InterviewLink Software Requirements

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**

Jira Epic	https://interviewlink.atlassian.net/jira/software/projects/KAN/boards/1/timeline?shared=&atlOrigin=eyJpIjoiODM0ZmE1OTM4NmVmNDYxNjkwNjJjNjJjZWQ4NmU3M2UiLCJwIjoiaiJ9
Notion	https://visshon.notion.site/Software-Requirements-ba86441f894b4077bb600eac7c77ca72?pvs=4
Delivery Date	Nov 15, 2023
Sponsor	Karanjot Singh (Mathworks)
Team Members	Anjanay Raina Garima Chopra Heemank Verma Vishnu Shon
Relevant Documents	https://www.figma.com/file/HxgnT9m5UkZopLloPqCWAE/InterviewLink?type=design&node-id=25-563&mode=design&t=rQRj2mlcknDzrDWV-0

Table of Contents

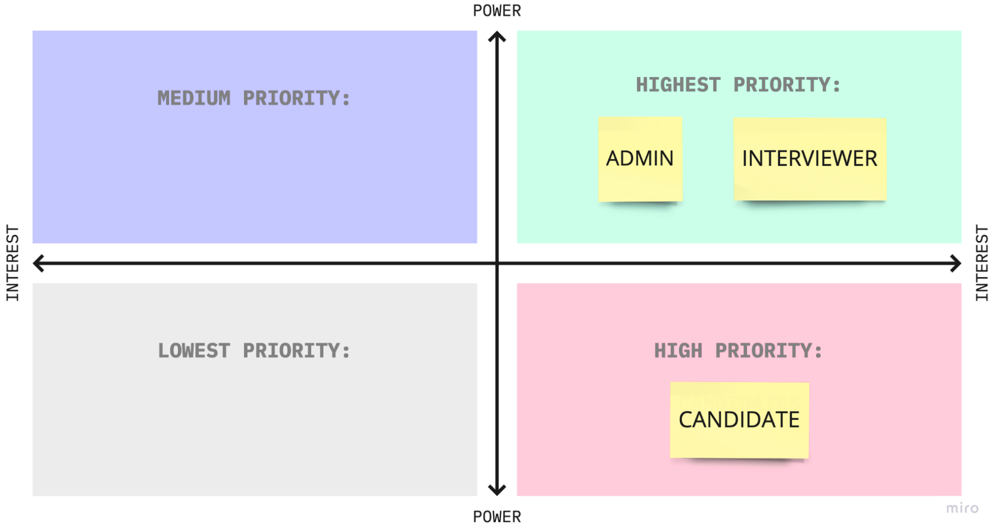
Introduction

1.1 Problem Statement

Creating a comprehensive interview management system that empowers administrators to efficiently schedule interviews, while providing interviewers with tools for effective facilitation and feedback collection, ultimately enhancing the candidate experience and decision-making process

1.2 Stakeholders

The user-friendly interface should be designed and developed with the needs of these stakeholders in mind, ensuring that it is easy to use and provides all the necessary features and functionalities.



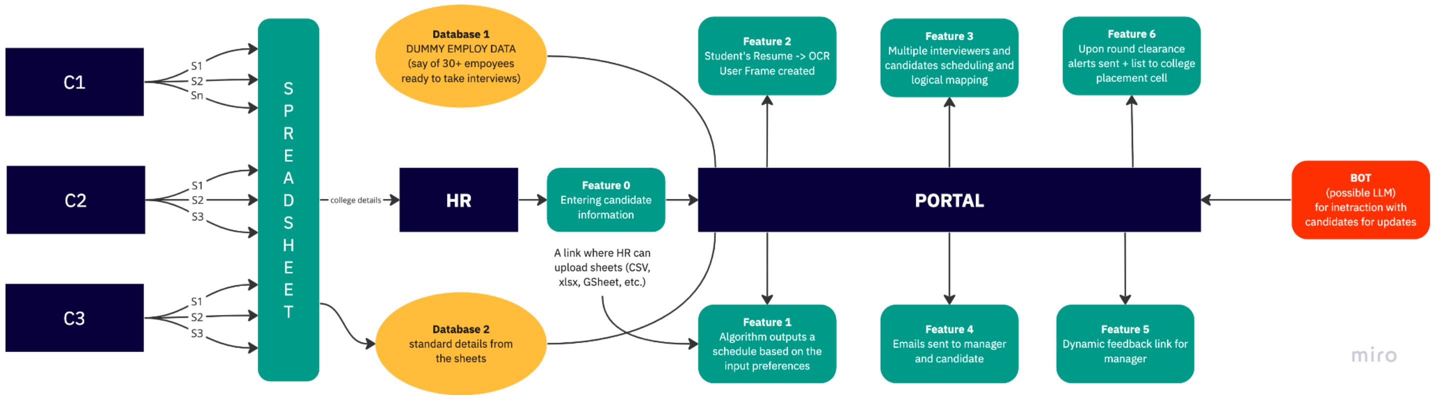
- Admins: The primary users of the system who manage the interview process, schedule interviews, manage feedback, and make the final decision on candidates. They are responsible for scheduling interviews and need to ensure that the interview process runs smoothly.
- Interviewers: Users who facilitate interviews and provide feedback on candidates.
- Candidates: Users who apply for jobs and participate in the interview process. They need to be able to check the status of their application.

