h n
oCOMP 3059 – Opportune

Software Requirements Analysis and Design Assignment

Leonid Serebryannikov
Michael Mocioiu

1.0 Introduction

1.1 Purpose

This document covers the main requirements of the systems behind Opportune. It provides a clear overview of what the systems should do, any limits they have, and what stakeholders will need/should expect. This helps keep everything organized and effective as we move into the design phase – establishing any necessary features, and laying out a guideline for the design.

1.2 Scope

This document describes the design and analysis of the Job Matchmaker system, the profiles that will be created for either employees or employers, managed, applied to, or reviewed. In this case, all aspects will be covered to seamlessly connect job seekers and employers: interface design, data handling, security, and interaction inside the system. Moreover, in this document we will also touch the constraints that are gonna face us during the development, design and post-deployment stages. We will explain assumptions, dependencies and look at possible use cases of our product. All of this will help us to design and build the project.

This explains what the proposed system will and will not do. Describe relevant benefits, objectives and goals. The description of scope should be consistent with the Project Plan.

2.0 System Overview

2.1 Project Perspective

Our project is a new self-contained system of a job-seeking family since we are niching down in the job market filling the market gap. We are not a follow-on member of a system family, though we will use their API's, since we are niching down and filling our system with features that satisfy our client's pain such as need in fast or even immediate employment. Also, we are not a replacement for those systems because they're focusing more on the full-time jobs and we do not touch that market.

The Project Perspective describes the context and origin of the system by defining whether the system is:

- a follow-on member of a system family
- a replacement for existing systems, or
- a new self-contained system.

2.2 System Context

The working group developed this system in the context that aims at targeting that particular part-time, one-time opportunity niche in the job-seeking industry. All the other existing job offering websites targeted full-time or long-term positions, which is different from this system, aimed at creating an interface between the seeker and short-term engagements for fast money or flexible schedules. It, therefore, aims at making the process easier for users who are supplementing their income, balancing commitments, or searching for work that fits into a flexible schedule. By addressing these needs, the system creates a strategic advantage in the specialized market of temporary employment.

The System Context describes the resulting software within the business case, including strategic issues in which the system is involved or which it specifically addresses.

2.3 General Constraints

1. The market for full-time jobs is occupied by big competitors such as linked in or indeed, so we had to specify our product and the best option for us was to focus on part-time and one-time job opportunities
2. Since the hiring market is a professional sphere, we will make the design more professional but at the same time simple
3. We don’t have a budget for this project, so we will face technical constraints such as dependency to a free hosting provider.
4. Our project will use linkedIn and Indeed API’s to fill our platform with job opportunities to apply, so we will have to stick with their using policies and their limits.

General Constraints identify any business or system constraints that will impact the manner in which the software is to be:

- specified
- designed
- implemented, or
- tested.

2.4 Assumptions and Dependencies

Assumptions:

- Job seekers and recruiters will accept the app, and it will be utilized actively
- Third-party services will be reliable
- The data from the users is valid

- Data and employment regulations will be followed
- Design of this platform will allow to scale it up later

Dependencies:

- Required technologies and tools are available
- Market demand facilitates opportunity
- Data will be stored with secure standards
- Continuous Platform Maintenance and Support
- Team members will be available as planned

List any assumptions that have been made during the initiation of the project. In addition, list any dependencies that may impact its success or the desired result.

3.0 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.1 <Functional Requirement or Feature #1>

- Must Allow for users to create/register accounts as Job seekers or employers
- Must allow for employers to publish job opportunities
- Must allow for users to upload images and documents
- Must allow for job Seekers to apply to job postings
- Must notify job seekers when employer accepted their application
- Must allow for employers to reject or approve applications
- Must allow for users to leave reviews on companies
- Must allow for job seekers and employers to communicate through messages
- Must allow for employers to edit, activate/deactivate, or remove job postings
- Must allow for users to upload task requests (odd-jobs or freelance)
- Must allow for Job seekers to search and filter through job postings

3.2 Use Cases

3.2.1 Use Case #1 ...

A full-time student will create a profile, showing their personality, work ethic and employment preferences. The student will then be allowed to search for part-time job availability, apply to whichever vacant position they choose, and receive notifications about the status of their applications.

