# OBSS CodeCamp 2023 Java Challengers

# “OBSS Hire App” Project

## High Level Design Report

İsmail Emre Deniz

08.08.2023

# 1. Introduction

The OBSS Hire App stands as a dynamic web application that serves as a bridge between OBSS HR personnel and aspiring job applicants. This sophisticated platform empowers HR professionals to efficiently create, oversee, and manage job openings, streamlining the entire process. Simultaneously, the application equips applicants to seamlessly apply for these opportunities, utilizing the information extracted from their LinkedIn profiles. An additional feature of paramount importance is the ability for HR personnel to meticulously sort and evaluate applicants based on the alignment of their profile and cover letter information with the opening, further enhancing the precision and effectiveness of the recruitment process.

# 2. Design Goals (Non-functional Requirements)

The primary design objectives of the OBSS Hire App encompass both Usability and Efficiency domains.

## a. Usability

The central aim of the OBSS Hire App is to deliver an intuitive and seamless user experience, enabling users to harness the system's capabilities effortlessly. Beyond meeting fundamental expectations, the OBSS Hire App introduces two key features that significantly amplify usability:

- Streamlined LinkedIn Integration: Applicants, constituting one of the two user categories within the system, can conveniently access the platform using their LinkedIn credentials through LinkedIn's established authentication process. By supplying their LinkedIn profile URL, the system autonomously populates their profile information in mere milliseconds. This automation extends to profile updates – a single button click enables users to synchronize any modifications made on their LinkedIn profiles, ensuring their profiles remain up to date.

- Empowered HR Personnel: Catering to the second user type, HR personnel are equipped with a powerful free-text search capability. This feature facilitates efficient exploration of applicants who have applied for posted job openings. Search results are intelligently ordered based on the alignment of their profiles and cover letters. HR personnel's initial search triggers a comprehensive scan of applicants' qualifications, consequently presenting the list in order of compatibility with the job opening's prerequisites.

## b. Efficiency

Efficiency stands as a pivotal design tenet of the OBSS Hire App, executed through meticulous selection of tools and frameworks to facilitate seamless functionality.

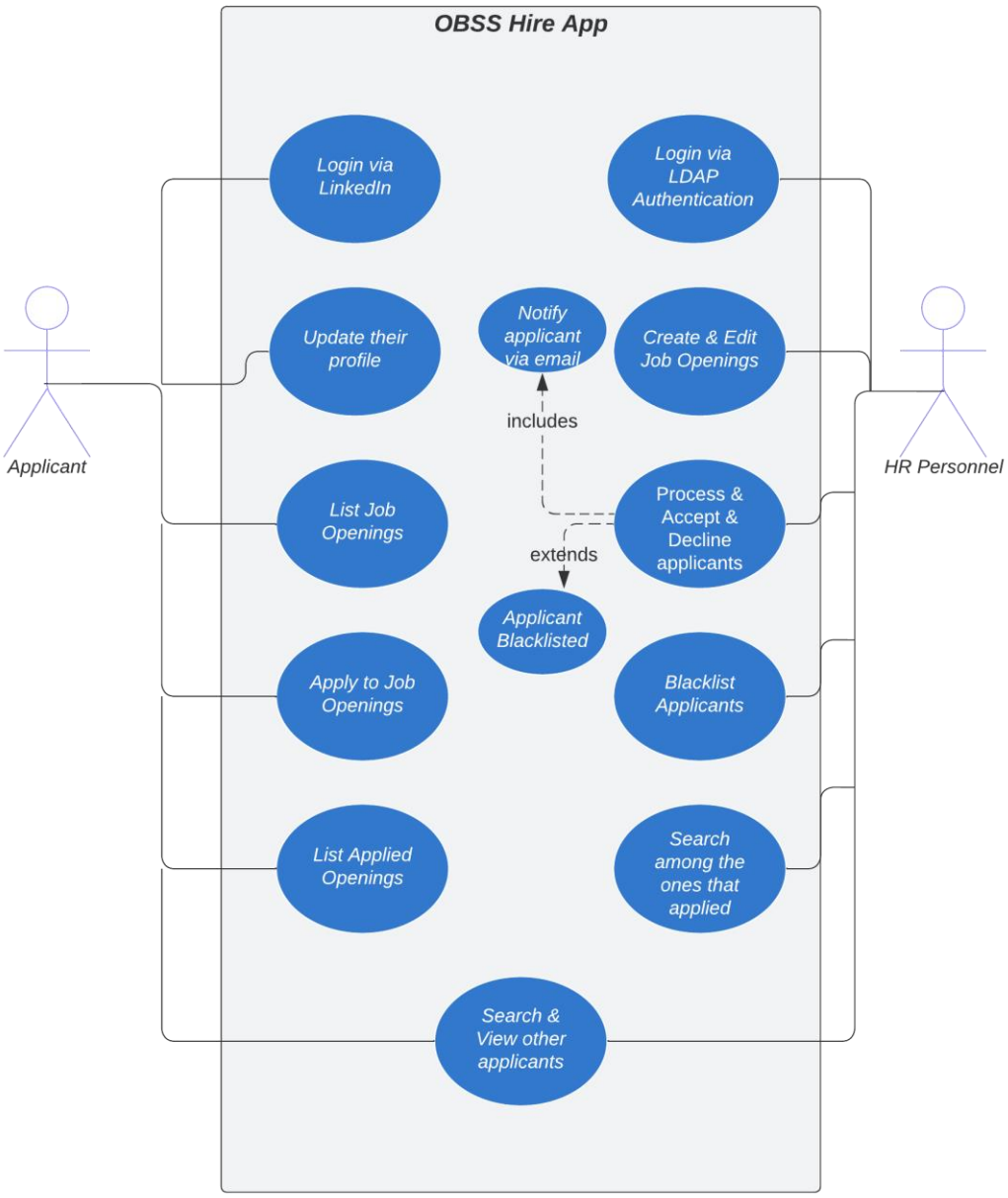
- Optimized LinkedIn Profile Retrieval: The expedient LinkedIn profile retrieval mechanism is accomplished using Selenium in conjunction with HTMLUnitDriver. Unlike ChromeDriver, HTMLUnitDriver eschews JavaScript and operates solely on base HTML, resulting in accelerated and resource-efficient profile scraping.

