coded using the (HTML, CSS, JS, Bootstrap, jQuery, Express Library Routes, Node JS, and Mongoose- No SQL) languages. And the web app will increase the level of the applicant registering overall when they find works, and help HRs in their Process.

6.2 Summary about the project

Human Resources Management System is a digital personal assistant to do tedious work for HR professionals. An HRMS helps manage employee functions like onboarding, and managing employee performance which lets HR focus on more important, strategic work. HRMS allows businesses to automate many processes and reduce errors, giving them a leg up against the competition and allowing them more time to perform necessary tasks. This type of platform eases communication, gives companies a way to monitor employees, and saves money over time. If a company has multiple locations or HR supervisors, they can access the HRMS from their various locations and get real-time updates on the status of various tasks relating to recruiting. When a user updates information within the HRMS, the changes are reflected across the platform and are stored in a secured cloud. For even greater ease of use, the main aim of HRM is to ensure the right people with the right skills for the right job position in an organization.

6.3 Achieved objectives

HRM to achieve the organizational objectives like to earn profit, growth, expansion, survival, diversification etc. via the HR functions namely planning, recruiting, selecting, placement, induct, train and develop the human resources and to arrange for performance appraisal of the employees as shown in Figure 6.1: Achieve Organizational Goals.The main objective of this study is: “To develop software that allows HR to access the information of employees & to help recruiters Tracking in an orderly and smooth manner”.

The Sub objectives of this study is:

1. To understand the current situation of the project scope and problem statement.

(This sub objective has been discussed in chapter 1)

2. To develop software that allows HRs to access the information of employees & to help recruiters in Tracing the jobs. (This sub objective has been discussed in Gap Analysis)

3. To validate a simple system for helping HRs. (This sub objective has been discussed in chapter 5).

1.Achieve Organizational Goals:

HRM function starts here. One major HRM objective is to fulfil organizational goals. Utilizing human resources to achieve business requirements and goals is very important for an effective HRM. Organizational objectives include workforce handling, staff requirements like hiring and onboarding, and retirement. To succeed at the organizational objectives, HR requires efficient planning and execution.

2.Work Culture:

When it comes to handling HRM effectively and following objectives, employee and work environment are the prior factors. Work culture plays an important role in defining HRM and business performance.

3.Team Integration:

One of the prime roles and objectives of HRM is to make sure the team coordinates efficiently. Easy communication is the need for teams at an enterprise. An HR here must ensure a tool to assist in making the integration easier and smooth. Ex: Onboarding table, and managing hierarchy.

Figure 6.1: Achieve Organizational Goals

This is why the role of HR has moved from a peripheral activity, considered only as the 'hiring', to become a core management. It's also why today's companies must embrace HR objectives across the organization. Success demands no less than the realization that a company's staff, the business structures supporting them, are at the core of today's competitive business.

6.4 Main contributions of the work

Work contribution means work completed to deliver trunk infrastructure in compliance with a condition of development approval or by an agreement related to the development of project**.**

"I can contribute to the project in several ways. As a team we learned from each other, we worked as a team to help each other in all matters of the project. It started out as my simple idea and evolved into a team of three working on it. We beat our submission deadline by one weeks, our system was discussed with our doctors". An HR department can help provide organizational structure and the ability to meet business needs by effectively managing the employee lifecycle, HR function helps an organization deliver its corporate strategy and objectives by effectively recruiting and developing people and managing their performance. HR objectives are delivered differently in different types of organizations. Recruitment refers to the process of identifying, attracting, interviewing, selecting, hiring and onboarding employees. One of HR's primary roles is

managing recruitment and Attending for an organization's staff. An HR professional holding a management position is often in charge of recruiting, and hiring new employees. This means the HR team is responsible for finding candidates who meet the necessary qualifications for specified positions and fully vetting a company's hires as shown as Figure 6.2: Objectives of HRMS.

Figure 6.2: Objectives of HRMS.

6.5 Limitation

During the construction of the system, we faced Some difficulties, such as the difficulty of collecting information's and determining the requirements, difficulty of determining the corporates and their department's structure and difficulty of determining the main functionalities of human resources system.

6.6 Future Work

There is a quote that says, "The end of things may be the beginning of better things." / Louis Pasteur [8].

The future work of the project that could improve its performance:

- Provide the website with a chat for easy communication
- Add more languages
- Integrate Payroll and Training process in the system to calculate all required salaries.

REFERENCES

1- Article:

[2] M. H. N. Nasir and S. Sahibuddin, "Critical success factors for software projects: A comparative study," Scientific research and essays, vol. 6, no. 10, pp. 2174-2186, 2011.

[3] L.-y. Shen, V. W. Tam, L. Tam, and Y.-b. Ji, "Project feasibility study: the key to successful implementation of sustainable and socially responsible construction management practice," Journal of cleaner production, vol. 18, no. 3, pp. 254-259, 2010

[8] G. Bordenave, "Louis Pasteur (1822–1895)," Microbes and infection, vol. 5, no. 6, pp. 553-560, 2003.

2- Conference:

[6] R. Ibrahim, "Formalization of the data flow diagram rules for consistency check," arXiv preprint arXiv:1011.0278, 2010.

3- Book:

[1] J. Bell and S. Waters, EBOOK: DOING YOUR RESEARCH PROJECT: A GUIDE FOR FIRST-TIME RESEARCHERS. McGraw-Hill Education (UK), 2018.

[4] J. Kuada, Research methodology: A project guide for university students. Samfundslitteratur, 2012.

[5] H. Gomaa, Software modeling, and design: UML, use cases, patterns, and software architectures. Cambridge University Press, 2011.

[7] J. Berger, "A comparison of testing methodologies," 2008.

4- Website:

[9] https://app.diagrams.net/

[10] https://plotly.com/javascript/

[11] https://mycolor.space/

[12] https://getbootstrap.com/

[13] https://discord.com/

Appendices

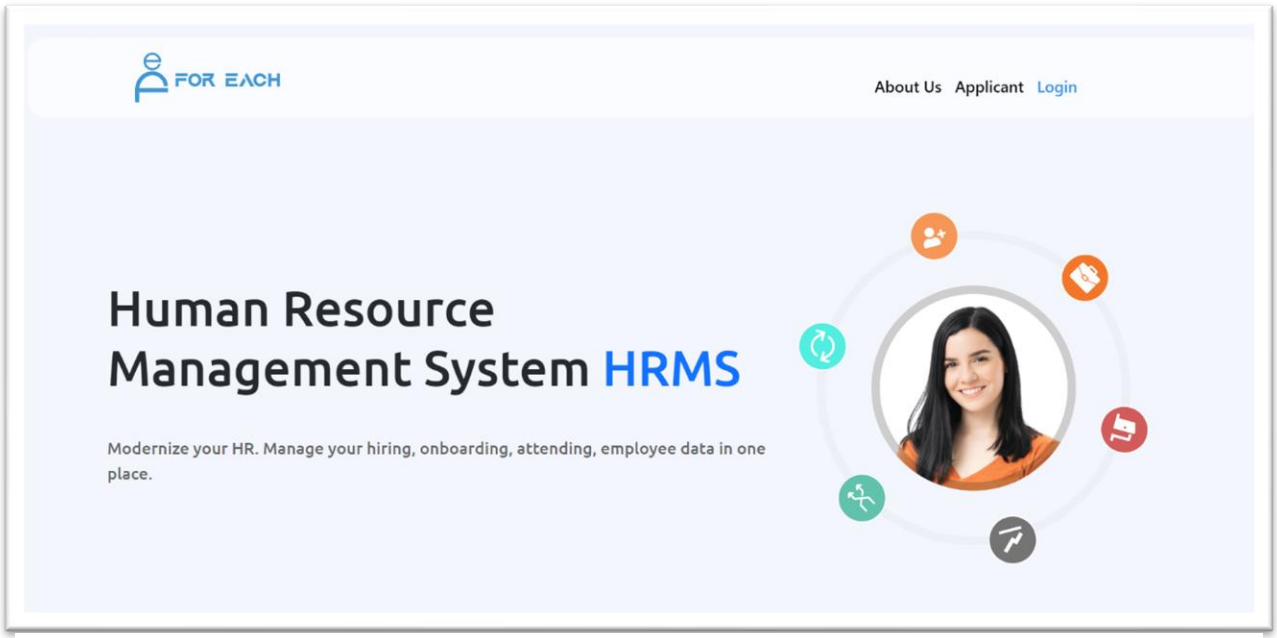


Figure A.1: Home page

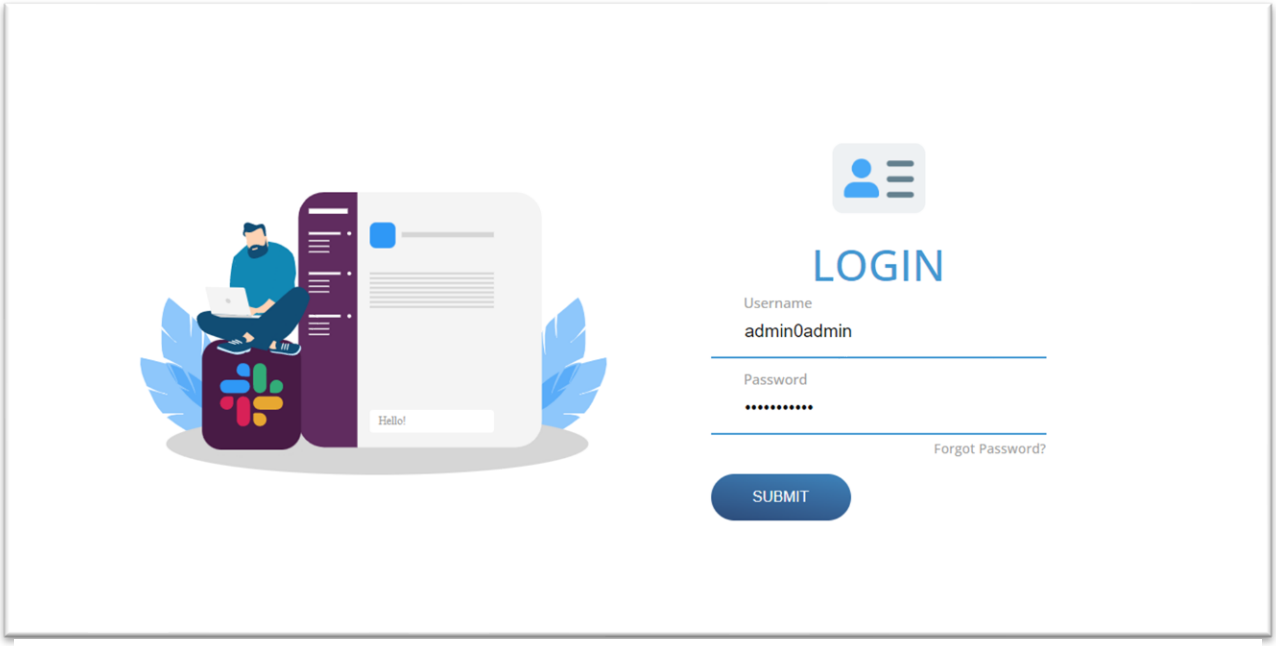


Figure A.2:Login Page

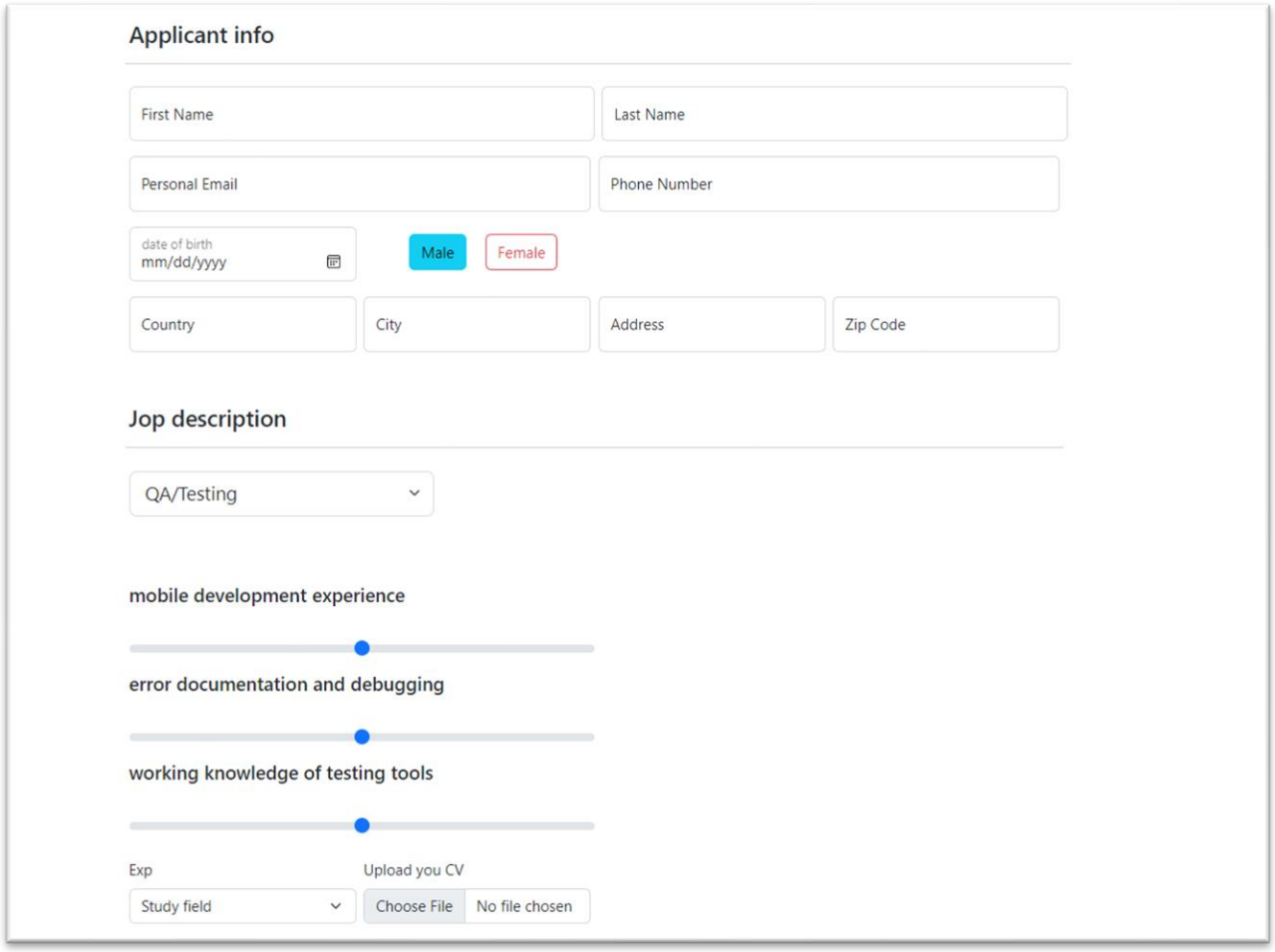


Figure A.3: Applicant Form