3.2.2 Use Case #2 ...

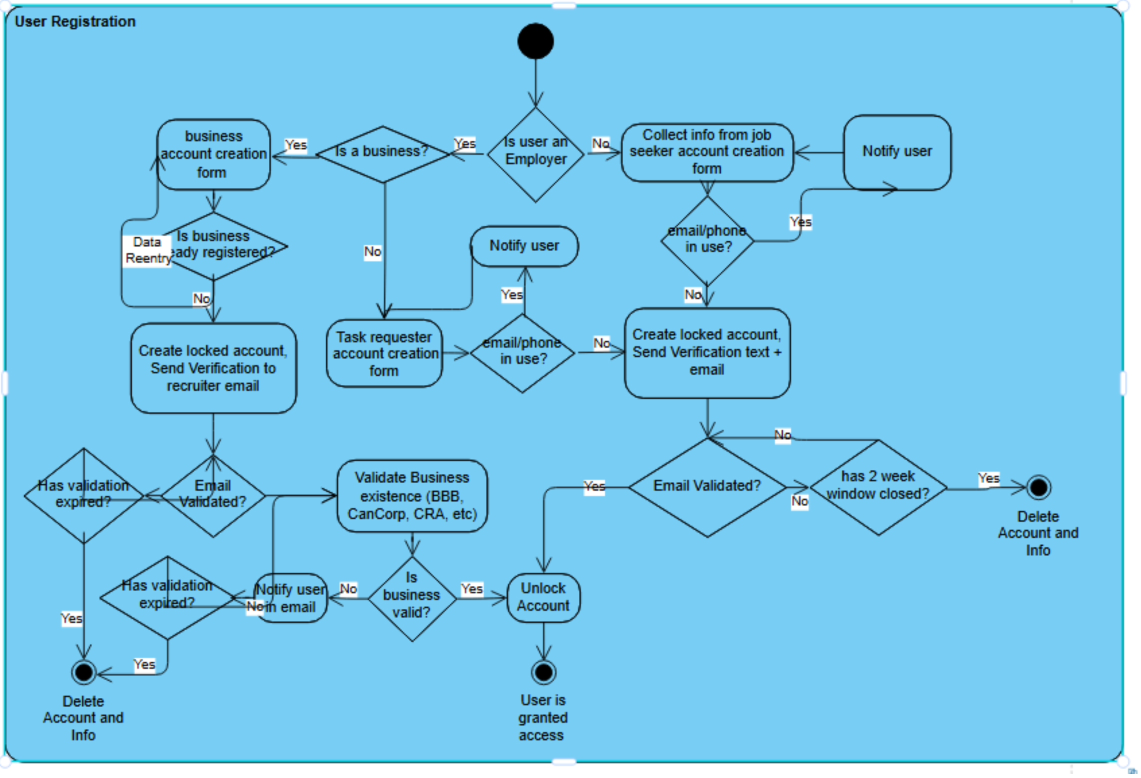
A company that is looking for physical help to set up an event for their employees. Company representative logs in to his employer account, create a job posting for event setup assistance, review the applications that come in, and select suitable candidates.