- Database Management with MongoDB: The pivotal requirement of enabling free-text search is met through the utilization of MongoDB for database management. This choice ensures robust and efficient handling of data for this specific functionality.

- Applicant Compatibility Ranking: To achieve efficient and precise ordering of applicants based on compatibility, MongoDB's potent search queries are harnessed. This approach guarantees swift and intelligently prioritized results, driven by the scoring established within the search index.

By pursuing these Usability and Efficiency objectives, the OBSS Hire App not only enriches user experience but also optimizes performance across its diverse functionalities.

## 3. Use Case Diagram and Explanations




- User Profiles
  - The application supports two user types: HR Personnel and Applicant.
- Authentication
  - HR Specialists must log in to the system using LDAP Authentication.
  - User definitions can be directly managed on the LDAP server.
  - Applicants should log in using their LinkedIn profiles.
- Job Posting Management
  - HR Personnel can create new job postings, activate them, and deactivate active postings.
  - While creating a job posting, HR Specialists can set future dates and times for automatic activation and deactivation.
- Job Application

    - Applicants can apply for job postings using their LinkedIn profiles.
    - Viewing job postings does not require login, but login is necessary during the application process.
    - The system should capture and store detailed LinkedIn profile information of applicants for flexible searches.
- Viewing Job Applications (Applicant)
    - When logged in, applicants can view details and statuses of their previous job applications.
- Viewing Job Applications (HR Specialist)
    - HR Specialists can view job applications for specific postings.
    - HR Specialists can view other job applications from an applicant who applied to a particular posting.
    - HR Specialists can access applicant profile details, process applications (accept or reject).
    - The system sends informative email notifications to applicants upon changes in application status.
- Blacklisting
    - HR Personnel can add an applicant to a blacklist.
- Free Format Text Search
    - HR Personnel can perform free text searches similar to Google search on collected applicant profiles.
    - This search can be conducted among applicants of a specific job posting or across all saved profiles.
- Ordering Applications by Suitability
    - The system must be capable of evaluating the compatibility between information in job applicants' profiles and the qualifications specified in job postings.
    - Applications should be ordered by suitability based on this evaluation.

# 4. Subsystem Decomposition


The OBSS Hire App is structured into four essential subsystem layers, each serving a specific purpose in ensuring the application's smooth operation and user interaction.

- Presentation Layer: This layer is the user's entry point into the OBSS Hire App. It provides user interfaces for both applicants and HR personnel. Applicants interact with the app to submit their job applications, while HR personnel use the interfaces to manage job postings and applications. The presentation layer ensures a user-friendly experience and facilitates easy navigation.

