InterviewLink

Software Architecture and Design

An Open Source Project built as a part of the course

CSE583: Software Development Using Open Source

By

Anjanay Raina (2020494)

Heemank Verma (2020064)

Garima Chopra (2020376)

Vishnu Shon (2020414)

Project Sponsor

Karanjot Singh (Mathworks)

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Introduction

The interview scheduling platform is an application designed to streamline and optimize the entire interview process for organizations. Leveraging a Microservices Architecture, this platform employs a modular approach, where distinct services handle specific functionalities, allowing for flexibility, scalability, and independent development.

It comprises three primary servers - a Next.js client for the presentation layer, a Django server for scheduling and resume analysis, and a PyBot server dedicated to the Python chatbot providing a conversational interface for candidates. The platform is empowered by a Neo4j database, ideal for handling interconnected data efficiently

Additional Google Cloud integrations with API for Google OAuth (for authentication), Google Calendar (for scheduling), Google Mail (for notification services) and Google Sheets (for managing data) will be used to provide seamless integration that enhances HR capabilities, enabling them to export and track data seamlessly. Integration with a CRM system facilitates efficient feedback collection. Node Mailer is employed for email communication, ensuring timely updates for all stakeholders.

Technology Stack

2.1 Database

Neo4j: Neo4j is a graph database that excels at managing highly connected data. Neo4j, in combination with the GraphQL plugin, allows for a more intuitive and natural schema definition.

GraphQl Integration: Utilizing the Neo4j GraphQL Plugin. This integration enables the use of GraphQL as the query language for interacting with the Neo4j database

2.2 Website Backend

Next.js: Next.js is a popular React framework that provides server-side rendering, allowing for efficient and SEO-friendly rendering of React components on the server side.

Apollo Server SDK: Apollo Server is a GraphQL server implementation that seamlessly integrates with various JavaScript frameworks, including Next.js.

2.3 Scheduler/OCR Server

Django and Libraries: Django offers a clean and pragmatic design, allowing for rapid development and deployment of web applications.

2.4 Chat Bot Server

Python Telegram Bot API: The Python Telegram Bot API is a library that allows for easy interaction with the Telegram Bot API. It provides a convenient way to develop and manage Telegram bots using Python.

Google Cloud NLP API: The powerful pre-trained models of the Natural Language API empower developers to easily apply natural language understanding (NLU) to their applications.

2.5 Frontend

Next.js: Next.js is a React framework that facilitates server-side rendering (SSR), static site generation (SSG), and client-side rendering.

