**Low-Level Architecture and Data Models**

**P05:HR MANAGEMENT SYSTEM**

**<team member names & ids>**

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| **22100289** | **Mohammad Yousuf** |
| **22100230** | **Javeria Tariq** |
| **22100260** | **Talha Nasir** |
| **22100303** | **Ali Adnan Arif** |
| **22100062** | **Aamina Mariam** |

**Table of Contents**

[1.](#_gjdgxs) Introduction 3

[2.](#_30j0zll) System Architecture 4

[2.1 Architecture Diagram—](#_1fob9te)As it is in the prototype code 4

[2.2 Architecture Diagram—](#_3znysh7)As it should-be 4

[3.](#_2et92p0) Data Models 5

[4.](#_tyjcwt) Tools and Technologies 6

[5.](#_3dy6vkm) Who Did What? 7

[6.](#_1t3h5sf) Review checklist 7

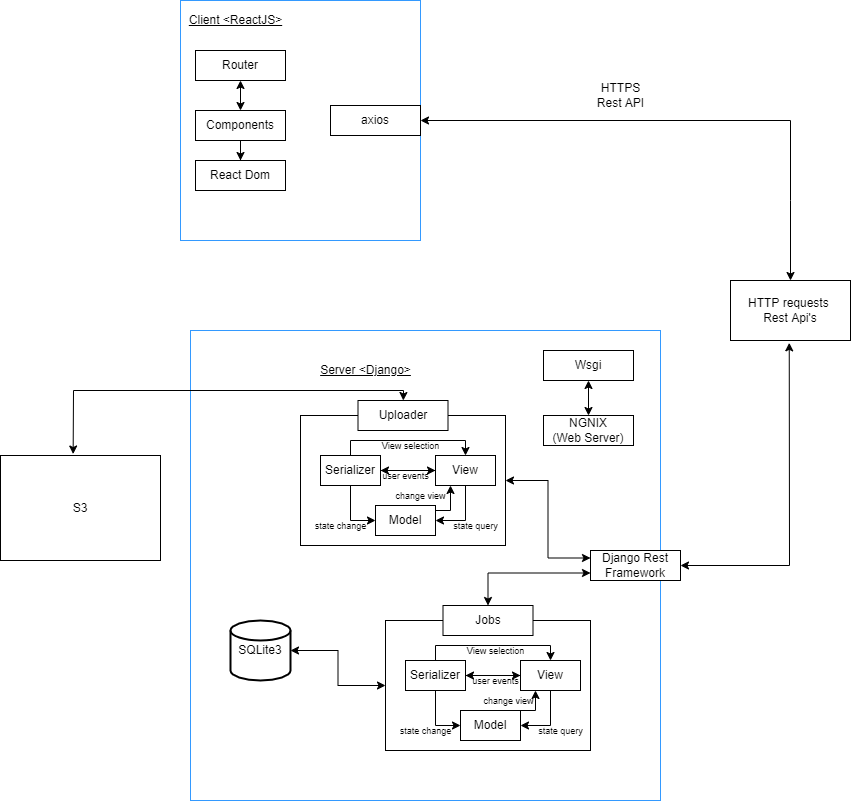
# Introduction

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.

# System Architecture

## Architecture Diagram—As it is in the prototype code



## Architecture Diagram—As it should-be

Both our architecture diagrams are essentially the same because we have coded the prototype like it should have been. We will not make any major changes to this architecture moving forward.

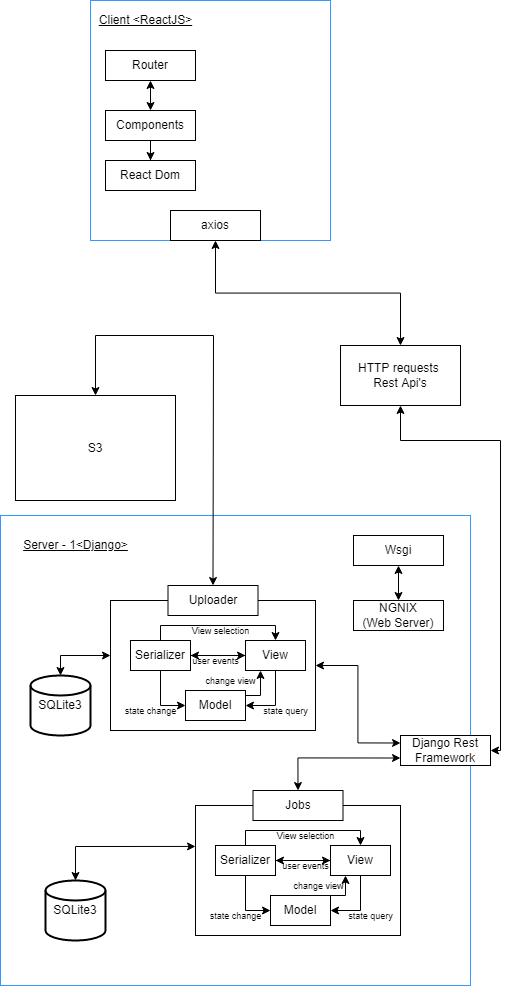
Since we are using different Django apps and these apps are completely independent modules, it makes it easier for us to detect any bugs or issues. This helps in easy maintenance of our code. We are using react components instead of repeating the code again along with interfaces which aids in reusing our code.

