

HRMS
SPROJ Report



Aamina Mariam 22100062
Javeria Tariq 22100230
Ali Adnan Arif 22100303
Mohammad Yousuf 22100289

Advisor: Waqar Ahmed
School of Science and Engineering
Lahore University of Management Sciences
30th April 2022

Acknowledgement and Dedication

We would like to acknowledge and express appreciation to our project Supervisor Mr. Waqar Ahmad for his continued guidance and support at different stages of the project and also for his invaluable feedback throughout the project.

Certificate

I certify that the senior project titled “**HRMS**” was completed under my supervision by the following students:

Mohammad Yousuf



Aamina Mariam



Javeria Tariq



Ali Adnan Arif

and the project deliverables meet the requirements of the program.

Date:

Advisor (Signature)

Date:

Table of Contents

Table of Contents	4
List of Figures	6
Introduction	8
Introduction	8
Objective and Scope	9
Development Methodology	9
Contributions	9
2. System Requirements	11
System Actors	11
Functional Requirements	11
Non-functional Requirements	13
3. System Architecture	14
Architecture Diagram	14
Architecture Description	15
Justification of the Architecture	16
Tools and Technologies	17
4. Requirements Specifications	18
4.1. Use Case Diagram	18
4.2 Use Cases	19
4.3 Class Diagram	32
4.4 Sequence Diagrams	33
5. Software Development Methodology and Plan	53
Software Process Selection	53
Gantt Chart	55
6. Database Design and Web Services	56
a. Database Design	56

b. API Specification	58
7. System User Interface	58
8. Project Security	70
Project Threats	70
Potential Losses	70
Security Controls	71
Static and Dynamic Security Scanning Tools	71
9. Risk Management	73
Potential Risks and Mitigation Strategies	73
10. Testing and Evaluation	78
11. Deployment Guidelines	79
12. Conclusion	80
Summary	80
Challenges	80
Future	80
13. Review checklist	81
14. References	82

List of Figures

Fig 1. Architecture Diagram.....	13
Fig 2. Use Case Diagram.....	17
Fig 3. Class diagram.....	31
Fig 4.1 Login Sequence Diagram.....	32
Fig 4.2 Logout Sequence Diagram.....	33
Fig 4.3 Change Password Sequence Diagram.....	34
Fig 4.4 Forgot Password Sequence Diagram.....	35
Fig 4.5 Job Promotion/Demotion Sequence Diagram.....	36
Fig 4.6 HR Dashboard Sequence Diagram.....	37
Fig 4.7 Manage Departments Sequence Diagram.....	38
Fig 4.8 Parsing CV's Sequence Diagram.....	39
Fig 4.9 Hierarchy Tree Sequence Diagram.....	40
Fig 4.10 Job Posting Sequence Diagram.....	41
Fig 4.11 Adding Announcements Sequence Diagram.....	42
Fig 4.12 Employee Data Analytics Sequence Diagram.....	43
Fig 4.13 Employee Requests Sequence Diagram.....	44
Fig 4.14 Employee Reviews Sequence Diagram.....	45
Fig 4.15 Employee Reviews Reminder Sequence Diagram.....	46
Fig 4.16 Hiring Portal Sequence Diagram.....	47
Fig 4.17 Notifications Sequence Diagram.....	48
Fig 4.18 Staff Directory Sequence Diagram.....	49
Fig 4.19 Hiring Candidates Sequence Diagram.....	50
Fig 4.20 Employee Request Accept/Denial Sequence Diagram.....	51
Fig 5a. Gantt Chart(Part 1).....	55
Fig 5b. Gantt Chart(Part 2).....	55
Fig 6. Database Tables.....	56
Fig 7.1 Login Screen.....	58
Fig 7.2a HR Portal Screen.....	58
Fig 7.2b HR Portal Screen.....	59
Fig 7.3 Employee Requests Screen.....	59

Fig 7.4 Employee Directory Screen.....	60
Fig 7.5 Adding Employees Popup.....	60
Fig 7.6 Announcements Screen.....	61
Fig 7.7 Add Announcement Popup.....	61
Fig 7.8 View Applications Screen.....	62
Fig 7.9 Hiring Portal Screen.....	63
Fig 7.10 Adding Job Screen.....	64
Fig 7.11a Job Portal Screen.....	65
Fig 7.11b Job Portal Screen.....	65
Fig 7.12 Add Statistics Screen.....	66
Fig 7.13 Employee Portal Screen.....	66
Fig 7.14 Employee Analytics Screen.....	67
Fig 7.15 Employee Portal Screen.....	68
Fig 7.16 Change Password Screen.....	69
Fig 8a Test Case 1.....	78
Fig 8b Test Case 2.....	78

1. Introduction

a. Introduction

Our project streamlines the complex process of managing and recruiting employees in an organization. We provide a "one stop solution" for Small and medium-sized enterprises (SMEs) that caters to the HR needs of an organization.

We have made a web application. The target user for this application can be any company that wants a Human Resource Management System with a CV ranking component.

b. Objective and Scope

The Human resource management system covers many HR aspects from application to management to promotions. The software keeps track of an organization's employees and provides analytics of their performance using relevant KPIs. The software combines a number of systems and processes to ensure the ease of management in human resources and business processes. The HRMS software helps HR professionals manage the modern workforce.

Our aim is to assist companies in running effectively and efficiently. The system is a suite of software that companies can use to regulate their internal HR functions. Employee data management, recruitment, benefits, training, talent management, employee engagement, and employee attendance include some of the features our software will provide.

c. Development Methodology

We used agile methodology to complete this project. We started off with a prototype and then added on it in each sprint in a modular fashion. This approach allowed us to work independently on each module and then combine all the modules together. The project took a total of 4 sprints. In the last sprint, we integrated all the modules together to form an integrated system.

d. Contributions

We made a list of everything that needed to be done and then assigned work based on our team members' preferences. The backend was largely done by Aamina and Yousuf, while the frontend was mostly done by Javeria and Adnan.

Our project streamlines the complex process of managing and recruiting employees in an organization. We provide a "one stop solution" for Small and medium-sized enterprises (SMEs) that caters to the HR needs of an organization.

Employees can update personal data without involving HR for simple tasks using our Project HRMS. This allows HR experts to focus on more important tasks. Storage is centralized because all of the data is in one location, reporting can be more efficient. HR professionals can streamline the recruitment process by filtering resumes based on role and other variables. Companies can improve employee engagement and reduce turnover by installing an HRMS. It also allows employees to participate effectively and productively in the overall direction of the company as well as the achievement of the organization's goals and objective

2. System Requirements

This chapter is briefly going to talk about our system actors and their jobs. For each actor their respective functional requirements will be mentioned in detail. And finally it will talk about the non functional requirements of our system.

a. System Actors

Actor Name	Description
HR manager	Will have access to information and statistics of all employees.
System Administrator	The manager of the website
Dept head	Will have access to the information of employees in their own department and will be able to enter information regarding each employee
Employee	Will have access to their information/analytics dashboard

b. Functional Requirements

Requirements	
Sr#	Requirement
HR Manager	
1	I want to be able to view monthly/quarterly employee performance statistics (completion of task on time, task quality, behavior rating)
2	I would like to send my emails straight through the system
3	I want to add new employees to the database
4	I want the system to be able to filter out potential candidates for a job and rank their resumes for me.
5	I want the system to use the CV scanner to parse an employee's resume.
6	I want to be able to transfer/promote/terminate employee roles from the system
7	I want to see company statistics such as the number of active employees, latest news , new hires, open job vacancies and more.
8	I want to have access to a timeline to see when changes to compensation occurred during an employees lifetime

9	I want to be able to search employees through filtering (by department, skills, education etc)
10	I want to monitor employees' working hours and absences, vacation and sick leave accruals and job contract terminations.
11	I want to have access to the company organization hierarchy
12	I would like to be able to add news and announcements (e.g., about new job postings)
	Department Head
1	I want to be able to search the content of database for the employees who are under my coverage
2	I want to be able to view monthly/quarterly employee performance statistics (completion of task on time, task quality, behavior rating) - Employee KPI tracking (e.g., open/completed tasks, overtimes).
3	I would like to send my emails straight through the system
4	I would like to approve or reject employee requests
5	I want to have access to a timeline to see when changes to compensation occurred during an employees lifetime
6	I want to be able to add comments for each employee to describe their achievements and flaws
7	I want to be able to search employees through filtering (by department, skills, education etc)
8	I want individual employee performance reviews to be scheduled automatically and for employees to be notified in advance about them.
9	I want to monitor employees' working hours and absences, vacation and sick leave accruals and job contract terminations.
	Employee
1	I want to be able to put in requests for sick days, personal days, vacation days, and volunteer hours
2	I want to be notified about the approval or rejection of my request
3	I want to view my own performance statistics through my dashboard
4	I want to have access to visualized reports of all my performance reviews
5	I want to have access to easy answers to employees' common HR-related questions.
	System Administrator
1	I want to have administrator level access to the system and its database.
	All Actors
1	I want to login & from the system with my username and password.

2	I want to see the appropriate user interfaces after logging in (according to my role)
3	I want to have access to automated notifications (eg: contract expiration)
4	I would like to be able to read recent news and announcements (e.g., about new HR policies)

c. Non-functional Requirements

Sr#	Requirements
1	The system shall not fail more than 3 times every 24 hours. In case of a failure, the system should restore to normal operations within 5 minutes of a failure.
2	Any interaction between the user and the system must have a maximum response time of 4 seconds. In cases where more time is required by the system, the system must display the progress
3	Personal data of employees shall only be available to the HR manager and System Manager.
4	Sensitive information of employees such as passwords and personal information shall be encrypted and must not be available to anyone except the system administrator.
5	The number of the simultaneous users of the system can accommodate shall be 50
6	Only 5000 employees can be added to the database.
7	The system shall be reliable i.e information entered should be stored successfully.
8	System shall distinguish authorization levels based on the actor. Authorization levels will be hierarchical (e.g Dept heads will be able to access information of their subordinates and the subordinates will be able to access data of the employees working under them respectively.)
9	The software shall be used on PCs and be functional via the internet using all major web browsers i.e Safari, Chrome and Firefox.
10	The website shall display errors/notifications in cases of failures For unexpected failures the users should be notified and they should be able to go back to the previous page.
11	At least 20% of the processor and RAM capacity shall be unused at peak load periods.

3. System Architecture

This chapter will define the structure and behavior of our system. The diagram gives an overview of our system and the main components of our system are also explained with an in-detail justification of the architecture

a. Architecture Diagram

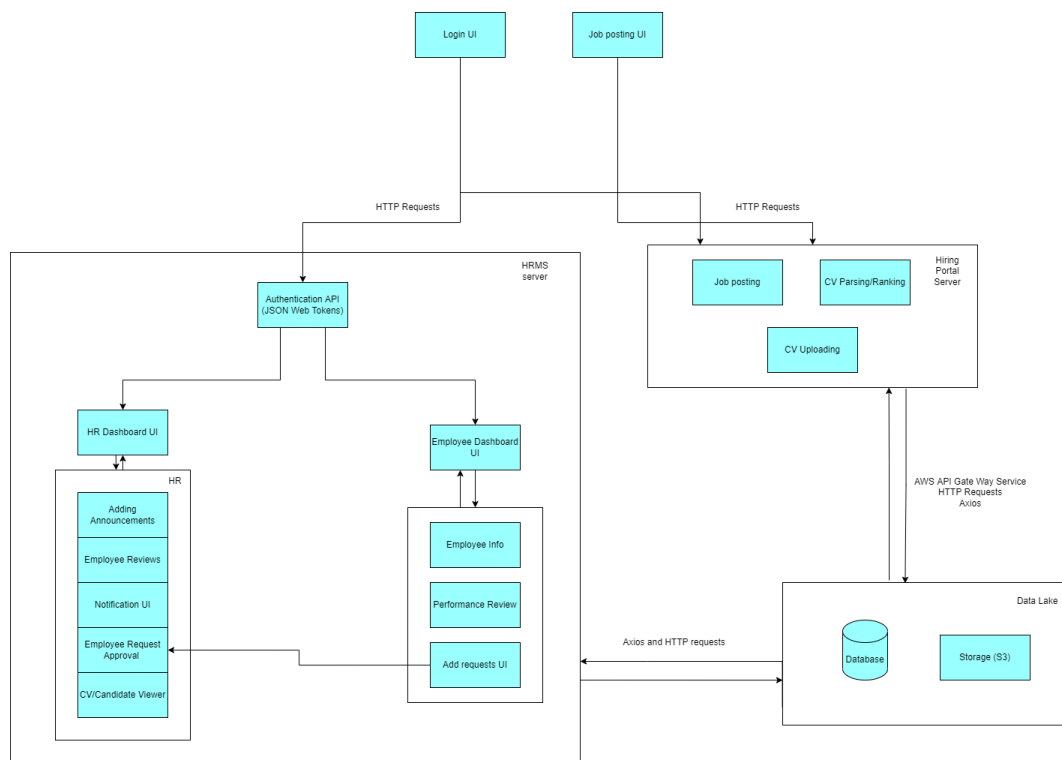


Fig 1. Architecture Diagram

b. Architecture Description

HRMS Server and Hiring Portal Server

This subsystem processes the user's requests and sends all of the required documents (based on the request) back to their browser. The required documents will consist of files such as JSON and XML. The way this subsystem works is that it calls the back-end infrastructures such as the database, etc. The server will be hosted on Amazon Web Services (AWS).

Database

The database subsystem contains information to be used for performing computations. The web application server can interact with the database to complete the information updates of the whole system such as adding, deleting, searching and organizing user data.

Data Lake

A data lake is a centralized repository that allows you to store all your structured and unstructured data. Since our system consists of data from two sources: employees and candidates, the data lake will act as a central repository for both of them. The application interacts with the data lake to provide analytics dashboards and for searching/querying data.

Login (JSON Web Tokens)

The JSON web tokens provide secure user authentication for the web application that allows a user to login according to their role.

Cloud Storage(S3)

S3 is a storage container for storing the CVs and resumes of applicants. When a job applicant uploads their resume/CV it gets stored in the S3 container.

c. Justification of the Architecture

We have chosen a hybrid between client server architecture and layered architecture. This architecture is specifically useful for our application since we require a separation of concerns between the client and the server along with the layered architecture separating the system functionality into layers. Furthermore, this also allows the candidate applying for a job to add their CV/resume without having to login to the internal HR portal of the company. This architecture is best suited for our application as it can cater to a large number of users without being overwhelmed and store and process data according to the requirements of the system. This also allows our system to scale both horizontally and vertically while ensuring the modularity of the system. Our architecture also ensures the security of sensitive user data like passwords by storing them after using bcrypt to hash the data. Json web tokens are being used for session management and authorization in our system.

- Moreover, the architecture uses a combination of server and serverless protocols. This strategy optimizes performance and cost of operating the system while ensuring that there is no single point of failure. The use of serverless protocols for handling data intensive workloads (like AI chatbot and search queries) ensures efficient utilization of resources while providing fast and reliable performance. We had decided to use a DynamoDB as our database as it provides a serverless, key-value NoSQL database for high-performance applications at scale. DynamoDB provides an efficient and scalable solution for building ad-hoc and on-demand dashboards required by our system.

Lastly, we went with having an architecture with multiple servers and only one database. This model doesn't store any data. When the client sends information to the web server, it is processed and written to the database, but managing this data takes place outside of the server. It's called stateless architecture. It's important to make our model reliable because if one server goes down, another one will take charge. So, in such a failure, all the requests will automatically go to the new server, without affecting the systems functioning. Thus, this

model is more reliable than a single server. However, if something happens to the database, the app will crash.

The hybrid approach we have used also makes the system architecture more complex. This added complexity makes the system and its various functionalities difficult for a new developer to understand without interacting with the system.

d. Tools and Technologies

- Backend: Django 3.8.2 or 4.0
- Frontend: React 17.0
- Databases: Amazon DynamoDB
- Servers/Serverless: AWS EC2

4. Requirements Specifications

In this we will be discussing the major use cases of our system and then explaining them in detail in section 4.2.

4.1. Use Case Diagram

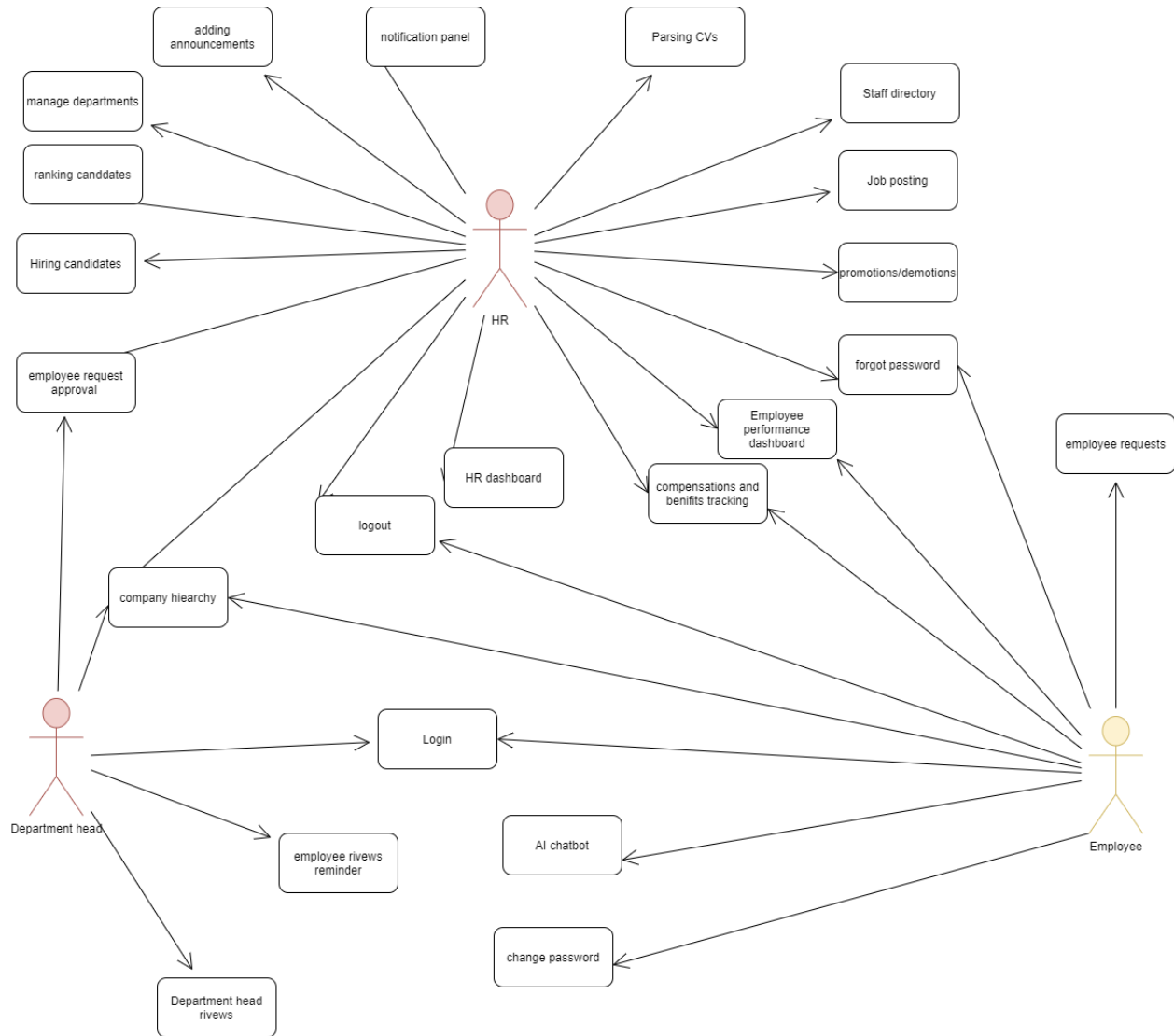


Fig 2. Use Case Diagram

4.2 Use Cases

4.2.1 Login

Identifier	UC-001
Purpose	A registered user login to system to access the functionality of system
Pre-conditions	A user must have an account and the system must be connected to the network.
Post-conditions	The user must be authorized to use the system.
Step #	Typical Course of Action
1.	Users enter their username and password
2.	The username and password are validated by the system.
3.	User is allowed access to system
Step #	Alternate Courses of Action
1.	In step 1, if the username or password is invalid, the system shows an error message
Step #	Exception Paths
	None

4.2.2 Logout

Identifier	UC-002
Purpose	User log off from the system
Pre-conditions	User must be logged in
Post-conditions	The user is logged out of the system
Step #	Typical Course of Action
1.	User clicks on the logout button and clicks confirm
Step #	Alternate Courses of Action
1.	None.
Step #	Exception Paths
1.	None.