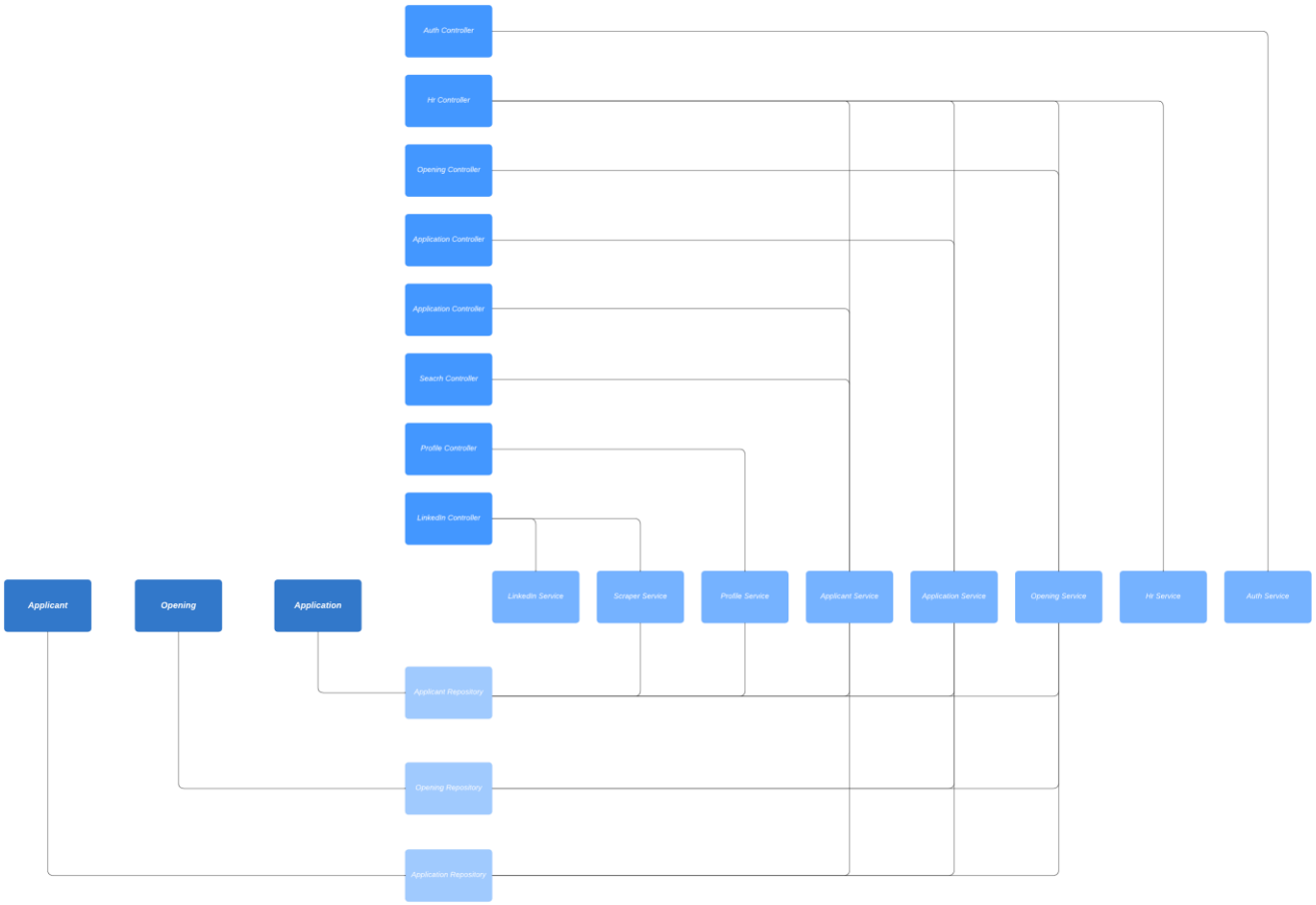
- Application Layer: Beneath the presentation layer, the application layer houses the core logic of the OBSS Hire App. It handles the business processes, such as processing job applications, matching applicants with job requirements, and managing job postings. This layer orchestrates the application's functionality and ensures that user actions are executed correctly.

- Database Abstraction Layer: The database abstraction layer acts as a bridge between the application layer and the database. It provides a standardized way for the application layer to communicate with the underlying database without needing to know the technical details of data storage and retrieval. This layer helps maintain data integrity and provides a separation of concerns between the application's logic and data management.

- Database Layer: At the core of the architecture is the database layer, where all the application's data is stored. This layer includes various collections that hold information such as applicant profiles, job postings, and applications. It is responsible for efficiently storing, organizing, and retrieving data to support the application's functionalities.

Together, these four layers form the backbone of the OBSS Hire App's architecture. They collaborate to deliver a functional, user-friendly, and efficient platform that simplifies job application and management processes for both applicants and HR personnel.

## 5. Final Object Design




- Model Objects and their attributes:
    - Applicant
        - Id: ObjectId
        - Email: String
        - FullName: String
        - LinkedInUrl: String
        - UrlId: String
        - About: String
        - AppliedOpeningIds: List<ObjectId>
        - Blacklisted: Boolean
        - ExperienceList: List<Experience>
        - EducationList: List<Education>
        - CertificationList: List<Certification>
    - Application
        - Id: ObjectId
        - ApplicantId: ObjectId
        - OpeningId: ObjectId
        - Status: ApplicationStatus
        - CoverLetter: String

- Opening
  - Id: ObjectId
  - HrId: ObjectId
  - Title: String
  - Explanation: String
  - Qualificaitons: String
  - Applicants: List<ObjectId>
  - IsActive: Boolean
  - ActivationDate: Date
  - DeactivationDate: Date

# 6. Used Technologies

## 6.1. Database

The foundation of the OBSS Hire App's data management is built upon MongoDB, a NoSQL database. The selection of MongoDB is underpinned by two key considerations. Firstly, the inherent structure of the application's data model lends itself to a non-relational approach, making the utilization of a traditional relational database unnecessarily complex. Secondly, MongoDB inherently supports a vital requirement of the application - free-text search. By harnessing MongoDB's robust search index and query capabilities, efficient searching across collections is seamlessly enabled.

## 6.2. Backend

The backend of the OBSS Hire App is empowered by Java Spring Boot, a framework known for its pragmatic and modular design. The decision to employ Java Spring Boot is informed by multiple factors. Its inherent façade-like architecture simplifies the development process, allowing for swift implementation of complex functionalities. Additionally, Java Spring Boot ensures streamlined integration and compatibility, enabling seamless interactions between various components of the application. The robustness of the framework and its vibrant developer community further solidify its position as the backend technology of choice.