Plus our system on a whole or a part of it can be easily extended easily because it is divided into different independent modules.

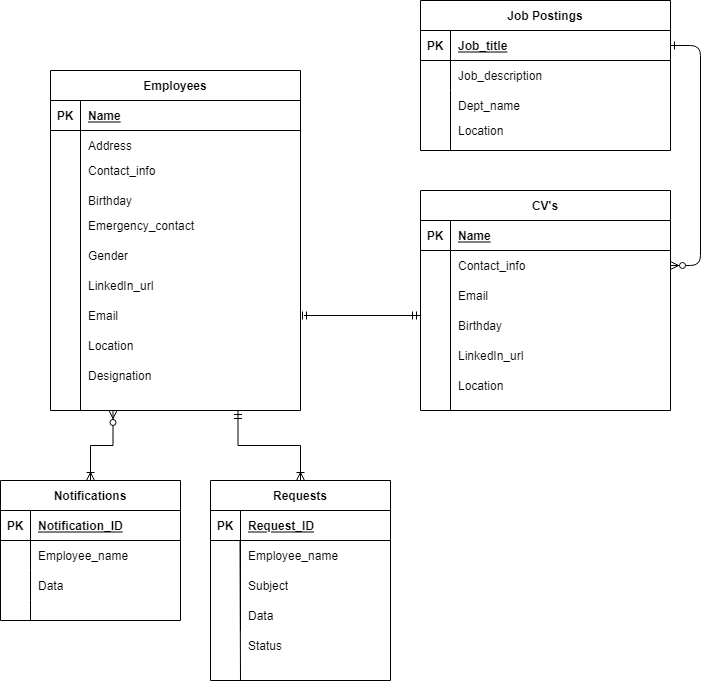
Lastly, our system is separated into distinct sections i.e each page of our system has separate functions. Plus, as mentioned above our system is divided into modules according to functionality so our system is a well-organized system and each part fulfills a meaningful and intuitive role while maximizing its ability to adapt to change.

The uploader module is only concerned with the CV’s that are uploaded. It stores all the uploaded CVs in an s3 instance. The rest of the job application form data is stored in a SQLite database. The Jobs module is used to post jobs. It stores all the data of the job posting in a different SQLite database.

Both these modules communicate with the frontend using Django Rest framework, HTTP requests and Rest API’s



# Data Models



All the three tables shown in this ER Diagram are from 3 different SQLite databases. The Employees table is part of the AWS server(the 2nd server). The Job postings and CV’s tables are part of the Django server.Since these tables are in different databases so they will be connected to each other through HTTP requests, Rest APIs and django rest framework.

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.

The Notifications and Requests tables are part of DynamoDB. They will store all current and previous notifications and requests. The status attribute in the requests table is for differentiating between accepted and rejected requests.

# Tools and Technologies

* Backend: Django 3.8.2 or 4.0
* Frontend: React 17.0
* Databases: Amazon DynamoDB orAmazon DocumentDB 3.6
* Chatbot services: AWS Lex V2
* Servers/Serverless: AWS EC2, AWS lambda

# Who Did What?

|  |  |
| --- | --- |
| **Name of the Team Member** | **Tasks done** |
| Aamina Mariam | Section 2, Section 3 |
| Javeria Tariq | Section 2, Section 4 |
| Ali Adnan Arif | Section 2 |
| Mohammad Yousuf | Section 2 |
| Talha Nasir | Section 1, Section 2 |

# Review checklist

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

|  |  |
| --- | --- |
| **Section** **Title** | **Reviewer Name(s)** |
| Section 1 | Javeria Tariq |
| Section 2 | Aamina Mariam, Mohammad Yousuf |
| Section 3 | Javeria Tariq, Talha Nasir |
| Section 4 | Ali Adnan Arif |