4.2.3 Forgot Password

Identifier	UC-003
Purpose	A registered user trying to get a new password.
Pre-conditions	A user must have an account.
Post-conditions	Password will be changed
Step #	Typical Course of Action
1.	Enter Username
2.	Enter password
3.	Validate Username through email.
4.	Reset password
5.	System will let the user know that their password has been changed.
Step #	Alternate Courses of Action
1.	None
Step #	Exception Paths
1.	If an invalid username is entered, the system shows an error message

4.2.4 Job Promotions/Demotions

Identifier	UC-004
Purpose	The HR manager wants to update the position/rank of an employee
Pre-conditions	The HR manager manager must be logged in
Post-conditions	The employee is promoted or demoted.
Step #	Typical Course of Action
1.	HR manager clicks manage employees on the HR portal
2.	A list of all employees will be presented to the HR manager
3.	HR manager will select the employee
4.	The profile of the employee will open up and the HR manager will be able to change/edit the information of the employee.

Step #	Alternate Courses of Action
1.	HR manager clicks manage employees on the HR dashboard
2.	A list of all employees will be presented to the HR manager
3.	HR manager will search for the name of the employee
4.	HR manager will select the employee
5.	The profile of the employee will open up and the HR manager will be able to change/edit the information of the employee.
Step #	Exception Paths
	None

4.2.5 HR Dashboard

Identifier	UC-005
Purpose	To provide an overview of the company and serve and give access to HR's functionality
Pre-conditions	The HR manager manager must have logged in
Post-conditions	The overview is provided
Step #	Typical Course of Action
1.	The HR staff goes to the website home page
2.	The homepage requests their id and password
3.	The user enters the information
4.	The system loads the users home page
5.	The user goes to their profile
6.	The user selects the HR dashboard option
7.	If the user has access, the Dashboard containing company information and HR functionality will show up
8.	The use case ends.
Step #	Alternate Courses of Action
1.	None
Step #	Exception Paths
1.	In step 7, if the user does not have access, then an error message is displayed

4.2.6 Hiring Candidates

Identifier	UC-006
Purpose	Hiring candidates as employees and adding them to the system
Pre-conditions	The HR Manager must be logged into the hiring portal.
Post-conditions	The candidate is hired and added as an employee.
Step #	Typical Course of Action
1.	The HR manager retrieves the CV and data of the candidate by searching for it in the job postings list.
2.	The HR manager chooses the desired applicant by clicking Hire.
3.	The candidate is added to the employee database
4.	The use case ends.
Step #	Alternate Courses of Action
1.	None
Step #	Exception Paths
1.	None

4.2.7 Manage Departments

Identifier	UC-007
Purpose	To add or remove departments or employees from the database
Pre-conditions	The user must be logged in from their HR account.
Post-conditions	Departments or employees will be added/removed
Step #	Typical Course of Action
1.	The user goes to the manage departments option
2.	The user clicks add or remove department
3.	The system displays a drop down menu where they can either add a new department or add an employee to a specific department
4.	The user clicks on the department that they want to add an employee to

5.	The user enters the information of the employee
6.	The user clicks add
7.	Dept Head receives a notification after the database has been updated.
8.	The use case ends
Step #	Alternate Courses of Action
1.	After step 2, the user can choose to add a new department.
2.	The user clicks the 'add button'
3.	A new department is added to the system
Step #	Exception Paths
1.	None

4.2.8 Change Password

Identifier	UC-008
Purpose	The user goes to their profile to change their password
Pre-conditions	The user has logged into their account successfully
Post-conditions	The password is changed in the system
Step #	Typical Course of Action
1.	The user goes to their profile
2.	The user selects the change password option
3.	The user confirms their new password
4.	If the user has successfully entered the new password twice, their new password is entered into the system and can be used to log-in in the future.
5.	The use case ends.
Step #	Alternate Courses of Action
	None

Step #	Exception Paths
1.	In step 7, if the passwords don't match, then an error message is displayed and the user is asked to re-enter the passwords.

4.2.9 Parsing CVs

Identifier	UC-009
Purpose	Parsing CV for shortlisting of suitable candidates
Pre-conditions	The CV's of candidates are in proper format
Post-conditions	The parsed CV's are then filtered
Step #	Typical Course of Action
1.	The candidate uploads the cv in the job posting
2.	The system will extract info from the CV
3.	They are then parsed according to the relevant job descriptions.
4.	The use case ends.
Step #	Alternate Courses of Action
1.	None
Step #	Exception Paths
1.	None

4.2.10 Staff directory

Identifier	UC-011
Purpose	For the HR manager to search for employees through filtering (by name, department , skills , education and project names they have done)
Pre-conditions	The HR manager has logged in.
Post-conditions	Employee profile can be selected

Step #	Typical Course of Action
1.	The user will navigate to the search option in their dashboard.
2.	The user can search by name & select any of the filters
3.	The user will then select the employee whose profile they want to visit
4.	Their profile is displayed
5.	The use case ends
Step #	Alternate Courses of Action
1.	In step 2, the user can choose to not select any filters (and not enter a name). Instead they may scroll down the employee list (arranged alphabetically)
Step #	Exception Paths
1.	In step 3, if the user does not have access to the employee's profile, an error message will be displayed

4.2.11 Employee requests

Identifier	UC-012
Purpose	Employees can apply for sick days, personal days, vacation days, and volunteer hours
Pre-conditions	The employee must be logged in and should be on the dashboard screen
Post-conditions	A request is sent to the manager/head of the team for approval
Step #	Typical Course of Action
1.	The user goes to the website home page
2.	The homepage requests their id and password
3.	The user enters the information
4.	The system loads the users home page
5.	The user clicks on the 'employee requests' option
6.	The user clicks on the apply for leave option
7.	A calendar pops up and the employee selects a start date
8.	The employee then selects an end date
9.	The employee adds their reasoning in a text box

10.	The employee clicks send and the application is sent to the department head
11.	The use case ends
Step #	Alternate Courses of Action
1.	In step 6, the employee will have a variety of other options (sick days, personal days, vacation days, and volunteer hours)
Step #	Exception Paths
1.	If the starting/ending date is not selected or a date from the past is selected the application will not be sent and an error message will be shown

4.2.12 Employee reviews

Identifier	UC-013
Purpose	For employee reviews to be added to their record .
Pre-conditions	The Immediate Manager must be logged in with their account and must be authorized to give reviews.
Post-conditions	Reviews for employees will be visible on the HR dashboard.
Step #	Typical Course of Action
1.	The user selects an employee from their department
2.	The user clicks on the ‘employee review’ option
3.	The user is taken to a new screen with the previous reviews.
4.	The user can wish to edit old reviews or add new ones
5.	The use case ends.
Step #	Alternate Courses of Action
1.	None
Step #	Exception Paths
1.	In step 3, if the user does not have the authority to edit the employee’s review, an error message will be shown

4.2.13 Employee performance analytics dashboard

Identifier	UC-014
Purpose	For users to see the performance analytics of themselves or others.
Pre-conditions	The user should be logged in and have access to the performance analytics of the individual whose dashboard they wish to see
Post-conditions	The dashboard is displayed with all relevant statistics
Step #	Typical Course of Action
1.	The user clicks on the performance analytics option
2.	The user is asked to enter the ID of the employee whose performance analytics they wish to see
3.	The system determines if the user has access to the analytics of the particular individual
4.	If the user has access, the dashboard is displayed.
5.	The use case ends.
Step #	Alternate Courses of Action
1.	In step 6, the customer can cancel the transaction and go directly to step 9.
Step #	Exception Paths
1.	In step 7, if the user does not have access, then an error message is displayed and execution proceeds to step 9.

4.2.14 Adding announcements

Identifier	UC-015
Purpose	For HR to add announcements to the notification panel of employees / department heads.
Pre-conditions	The user should be logged in from their HR account
Post-conditions	The announcement is posted
Step #	Typical Course of Action
1.	The user goes to the announcements option
2.	The user clicks on add announcement
3.	The user enters the message to be posted

4.	The use case ends.
Step #	Alternate Courses of Action
1.	None
Step #	Exception Paths
1.	None

4.2.15 Notification panel

Identifier	UC-016
Purpose	For users to be able to read recent news and announcements (e.g., about new HR policies or request approval / rejection)
Pre-conditions	User must be logged in
Post-conditions	Recent news and announcements are displayed
Step #	Typical Course of Action
1.	User must click on the notification panel on the top right
2.	The recent announcements are displayed
3.	The user clicks on an announcement of their choice
4.	The announcement is fully shown on a separate screen
5.	The use case ends.
Step #	Alternate Courses of Action
1.	After step 2, the user can decide to move straight to step 5..
Step #	Exception Paths
1.	After step 1, if there are no new announcements, then the user moves to step 5

4.2.16 Employee request approval/denial

Identifier	UC-017
Purpose	For manager to approve or deny employee requests regarding sick days, personal days, vacation days, and volunteer hours
Pre-conditions	User must be logged with their department head account

Post-conditions		Request will be approved or denied
Step #	Typical Course of Action	
1.	Users must click on the notification panel.	
2.	The recent announcements are displayed.	
3.	The user clicks on an announcement regarding employee request	
4.	The user clicks either accept or reject	
5.	The result is forwarded to the employee's notification panel	
6.	The use case ends	
Step #	Alternate Courses of Action	
1.	After step 4, the user can decide to ignore the request.	
Step #	Exception Paths	
1.	After step 1, if there are no new announcements, then the user moves to step 5	

4.2.17 Employee reviews reminder

Identifier		UC-018
Purpose		To act as a reminder for managers/department heads to add a performance review for their employee
Pre-conditions		User must be logged in with their department head account
Post-conditions		Employee Review section will be updates
Step #	Typical Course of Action	
1.	Users must click on the notification panel.	
2.	The recent announcements are displayed.	
3.	The user clicks on an announcement regarding employee performance review reminder	
4.	The user is taken to the employee profile where they can update the performance of the employee using KPIs.	
5.	The use case ends	
Step #	Alternate Courses of Action	
1.	After step 2, the user can decide to ignore/delete the notification.	
Step #	Exception Paths	
1.	After step 1, if there are no new announcements, then the user moves to step 5	

4.2.18 Hierarchy Tree

Identifier	UC-019
Purpose	To display information of the organization's hierarchy to the HR manager
Pre-conditions	The HR manager needs to be logged in
Post-conditions	The tree is displayed
Step #	Typical Course of Action
1.	The HR manager clicks on the company hierarchy button
2.	The organization's hierarchy is displayed.
3.	The use case ends.
Step #	Alternate Courses of Action
1.	None
Step #	Exception Paths
1.	None

4.2.19 Hiring portal/ Collecting CVs

Identifier	UC-020
Purpose	To collect CVs and other information relevant to available jobs
Pre-conditions	User must be logged with their HR manager account
Post-conditions	User will have collected the CVs and other information
Step #	Typical Course of Action
1.	User must click on the hiring portal panel
2.	The uploaded cvs are displayed
3.	The use case ends
Step #	Alternate Courses of Action
1.	None
Step #	Exception Paths
1.	None

4.2.20 Job posting

Identifier		UC-021
Purpose		To post available jobs on HR portal
Pre-conditions		User must be logged with their HR manager account
Post-conditions		Available jobs will be posted on HR portal
Step #	Typical Course of Action	
1.	User must click on the hiring portal panel	
2.	User must click on 'add job'	
3.	User must fill the information regarding requirements of job	
4.	User must click confirm	
5.	The use case ends	
Step #	Alternate Courses of Action	
1.	None	
Step #	Exception Paths	
1.	None	

4.3 Class Diagram

The User class defines the user along with the relevant details,role and password. HR Employee and Department Head are all users. It also allows the user to change the password.

The HR class defines the roles of the HR such as posting jobs, recruiting employees and notifications etc. Only the HR is allowed to view and use these functions and views.

The employee class allows the employee to add requests and view analytics and personal details.The HR can also login as an employee to view their details.