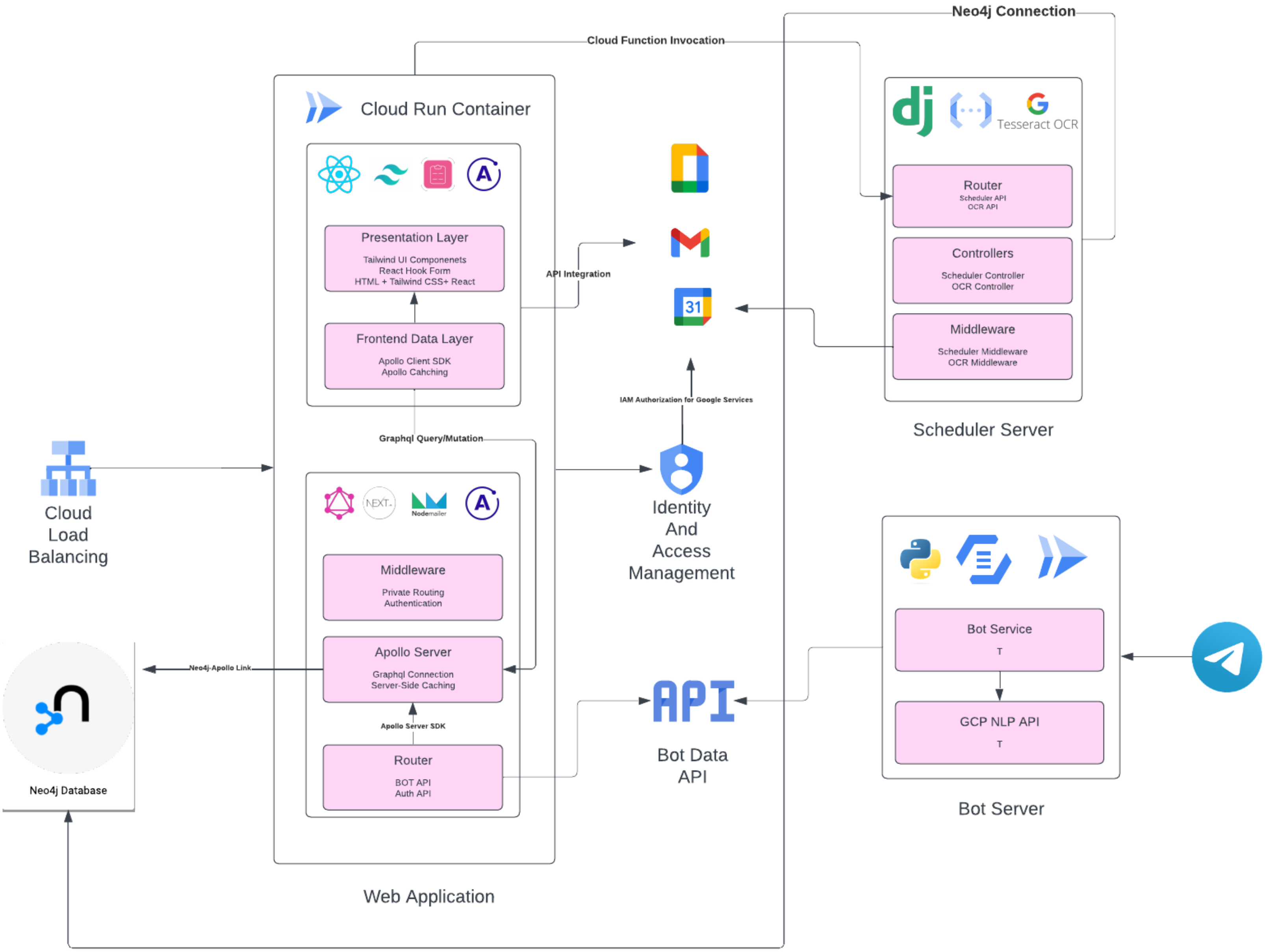
Apollo Client SDK: Apollo Client is a comprehensive state management library designed for managing GraphQL data in a front-end application.

Tailwind UI: Tailwind UI Components is a curated collection of pre-designed, responsive, and customizable user interface elements built on top of the popular Tailwind CSS framework.

High-Level Architecture

The interview scheduling platform is designed as a distributed system, composed of various interconnected components that work together seamlessly to optimize the interview process. The architecture is divided into several key layers, each responsible for specific functionalities:

3.1 Architecture Diagram



Database Design

4.1 Database

Neo4j

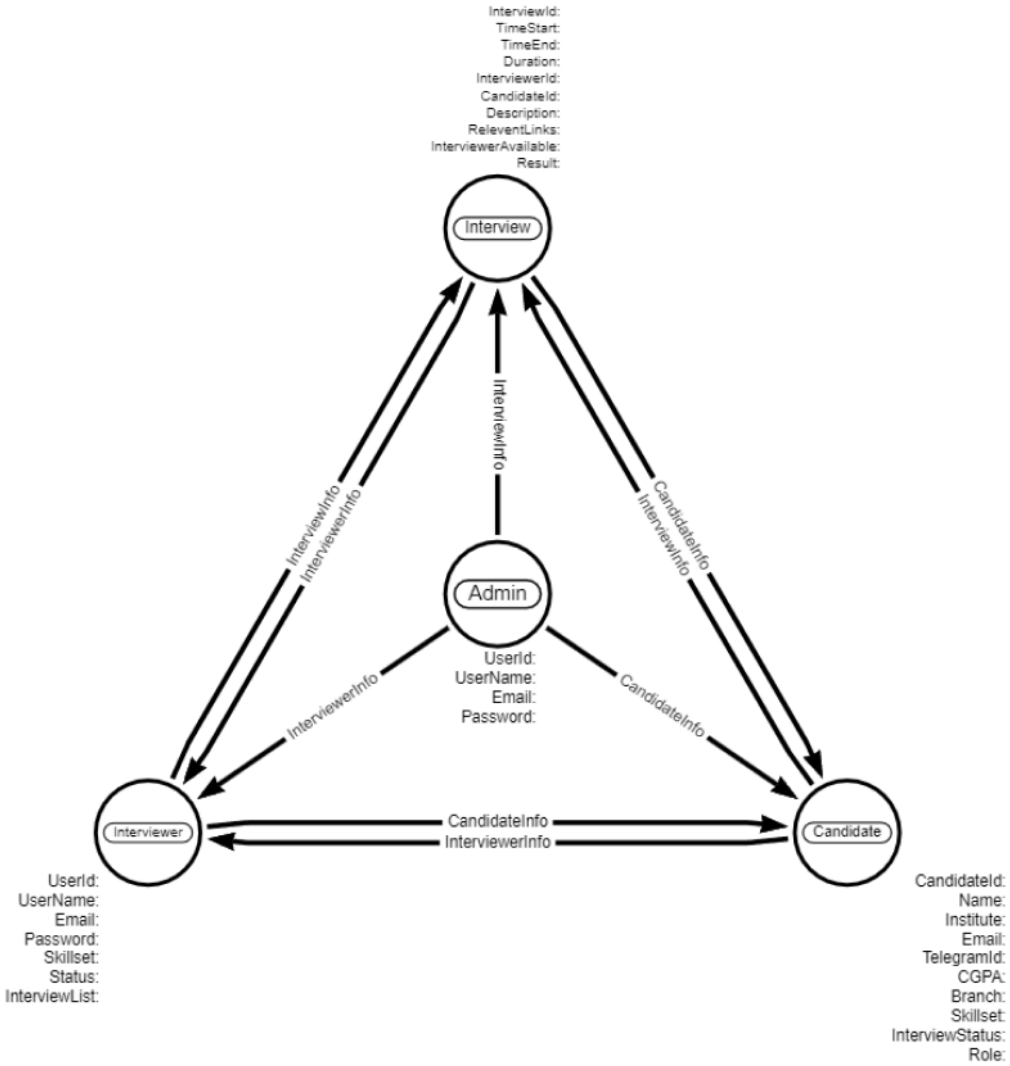
- Neo4j is a graph database that excels at managing highly connected data. Given the nature of interview scheduling, which involves complex relationships between candidates, feedback, interviewers, and scheduling details, a graph database is an excellent choice. It allows for efficient traversal of relationships, making it well-suited for this application.
- Neo4j, in combination with the GraphQL plugin, allows for a more intuitive and natural schema definition. This means that the data model can be designed in a way that closely mirrors the actual relationships and entities involved in interview scheduling
- Neo4j provides robust authorization features, including field-level authorization. This means that access to specific data fields can be tightly controlled, ensuring that only authorized users can view or modify sensitive information.

GraphQl Integration

- Utilizing the Neo4j GraphQL Plugin. This integration enables the use of GraphQL as the query language for interacting with the Neo4j database. It provides a more flexible and expressive way to retrieve and manipulate data.
- GraphQL is particularly advantageous when dealing with highly nested data structures, as is often the case in interview scheduling. It allows for precise and efficient retrieval of only the required information, avoiding over-fetching or

under-fetching of data. This capability ensures optimal performance and responsiveness in managing complex interview-related relationships.

4.2 Schema



Web Application Design (Client+Server)

4.1 Framework and Libraries

Next.js

- Next.js is a powerful React framework that offers server-side rendering (SSR) and static site generation (SSG). It's chosen as the primary technology for both the client and the backend due to its efficiency in rendering and SEO-friendly capabilities. The client-side benefits from dynamic, interactive components, while the backend takes advantage of SSR for fast page loads and better SEO.

Tailwind CSS and UI Component Library

- Tailwind UI Components offers a curated collection of pre-designed, responsive, and customizable user interface elements. These components are integrated into the platform to accelerate development and ensure a consistent design language across the application. They are highly customizable, allowing for seamless alignment with the project's specific requirements.

React Hook Form (User Input)

- React Hook Form is employed for managing user input within forms. It offers a straightforward and efficient way to handle form validation, submission, and data management. This library enhances the user experience by ensuring smooth and error-free input interactions.

Node Mailer (Email Notifications)

- Node Mailer is integrated for sending email notifications to stakeholders. It provides a straightforward way to automate and manage email communication, ensuring timely updates and notifications for candidates, interviewers, and administrators.