Functional Requirements

2.1 Functional Goals

The following functionality goals are expected from the software:

- Admins can manage multi-level interviews. Minimize wait times for the interview using scheduling based on manager skills and time slot. Process candidate's applications. Admin can schedule interviews for an interviewer and candidates.
- Interviewers can facilitate interviews. Provide a communication layer between interviewers and candidates. Provide Admin tools to help make the final decision. Provide a mechanism for collecting interviewer feedback and analyzing
- Candidates can get an update on the status of their interview.


C1
S1
S2
Sn
C2
S1
S2
S3
C3
S1
S2
S3
SPREADSHEET
college details
Database 1
DUMMY EMPLOY DATA
(say of 30+ empoyees
ready to take interviews)
HR
Feature 0
Entering candidate information
A link where HR can upload sheets (CSV, xlsx, GSheet, etc.)
Database 2
standard details from the sheets
PORTAL
Feature 2
Student's Resume -> OCR
User Frame created
Feature 3
Multiple interviewers and candidates scheduling and logical mapping
Feature 6
Upon round clearance alerts sent + list to college placement cell
Feature 1
Algorithm outputs a schedule based on the input preferences
Feature 4
Emails sent to manager and candidate
Feature 5
Dynamic feedback link for manager
BOT
(possible LLM)
for inetraction with candidates for updates
miro

2.2 Success Metrics

Title	User Story	Actor	Priority
Auto Scheduling	*Admins can schedule multiple interviews with a single button click.*	*Admins*	High
Multi-Level Hiring Process	*The company can take multiple interviews of a single candidate with automatic multi-level scheduling.*	*Interviewers /Candidates*	*High*

Bulk Candidate Upload	*Admins can upload candidate information by importing a CSV.*	*Admins*	*Medium-low*
Interview Dashboard	*Interviewers can facilitate interviews and share feedback on the candidate with the admin.*	*Interviewers /Admin*	*High*
Integration with CRM	*Interviewers and Candidates can provide feedback for the interview process.*	*Interviewers /Candidates*	*Low*
Google Sheets Export	*Admins can export interview status in a Google sheet format.*	*Admins*	*Medium-low*
Google Calendar Integration	*The platform can fetch time slots from the Google calendar of the interviewers.*	*Interviewers*	*Medium*
Notification and Malining System	*Interviewers and Candidates can receive emails and SMS notifications about the status and timings of the interview.*	*Interviewers /Candidates,*	*Medium*
Candidate NLP Bot	*Candidates can get updates on the status of their applications and information about the company from the chatbot.*	*Candidates*	*Medium-Low*
Test Implementation	*Implementing UI and Unit Testing for the platform.*	*Interviewers /Candidates/ Admin*	*Low*

Non-Functional Requirements

3.1 Tech Stack

Scheduler Platform

- The backend of our scheduler platform will be powered by Apollo GraphQL Server.
- Scheduler The backend of our live scheduler platform will be built using node js
- Frontend Next.js serves as the frontend framework, providing server-side rendering for enhanced performance and SEO optimization.
- Query GraphQL serves as our query language, enabling precise data retrieval from the server. Its flexibility allows clients to request only the information they need.
- Database Neo4j, our chosen database, excels in handling highly connected data, making it an ideal choice for managing interview-related information.