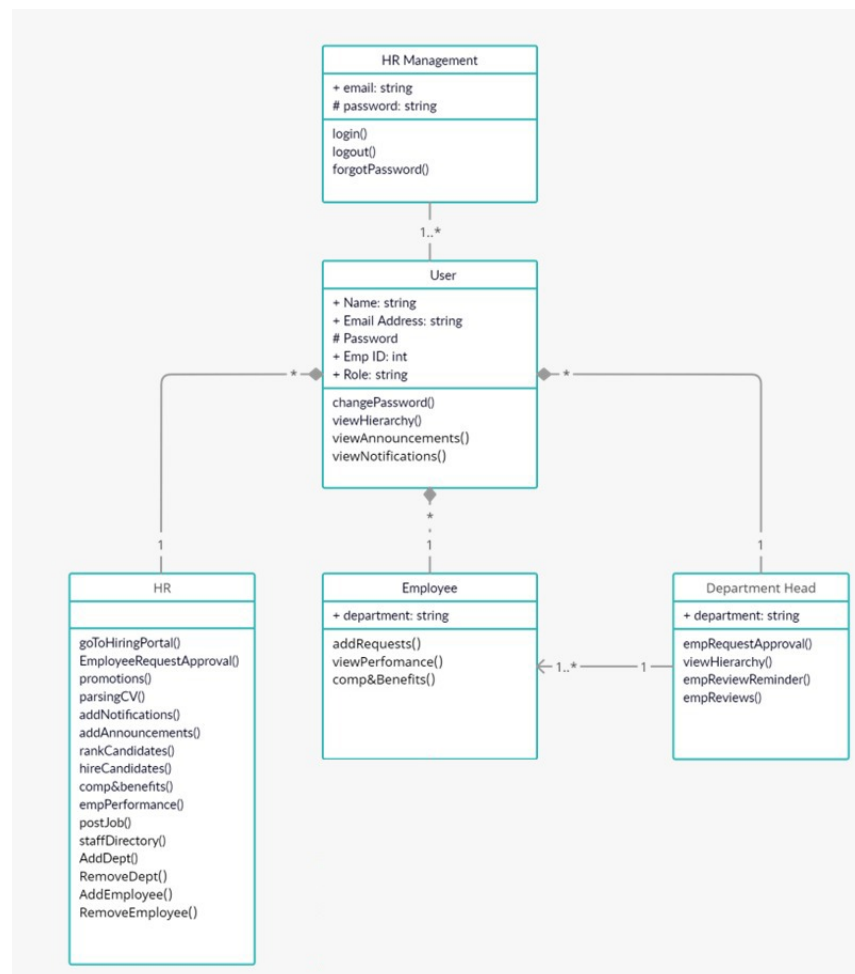


Fig 3. Class diagram

4.4 Sequence Diagrams

4.4.1 Login

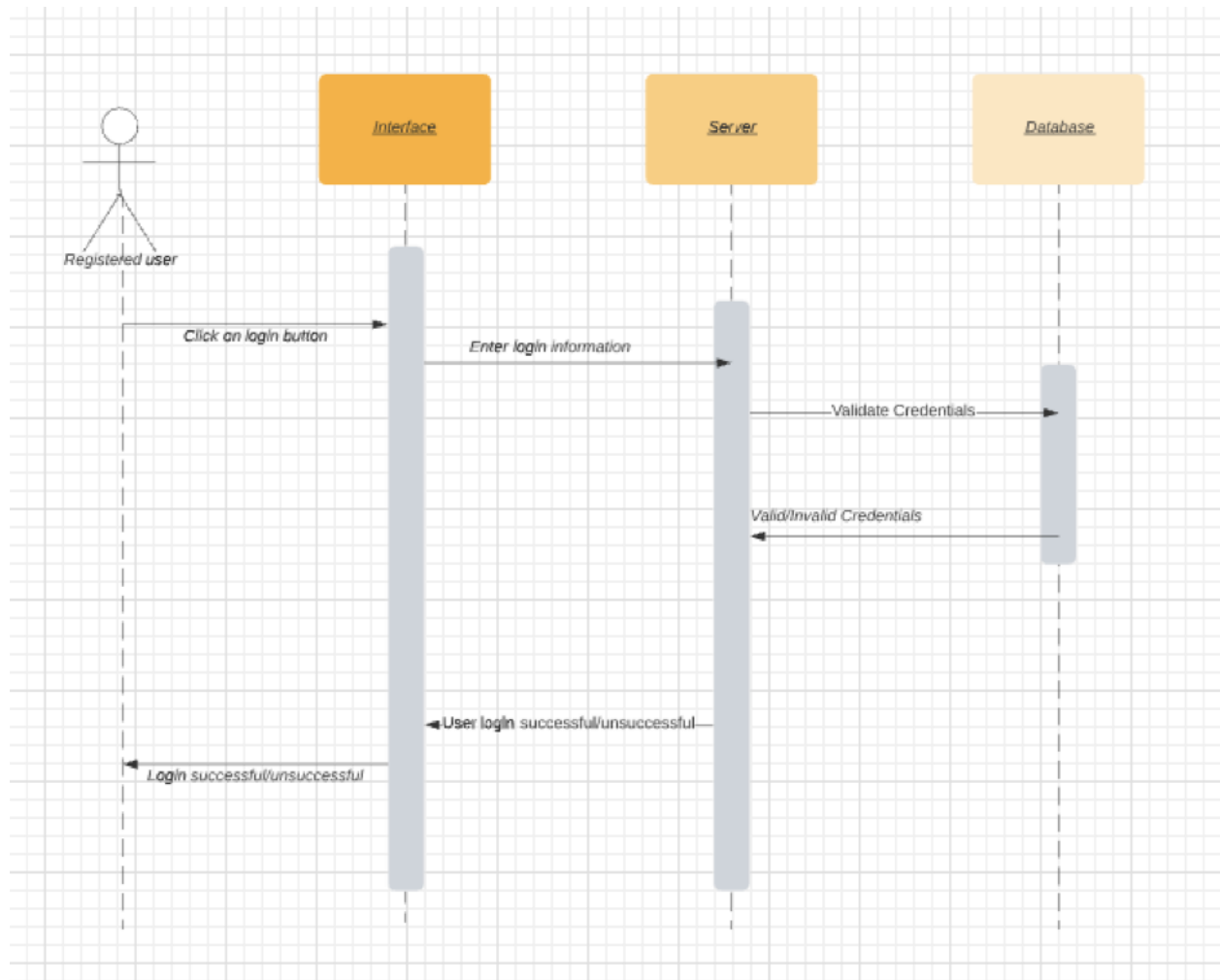


Fig 4.1 Login Sequence Diagram

4.4.2 Logout

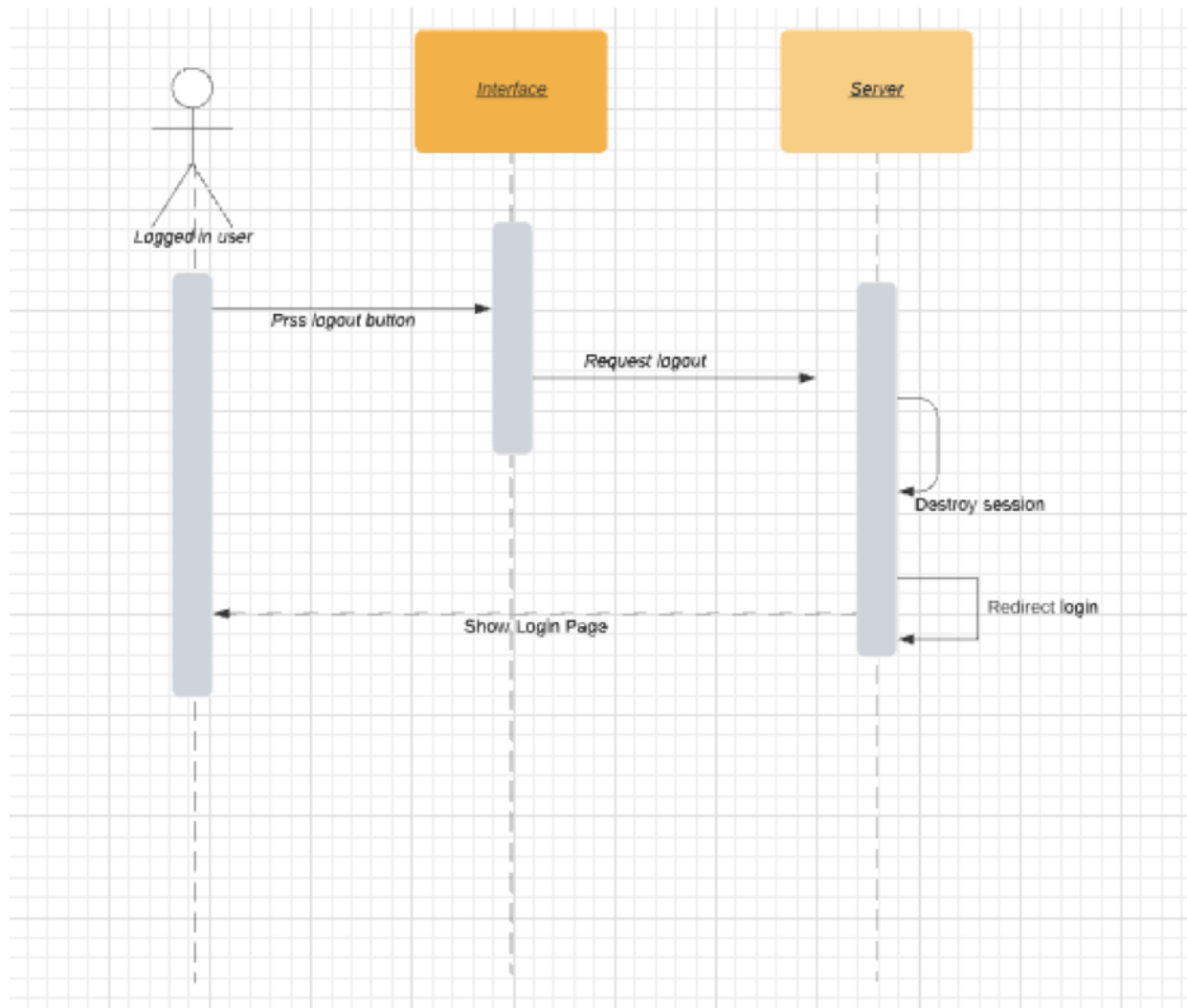


Fig 4.2 Logout Sequence Diagram

4.4.3 Change Password

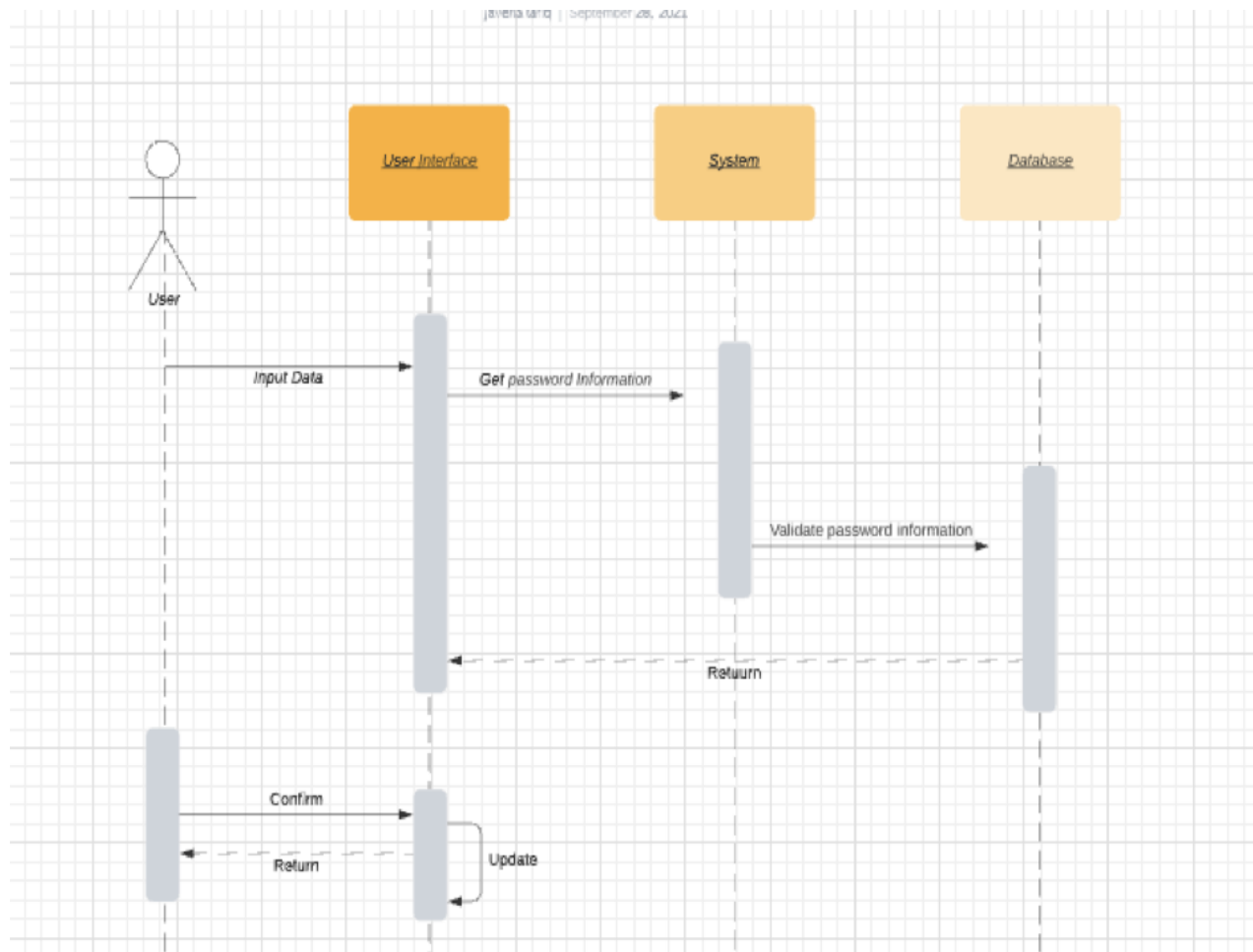


Fig 4.3 Change Password Sequence Diagram

4.4.4 Forgot Password

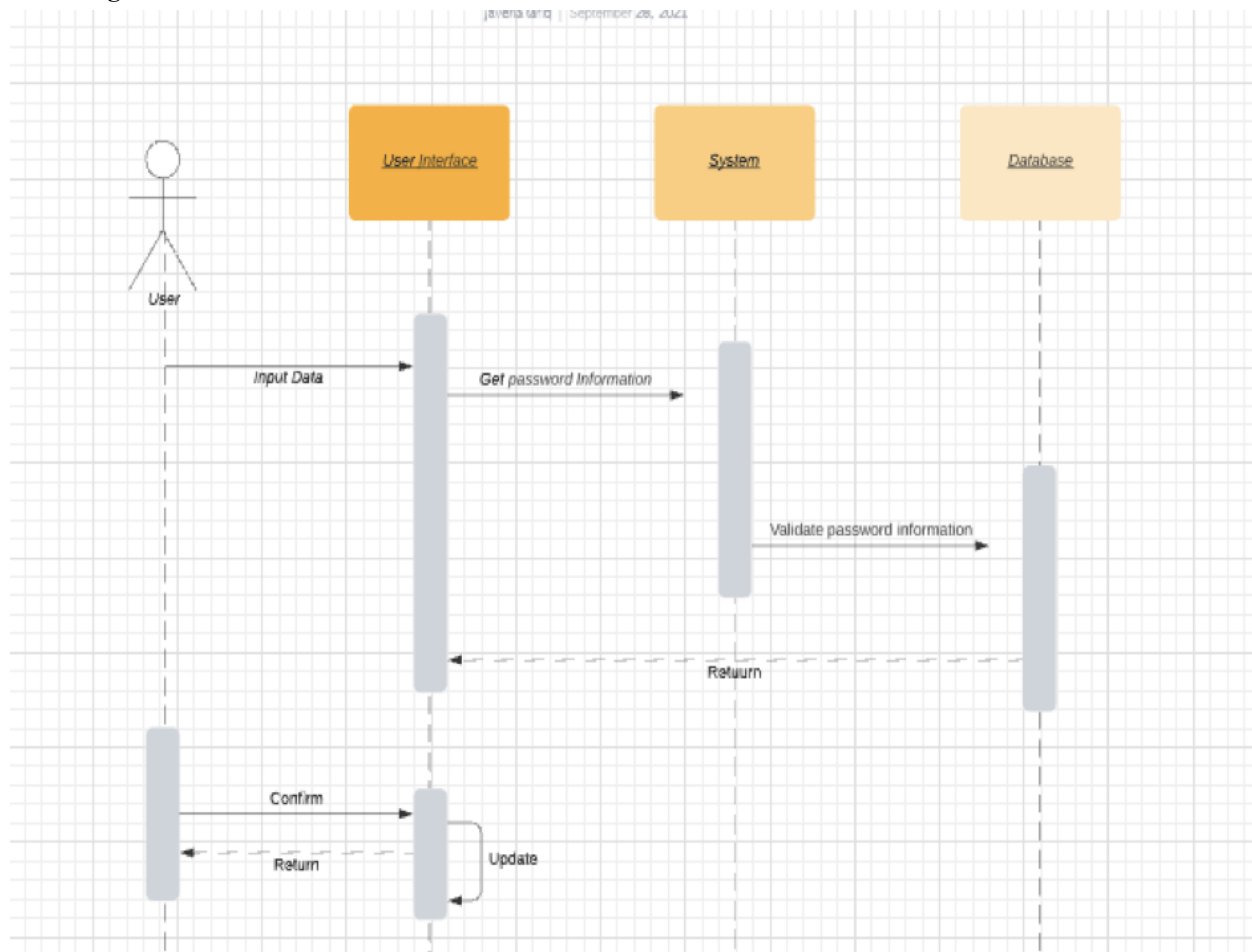


Fig 4.4 Forgot Password Sequence Diagram

4.4.5 Job promotions/demotions

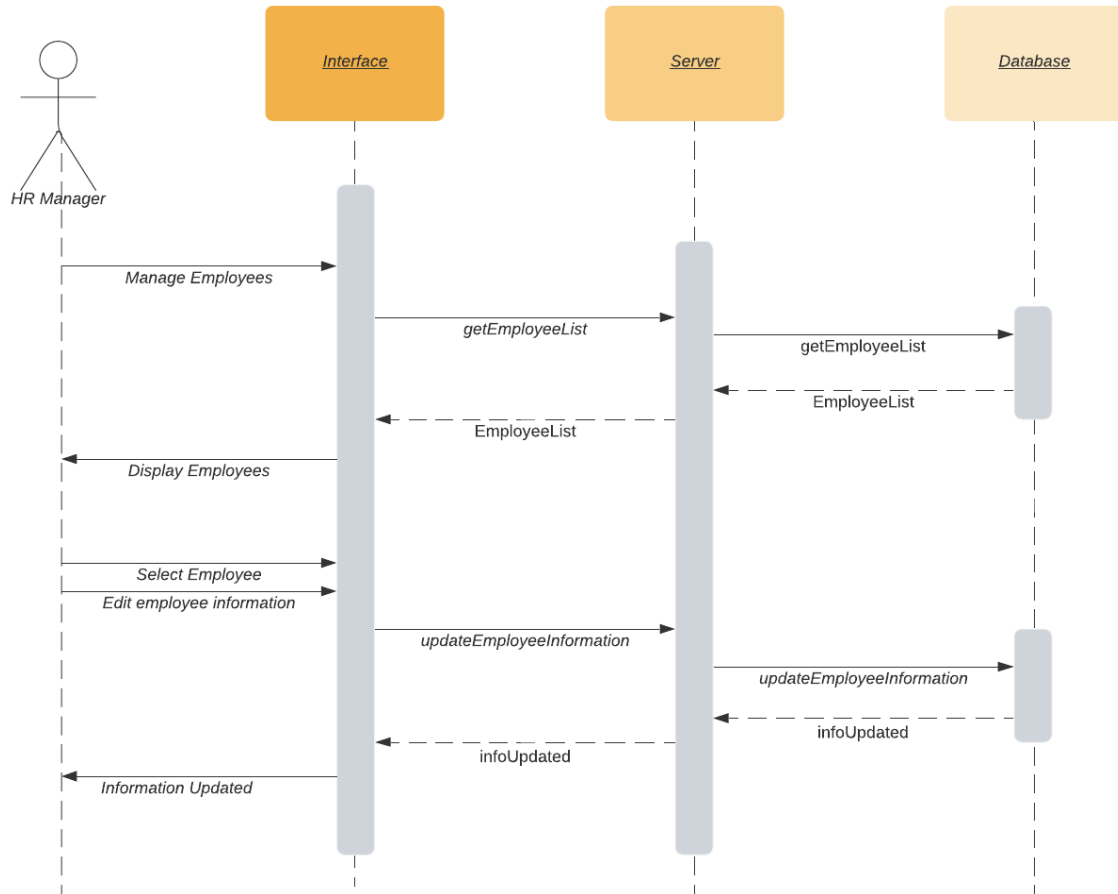


Fig 4.5 Job Promotion/Demotion Sequence Diagram

4.4.6 HR Dashboard

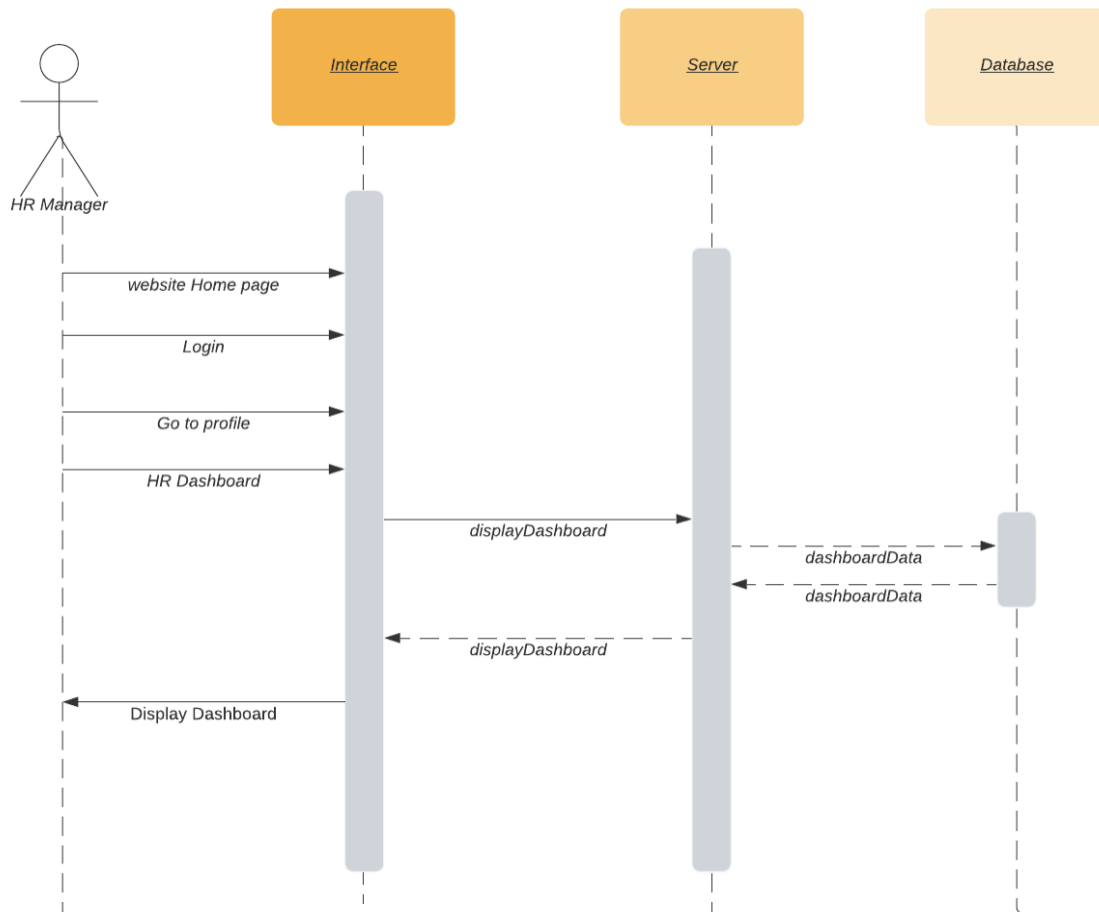


Fig 4.6 HR Dashboard Sequence Diagram

4.4.7 Manage Departments

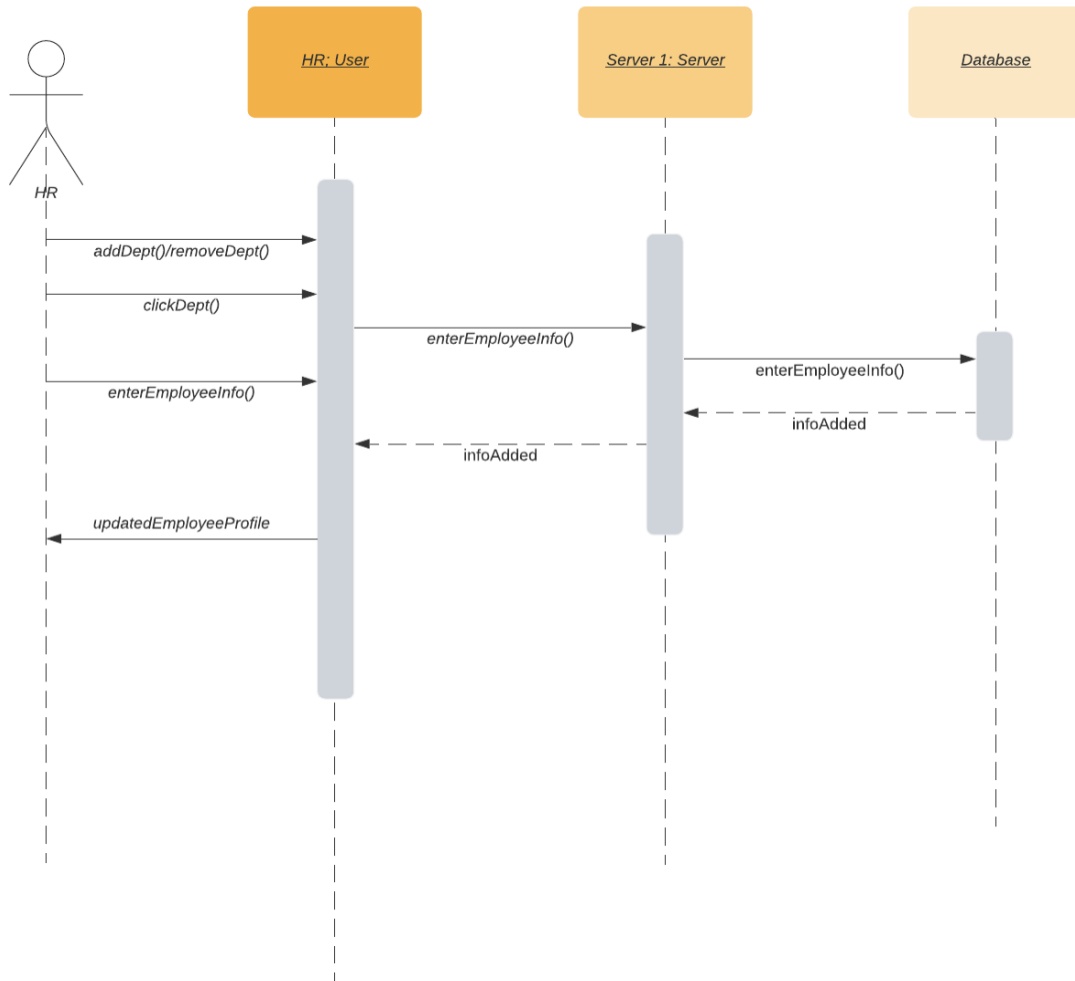


Fig 4.7 Manage Departments Sequence Diagram

4.4.8 Parsing CV's

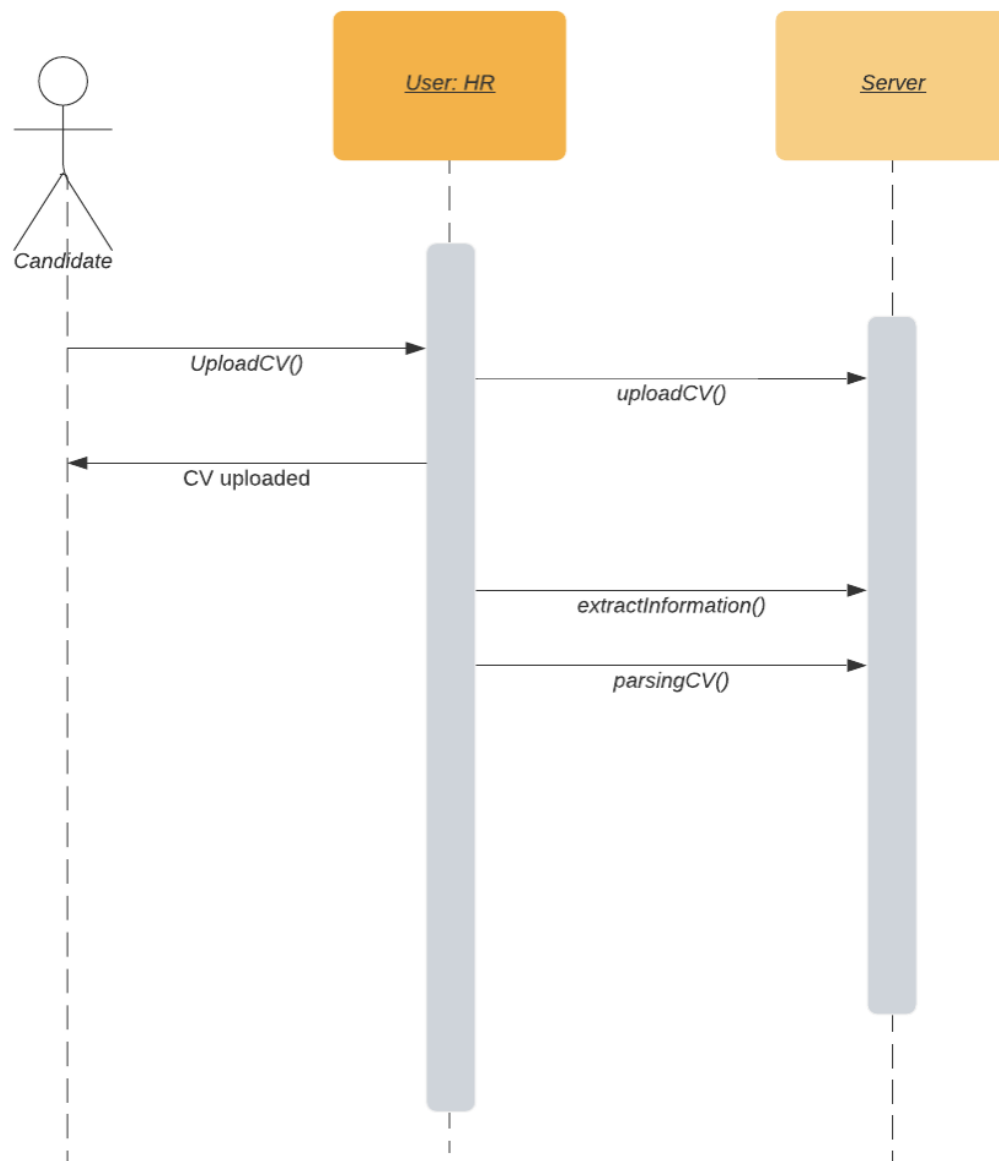


Fig 4.8 Parsing CV's Sequence Diagram

4.4.9 Hierarchy Tree

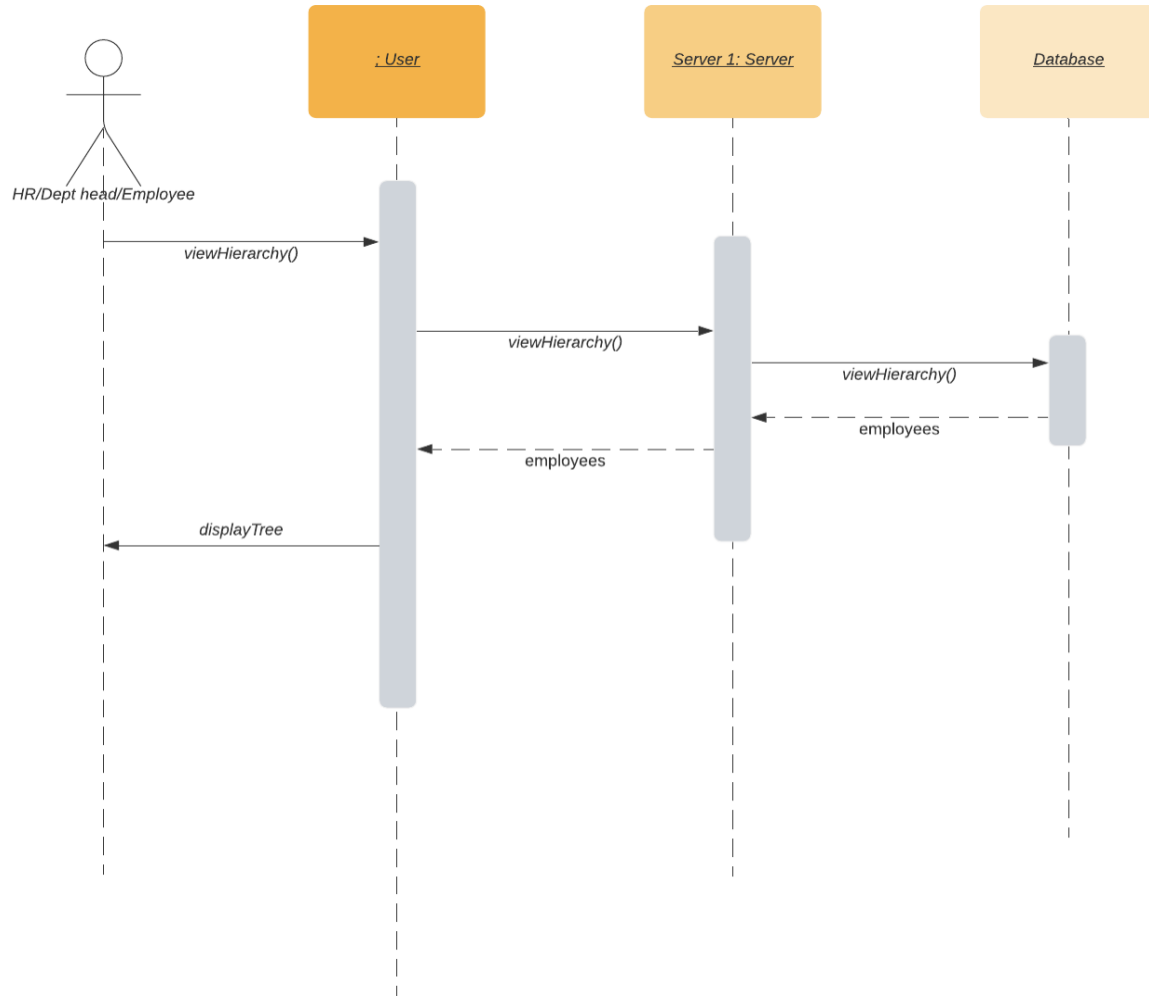


Fig 4.9 Hierarchy Tree Sequence Diagram

4.4.10 Job Posting

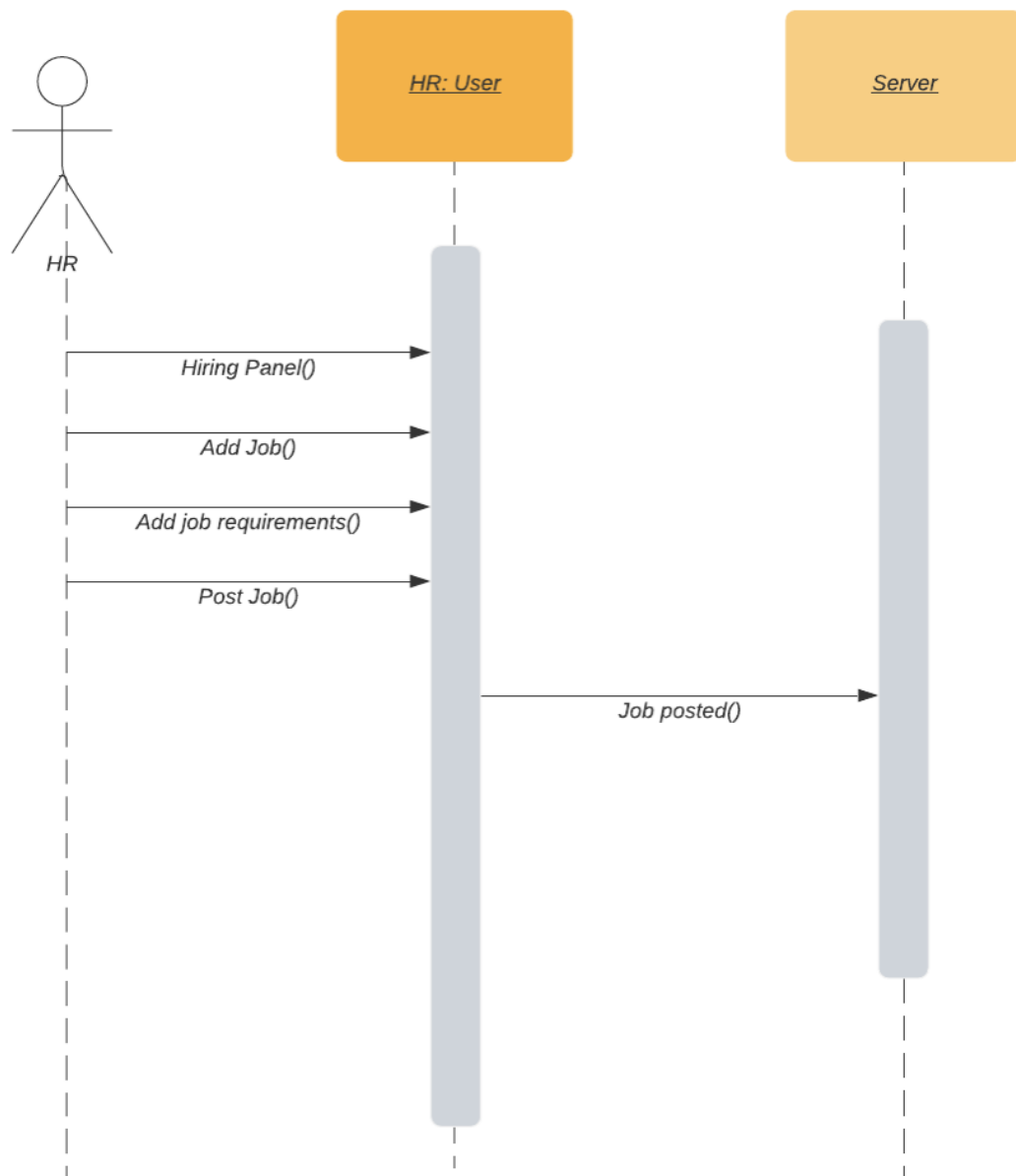


Fig 4.10 Job Posting Sequence Diagram

4.4.11 Adding Announcements:

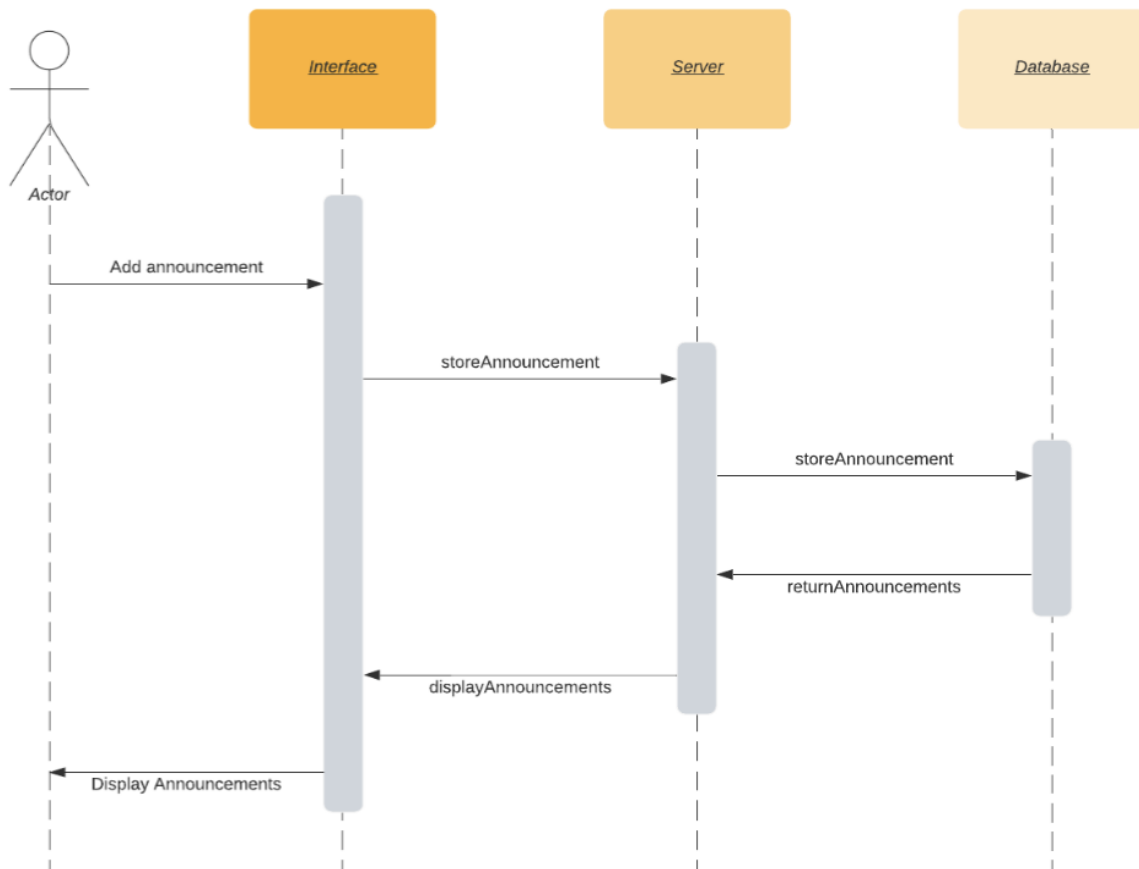


Fig 4.11 Adding Announcements Sequence Diagram

4.4.12 Employee Data Analytics:

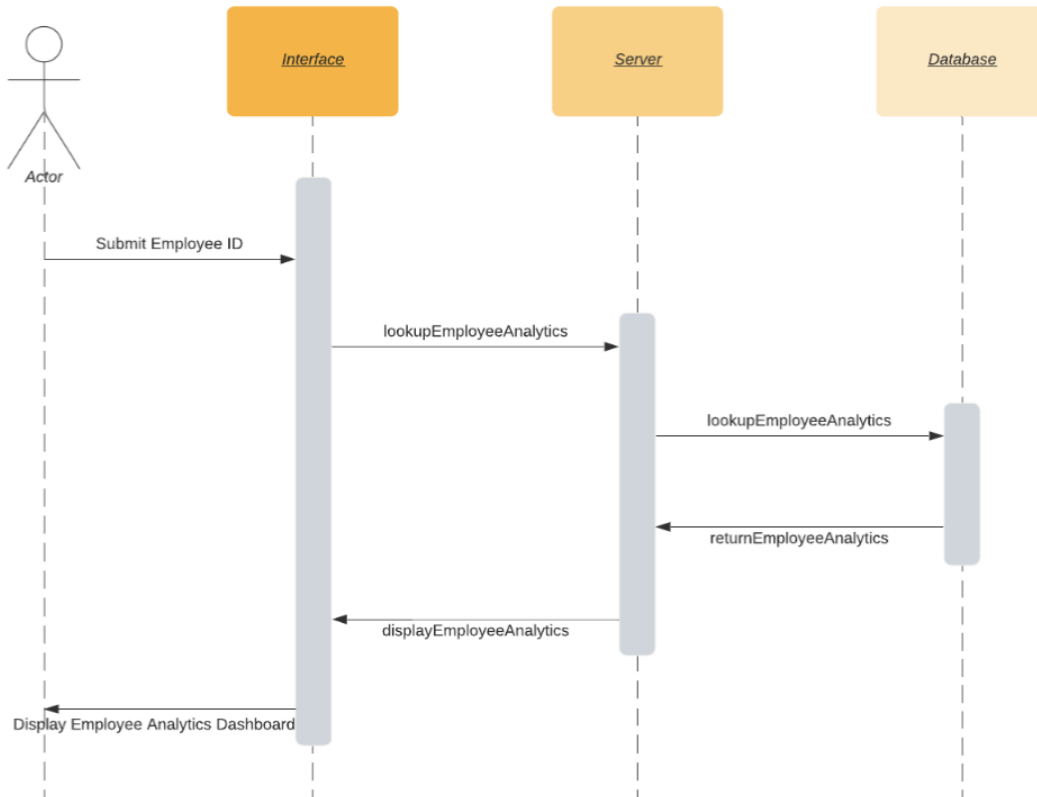


Fig 4.12 Employee Data Analytics Sequence Diagram

4.4.13 Employee Requests:

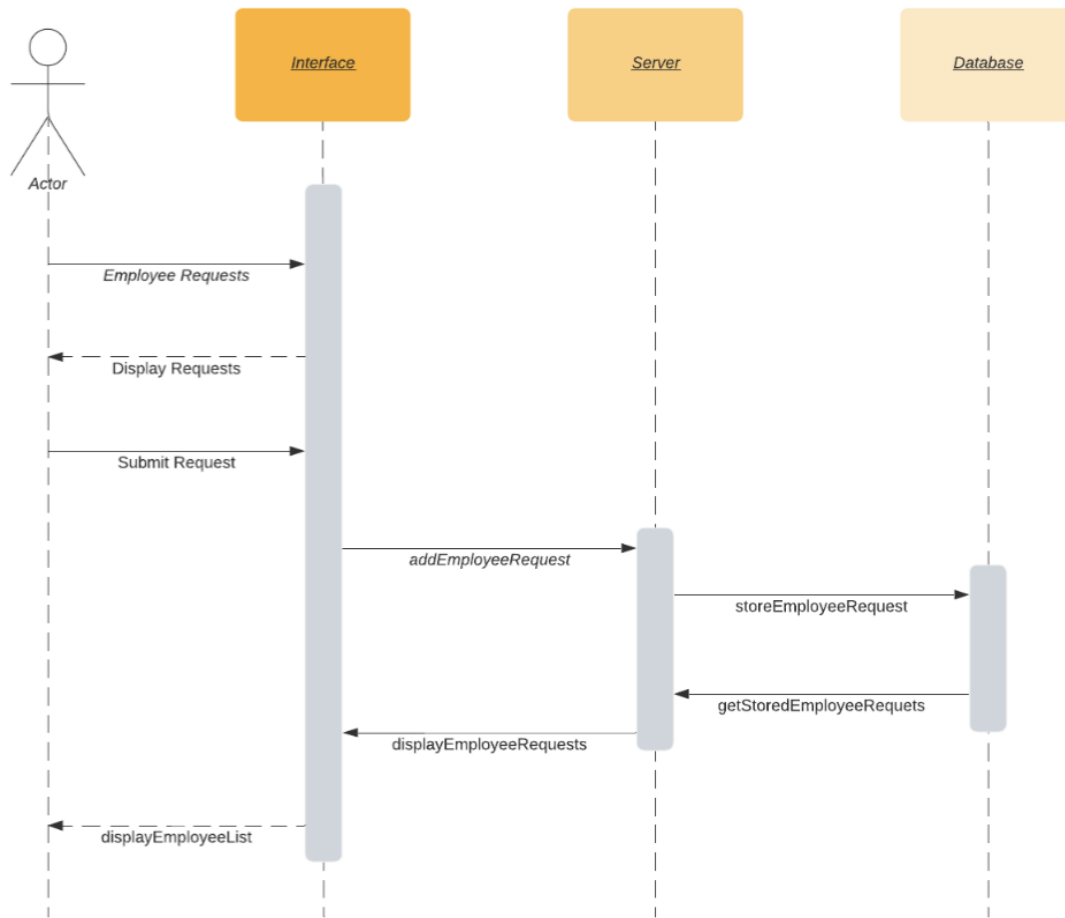


Fig 4.13 Employee Requests Sequence Diagram

4.4.14 Employee Review:

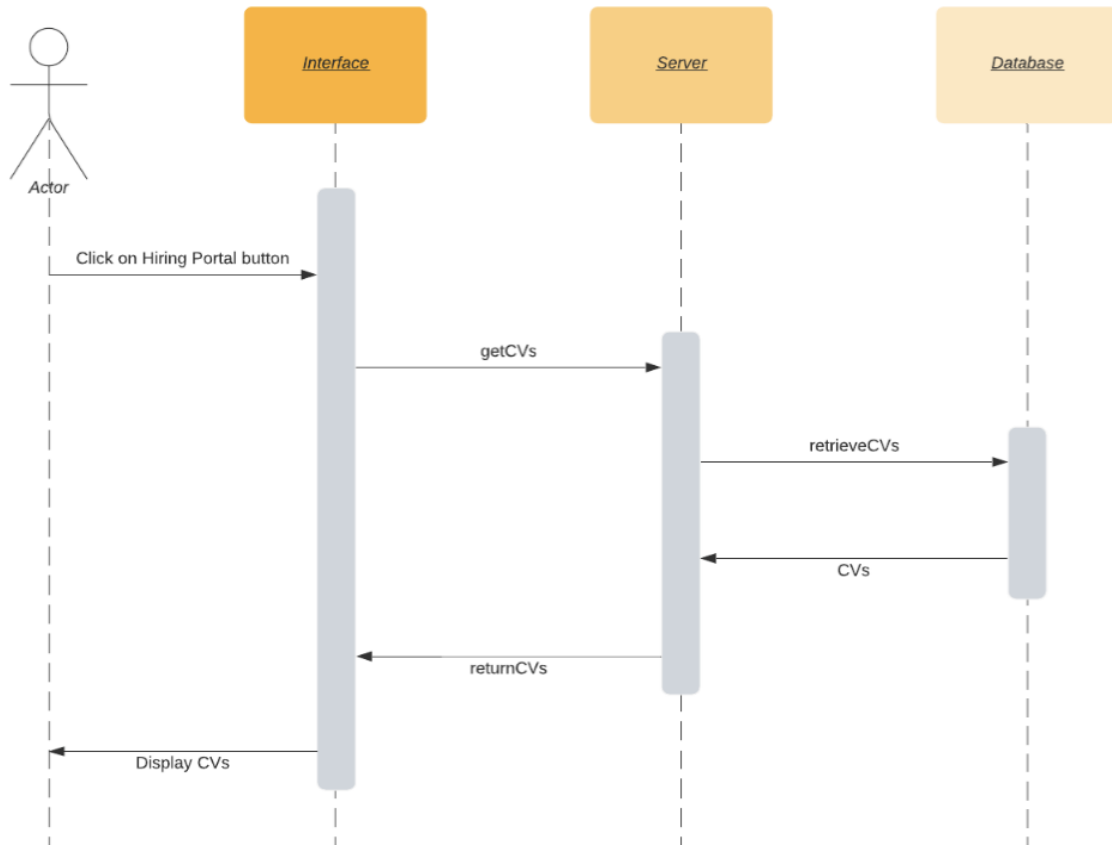


Fig 4.14 Employee Reviews Sequence Diagram

4.4.15 Employee Reviews Reminder

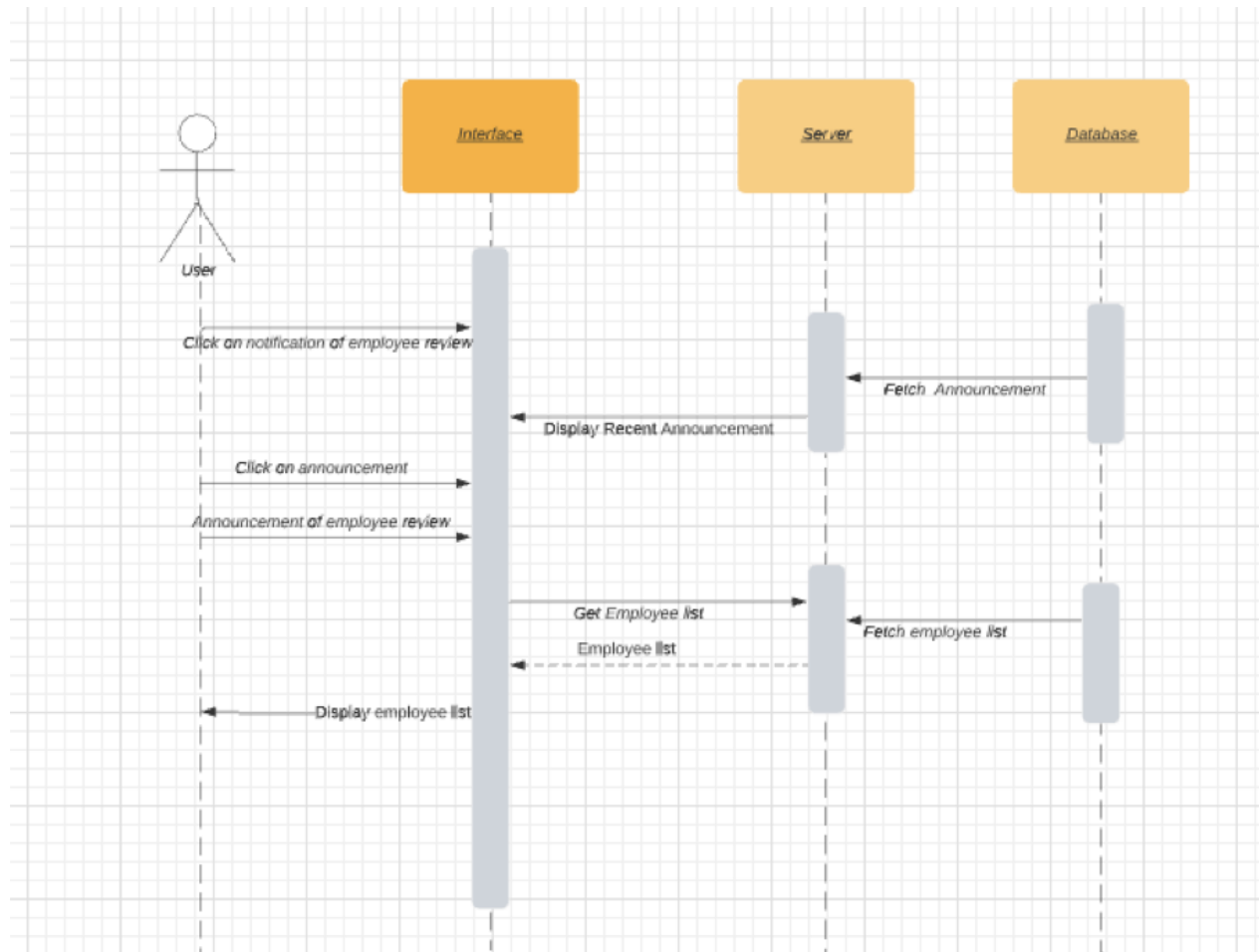