3.2.3 Use Case #3 …

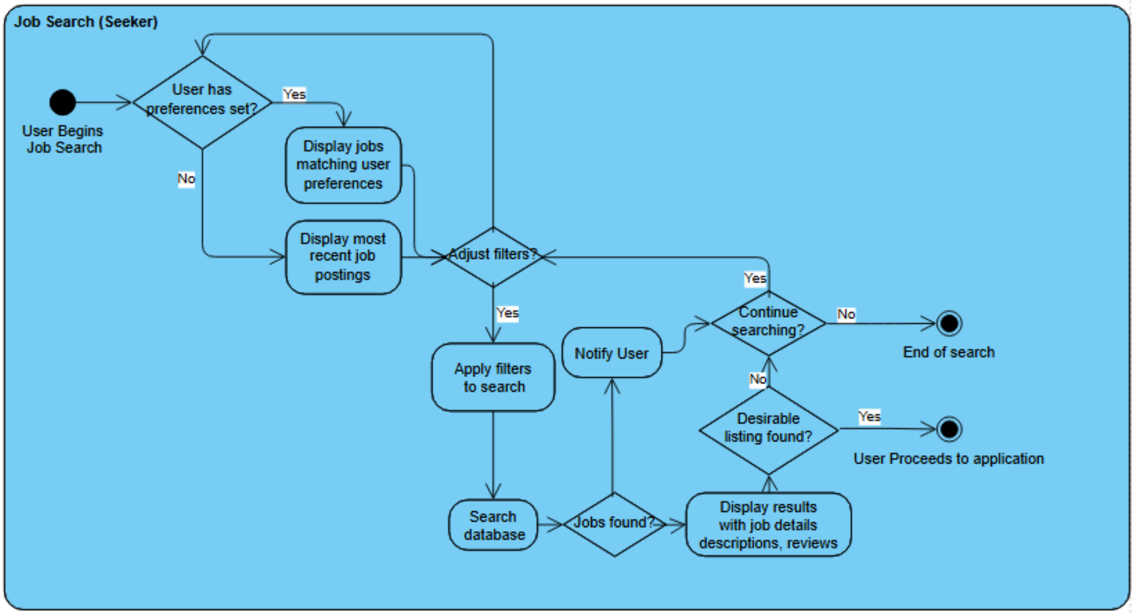
A small business owner that needs temporary assistance during a peak season will create a job posting. The owner will specify requirements, such as hours, and start date. They can view applicant profiles, check their reviews from previous jobs on the platform, and message through email potential candidates to coordinate interviews or clarify details before making a final selection.

3.3 Data Modelling and Analysis

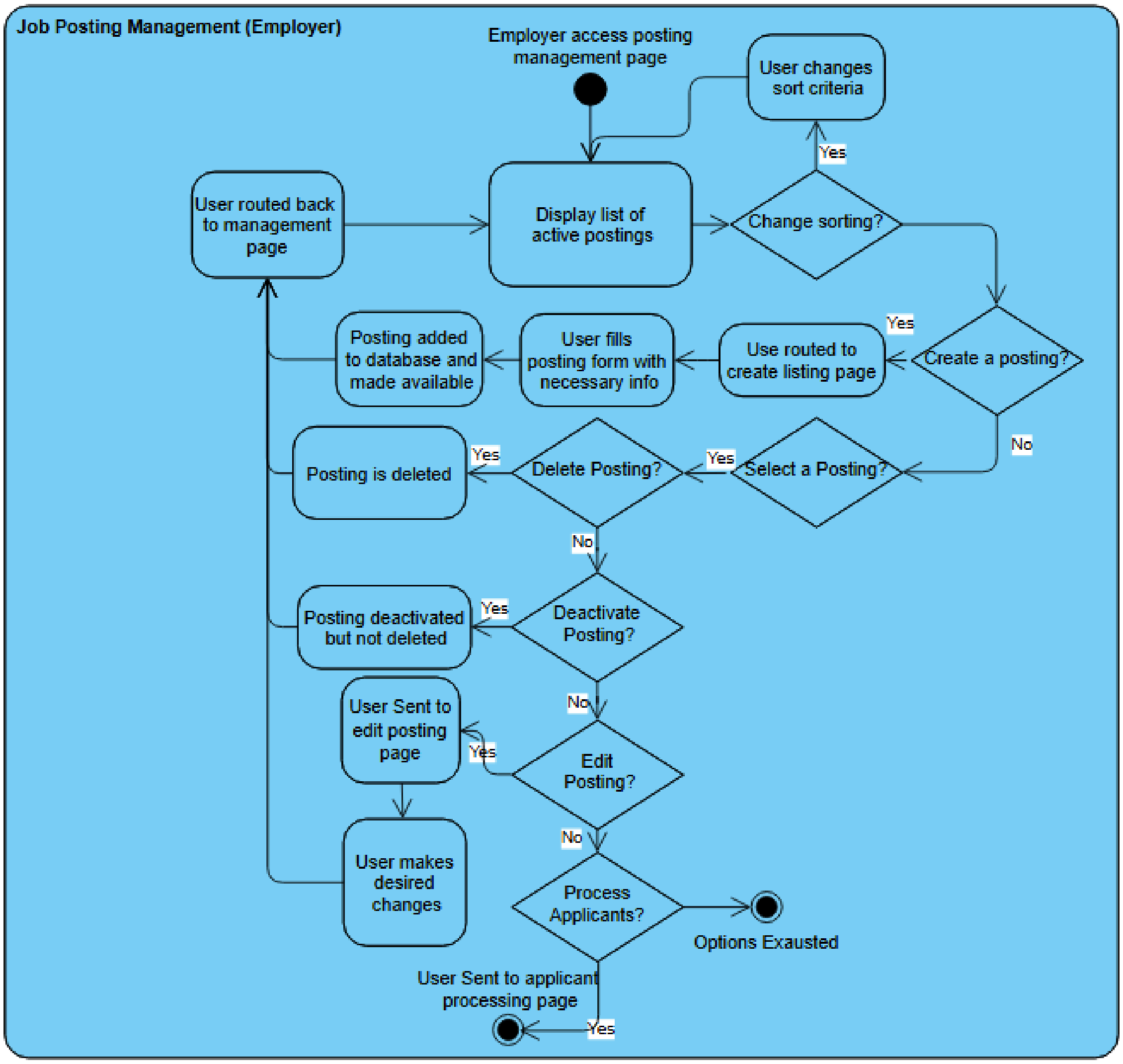
1. Activity Diagrams:
   a. Registration