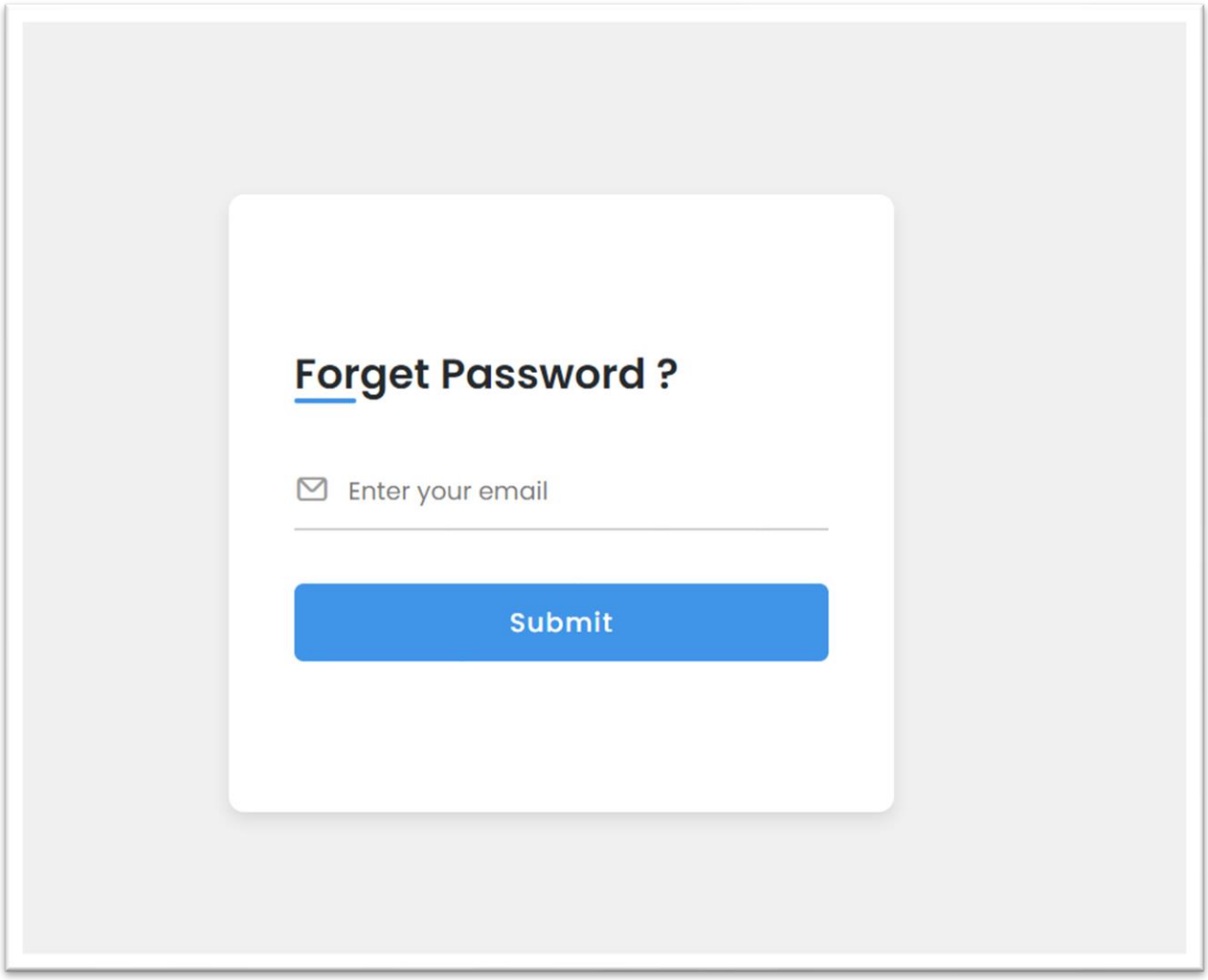


Figure A.4: Forgot Password

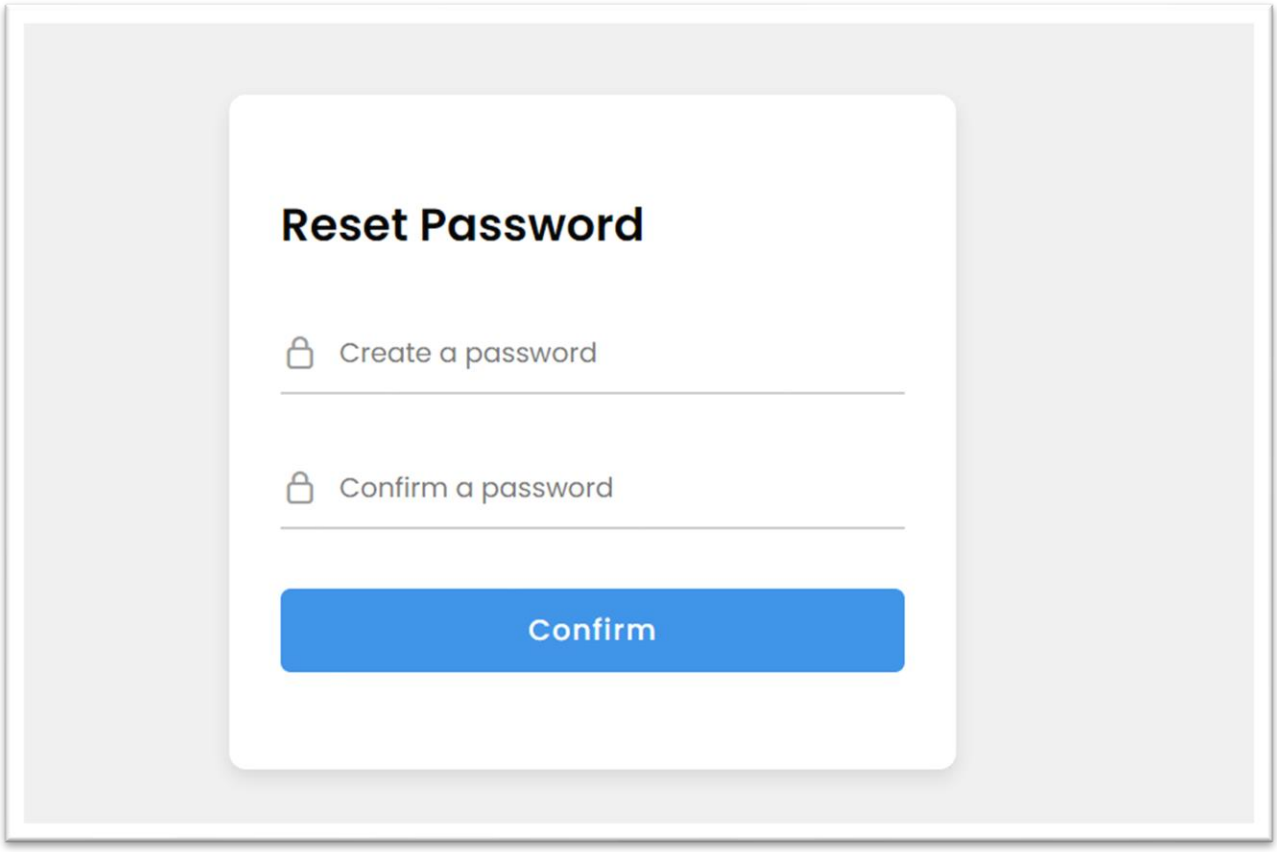


Figure A.5:Reset Password

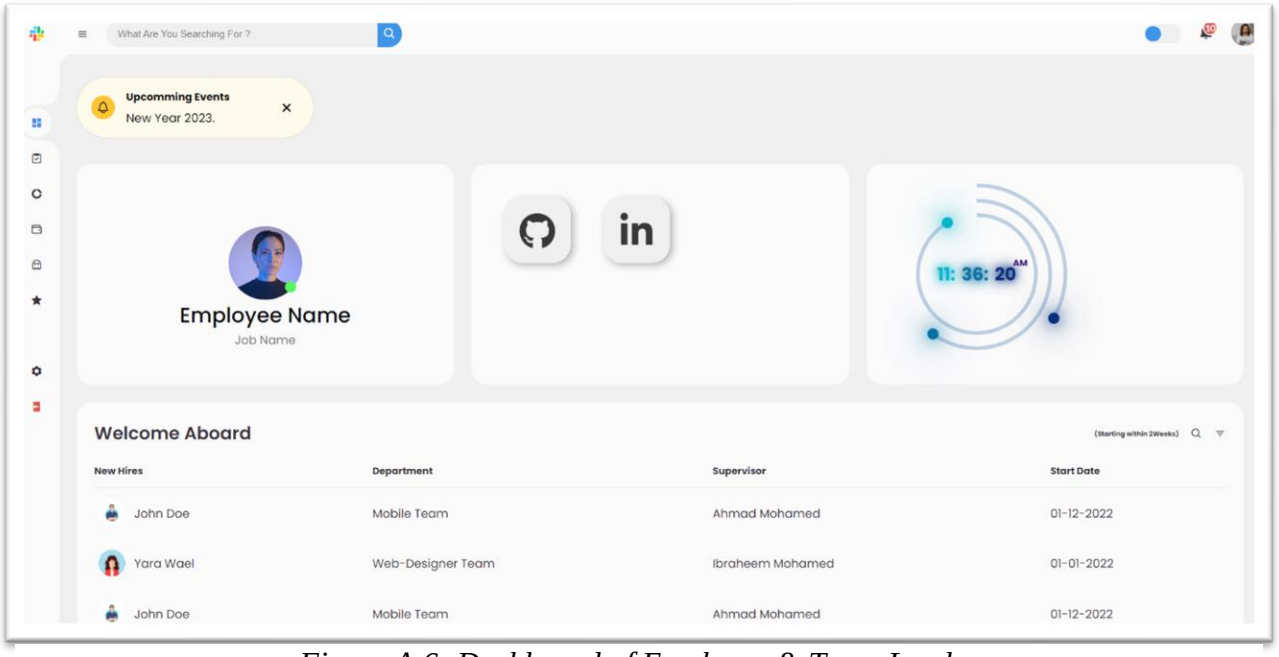


Figure A.6: Dashboard of Employee & Team Leader

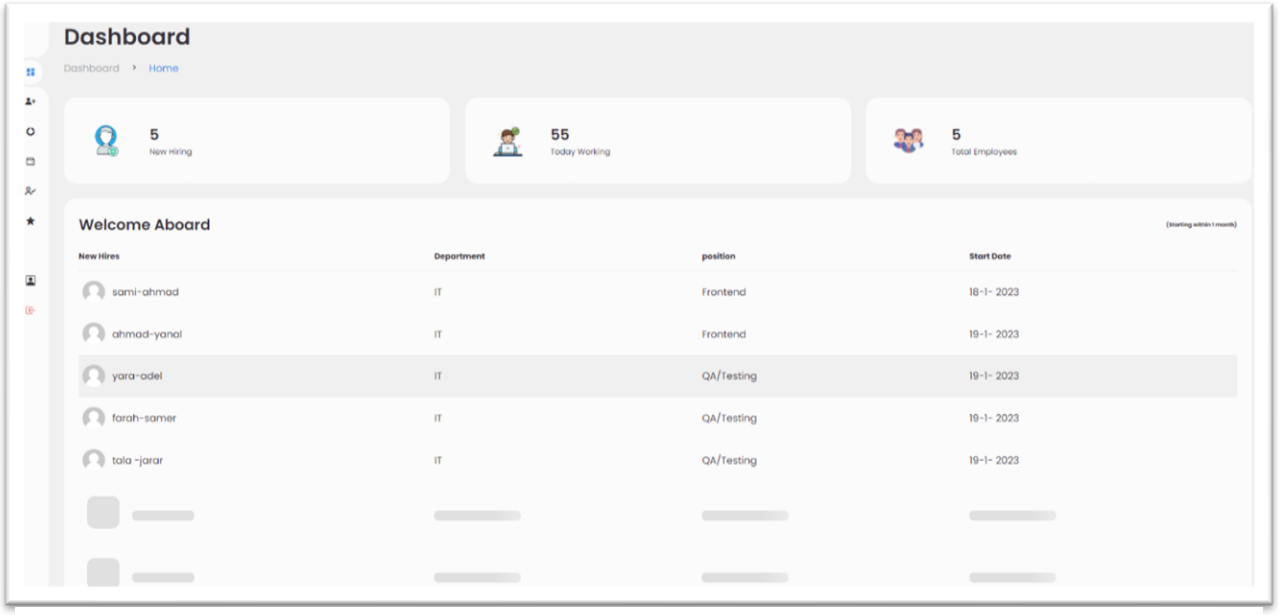


Figure A.7: Dashboard of HR & Manager

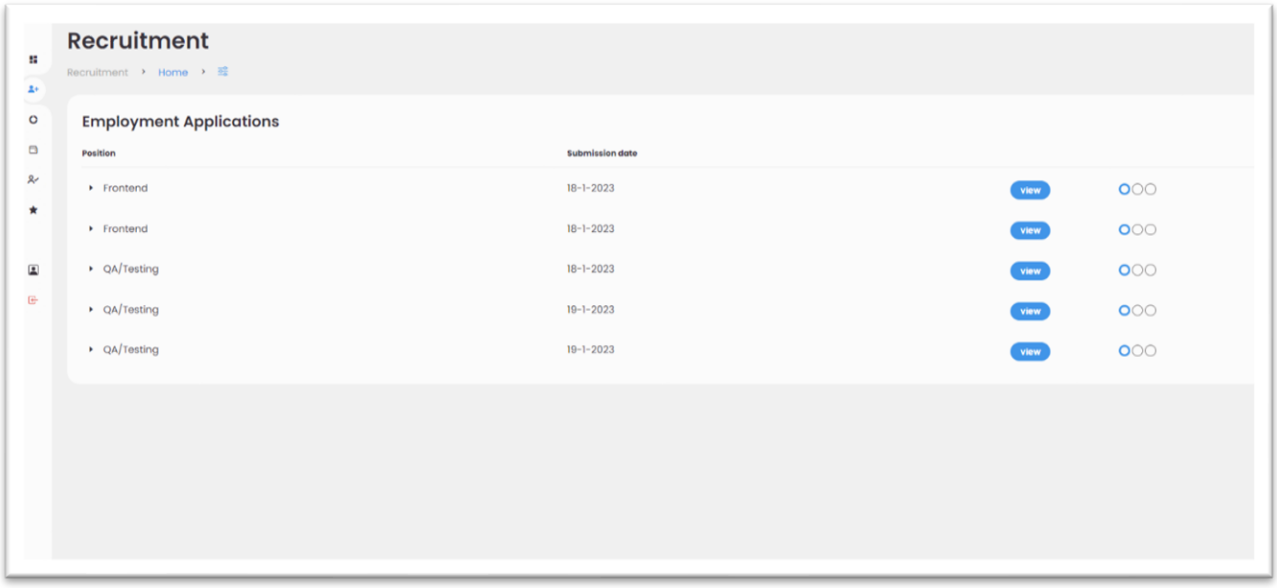
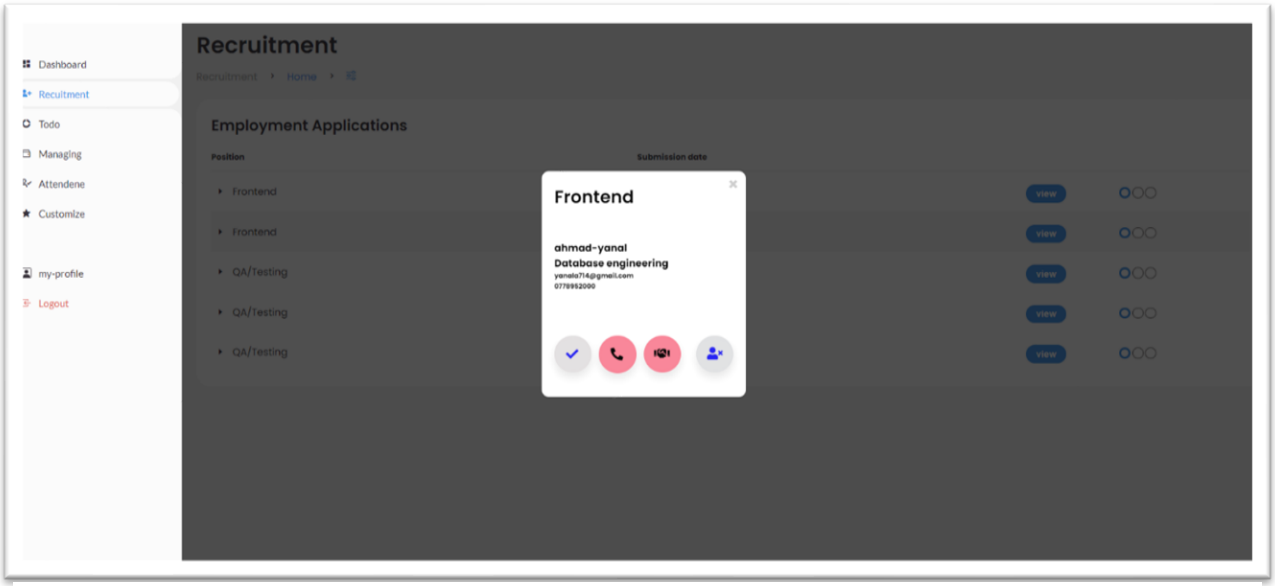