4.2 State Management

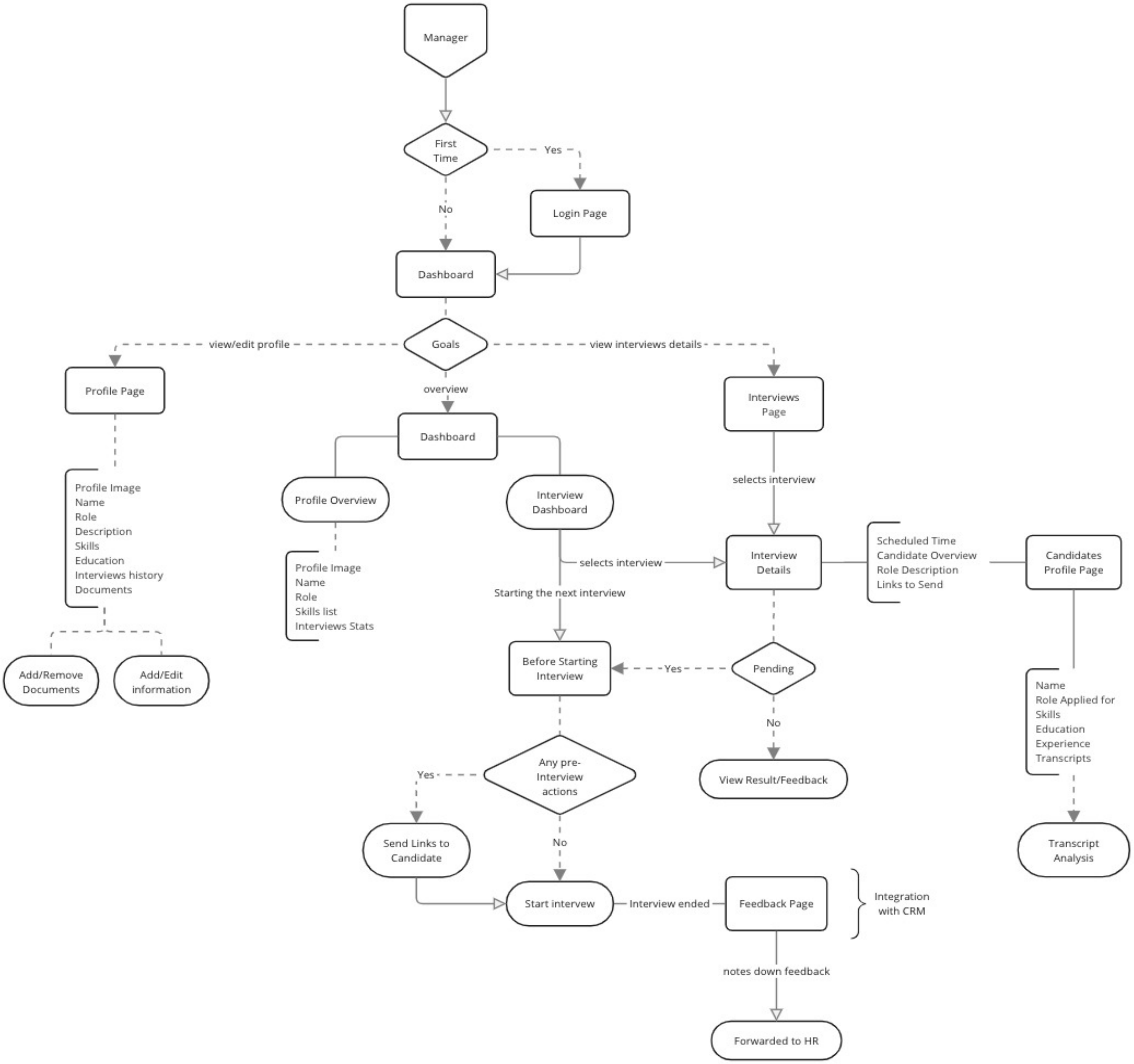
Apollo Client
(Querying Data, State Management, and Caching)

- Apollo Client is a comprehensive state management library designed specifically for GraphQL. It facilitates data fetching and management in the front end.