Chatbot

- Our chatbot is built using the pyTelegramBotAPI framework.
- Database Neo4j serves as the database that allows for personalized and context-aware responses, enhancing the user experience.
- Platforms Google Colab provides a cloud-based environment for developing and running Python code.

Others

- Our cache server will be built using Node.js, which optimizes performance by temporarily storing frequently accessed data.
- Cache store Redis DB serves as the cache store, Its in-memory data structure store supports various data types, making it highly effective for caching purposes.
- Testing framework Cypress is our chosen testing framework, and provides a comprehensive suite of tools for writing, running, and debugging tests.
- Hosting and load balancing Google Cloud Suite provides a reliable and scalable infrastructure for hosting our platform.
- CRM SugarCRM serves as our Customer Relationship Management system, enabling efficient management of interactions with candidates, interviewers, and administrators.

3.2 Design Strategy

- Given our adherence to MathWorks' Parula design system for developing this internal tool, our strategy is to seamlessly integrate its core principles while allowing for necessary deviations to optimize the user experience.
- By aligning with the Parula design, we aim to maintain a consistent and familiar interface for MathWorks employees, ensuring an intuitive navigation experience. This includes leveraging standardized colour schemes, typography, and layout guidelines that reflect the MathWorks brand identity.
- In enhancing the Parula design system, we plan to focus on user-centric features, such as an intuitive interview scheduling interface, candidate management dashboard, and a streamlined communication channel.
- While we will adhere to MathWorks' design standards, we may introduce slight modifications to optimize task flows and facilitate seamless interactions

3.3 Performance Requirements

The system will be utilized as an internal tool for MathWorks so the performance is expected to be optimum, with a response time within a **maximum of 4 seconds** for critical operations during load testing.

- At peak periods, the system must be able to accommodate up to **120 interviewers**, **50 administrators**, and up to **300 candidates** each day.
- During peak usage, the system should be able to support at least **200 concurrent sessions.**
- Users from **East Asia**, **West Europe**, the **United States**, **Japan**, and **Australia** must be served adequately by the system because these are the critical locations for MathWorks operations.
- Because we are using **Google Cloud Suite** to host our project, It will be capable of operating up to 10 instances concurrently to meet peak traffic.
- **Leveraging Cloud Run** will give us the flexibility to run multiple container instances simultaneously, allowing us to dynamically adjust resources based on demand.
- **Cache servers** play a pivotal role in optimizing response times by storing frequently accessed data, reducing the strain on our backend systems.
- **Content Delivery Networks (CDNs)** enable us to distribute content globally, ensuring swift access for users across different geographic locations.

The goal is to assess how the system responds to this load, particularly in terms of response times for critical operations.

- The Testing approach followed involves simulating real-world conditions, with up to 300 concurrent users, including 120 interviewers, 50 administrators, and 300 candidates accessing the system during peak hours.
- We'll closely monitor resource utilization, including CPU, memory, and network usage, to ensure that the system operates within efficient thresholds.