Figure A.8: Recruitment Process "1"



FigureA.9: Recruitment process "2"

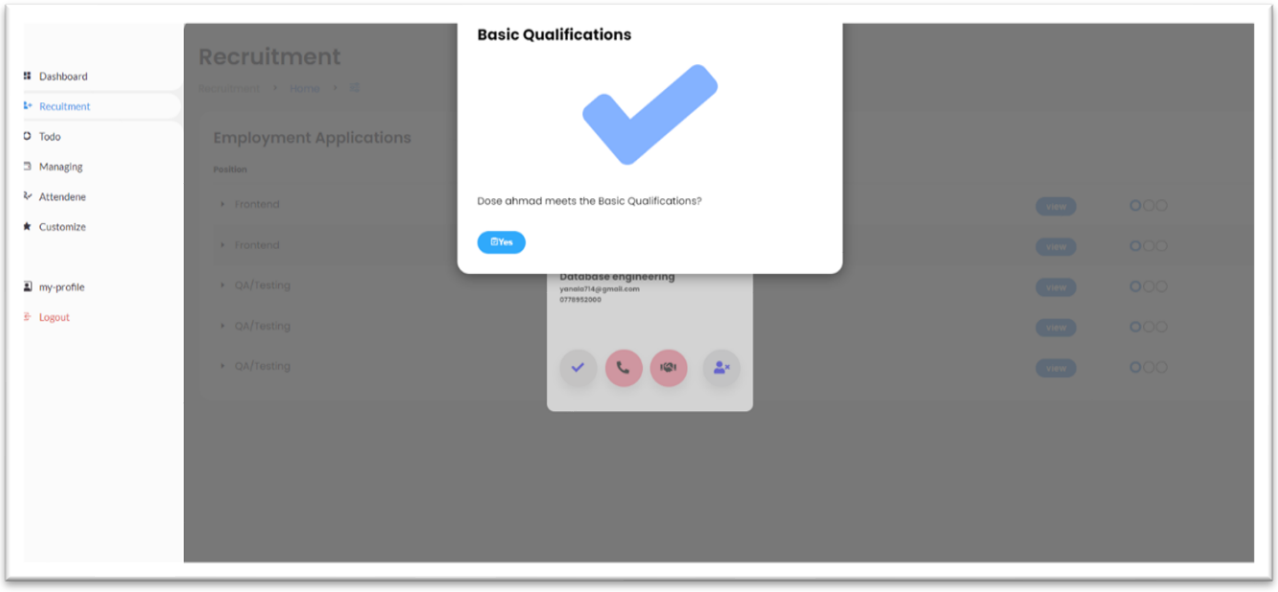


Figure A.11: Recruitment Process "3"

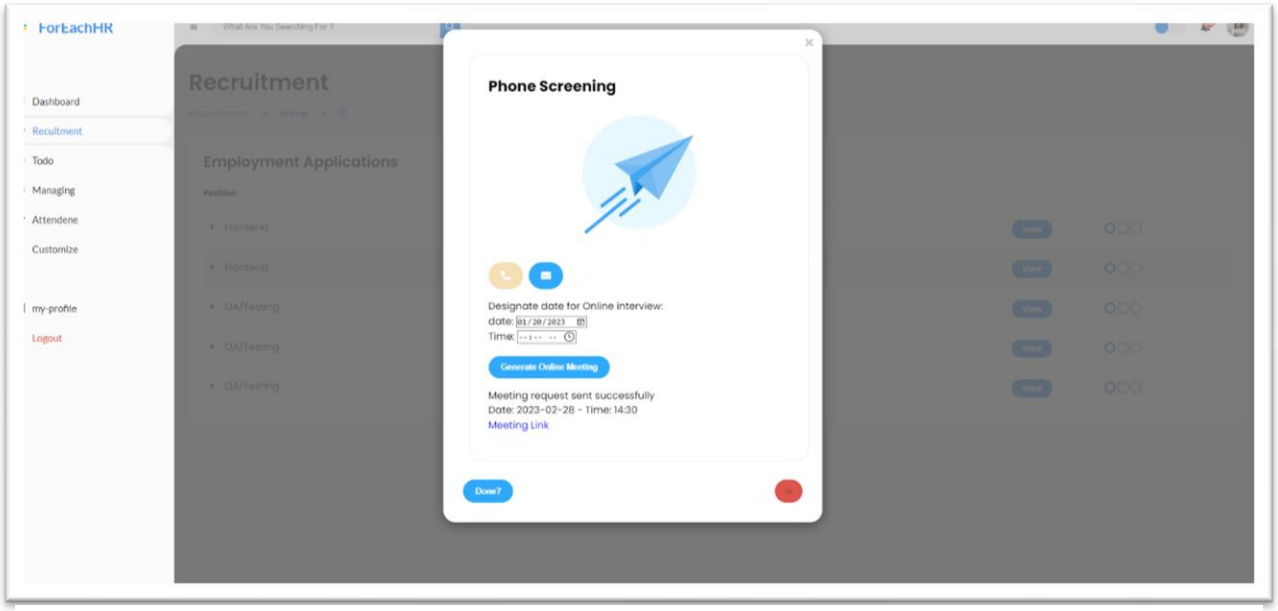


Figure A.12: Recruitment Process "4"

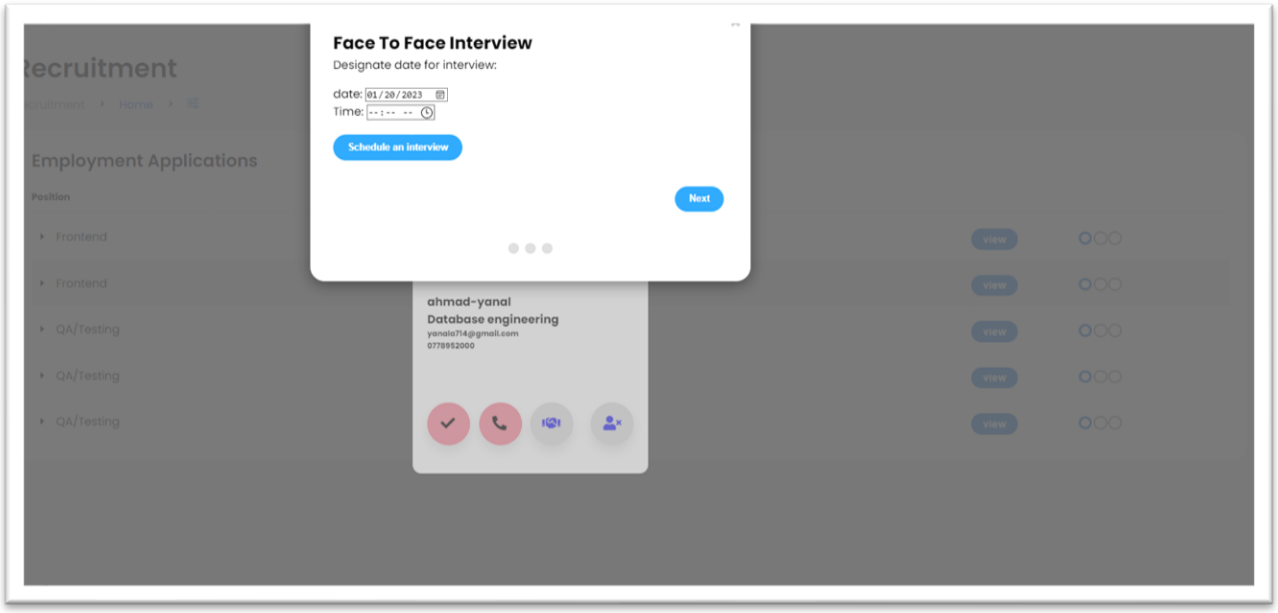


Figure A.13: Recruitment Process "5"