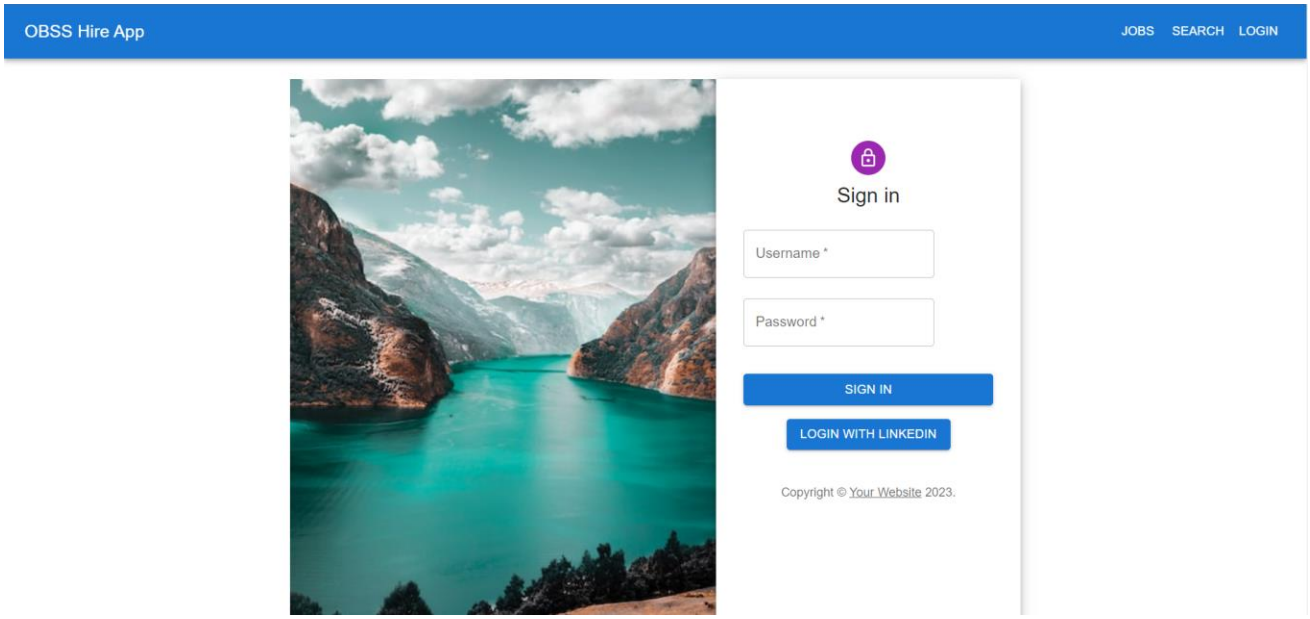
Furthermore, for the LinkedIn Scrape Service, the OBSS Hire App leverages Selenium and HTMLUnitDriver dependencies. These tools are selected for their efficiency in web scraping, enabling rapid and accurate extraction of data from LinkedIn profiles.

## 6.3. Frontend

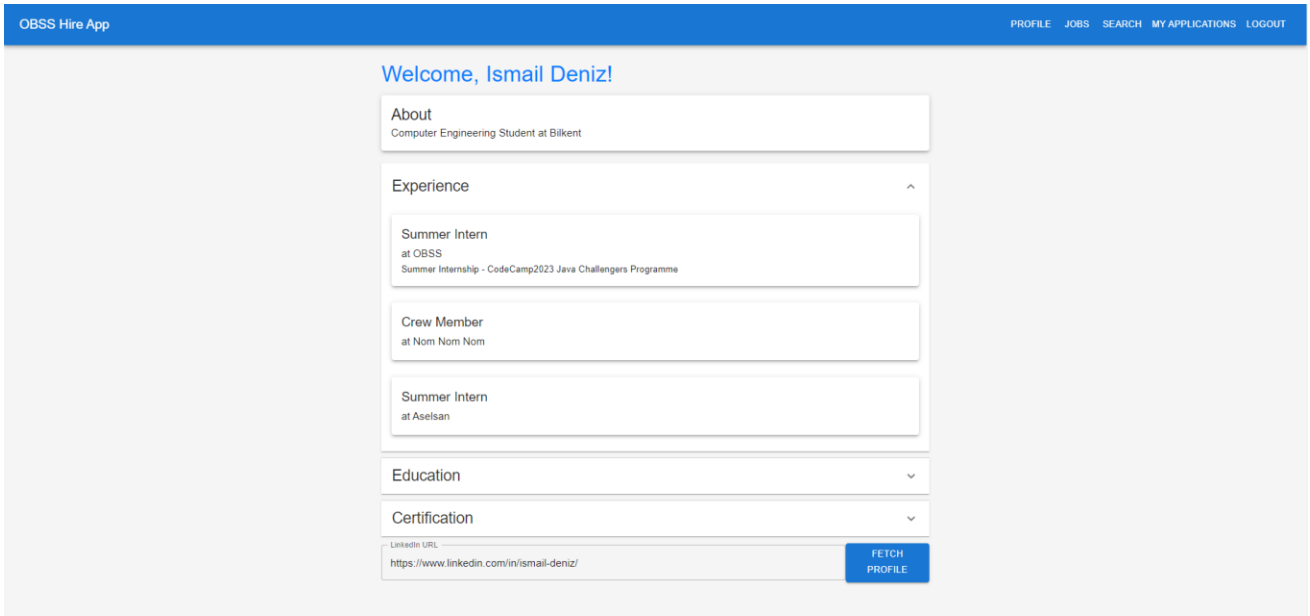
The frontend of the OBSS Hire App is crafted using React.js, a dynamic and popular JavaScript library. The choice of React.js as the frontend technology is guided by several strategic factors. One of the primary motivations is the development team's familiarity with React.js, ensuring smoother implementation and maintenance. React.js also offers exceptional modularity and reusability, facilitating the creation of interactive and responsive user interfaces. Its virtual DOM mechanism enhances rendering speed, leading to a seamless user experience. Overall, React.js aligns with the project's goals of user-centric design and efficient development processes.