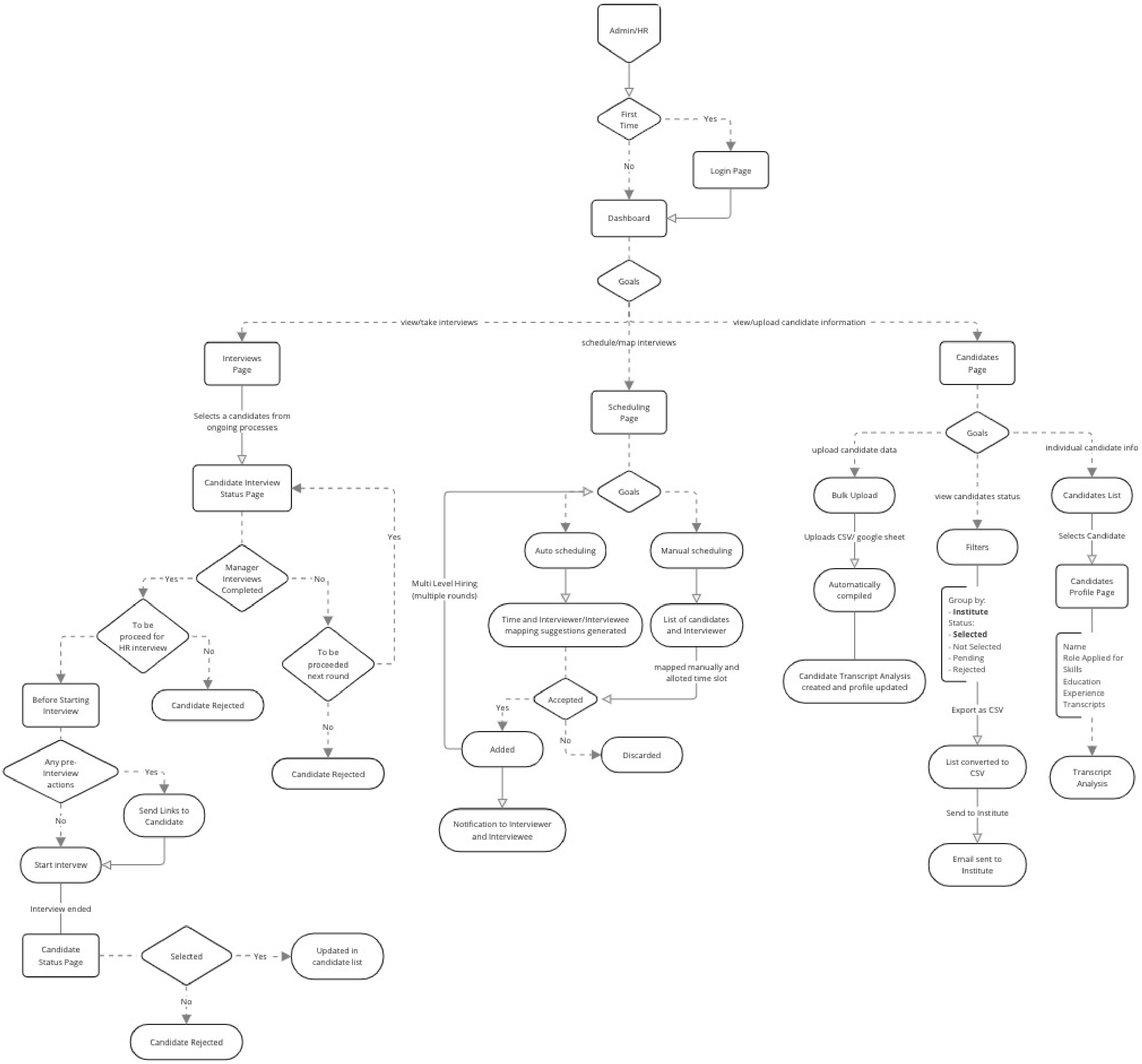
Apollo Client also handles caching, reducing redundant network requests and enhancing performance. It provides a unified interface for querying, caching, and updating data from the GraphQL API.