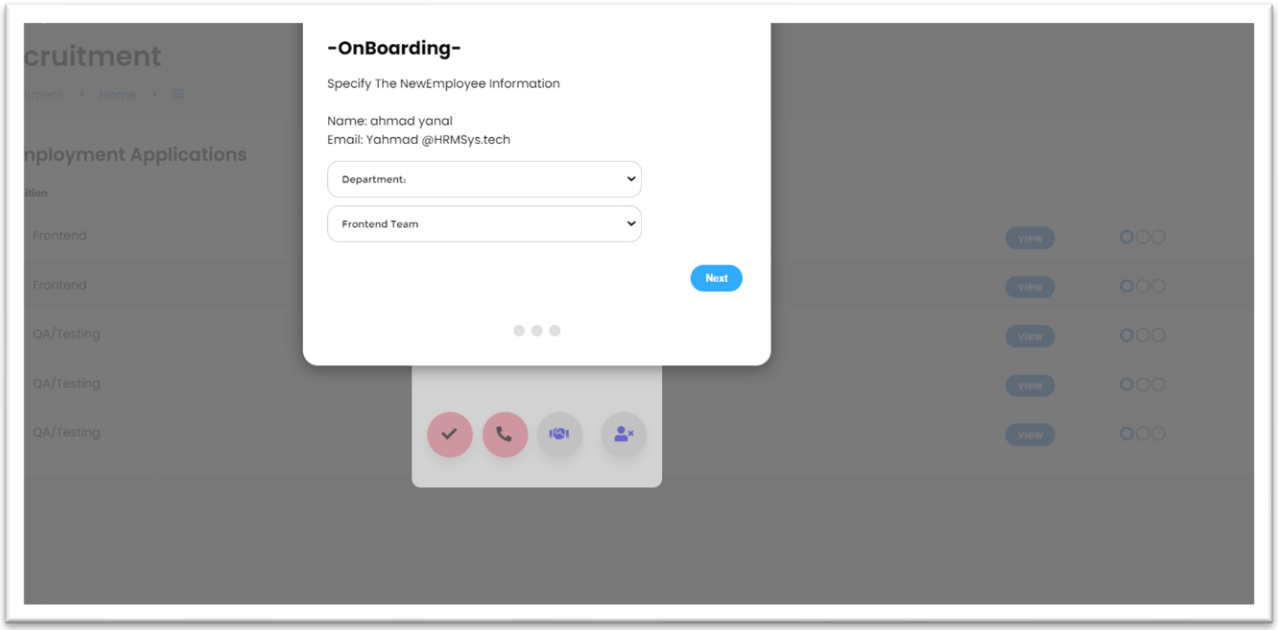


Figure A.14: Recruitment Process "6"

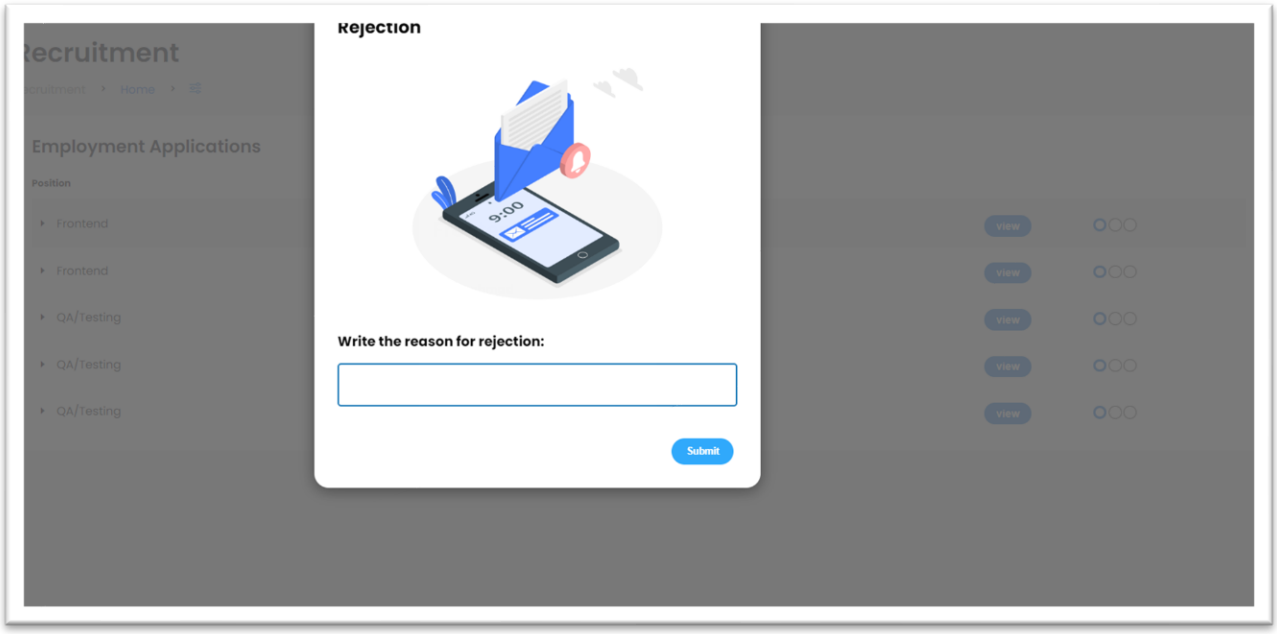


Figure A.15: Recruitment Process "7"