# 7. Screenshots from the app

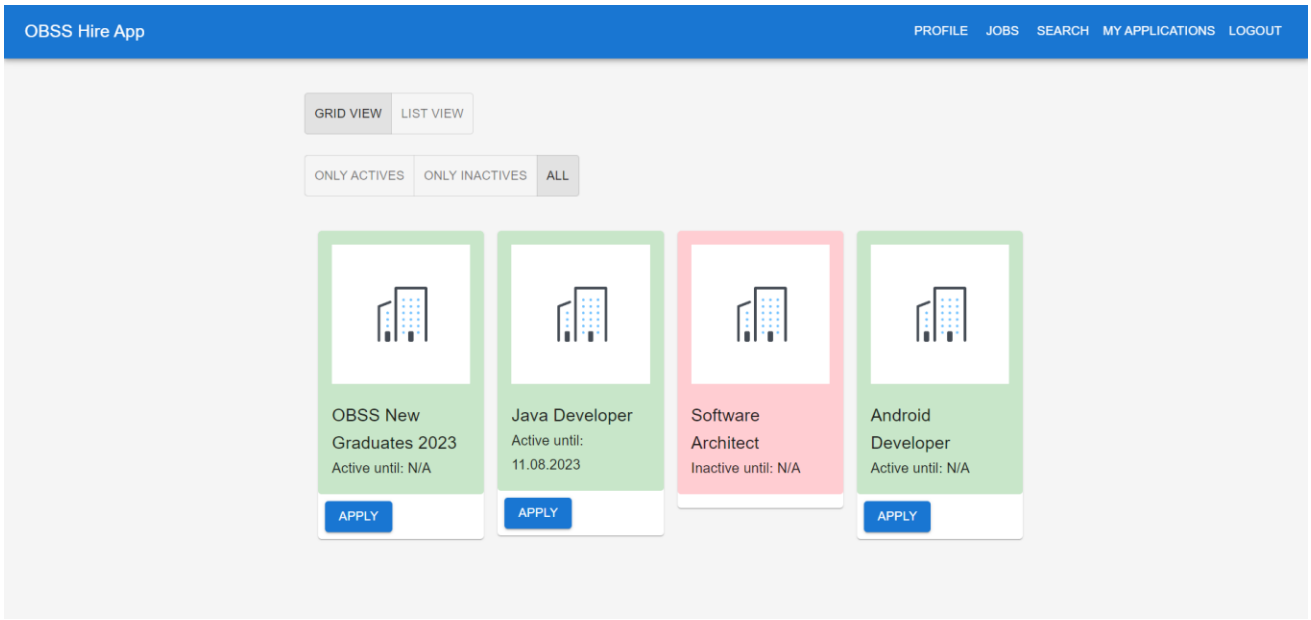
## 7.1 Login Screen



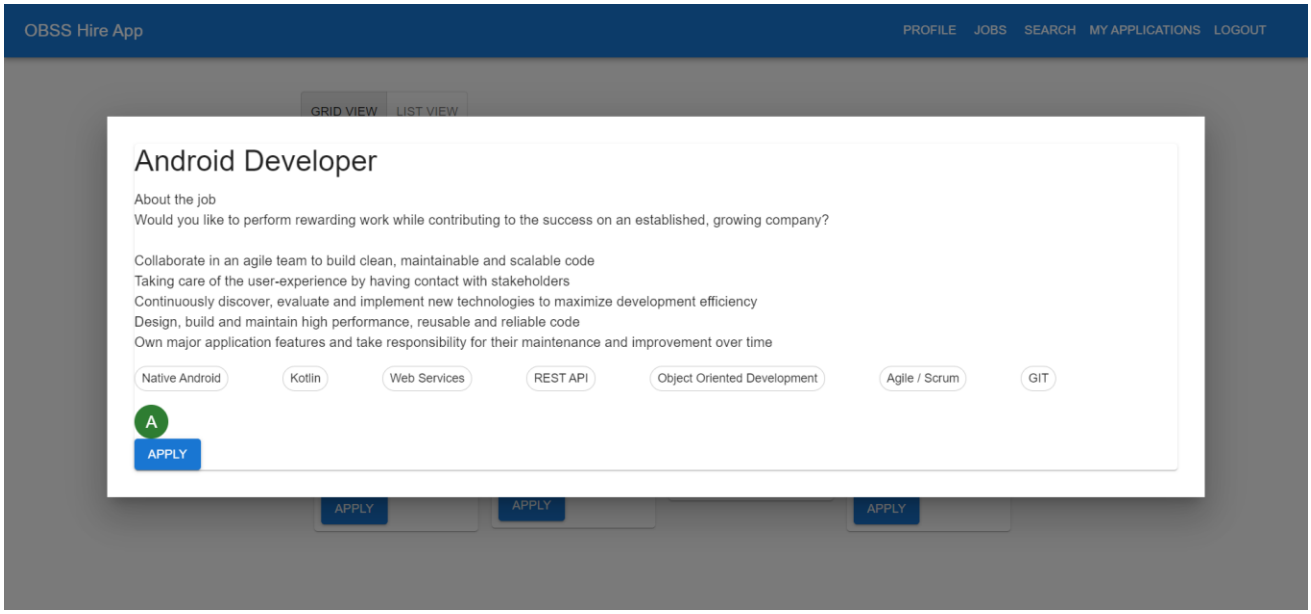

## 7.2 Profile Screen



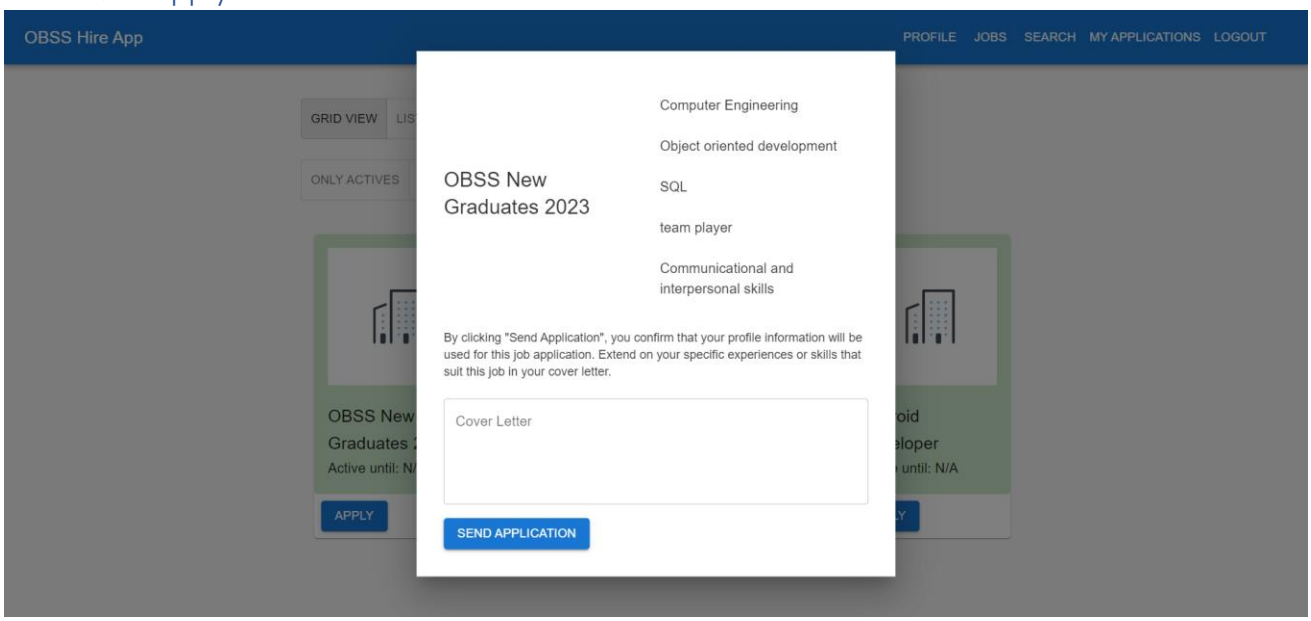