Fig 4.15 Employee Reviews Reminder Sequence Diagram

4.4.16 Hiring Portal:

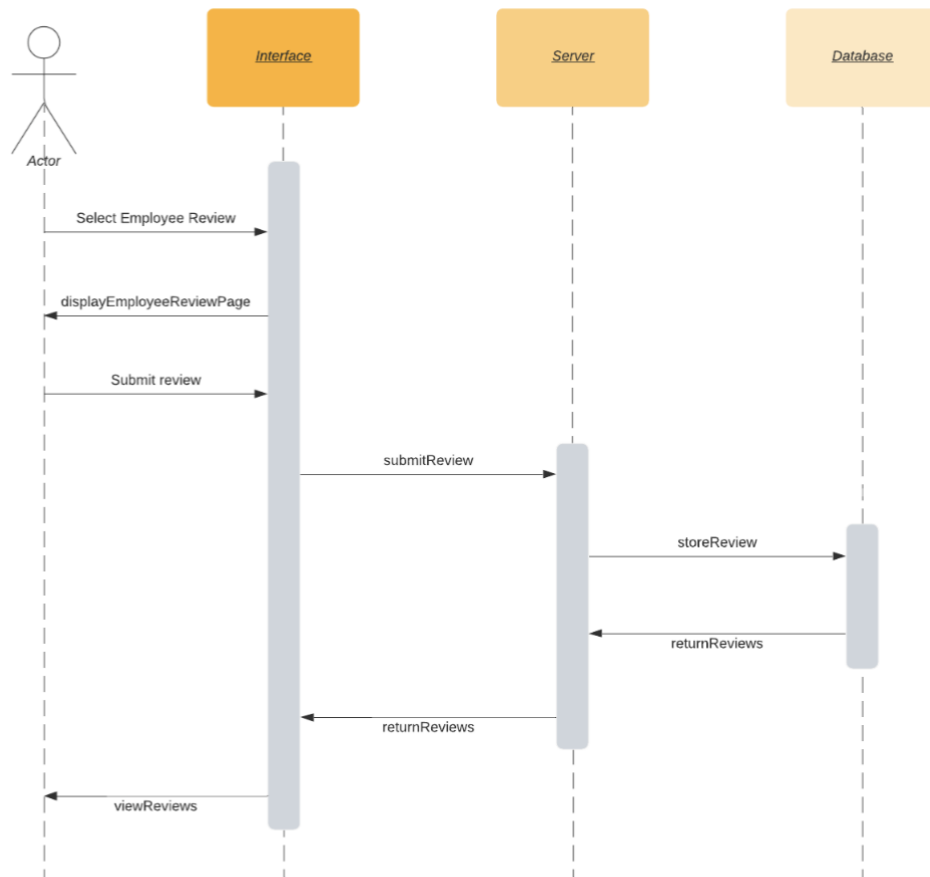


Fig 4.16 Hiring Portal Sequence Diagram

4.4.17 Notifications:

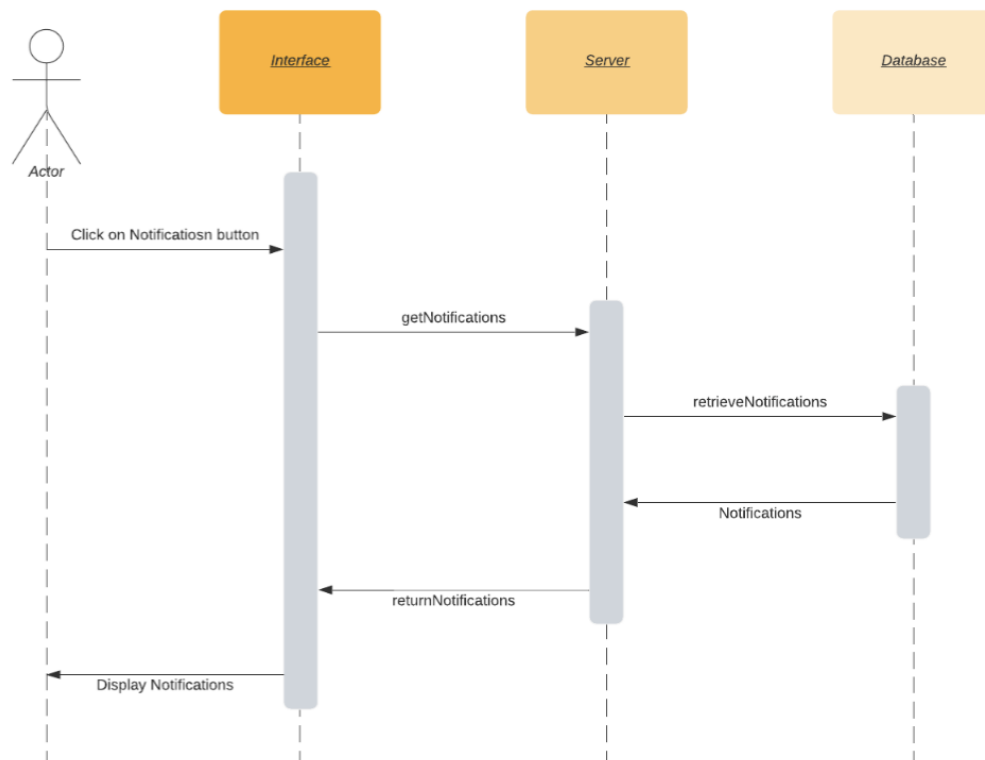


Fig 4.17 Notifications Sequence Diagram

4.4.18 Staff directory:

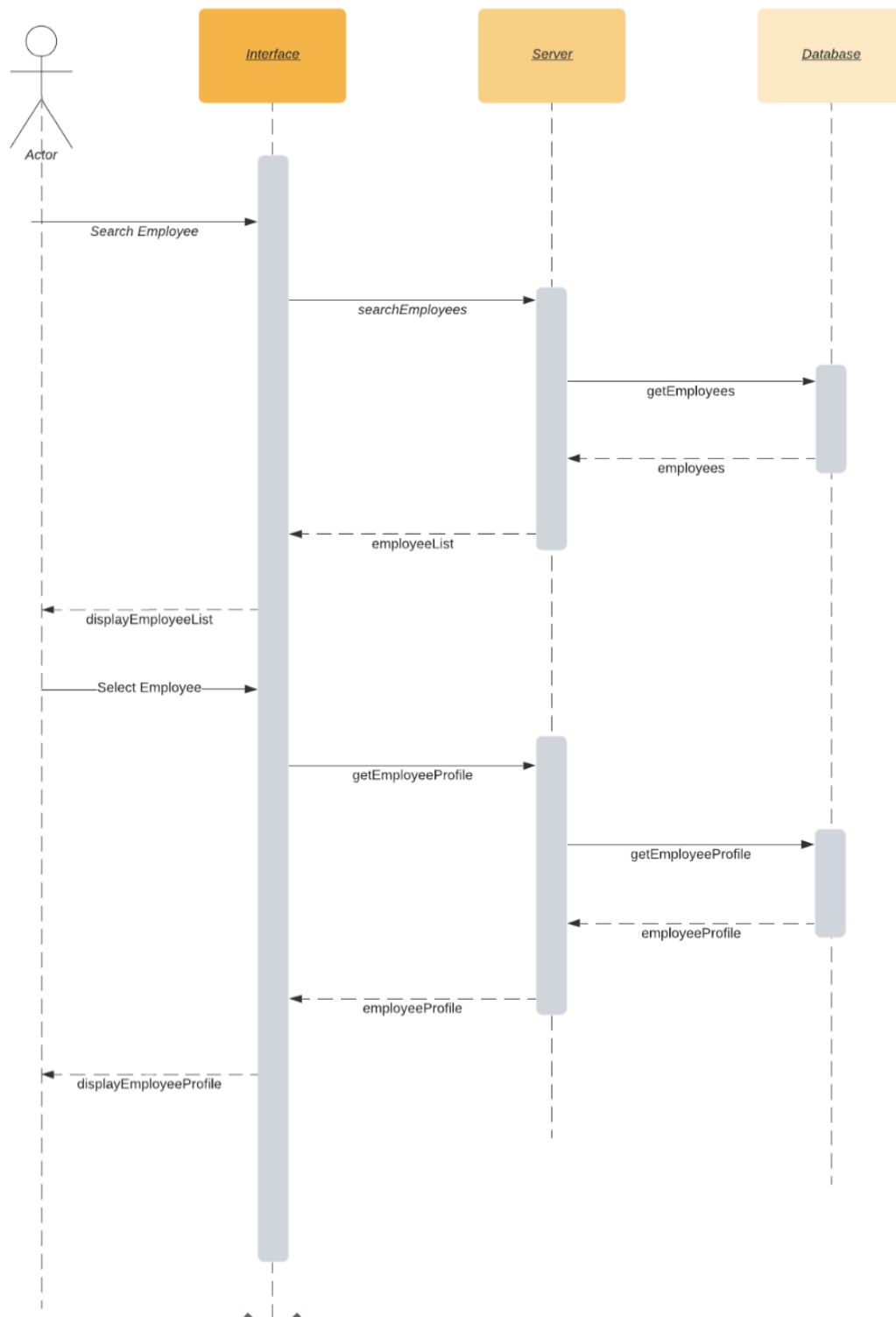


Fig 4.18 Staff Directory Sequence Diagram

4.4.19 Hiring Candidates

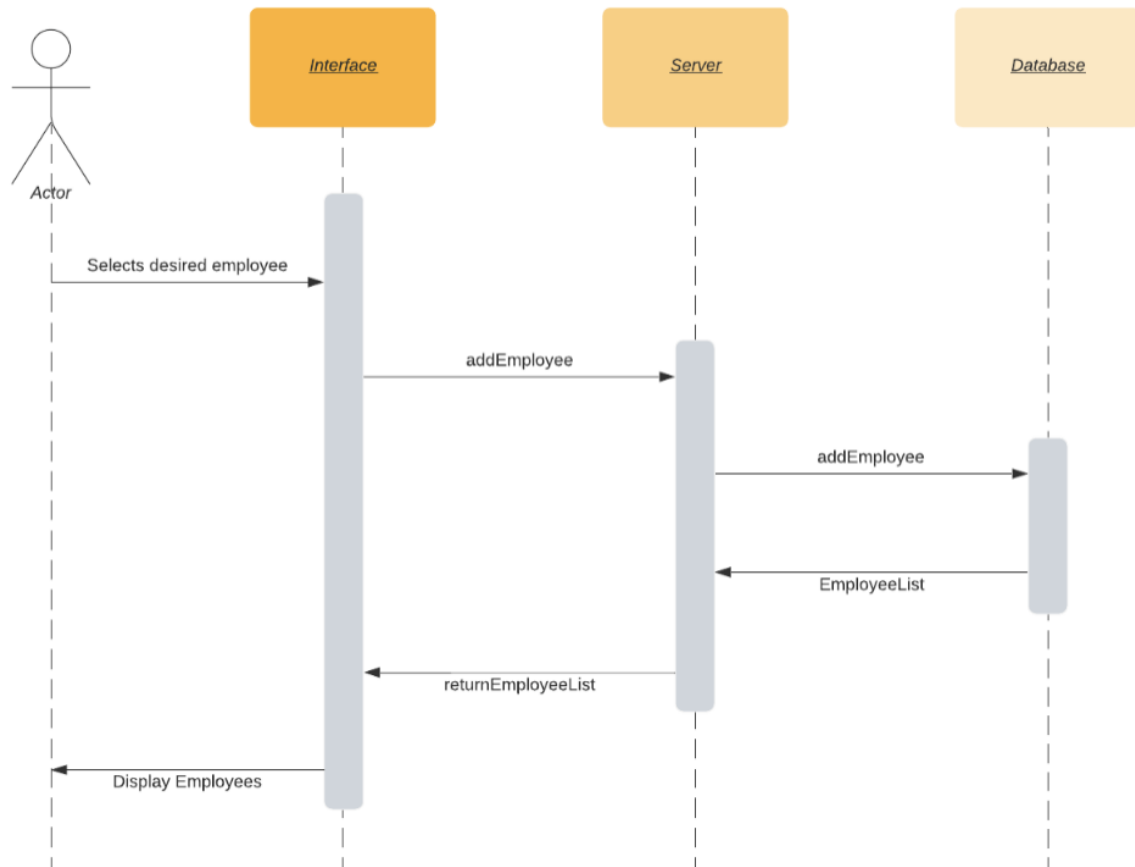


Fig 4.19 Hiring Candidates Sequence Diagram

4.4.20 Employee Request Accept/Denial

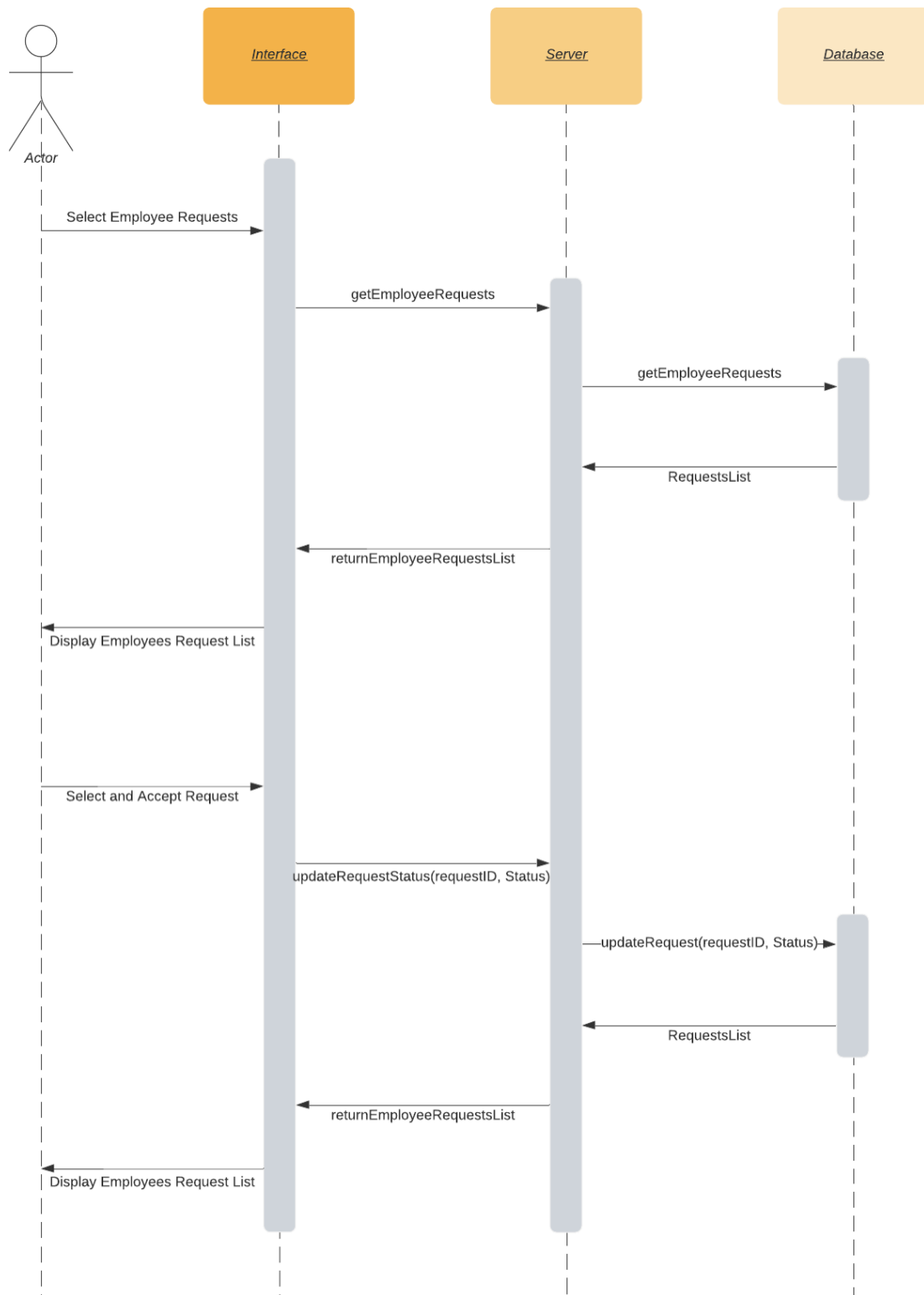


Fig 4.20 Employee Request Accept/Denial Sequence Diagram

5. Software Development Methodology and Plan

This chapter discusses the two software processes we had in mind and how we decided on a process to use with complete justification of our choice.

a. Software Process Selection

The Waterfall Model methodology is followed in a sequential order. The project development team only moves to the next phase of development after the previous step has been completed. The thorough upfront preparation of Waterfall yields precise project plans. Clients will know exactly what will be delivered because the project scope, cost, and deadline are all well-defined. With a clear framework, even if a team encounters turnover, a new member may step in and contribute without sidetracking the timetable. However in the Waterfall model deviation from the original plan is difficult with a strict layout, testing is completed at the end of the Waterfall project, and the final quality assurance process takes a long time. If a client's needs change while a Waterfall project is in progress, it cannot be handled.