3.4 Security Requirements

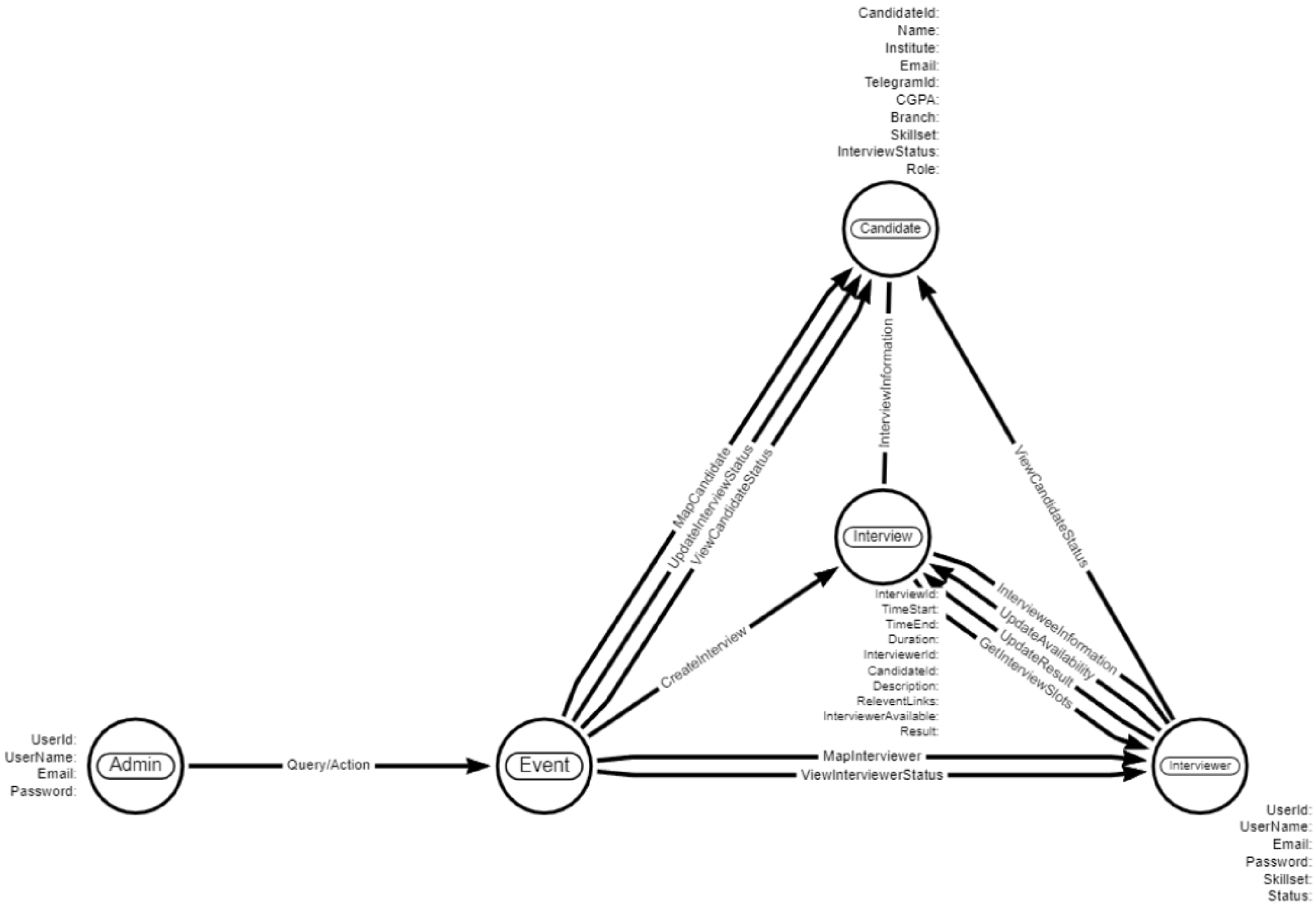
- Employing a multi-layered approach to authentication and access control. We implement JWT (JSON Web Token) authentication methods to securely transmit information between the client and server, further fortifying our verification process. This token-based system ensures that only authenticated and authorized users can access the platform's functionalities.
- GraphQL schema incorporates field-level authorization mechanisms. This fine-grained control enables us to restrict access to specific data fields, ensuring that sensitive information remains strictly accessible only to those with the appropriate permissions. This robust authorization framework is complemented by data write-only rules, guaranteeing that users can only modify data within their designated scope.
- The Neo4j database will store data in an encrypted form while it is at rest. This extra security measure ensures that the data will remain incomprehensible even if unauthorized access is obtained.

3.5 Legal Requirements

- In compliance with legal standards, MathWorks maintains ownership of the intellectual property associated with this scheduling platform. This encompasses any proprietary code, algorithms, and unique features developed specifically for the platform. Additionally, the licenses associated with any third-party libraries or open-source components used will apply. Strict adherence to copyright laws and licensing agreements will be enforced to safeguard MathWorks' intellectual property.

- All data collected and processed by the scheduling platform will be handled in compliance with relevant data protection laws, including GDPR and other applicable regulations. This entails obtaining explicit user consent for data collection, storage, and processing. Personal data will be safely stored, with only authorized persons having access. In addition, safeguards will be put in place to stop data breaches, unauthorized access, and data loss.