## 7.3 Job Openings Screen



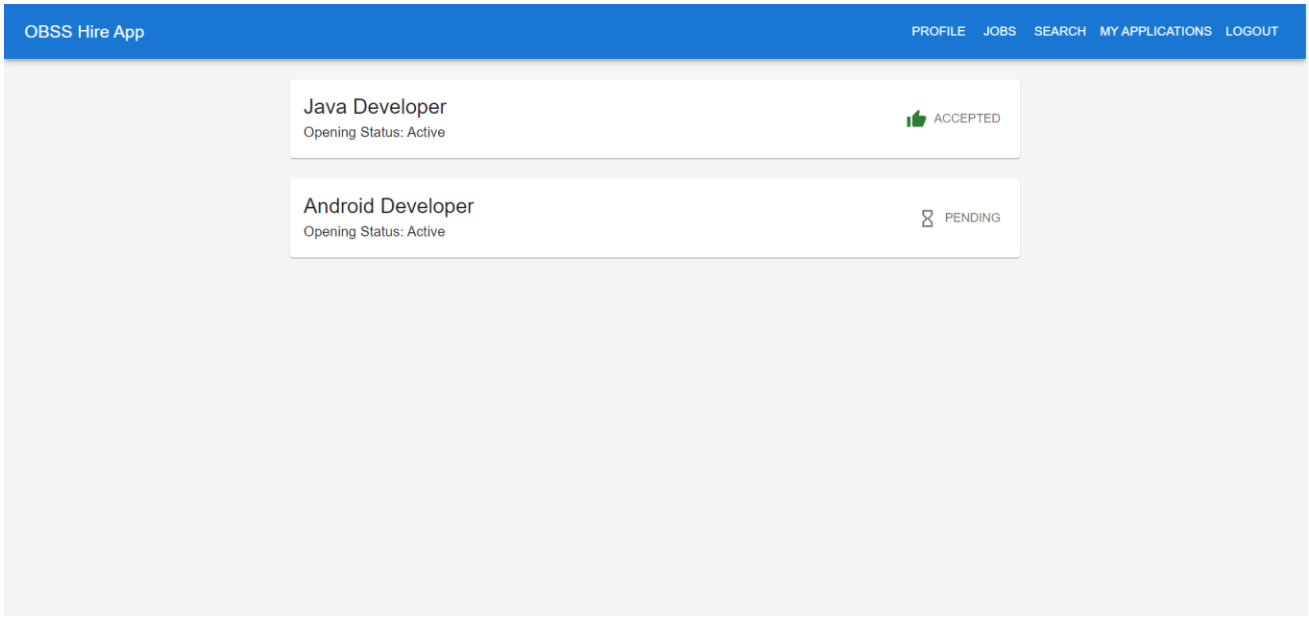

## 7.4 Job Opening Information Screen



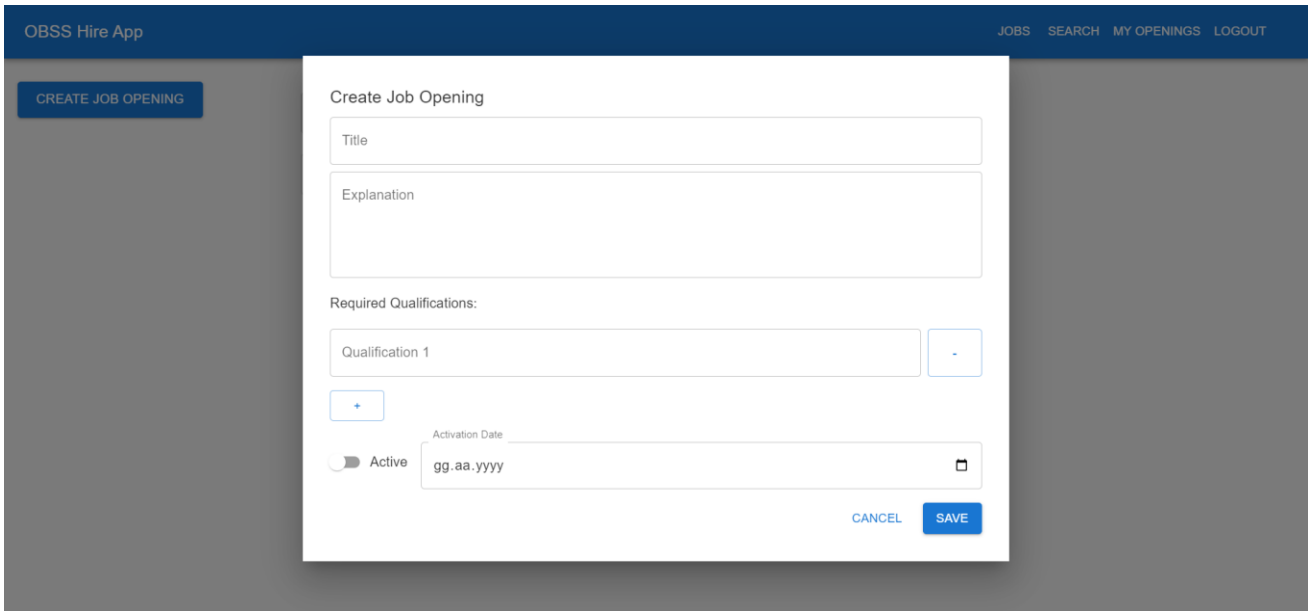
## 7.5 Apply Screen



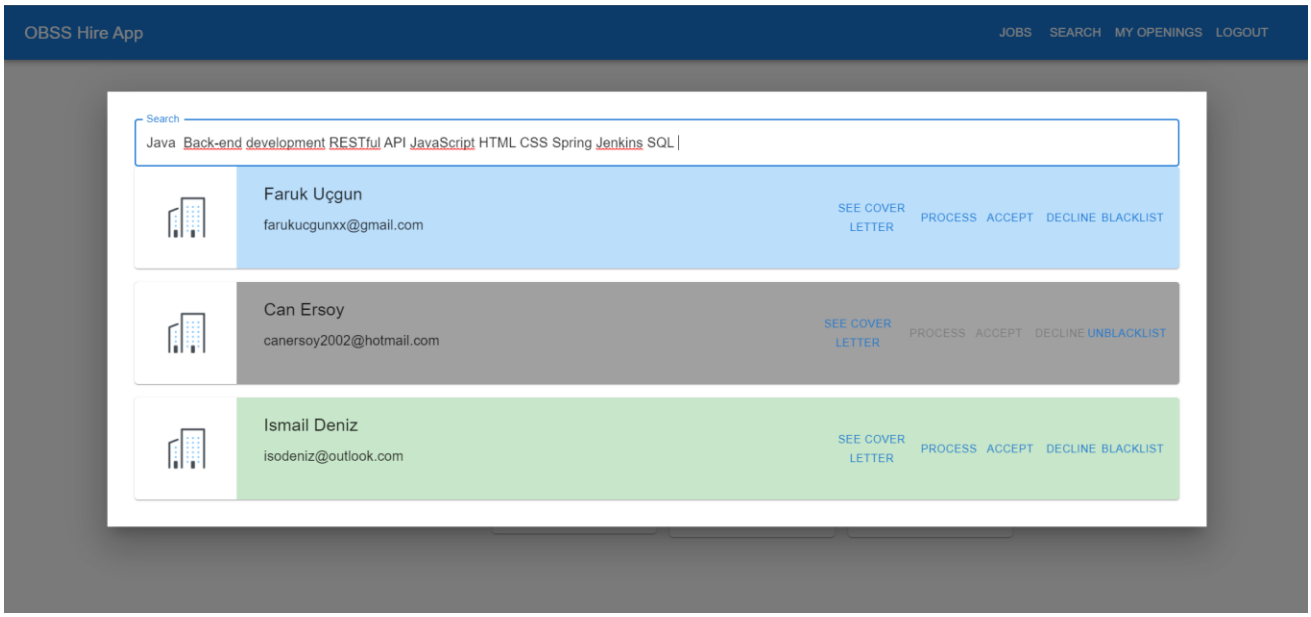

## 7.6 My Applications Screen