Agile methodology encourages flexible, rapid progress using iterative development, delivering pieces of the project along the way to ensure customer needs are met. Frequent delivery allows the customer to provide constant feedback, resulting in a higher-quality product. The methodology breaks down large tasks to be completed in specific time frames. The development and testing activities are being done at the same time. The Agile methodology is known for its flexibility and Agile allows changes in project development requirements. Agile also performs testing concurrently with software development. Agile, on the other hand, does not comply with a strict timeline, which might be tough to manage under pressure. Project requirements that change may generate issues in other sections of the company. Agile requires a dependable team. Wasted time and money could occur from a weak link in the Agile team or management.

Our Selection:

Our group worked with the agile model. The waterfall model is not ideal for projects where the requirements are not completely decided at the beginning. That is the case in our project since we will be updating the requirements in the later phases. Our project does not have

strict regulatory requirements and there is a lot of room to make changes, this pushed us away from the Waterfall model.

Justification of Selection:

- Since the water model has defined requirements, it leaves less room for creativity which our team would not prefer. Agile however, allows us to adjust requirements and priorities along the way. Our team is trying to implemented some innovative features that do not fully exist in any form today so the agile model was ideal for that.

Project Context Analysis

	1 (Low)	2	3	4	5 (High)
Potential loss due to defect/bugs			-		
Developers' experience/skills				-	
Rate of requirements change				-	
Team Size (5,10,25,50,100+)	-				
Organization culture (adaptive to change)			-		
Pressure to develop early releases			-		

Business staff's commitment to work extensively with development team					-
Developers experience with similar systems		-			
Availability of reusable components				-	

b. Gantt Chart

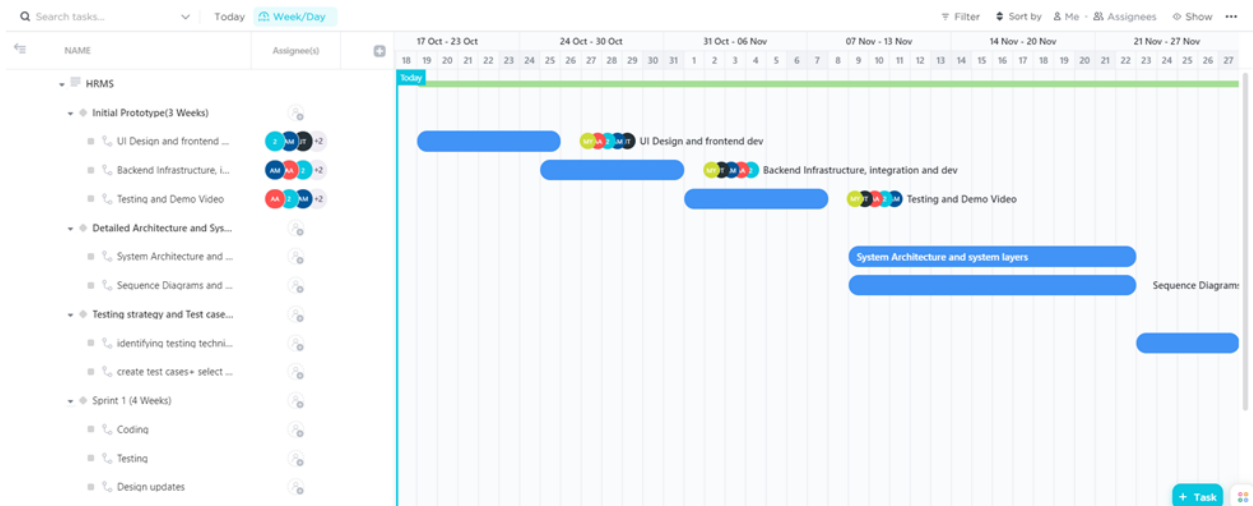


Fig 5a. Gantt Chart (Part 1)

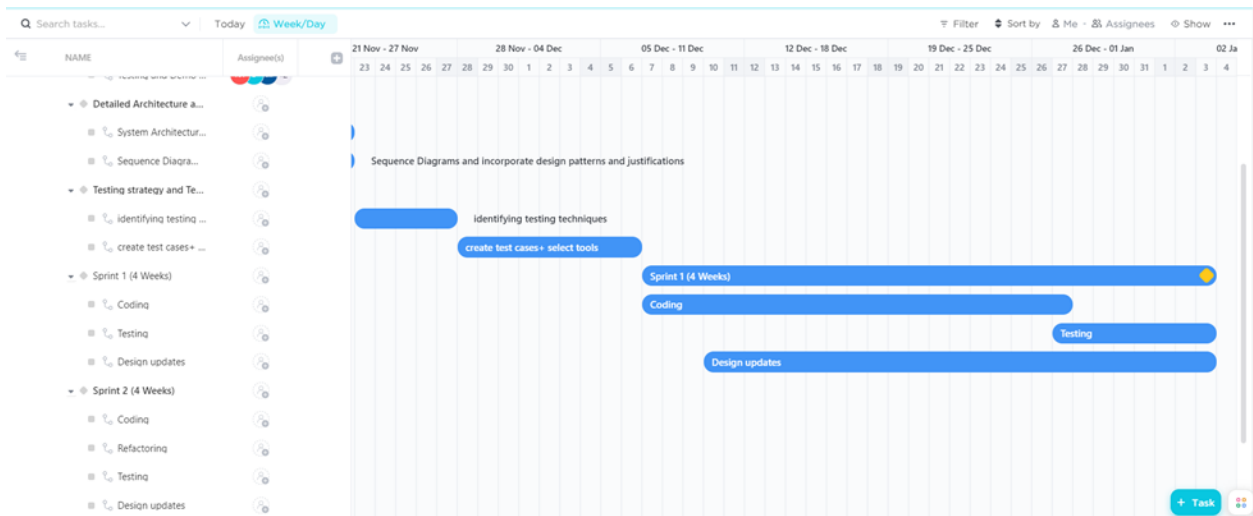


Fig 5b. Gantt Chart (Part 2)

6. Database Design and Web Services

This chapter describes our database, its tables and their attributes.

a. Database Design

We have chosen a NoSQL database, DynamoDB. It is auto scaleable and has a low latency.

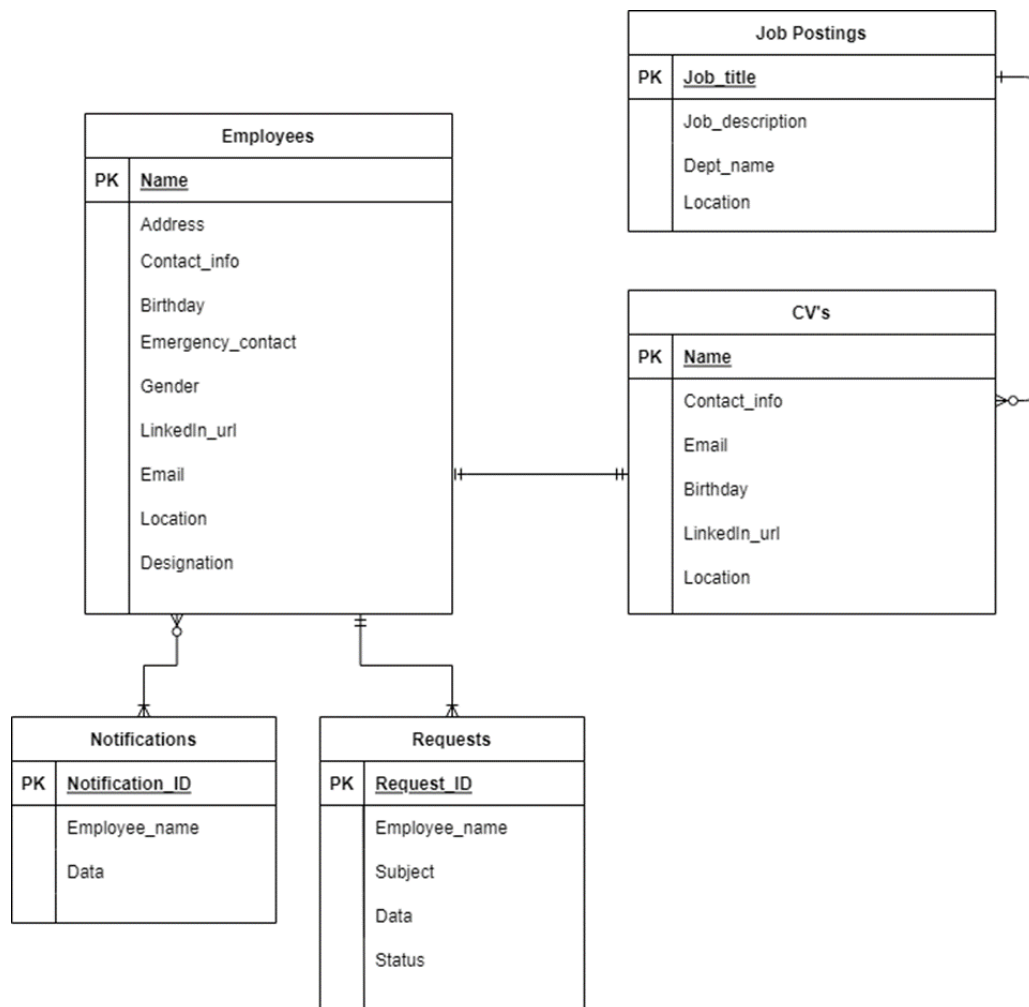


Fig 6. Database Tables

All the three tables shown in this ER Diagram are part of AWS Dynamodb. The Employees table is part of the HRMS node server. The Job postings and CV's tables are part of the Job Application node server.

The Employees table will have complete information about the current employees of the company. All the attributes will be stored as strings.

The Job postings table will store the information about all the active job postings.

The CV's Table will contain the data of the application form such as the applicant's name, their contact information, their location and their LinkedIn profile url. The CV documents will be stored in amazon s3 storage.

b. API Specification

We are not using any external API's. All of the API's being used are made by us.

7. System User Interface

Our application consists of 3 parts: an HR portal, Employee and a jobs portal for candidates.