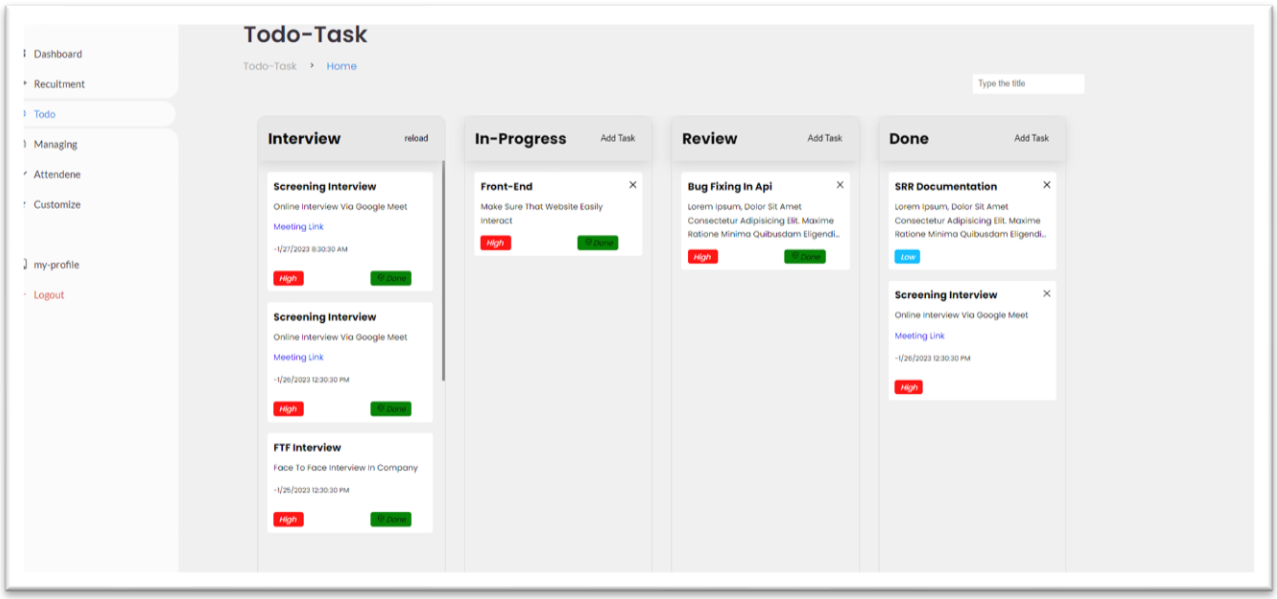


Figure A.16: To-do for HR & Manger

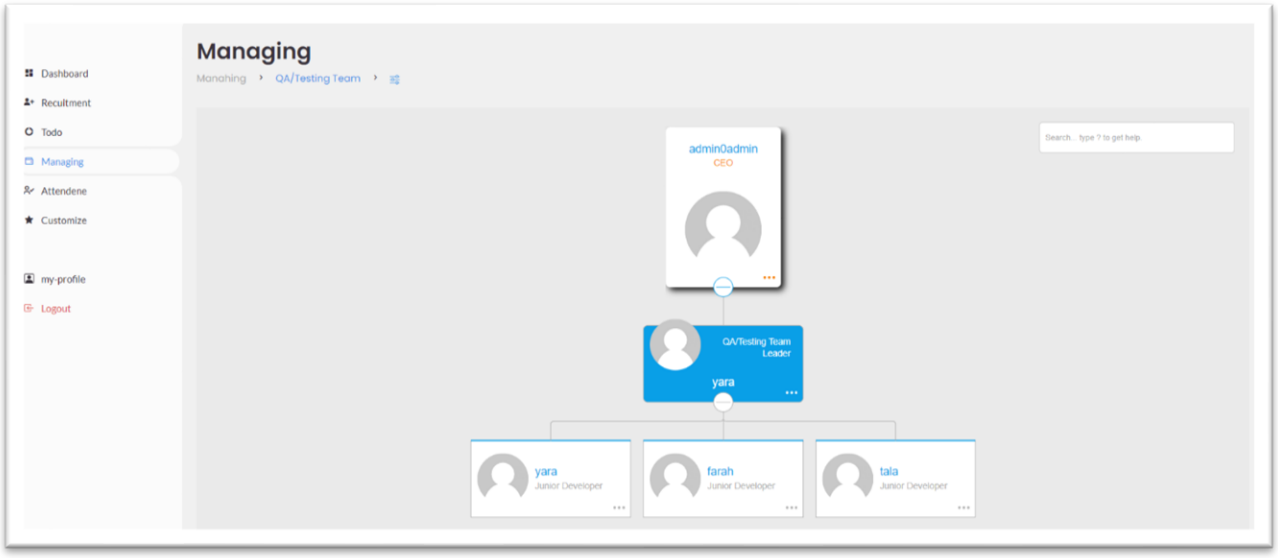


Figure A.17: Managing Screen

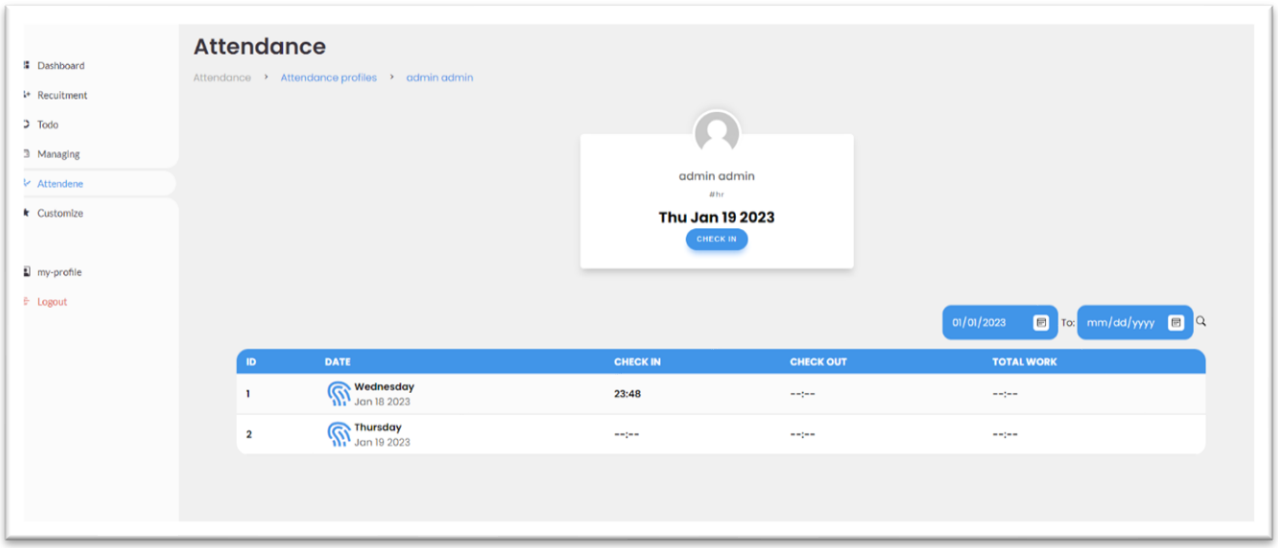


Figure A.18: Attendance Process

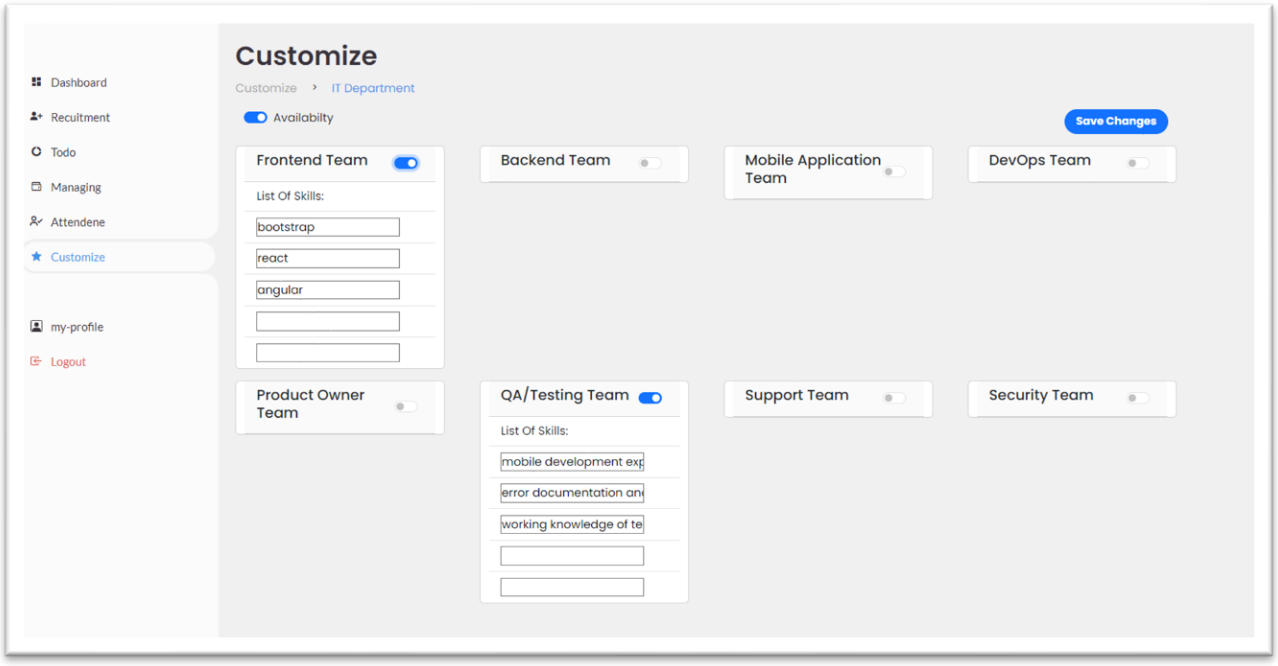


Figure A.19: Customize Applicant

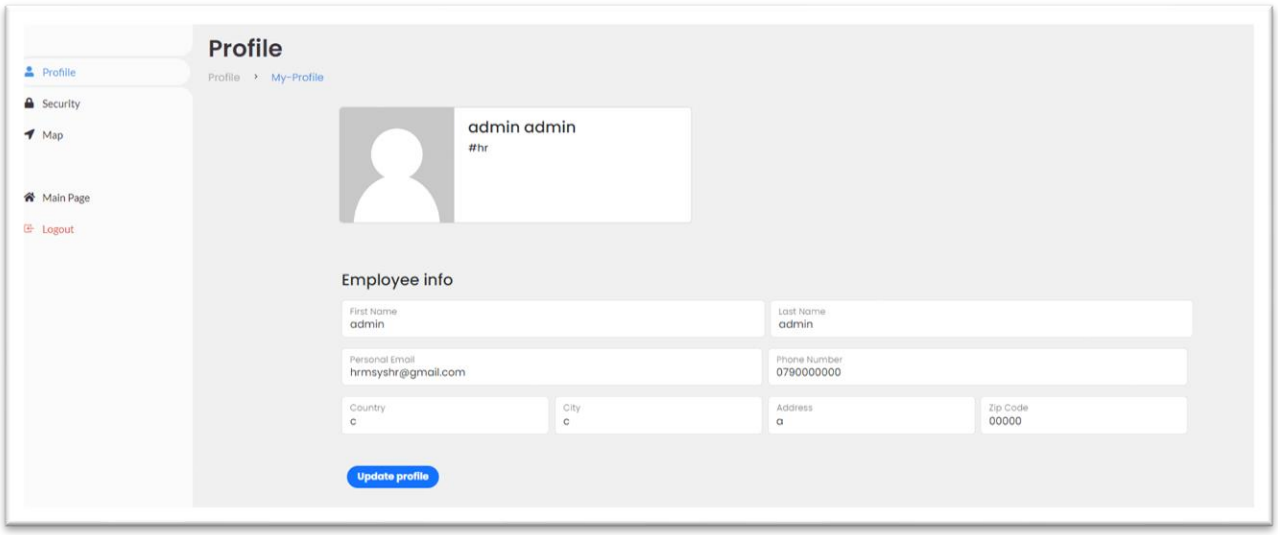


Figure A.20: Profile Screen

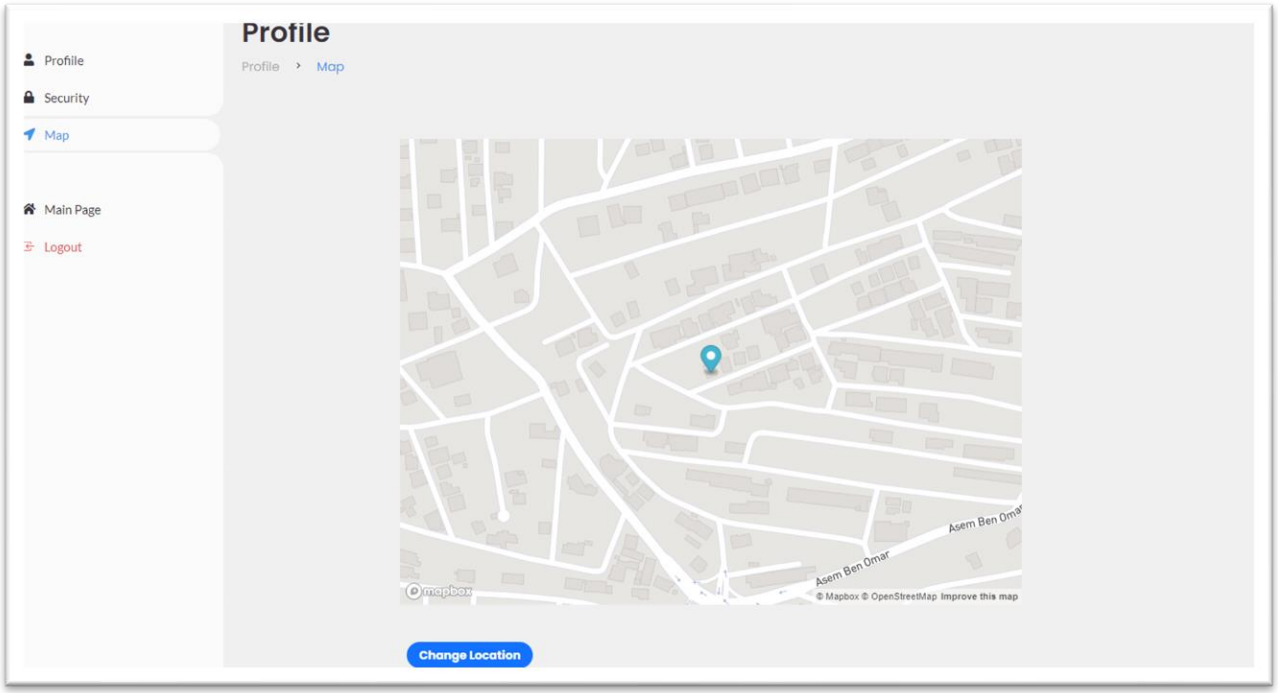


Figure A.21: Map Screen

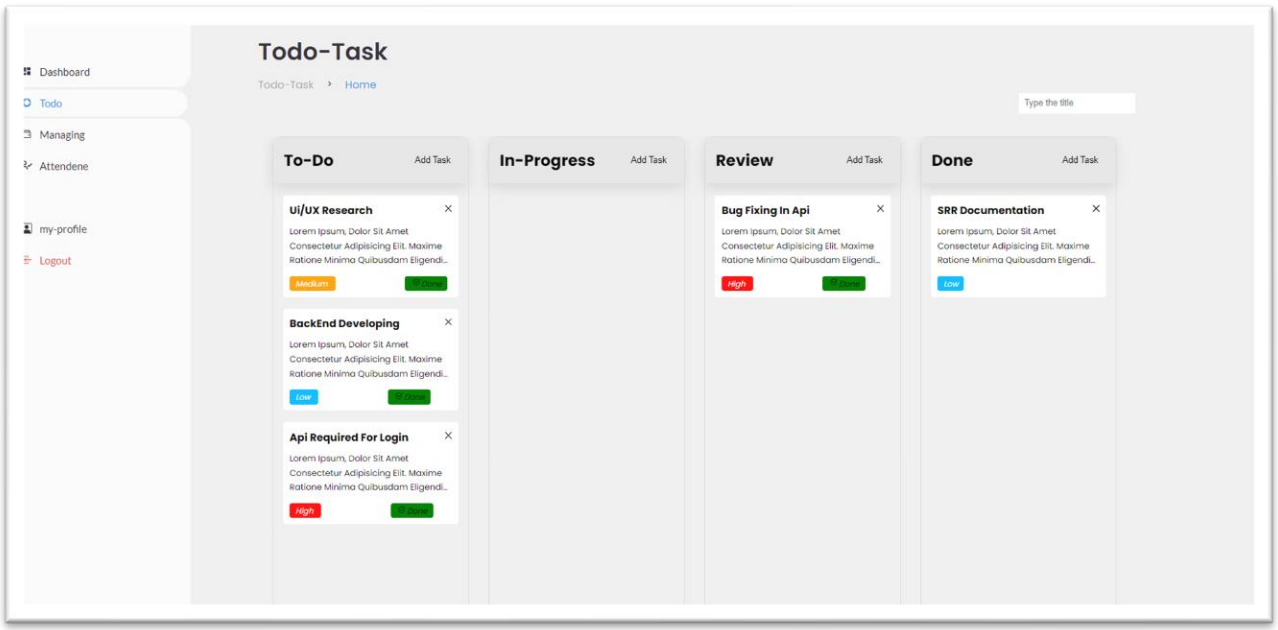
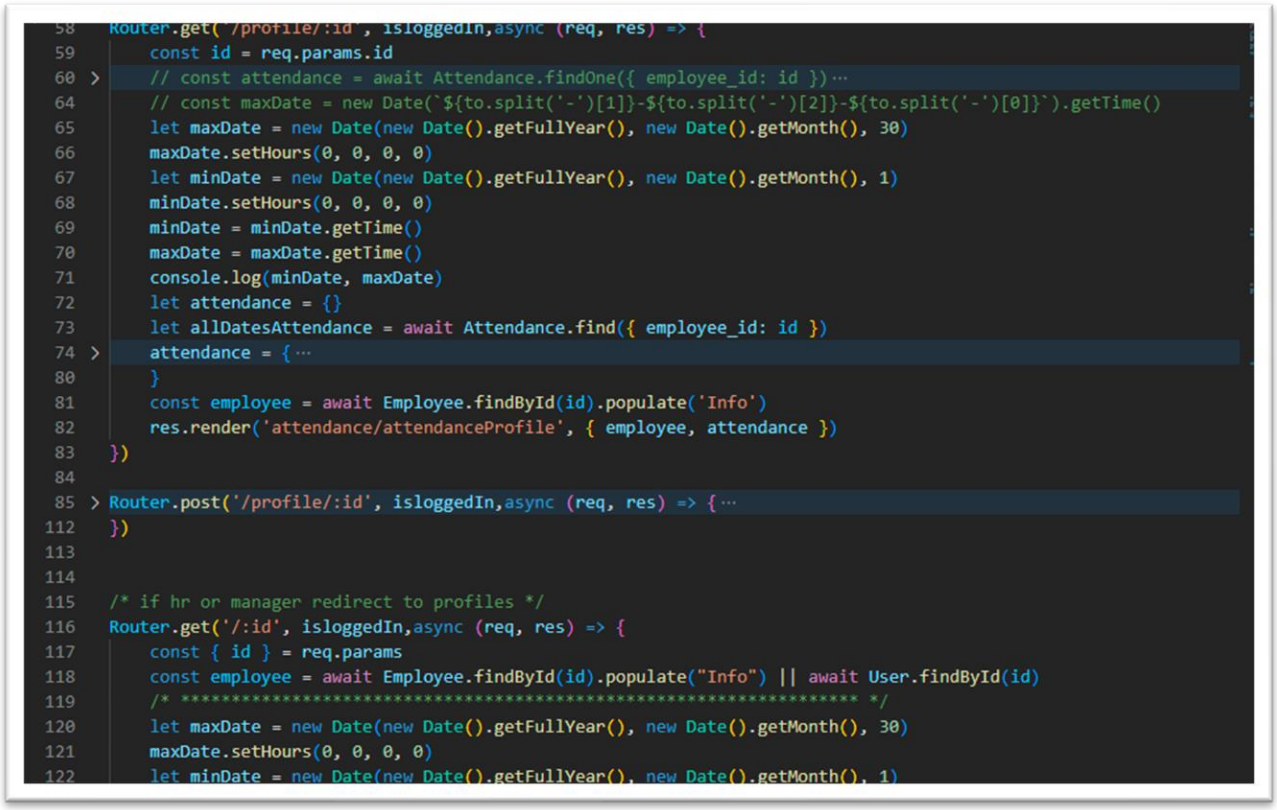


Figure A.22:To-do for Employee & Team Leader



```
Router.get('/profile/:id', isloggedIn,async (req, res) => {
    const id = req.params.id
    // const attendance = await Attendance.findOne({ employee_id: id }) ···
    // const maxDate = new Date(`${to.split('-')[1]}-${to.split('-')[2]}-${to.split('-')[0]}`).getTime()
    let maxDate = new Date(new Date().getFullYear(), new Date().getMonth(), 30)
    maxDate.setHours(0, 0, 0, 0)
    let minDate = new Date(new Date().getFullYear(), new Date().getMonth(), 1)
    minDate.setHours(0, 0, 0, 0)
    minDate = minDate.getTime()
    maxDate = maxDate.getTime()
    console.log(minDate, maxDate)
    let attendance = {}
    let allDatesAttendance = await Attendance.find({ employee_id: id })
    attendance = { ···
    }
    const employee = await Employee.findById(id).populate('Info')
    res.render('attendance/attendanceProfile', { employee, attendance })
})

Router.post('/profile/:id', isloggedIn,async (req, res) => { ···
})


/* if hr or manager redirect to profiles */