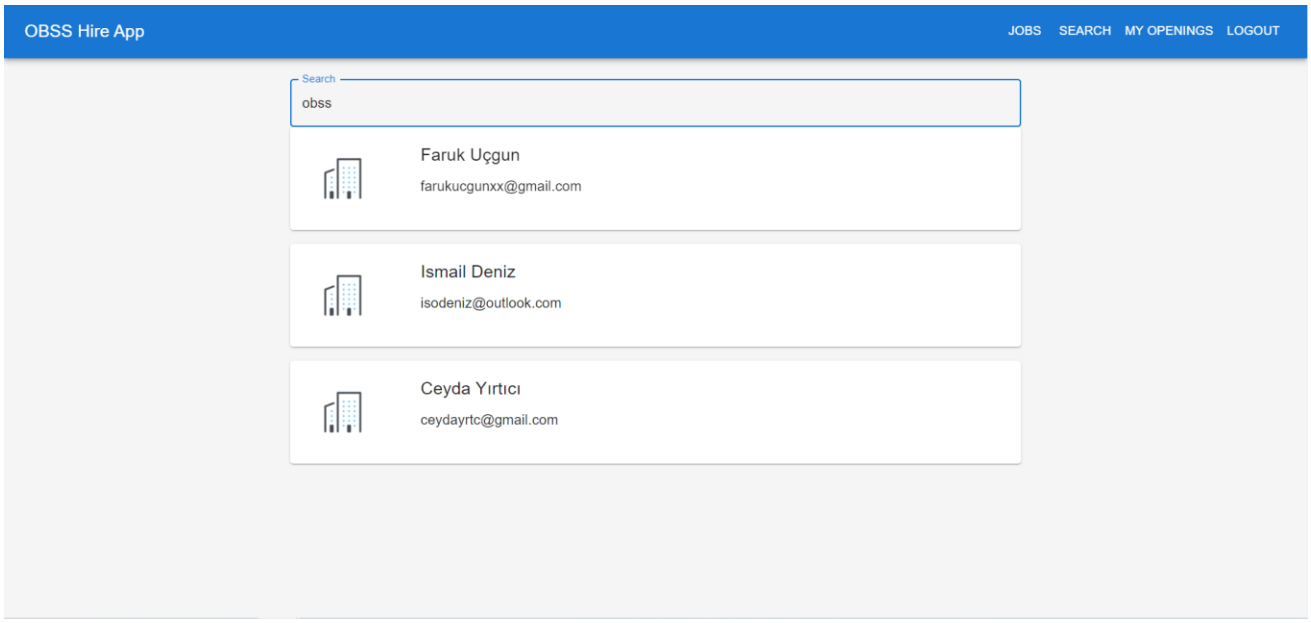
## 7.7 Create Opening Screen




## 7.8 See Applicants Screen



## 7.9 Search Screen



# References


- MongoDB Documentation: https://docs.mongodb.com/
- Spring Boot Documentation: https://spring.io/projects/spring-boot
- Selenium Official Documentation: https://www.selenium.dev/documentation/en/
- HTMLUnitDriver: https://htmlunit.sourceforge.io/
- React.js Documentation: https://reactjs.org/docs/getting-started.html