Login screen:

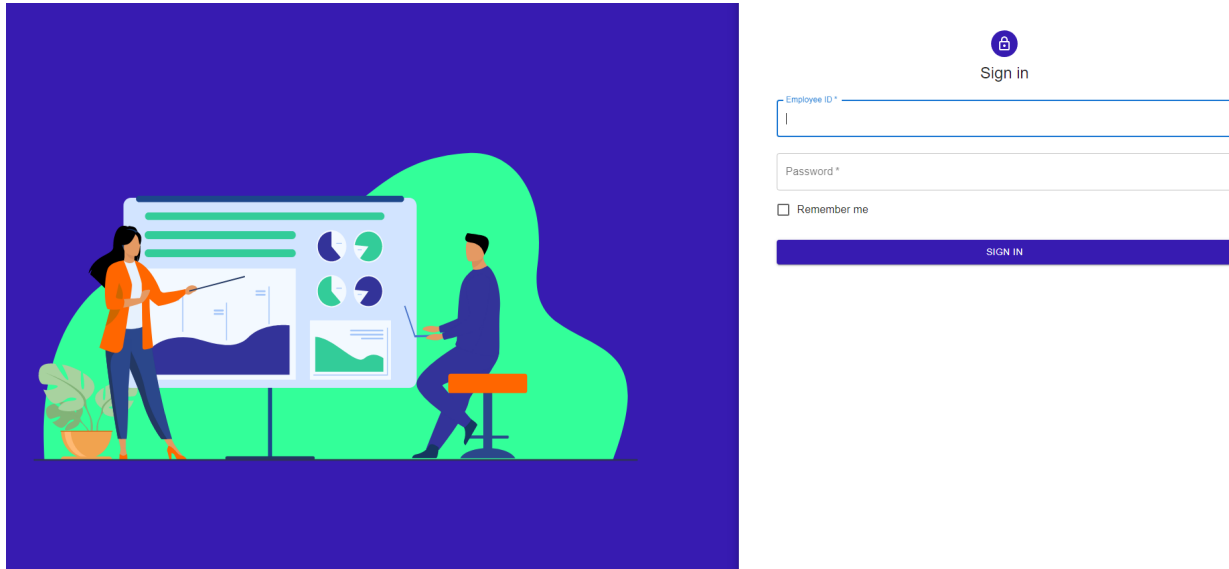


Fig 7.1 Login Screen

HR portal when user logs in as HR:

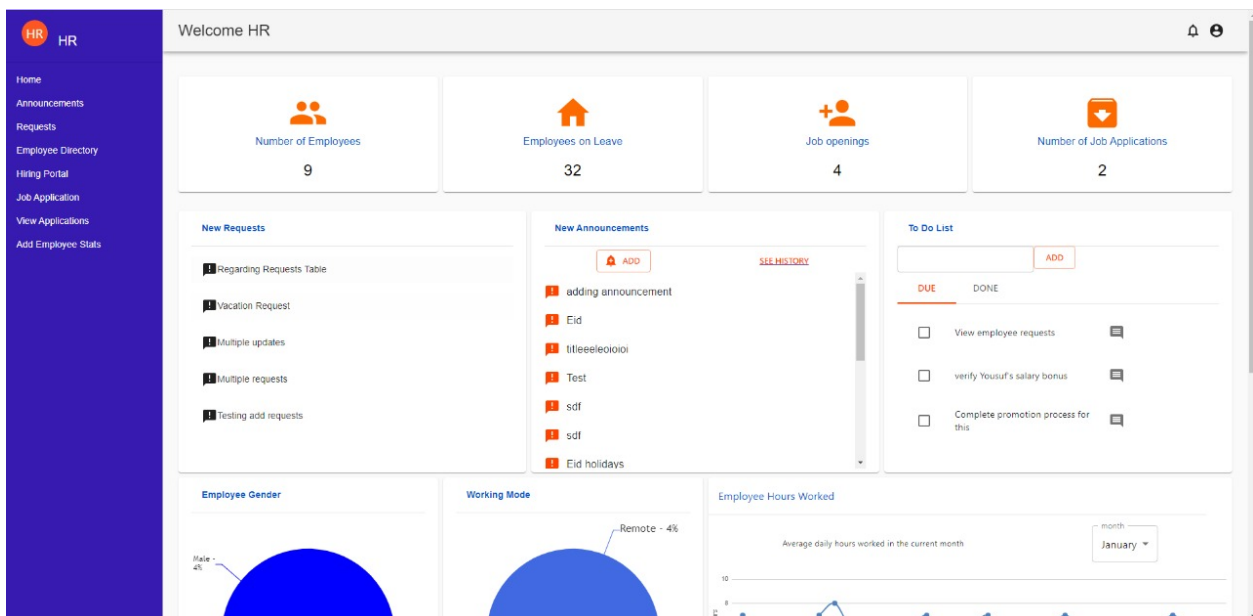


Fig 7.2a HR Portal Screen

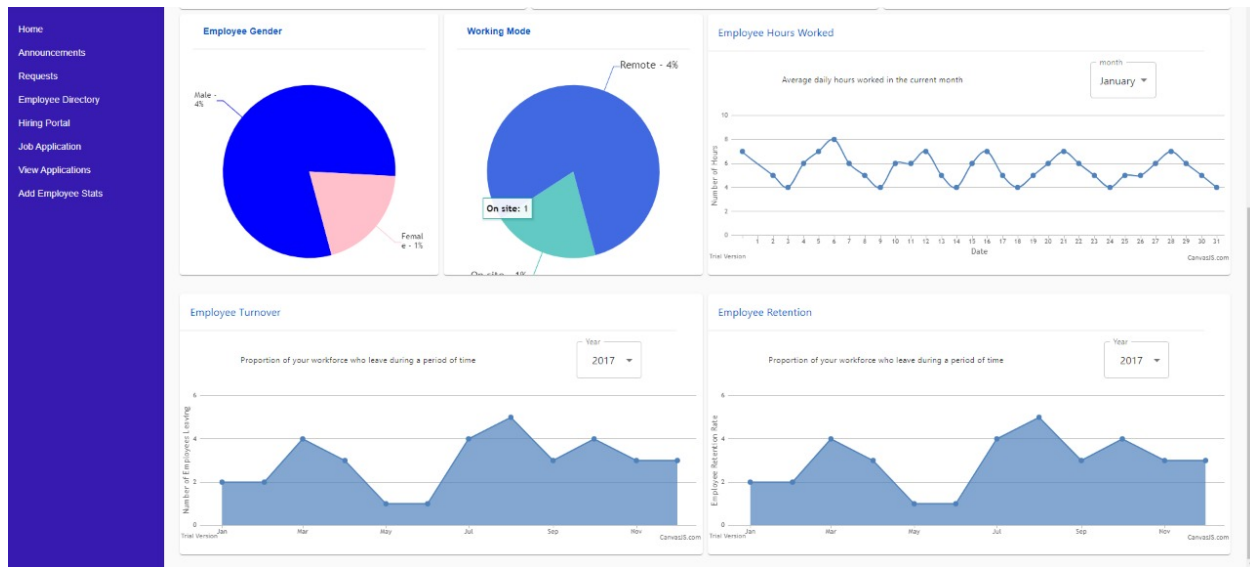


Fig 7.2b HR Portal Screen

This allows HR to monitor real-time analytics, view requests, add announcements, requests, job postings, job applications and number of employees. It also has a navigation bar and a header bar to logout and change passwords.

Employee Requests:

The Employee Requests Screen displays a table of requests with the following columns: Employee ID, Type, Title, Status, Request, and Date. The table includes 5 rows of data. Below the table, there are buttons for 'APPROVE SELECTED REQUEST' and 'DENY SELECTED REQUEST'.

Employee ID	Type	Title	Status	Request	Date
22100303	type	Regarding Requests Table	Denied	This is one test request made manually, bus iake around map funtion call kar do thx bye.	2022-01-01
22100270	Leave	Vacation Request	Approved	I want a holiday vacation	2022-01-01
134567	Other	Multiple updates	Approved	Testing multiple updates	2022-01-01
22100062	Other	Multiple requests	Approved	Testing multiple requests by same employee	2022-01-01
256789	Other	Testing add requests	Denied	Add requests is working fine	2022-01-01

Fig 7.3 Employee Requests Screen

A list containing all the requests by employees with an option to accept or deny these requests by the HR.

Employee directory:

<input type="checkbox"/>	Employee ID	Full Name	Address	Contact	Department	Designation	Level	Gender	Date of Birth	Remain
<input type="checkbox"/>	HR	HR								
<input type="checkbox"/>	22100270	Jibran Ahmed	117/8 c Block SNOOCHS.L...	03009401582	Finance	Manager	Mid	Male		
<input type="checkbox"/>	10005898	Ahmed	4 Downing street	03115268982	Product	Product Associate	Associate	Male		
<input type="checkbox"/>	22100289	Mohammad Yousuf	114/8 E Block SNOOCHS.L...	03114060982	Data	ML manager	Manger	Male		
<input type="checkbox"/>	22100304	Yousuf	Lahore	0321312882882	CS	Software Engineer	422	Male		418
<input type="checkbox"/>	22100305	Javeria	Lahore	12341242312322	CS	Software Engineer	422	Female		418
<input type="checkbox"/>	67382716	Shah Hussain Qadri	331 E8 Johar Town, Lahore...	03317483928	Productions	Head of Productions	1			60
<input type="checkbox"/>	62819276	David Bowie	78 Provot Drive, SOmeplice...	009836528829	Marketing	Associate	1			70
<input type="checkbox"/>	root	root								

Fig 7.4 Employee Directory Screen

A list containing all the data of employees e.g. name and date of birth etc with a button to add employees and their details.

Employee ID *

Name *

Department *

Level *

Role

Contact *

Designation *

Address *

Date Joined *

Email *

Password *

Gender

Working mode

ADD

Fig 7.5 Adding Employees Popup

Announcements:

Title	Posted On	Full Name	Department	Announcement
adding announcement		Mohammad Yo...	sdf	ABCD
Eid		Mohammad Yo...	HR	Eid holidays will be in June
titleeeleololo		yOusif	CS	data
Test		Test	Test	Test
sdf		Mohammad Yo...	sdf	ABCD
sdf		sdf	sdf	ABCD
Eid holidays		Aamina	HR	Eid holidays in July
TEST		Adnan	CS	Test
NEW		NEW	CS	TESTT
Cafeteria open			Logistics	repair maintenance of cafeteria completed
Test		Aamina	CS	Testing Announcements
Manual Created Announcement		Adnan	CS	TESTING123
Test		Aamina	CS	Testing Announcements
Test		Aamina		✓ New Announcement added! ✕ rments

Fig 7.6 Announcements Screen

Announcements page with the details of the announcements along with options to add announcements as well.

Name *

Department *

Title *

Announcement *

ADD

Fig 7.7 Add Announcement Popup

View Applications:

The screenshot displays the 'View Applications' interface of an HR system. On the left is a dark blue sidebar with navigation links: Home, Announcements, Requests, Employee Directory, Hiring Portal, Job Application, View Applications (highlighted), and Add Employee Stats. The main content area has a light gray header with 'Welcome HR' and notification icons. Below the header is a search bar labeled 'Enter keywords seperated by commas:' with a placeholder 'keywords *'. A table lists two job applications with columns for checkboxes, date, name, city, state/province, phone number, LinkedIn, email, job applied, CV, and Match. The first application is for 'Salesforce Project Manager' by Aamina Ah... and the second is also for 'Salesforce Project Manager' by Mohamma... Both have 'VIEW CV' and 'GET MATCH' buttons. Below the table is a 'Matches' section with fields for 'Match: %' and 'Words matched:'. A pagination indicator '1-2 of 2' is visible at the bottom right of the table area.

<input type="checkbox"/>	date	Name	City	State/Provi...	Phone Number	Linkedin	email	job applied	CV	Match
<input type="checkbox"/>	Sat May 1...	Aamina Ah...	Lahore	Punjab	+923114060592	ewrew	65	Salesforce Project Manager	VIEW CV	GET MATCH
<input type="checkbox"/>	Sat May 1...	Mohamma...	Lahore	Punjab	+923114060592	abc	g@gmail.c...	Salesforce Project Manager	VIEW CV	GET MATCH

Fig 7.8 View Applications Screen

A view application page where the HR can view applications for jobs. The HR also has the option to view the CV by clicking view CV. The HR can also scan a candidate's CV with words like Javascript, python etc without looking at the CV by adding keywords and clicking get match. This feature can save time for HR/hiring managers as they will not have to read through each and every CV thus saving time.

Hiring Portal:

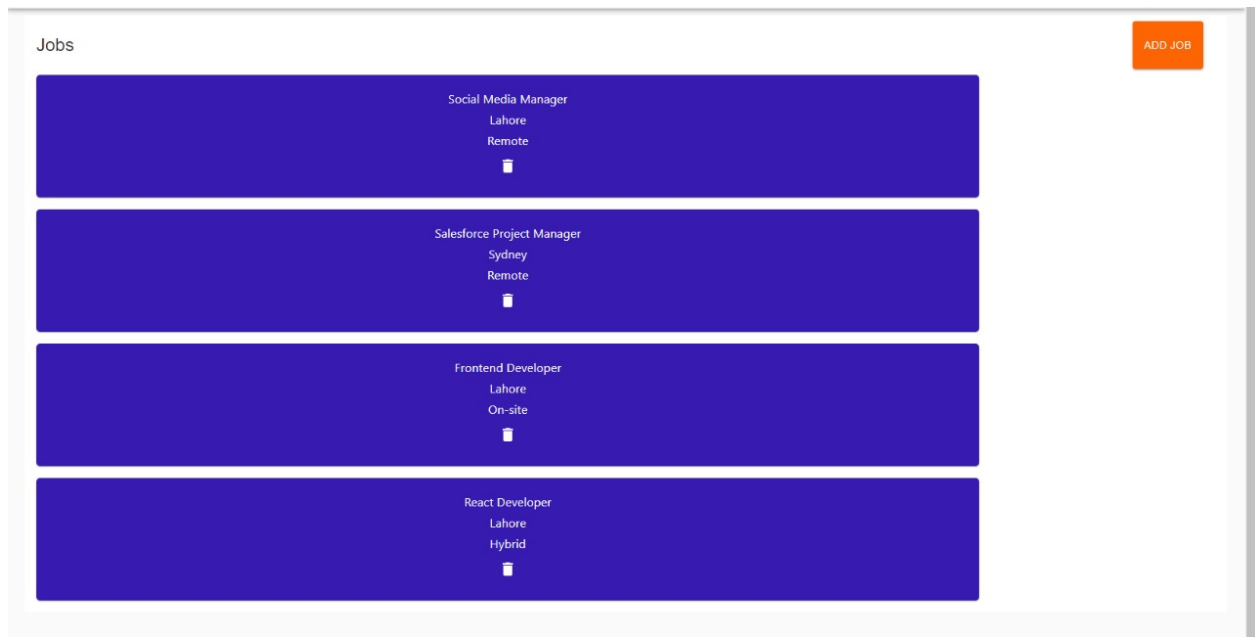


Fig 7.9 Hiring Portal Screen

A hiring portal where all jobs posted can be viewed by the HR. The HR can also add jobs by clicking the add job button or delete jobs by clicking the trash icon.

Add Job

Job Title *

Job Description

Department name *

Location *

Job type

Back

Submit

Fig 7.10 Adding Job Screen

Add a job portal where a job can be added along with details of the job.

Jobs portal:

Job openings
Social Media Manager Lahore Remote
Salesforce Project Manager Sydney Remote
Frontend Developer Lahore On-site
React Developer Lahore Hybrid

Fig 7.11a Job Portal Screen

A jobs portal where all the jobs can be seen by candidates. A candidate can apply for a job by clicking on a job posting.

Job:Social Media Manager	Location:Lahore(Remote)
--------------------------	-------------------------

Job Description

Job brief We are looking for a talented Social media manager to administer our social media accounts. You will be responsible for creating original text and video content, managing posts and responding to followers. You will manage our company image in a cohesive way to achieve our marketing goals. As a Social media manager, we expect you to be up-to-date with the latest digital technologies and social media trends. You should have excellent communication skills and be able to express our company's views creatively. Ultimately, you should be able to handle our social media presence ensuring high levels of web traffic and customer engagement. Responsibilities Perform research on current benchmark trends and audience preferences Design and implement social media strategy to align with business goals Set specific objectives and report on ROI Generate, edit, publish and share engaging content daily (e.g. original text, photos, videos and news) Monitor SEO and web traffic metrics Collaborate with other teams, like marketing, sales and customer service to ensure brand consistency Communicate with followers, respond to queries in a timely manner and monitor customer reviews Oversee social media accounts' design (e.g. Facebook timeline cover, profile pictures and blog layout) Suggest and implement new features to develop brand awareness, like promotions and competitions Stay up-to-date with current technologies and trends in social media, design tools and applications Requirements and skills Good to have Social Media Manager skills Proven work experience as a Social media manager Hands on experience in content management Excellent copywriting skills Ability to deliver creative content (text, image and video) Solid knowledge of SEO, keyword research and Google Analytics Knowledge of online marketing channels Familiarity with web design Excellent communication skills Analytical and multitasking skills BSc degree in Marketing or relevant field

Job Application

Fig 7.11b Job Portal Screen

The candidate will be redirected to a form with the job application details and description. The candidate can then enter their details and submit the form.

Add stats form:

The screenshot shows a web application interface for an HR manager. On the left is a dark blue sidebar with a logo and a list of navigation items: Home, Announcements, Requests, Employee Directory, Hiring Portal, Job Application, View Applications, and Add Employee Stats. The main content area has a header 'Welcome HR' and a form titled 'Add stats form'. The form contains five input fields: 'Employee ID *', 'Rating' (a dropdown menu), 'Team Score *', 'Hours Worked *', and 'Comments *'. At the bottom of the form are two orange buttons: 'Back' and 'Submit'.

Fig 7.12 Add Statistics Screen

A form to add stats where an HR manager can add stats and ratings for an employee and they will show up on the employee's analytics dashboard.

Employee Portal:

The screenshot shows an 'Employee Portal' for a user named Jibran Ahmed. The left sidebar is dark blue with the user's name and a list of navigation items: Home, My Performance Analytics, Add Requests, and Change Password. The main content area has a header 'Welcome Jibran Ahmed' and three large buttons: 'Performance statistics', 'change password', and 'Add requests'. Below these are three panels: 'Requests', 'Announcements', and 'To Do List'. The 'Requests' panel lists five items with exclamation mark icons: 'Regarding Requests Table', 'Vacation Request', 'Multiple updates', 'Multiple requests', and 'Testing add requests'. The 'Announcements' panel lists four items with exclamation mark icons: 'adding announcement', 'Eid', 'titleeeleoioio', and 'Test'. The 'To Do List' panel has a search bar with an 'ADD' button, a 'DUE' tab, and three tasks: 'View employee requests', 'verify Yousuf's salary bonus', and 'Complete promotion process for this'. Each task has a checkbox and a message icon.

Fig 7.13 Employee Portal Screen

The employee can view announcements, requests, make a to do list and view performance stats added by the HR.