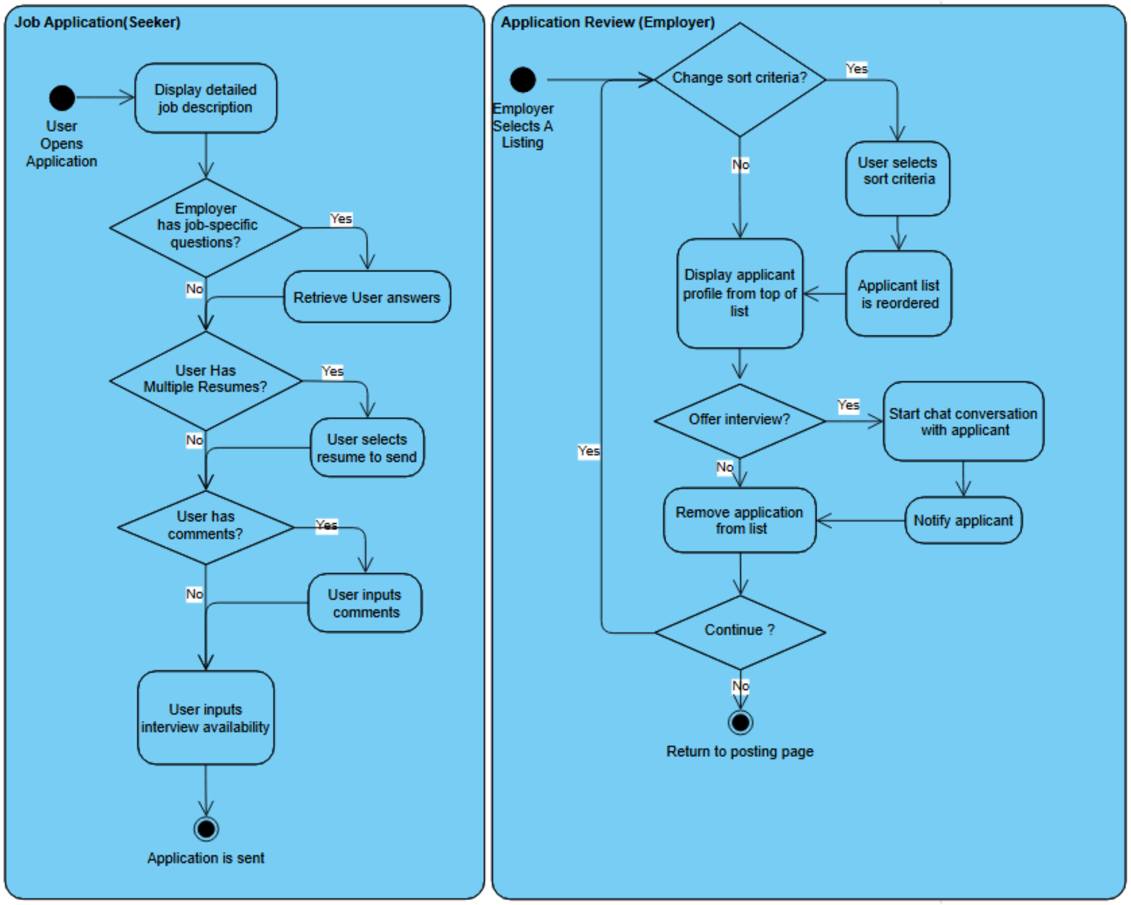
   b. Job Search



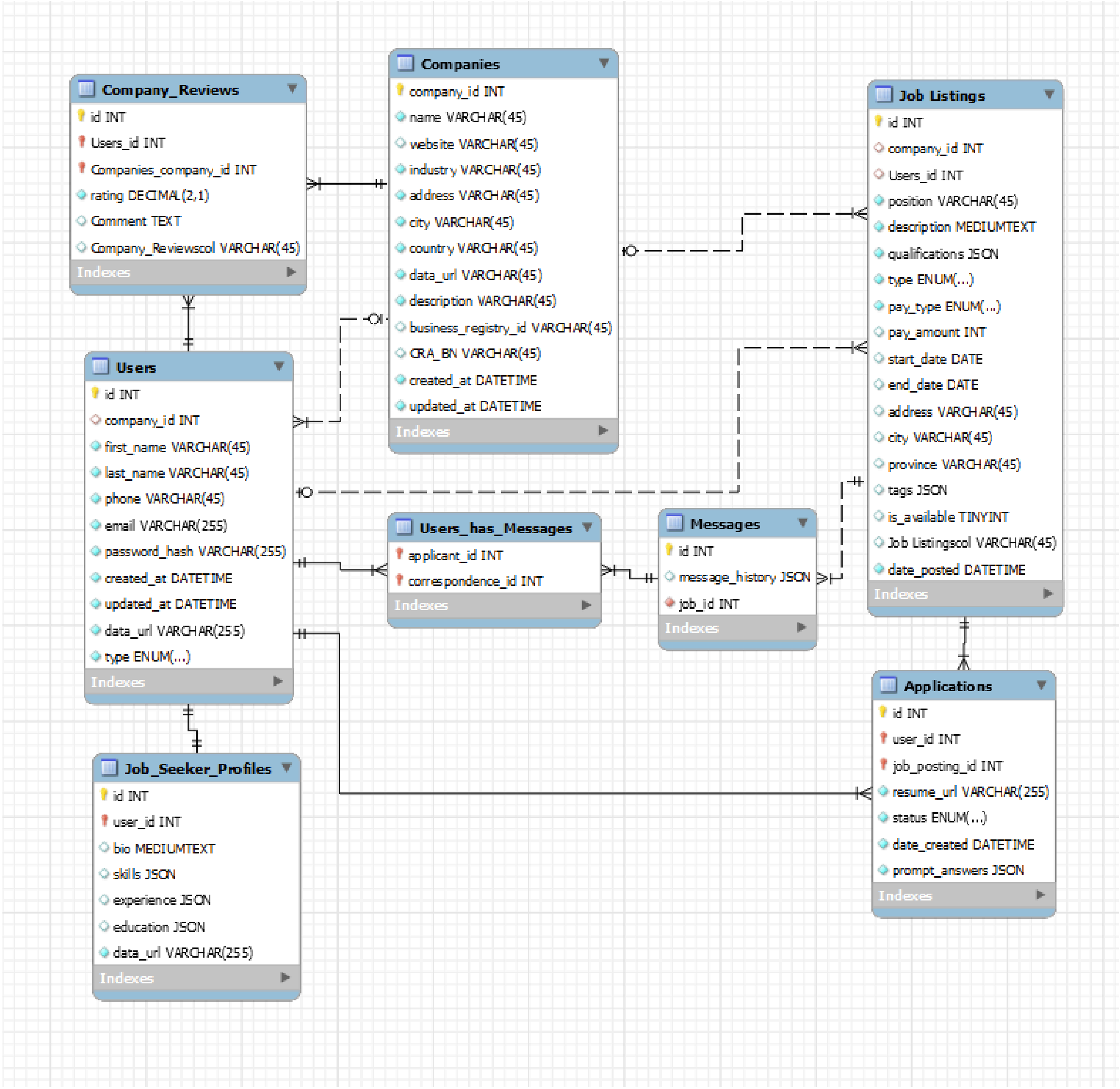
c. Job Posting Management


Job Posting Management (Employer)
Employer access posting management page
User changes sort criteria
Yes
User routed back to management page
Display list of active postings
Change sorting?
Yes
Posting added to database and made available
User fills posting form with necessary info
Use routed to create listing page
Create a posting?
No
Posting is deleted
Yes
Delete Posting?
Yes
Select a Posting?
No
Posting deactivated but not deleted
Yes
Deactivate Posting?
No
User Sent to edit posting page
Yes
Edit Posting?
No
User makes desired changes
Process Applicants?
Options Exausted
User Sent to applicant processing page
Yes