Further Requirements



4.1 Use cases

<u>Candidate applications bulk upload</u>

- Actor: HR (Admin)
- Description: The HR can upload a bunch of applications provided in a particular format from CSV to our database through a dashboard.
- Entry condition
  - The HR must log into the platform.
  - The CSV must be in a certain format
- Event flow

    - The HR gets data of applications for universities.
    - He/She sanitizes the data of applicants into the desired format.
    - Bulk uploads the data into our software
- Exceptional requirements
    - Remove specific applicants after bulk upload

User frame creation

- Actor: HR (Admin)
- Description: The HR can create a user frame which contains his skills and a basic overview from the resume and information of the applications.
- Entry condition
    - The HR must log into the platform.
    - The applications must be finalized
- Event flow
    - The HR finalizes the applicants and reviews their data.
    - The applications are analyzed and user frames are created.
- Exceptional requirements
    - Remove specific applicants

Auto scheduling (Live Scheduling)

- Actor: HR (Admin)
- Description: The HR can facilitate auto-scheduling of the applications with the managers based on available time slots and matching skills.
- Entry condition
    - The HR must log into the platform.
    - There must be at least one application.
    - There must be at least 1 free slot per manager.
- Event flow
    - The HR uploads the application information and sets constraints.
    - The scheduling server gives optimal scheduling suggestions based on available slots.
    - HR gives the final green light to the interview.
- Exceptional requirements
    - Change particular scheduled interviews.

Upcoming interviews and candidate profiles panel

- Actor: Managers (interviewers)
- Description: The Managers can see their upcoming or scheduled interviews and view candidate profiles.

- Entry condition
    - The manager must be logged in.
    - There should be at least one scheduled interview.
- Event flow
    - The manager can see the interviews that are scheduled for him and select an interview to begin.
    - The manager can see the candidate's profile for the next interview.
- Exceptional requirements
    - Change particular scheduled interviews.

Send link to participant

- Actor: Managers (interviewers)
- Description: The Managers (interviewers) can share meet link with the participant for an interview and share additional test links
- Entry condition
    - The manager must log into the platform.
- Event flow
    - The manager selects the candidate for the interview.
    - They send the meet link to the candidate by clicking the button.
    - They share additional links (for tests etc.) with the candidate through a chatbot channel.
- Exceptional requirements
    - The candidate is logged in with the chatbot

Provide feedback for the participant

- Actor: Managers (interviewers)
- Description: The Managers (interviewers) can provide feedback on the whole interview through this side tool to help HR in making final decisions.
- Entry condition
    - The manager must log into the platform.
    - The interview must be going on
- Event flow
    - The manager shares the meet link with the candidate through the platform.
    - They carry on with the interview and provide feedback on the candidate through our side portal
- Exceptional requirements
    - none

Verdict portal

- Actor: HR (Admin)
- Description: The HR can make a final decision on whether the candidate moves forward with the interview or not.
- Entry condition
    - The HR must log into the platform.
    - The feedback has been provided by the interviewer
- Event flow
    - The manager provides the feedback for the candidate.
    - The feedback is visible to the HR through the portal.
    - The HR selects if the candidate goes forward or not.
- Exceptional requirements
    - If the candidate was in the final round then the candidate must be informed of selection.
    - If a candidate is rejected they must be informed of the rejection.
    - Otherwise the candidate will be informed of the next round of details

Google Sheet exports

- Actor: HR (admins)
- Description: The HR can export candidate status details in a sheets format.
- Entry condition
    - The manager must log into the platform.
- Event flow
    - The HR clicks the export icon from the dashboard.
    - A sheet format is exported which is mailed to HR.
- Exceptional requirements
    - none

Chatbot

- Actor: Candidates
- Description: They can get the status of their interview and updates using the telegram chatbot.
- Entry condition
    - The candidate must log into telegram.
- Event flow
    - The candidates can ask questions in natural language to get the status of their interviews.
- Exceptional requirements
    - The candidate wants to ask a specific question to HR

Next round allocation

- Actor: HR (Admin)
- Description: The HR can facilitate auto-scheduling of the applications for the next round of interview process
- Entry condition
  - The HR must log into the platform.
  - There must be at least one application.
  - There must be at least 1 free slot per manager.
- Event flow
  - The HR provides the final decision for a candidate.
  - The scheduling server gives optimal scheduling suggestions based on available slots for the next round.
  - HR gives the final green light to the interview.
- Exceptional requirements
  - Change particular scheduled interviews.