Employee analytics:

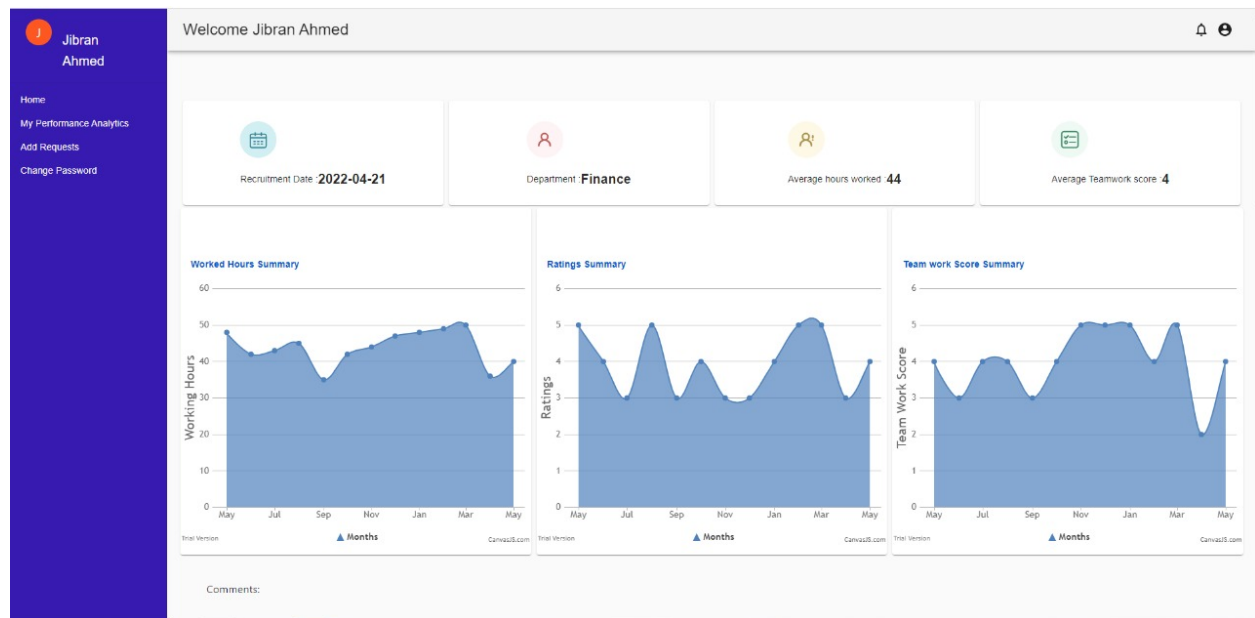
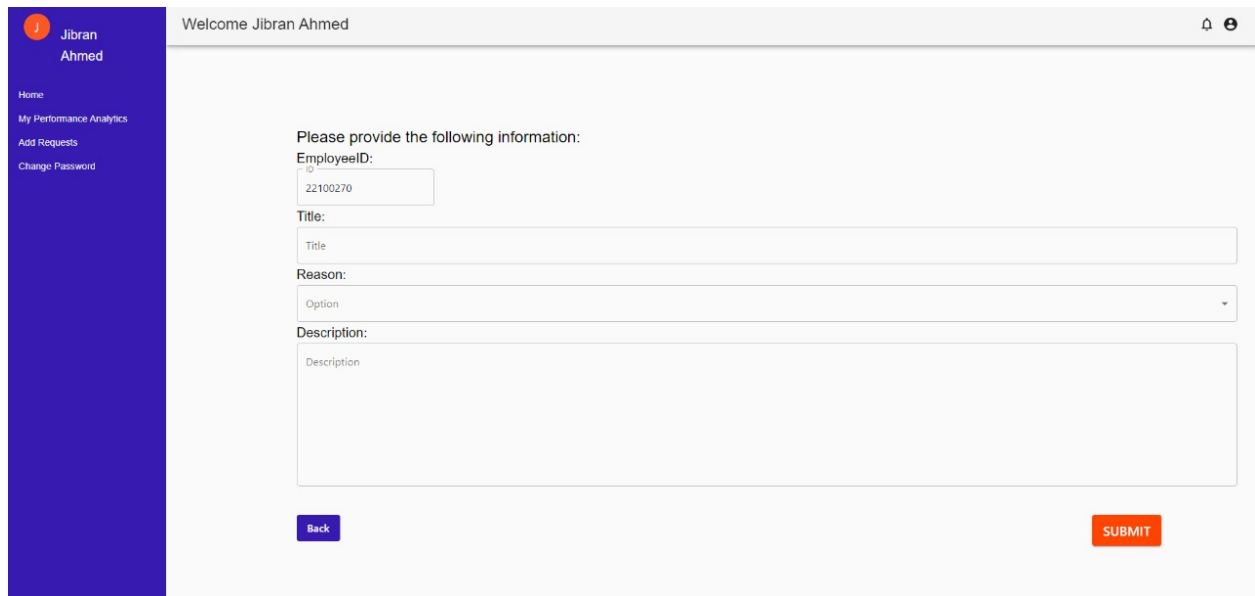


Fig 7.14 Employee Analytics Screen

an employee analytics page where an employee can view their teamwork scores, hours worked and comments by the HR.

Employee Requests:



The image shows a web application interface for an employee portal. On the left is a dark blue sidebar with a user profile at the top (Jibran Ahmed) and a list of navigation links: Home, My Performance Analytics, Add Requests, and Change Password. The main content area has a light gray header with a welcome message 'Welcome Jibran Ahmed' and notification icons. Below the header, the form is titled 'Please provide the following information:'. It contains four fields: 'EmployeeID' (a text box with '22100270'), 'Title' (a text box), 'Reason' (a dropdown menu with 'Option' selected), and 'Description' (a large text area). At the bottom of the form are two buttons: a blue 'Back' button and an orange 'SUBMIT' button.

Welcome Jibran Ahmed

Please provide the following information:

EmployeeID: 22100270

Title:

Reason: Option

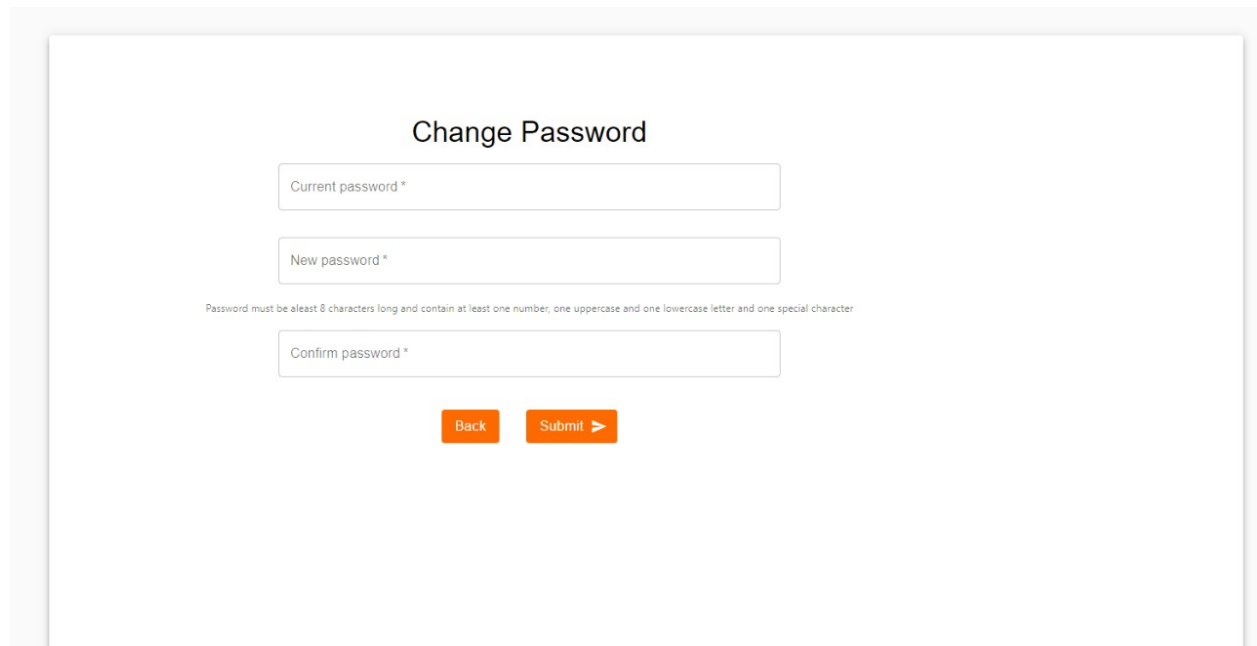
Description:

Back SUBMIT

Fig 7.15 Employee Portal Screen

A requests portal where an employee can add things like maternity leave and vacations leave etc. These requests are then sent to the HR for approval.

Change password:



The image shows a web form titled "Change Password". It contains three input fields: "Current password *", "New password *", and "Confirm password *". Below the "New password *" field, there is a small text string: "Password must be atleast 8 characters long and contain at least one number, one uppercase and one lowercase letter and one special character". At the bottom of the form, there are two orange buttons: "Back" and "Submit >".

Change Password

Current password *

New password *

Password must be atleast 8 characters long and contain at least one number, one uppercase and one lowercase letter and one special character

Confirm password *

Back Submit >

Fig 7.16 Change Password Screen

A change password page where an employee can change their password.

8. Project Security

In this we are discussing the top 3 security risks and their controls for our system[1][2][3][4]. Also, the potential losses for our system if we do not implement the controls. Plus, we will also be discussing some testing tools suitable for our system.

a. Project Threats

1. Broken Access Control

- a) Violation of the principle of least privilege or deny by default
- b) Accessing API with missing access controls for POST, PUT and DELETE.
- c) Elevation of privilege. Acting as a user without being logged in or acting as an admin when logged in as a user.

2. Cryptographic failures

All employee passwords are encrypted and then stored. If the cryptographic algorithms are weak or old it can lead to attackers gaining access to our system.

3. Identification and Authentication Failures

The attacker has a list of valid usernames and passwords they can access our system without raising any red flags

b. Potential Losses

1. Broken Access Control

- a) An attacker might be able to change or delete content
- b) Perform unauthorized functions
- c) Attackers can take over site administration.

2. Cryptographic failures

As with broken access control the attacker can change or delete information or perform unauthorized actions

3. Identification and Authentication Failures

If the attacker can access our system without raising any red flags and have access to all of the company information it can be dangerous for the security and privacy of employees.

c. Security Controls

For Broken Access Control we can:

- a) Minimize Cross-Origin Resource Sharing (CORS) usage.
- b) Rate limit API and controller access to minimize the harm from automated attack tooling.

For Cryptographic Failures we can:

- a) Store passwords using strong adaptive and salted hashing functions with a work factor (delay factor), such as scrypt or bcrypt
- b) Always use authenticated encryption instead of just encryption.

For Identification and Authentication failures we can:

- a) Implement multi-factor authentication
- b) Enforce a password policy to prevent users from setting weak passwords
- c) Use session tokens

All of these are protective controls

d. Static and Dynamic Security Scanning Tools

For Static Testing we chose Coverity because it is open source and it is compatible with JavaScript and NodeJs, 2 of the major technologies used in our system. Its fast and accurate incremental analysis runs in the background to minimize disruption. It can comprehensively track and manage compliance through a wide range of security, quality, data protection, and safety standards. [5]

And for Dynamic testing we chose APIsec. It automatically creates and runs thousands of test cases tailored to your unique API architecture - integrated directly into your workflow. Unlike traditional security solutions that look for common security issues, APIsec pressure-tests the entire API, including hidden business logic flaws responsible for most breaches.[6]

9. Risk Management

Potential Risks and Mitigation Strategies

Sr.	Risk Description	Mitigation Strategy
1.	Server failure	There are two ways to fix this issue: to store data in all the accessible databases or distribute it evenly among them
2.	Broken Authentication	Configure multi-factor authentication whenever possible. The more hoops an attacker has to jump, the harder it is to get into your system. Also paying attention to session management and setting application timeouts correctly. When a user closes a browser, log them out of the system. Any time a user leaves a session while still logged in, the entire system remains vulnerable.
3.	Sensitive data exposure	<p>Sensitive data exposure can be prevented by</p> <ul style="list-style-type: none">● Encrypting data in transit and at rest.● Using the latest encryption algorithms.● Disabling auto-complete on forms that collect data.● Disabling caching on forms that collect data.

4.	Broken Access Control	<p>It is important to control who has access to which portion of the website. Access control is only effective if it is enforced in trusted server-side code, where the attacker cannot modify the access control authentication.</p> <ul style="list-style-type: none"> • Deny access to standard functions. • Use access control lists and role-based authentication mechanisms. • Don't just hide features.
5.	Security Misconfiguration	<p>Security misconfiguration can be mitigated by not allowing default configuration in our website. It can be prevented by</p> <ul style="list-style-type: none"> • Disabling administration interfaces • Disabling use of default accounts/passwords. • Configuring the server to prevent unauthorized access, directory listing, etc. • Considering periodically performing scans and audits to help detect future configuration errors or missing fixes.

6.	Cross Site Scripting	<p>In general, effective prevention of cross site scripting vulnerabilities can involve a combination of the following measures:</p> <ul style="list-style-type: none"> ● Sanitize User Input <ol style="list-style-type: none"> 1. Validate to detect potentially malicious input from users. 2. Encrypt the output to prevent potentially malicious user-supplied data from triggering the autoload and execute behavior by a browser ● Limit use of user provided data <ol style="list-style-type: none"> 1. Only use when it's necessary 2. Utilize content security policy By providing additional layers of protection against cross site scripting attacks.
----	----------------------	--

7.	Insufficient logging and monitoring	<p>Depending on the risk of the data being stored or processed by the application:</p> <ul style="list-style-type: none"> • Ensure that all server-side login, access control, and access validation errors can be logged with sufficient user context to identify suspicious or malicious accounts and for a sufficient period of time to allow for deferred forensic analysis. • Ensure that high-value transactions have an audit trail with integrity checks to prevent tampering or deletion, such as add-only database tables and the like. • Provide effective monitoring and alerts to detect and address suspicious activity in a timely manner.
8.	Insecure Direct Object References	<p>Perform proper and consistent user authorization and whitelist the choices. More often than not, however, the whole problem can be avoided by storing the data internally.</p>
9.	Unvalidated Redirects and Forwards	<p>This can be prevented by</p> <ul style="list-style-type: none"> • Avoiding redirects • Providing a static list of valid locations to redirect to.

10.	Unrestricted File Upload	<ul style="list-style-type: none">● Never accept a filename and its extension directly without an allow list filter.● The application should filter and check content for all files uploaded to the server. Files must be carefully analyzed and validated before being made available to other users. When in doubt, the file should be discarded.● It is necessary to have a list of the only allowed extensions on the web application. And the file extension can be selected from the list.
-----	--------------------------	--

10. Testing and Evaluation

Throughout the course of the project, we have manually tested our system. We mostly used Postman to test our API's. Two of the test cases that we created are mentioned below. Since our application was very heavy and our machines could not handle the load of automated testing, we did not carry out any automated tests.

View Performance:

Test Case ID	T03	Use Case Name	View Performance				
Test Created by	Aamina	Test executed by					
Test Case Priority	High	Test Case Objectives	Check the functionality of the View Performance use case for th employees				
Test browser/platform	OS: Windows 10, Browser: Google Chrome etc.						
Pre-conditions	The user is logged in using their employee credentials.						
Post-conditions	The user can view their how performance has been according to their Dept head/ HR manager.						
Step No	User actions		Inputs	System response	Expected Outputs	Actual Output	Test Result (Pass/Fail)
1	The user chooses to view their performance		Click View Performance	The performance dashboard id opened	All the ratings and stats for that user are displayed		
Test Case Execution Result		Passed	Failed	Not Executed			
Test Case Variations (Test cases with different inputs go here)							
Test Case ID	T01	T02	T03	T04	T05	T06	
Pre-conditions	User logged in	User logged in	User Logged in	User Logged in. User is a new employee and their performance review has not	User Logged in	User logged in	
Post-conditions							
Step No	Inputs		System response		Expected Outputs		Test Result (Pass/Fail)
1	Click View Performance		Click View Performance		Click View Performance		
Expected Output	The ratings and stats are displayed		The ratings and stats are displayed		An empty screen is opened		
Test Case Execution Result		Not executed	Not executed	Not executed	Not executed	Not executed	Not executed

Fig 8a Test Case 1

Add Request:

Test Case ID	T04	Use Case Name	Add Request				
Test Created by	Adnan	Test executed by					
Test Case Priority	High	Test Case Objectives	check the functionality of the add request use case for the employees				
Test	OS: Windows 10, Browser: Google Chrome etc.						
Pre-conditions	The employee actor has been authenticated and logged in						
Post-conditions	The HFI manager is able to view the requests added by the specific employee in their own database						
Step No	User actions		Inputs	System response	Expected Outputs	Actual Output	Test Result (Pass/Fail)
1	The user clicks on Requests		Click Requests		The system shows the previous requests in the list		Pass
2	The user clicks on add request button		Click Add Request		The system shows the request input form		Pass
3	The user fills in the particular details of the request.		type: "Kindly review my overtime hours"		The system shows the details on the type of request is selected		Pass
4	User selects the type of details		Clicks type		The system shows the new request in the list		Pass
5	The user submits the request		Click Submit				Pass
Test Case Execution Result		Passed	Failed	Not Executed			
Test Case Variations (Test cases with different inputs go here)							
Test Case ID	T01	T02	T03	T04	T05	T06	
Pre-conditions	Logged in user id = 15	Logged in user id = 16	Logged in user id = 19	Logged in user id = 38	Logged in user id = 15	Logged in user id = 98	
Post-conditions							
Step No	Inputs		System response		Expected Outputs		Test Result (Pass/Fail)
1	Click Add Request		Click Add Request		Click Add Request		
Expected Output	1. Leave 2. Leave 3. Maternity leave		previous requests so list will be en		1. Leave2. Leave		
2	Click Add Request		Click Add Request		Request input form opens up		
Expected Output	Request input form opens up		Request input form opens up		Request input form opens up		
3	1. Sick leave		2. Maternity leave		3. Vacation		
Expected Output	-		-		-		
4	Click Submit		Click Submit		Click Submit		
Expected Output	1. Vacation 2. Paternity leave 3. Maternity leave		1. Vacation 2. Paternity leave 3. Maternity leave		1. Vacation 2. Paternity leave 3. Maternity leave		
Test Case Execution Result		Passed	Passed	Passed	Passed	Passed	Passed

Fig 8b Test Case 2

11. Deployment Guidelines

Prerequisites:

- Node js should be installed on PC
- env file
- pdf services api credentials
- git

Deployment:

1. open terminal and type: `git clone https://github.com/aaminamariam/P05.git`
2. then type: `git checkout nevdev`
3. Navigate to P05>P05-HR Management System>Development>Sprint-1
4. on the terminal type: `npm i`
5. then type: `npm run devStart`
6. open another terminal or on a separate server and open the same folder and type : `npm run server2`
7. on another terminal Navigate to P05>P05-HR Management System>Prototype
8. then type: `npm i`
9. then type: `npm run build`
10. then type: `serve -s build (port number)`

We were unable to deploy since our free trial ended on AWS

12. Conclusion

a. Summary

This project aims to simplify the complex HR process of recruitment and managing employees for HR. It simplifies recruitment by making it easier for HR to filter employees. It also makes it easier for HR to manage current employees by giving HR real-time statistics and details of each employee.

b. Challenges

1. Parsing CVs and filtering them was a huge challenge since it was difficult to make an automated system for parsing and filtering through pdfs.
2. Working with multiple servers was also a challenge as it was difficult to figure out the correct routing mechanism for each of them
3. Analytics and graphs were tricky to manage since they required a lot of data and made the application heavier.
4. The UX was also complex to understand as HR use-cases for each organization vary.

c. Future

The system is built in a modular format so it can be edited and added on easily without major changes needed to the databases or the system.

13. Review checklist

Before submission of this report, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

Chapter/Section Name	Reviewer Name(s)
System Requirements, Requirement Specifications	Ali Adnan Arif
Introduction, System User Interface, Deployment Guidelines, Conclusion	Mohammad Yousuf
Introduction, Software Development Methodology and Plan, Risk Management	Javeria Tariq
System Architecture, Database Design and Web services, Project Security, Testing and Evaluation	Aamina Mariam

14. References

- 1) <https://owasp.org/www-project-top-ten/>
- 2) https://owasp.org/Top10/A01_2021-Broken_Access_Control/
- 3) https://owasp.org/Top10/A02_2021-Cryptographic_Failures/
- 4) https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/
- 5) <https://www.synopsys.com/software-integrity/security-testing/static-analysis-sast.html>
- 6) <https://www.apisec.ai/>