Router.get('/:id', isloggedIn,async (req, res) => {
    const { id } = req.params
    const employee = await Employee.findById(id).populate("Info") || await User.findById(id)
    /* ********************************************************************** */
    let maxDate = new Date(new Date().getFullYear(), new Date().getMonth(), 30)
    maxDate.setHours(0, 0, 0, 0)
    let minDate = new Date(new Date().getFullYear(), new Date().getMonth(), 1)
```

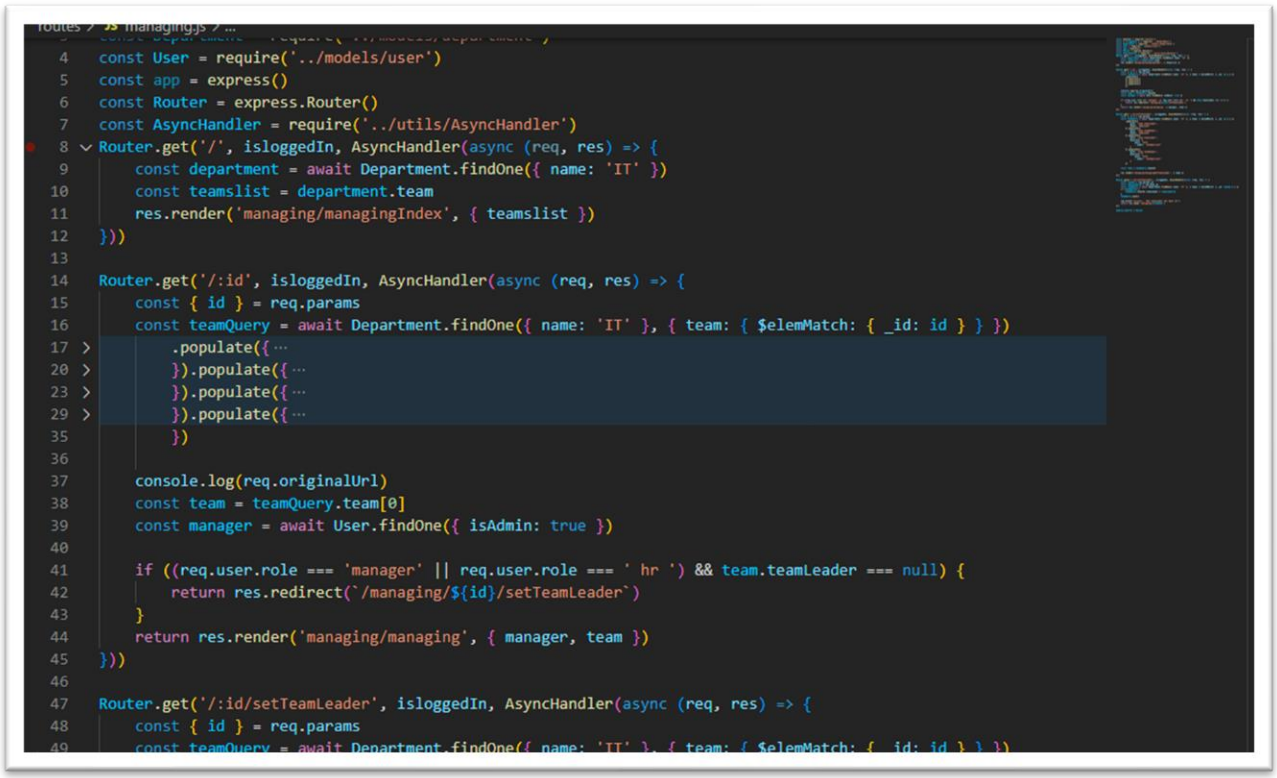
Figure A.23: Backend-Attendance

```
routes > JS customize.js > Router.get('/') callback
const User = require('../models/user')
const Department = require('../models/department')

Router.get('/', async (req, res) => {
    const user = await User.findOne({ isAdmin: true })
    const department = await Department.findOne({ name: "IT" })


    if (!user.applicantForm) {
        let positionsName = []
        const applicantForm =
        {
            ApplicationAvailability: true,
            positions: []
        }
        applicantForm.ApplicationAvailability = true
        department.team.forEach(team => {
            applicantForm.positions.push({
                name: team.name,
                availability: true,
                skills: [],
            })
        })
        user.applicantForm = applicantForm
        user.save()

        return res.render('customize/customize', { applicantForm })
    } else {
        const applicantForm = user.applicantForm
        return res.render('customize/customize', { applicantForm })
    }
})
```