4.3 User Flow Diagram

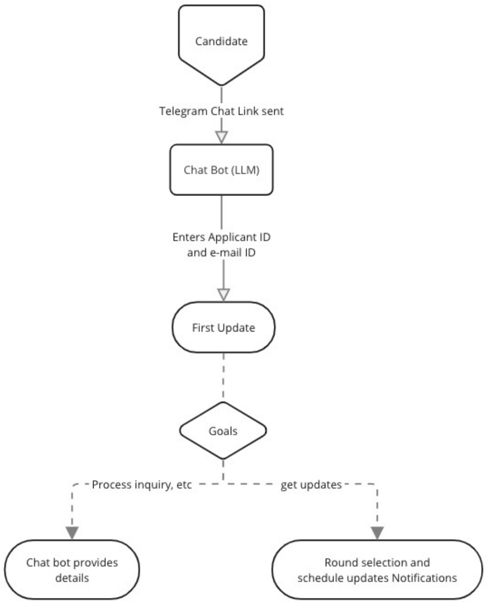
➔ Manager /Interviewer Userflow



➔ Admin/HR Userflow



- ➔ Candidate Userflow


Candidate
Telegram Chat Link sent
Chat Bot (LLM)
Enters Applicant ID and e-mail ID
First Update
Goals
Process inquiry, etc
get updates
Chat bot provides details
Round selection and schedule updates Notifications

4.4 Database Connection

Apollo Server

- ➔ Apollo Server serves as the bridge between the front end and the Neo4j database. It's a GraphQL server that handles queries and mutations from the client, enabling efficient communication with the database. This layer ensures that data retrieval and manipulation are streamlined and optimized.

Authentication from Google IAM

- ➔ Google OAuth is implemented for secure user authentication. It allows users to log in using their Google credentials, adding an extra layer of security to the platform. This integration ensures that only authorized individuals can access specific functionalities, safeguarding user data and login information.

4.5 Integrations

Integration with Google Services (Gmail, Google Sheets, Google Calendar)

- ➔ The platform integrates with various Google services to enhance functionality. This includes Gmail for email communication, Google Sheets for data export and tracking, and Google Calendar for scheduling and managing interviews. These integrations streamline administrative tasks and provide a seamless experience for users.

Scheduler Server Design

4.1 Technology Used

Backend Framework (Django)

➔ Django is a high-level Python web framework known for its simplicity, scalability, and robustness. It is chosen as the backend framework for the Scheduler Server due to its extensive built-in features and the efficiency it brings to web development.

OCR Library (Tesseract OCR)

➔ Tesseract OCR is a powerful open-source library for Optical Character Recognition (OCR). It's employed to extract text data from resumes, allowing for the creation of candidate profiles.

4.2 Component Library

Neo4j Python driver

➔ The Neo4j Django driver is utilized to connect Django with the Neo4j database. This driver enables seamless interaction between Django models and the Neo4j graph database.

Google Calendar API (for Scheduler API)

➔ The Google Calendar API is integrated to access the availability of interviewers and schedule interviews based on their calendar data.

4.3 Service Description

Scheduler API

➔ This API is responsible for scheduling interviews based on the availability of interviewers retrieved from their Google Calendar.

➔ Inputs: Candidate Data, Time Space for Scheduling

➔ Interviewer Availability and Details (retrieved from Google Calendar API)

OCR API

➔ This API uses Tesseract OCR to extract text data from resumes. This data is then used to create candidate profiles.

➔ Inputs: Resume (File /PDF URL)

Chatbot Server Design

4.1 Technology Use

Backend Framework (Python Telegram)

➔ The Python Telegram Bot API is employed to develop the chatbot server. This framework provides a straightforward way to integrate Telegram's bot functionality into Python applications. It's chosen for its ease of use and robust features for creating interactive chatbots.

OCR Library (Tesseract OCR)

➔ The Google Cloud Natural Language Processing (NLP) service is integrated to enhance the chatbot's communication capabilities. This service provides advanced text analysis, including sentiment analysis, entity recognition, and syntax analysis, enabling the chatbot to process and understand user messages effectively.

4.3 Service Description

Backend Connection

➔ The chatbot is connected with the database through next app, utilising BOT api from the web application server to get the data flow.

Candidate Interaction

➔ The chatbot interacts with candidates, offering updates on interview schedules, providing relevant information about the interview process, and addressing any queries or concerns they may have. Leveraging Google's NLP capabilities, the chatbot is equipped to understand and process user messages, ensuring accurate responses and seamless interactions.