Figure A.24: Backend-Customize

```
routes > JS dashboard.js > ...
49  Router.get('/:id',isloggedIn, AsyncHandler(async (req, res) => {
50
51      let {role} = req.user
52      const { id } = req.params
53      let currentDate = new Date;
54      const thisMonth = new Date(currentDate.setMonth(currentDate.getMonth()))
55      let lasMonth = new Date(currentDate.setMonth(currentDate.getMonth() - 1))
56      thisMonth.setDate(thisMonth.getDate())
57
58      const aboard = await Employee.find({
59          'startDate': {
60              "$gte": lasMonth,
61          }
62      }).populate('Info').populate('department')
63      console.log(role)
64      const hr = await User.findById(id)
65      console.log(hr)
66      if (hr) {
67          const totalEmp = (await Employee.find()).length
68          const newHiring = aboard.length
69          return res.render('dashboard/DashboardHM', { totalEmp, newHiring, aboard })
70      } else {
71          const event = await upComingEvents()
72          let  {summary, start}  = event[0]
73          if(!event){
74              summary="", start=""
75          }
76          console.log(event[0])
77          const emp = await Employee.findById(id).populate('Info').populate('department')
78          return res.render('dashboard/DashboardET', { emp, aboard, summary, start })
79      }
```

Figure A.25: Backend-Dashboard



```
routes > JS managingjs > ...
4   const User = require('../models/user')
5   const app = express()
6   const Router = express.Router()
7   const AsyncHandler = require('../utils/AsyncHandler')
8   Router.get('/', isloggedIn, AsyncHandler(async (req, res) => {
9       const department = await Department.findOne({ name: 'IT' })
10      const teamslist = department.team
11      res.render('managing/managingIndex', { teamslist })
12  }))
13
14  Router.get('/:id', isloggedIn, AsyncHandler(async (req, res) => {
15      const { id } = req.params
16      const teamQuery = await Department.findOne({ name: 'IT' }, { team: { $elemMatch: { _id: id } } })
17          .populate({ ···
20          }).populate({ ···
23          }).populate({ ···
29          }).populate({ ···
35          })
36
37      console.log(req.originalUrl)
38      const team = teamQuery.team[0]
39      const manager = await User.findOne({ isAdmin: true })
40
41      if ((req.user.role === 'manager' || req.user.role === ' hr ') && team.teamLeader === null) {
42          return res.redirect(`/managing/${id}/setTeamLeader`)
43      }
44      return res.render('managing/managing', { manager, team })
45  }))
46
47  Router.get('/:id/setTeamLeader', isloggedIn, AsyncHandler(async (req, res) => {
48      const { id } = req.params
49      const teamQuery = await Department.findOne({ name: 'IT' }, { team: { $elemMatch: { _id: id } } })
```

Figure A.26: Backend-Managing

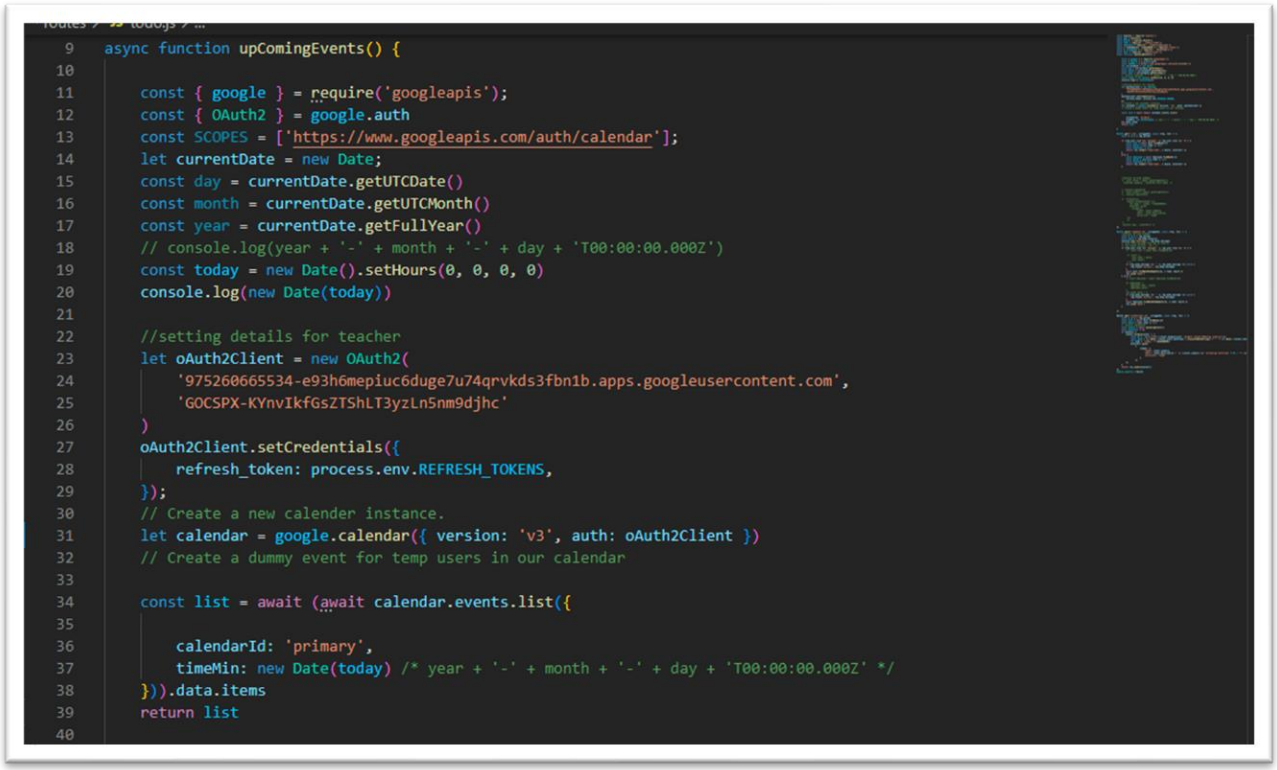
routes > JS profile.js > ...

```
35
36
37
38  Router.get('/:id', isloggedIn, AsyncHandler(async (req, res) => {
39      const { id } = req.params
40      if (req.user.role === 'hr' || req.user.role === 'manager') {
41          const hr = await User.findById(id)
42          const Info = hr.Info
43          const image = hr.image
44          const userId = hr.id
45          return res.render('profile/profile', { Info, image, userId })
46      } else {
47          const employee = await Employee.findById(id).populate('Info')  /* or dine in User Model */
48          const Info = employee.Info
49          const image = employee.image
50          const userId = employee.id
51          return res.render('profile/profile', { Info, image, userId })
52      }
53
54  }))
55
56  Router.post('/:id', isloggedIn, upload.single('image'), AsyncHandler(async (req, res) => {
57      const { id } = req.params
58      const body = req.body
59
60      if (req.user.role === 'hr' || req.user.role === 'manager') {
61          const hr = await User.findById(id)
62          if (req.file) {
63              console.log(req.file.path)
64              const image = req.file.path.replace('/upload/', '/upload/w_200,h_200/') /* req.file.path */
65              await hr.updateOne({ image: image })
66
```

FigureA.26: Backend-Profile

```
146     res.render('users/newForm', { applicantForm, fieldName })
147 }))
148
149
150 router.post('/new', upload.single('cv'), AsyncHandler(async (req, res) => {
151     const { firstName, surName, address, zipCode, applyingFor, linkedinProfile, githubProfile, dateOfBirth,
152         expectedSalary, department, personalEmail, gender, portfolioUrl, phone, skills, field } = req.body
153     const cv = req.file.path.replace('/upload/', '/upload/fl_attachment/')
154     console.log(req.body)
155     const age = derivedAge(dateOfBirth)
156     //add /fl_attachment/ to cv
157     const applicantDate = applyingDate()
158     const newapplicant = new newApplicant({
159         firstName: firstName, //
160         surName: surName,//
161         address: address,//
162         zipCode: zipCode,//
163         applyingFor: applyingFor,
164         linkedinProfile: linkedinProfile,
165         githubProfile: githubProfile,
166         phone: phone,//
167         age,//
168         applyingDate: applicantDate,//
169         personalEmail: personalEmail,
170         cv,
171         requestStatus: 'inProgress',
172         skills: skills,
173         dateOfBirth: dateOfBirth,
174         gender: gender,
175         field: field,
176
177     })
```

Figure A.27: Backend-recruitment



```
async function upComingEvents() {

    const { google } = require('googleapis');
    const { OAuth2 } = google.auth
    const SCOPES = ['https://www.googleapis.com/auth/calendar'];
    let currentDate = new Date;
    const day = currentDate.getUTCDate()
    const month = currentDate.getUTCMonth()
    const year = currentDate.getFullYear()
    // console.log(year + '-' + month + '-' + day + 'T00:00:00.000Z')
    const today = new Date().setHours(0, 0, 0, 0)
    console.log(new Date(today))

    //setting details for teacher
    let oAuth2Client = new OAuth2(
        '975260665534-e93h6mepiuc6duge7u74qrvkds3fbn1b.apps.googleusercontent.com',
        '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'
    )
    oAuth2Client.setCredentials({
        refresh_token: process.env.REFRESH_TOKENS,
    });
    // Create a new calender instance.
    let calendar = google.calendar({ version: 'v3', auth: oAuth2Client })
    // Create a dummy event for temp users in our calendar

    const list = await (await calendar.events.list({

        calendarId: 'primary',
        timeMin: new Date(today) /* year + '-' + month + '-' + day + 'T00:00:00.000Z' */
    })).data.items
    return list

```

Figure A.28: Backend-Todo

```
router.get('/register', isloggedIn, (req, res) => {
    res.render('users/register')
})

router.post('/register', isloggedIn, AsyncHandler(async (req, res) => {
    //auth the currentUser middleware if he is manager or hr only
    const { username, email, password, role } = req.body;
    if (req.body.role && req.body.role === 'hr') {
        if (await Employee.findByUsername(username)) {
            throw new Error('A user with the given username is already registered')
        } else if (await Employee.findOne({ email })) {
            throw new Error('A user with the given email is already registered')
        } else {
            const user = new User({ email, username, role });
            const registeredUser = await User.register(user, password);
            req.flash('success', `hr - ${username} has been registered`)
        }

    } else {
        if (await User.findByUsername(username)) {
            throw new Error('A user with the given username is already registered')
        } else if (await User.findOne({ email })) {
            throw new Error('A user with the given email is already registered')
        } else {
            const user = new Employee({ email, username, role });
            const registeredEmployee = await Employee.register(user, password);
            req.flash('success', `new ${role} -${username} has been registered`)
        }

    }
    res.redirect('/login')
```

Figure A.29: Backend-login