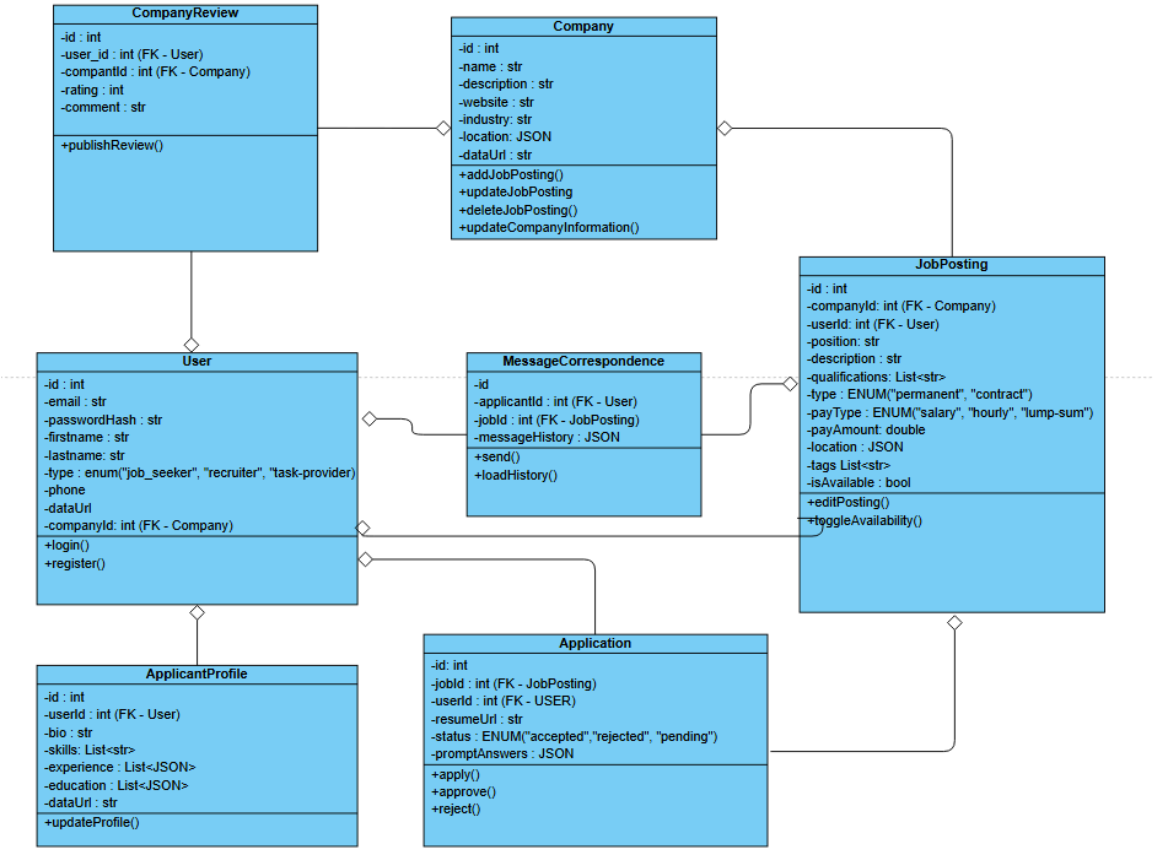
d. Job Application and Application Review


Job Application(Seeker)
User Opens Application
Display detailed job description
Employer has job-specific questions?
Yes
No
Retrieve User answers
User Has Multiple Resumes?
Yes
No
User selects resume to send
User has comments?
Yes
No
User inputs comments
User inputs interview availability
Application is sent
Application Review (Employer)
Employer Selects A Listing
Change sort criteria?
Yes
No
User selects sort criteria
Applicant list is reordered
Display applicant profile from top of list
Offer interview?
Yes
No
Start chat conversation with applicant
Notify applicant
Remove application from list
Continue ?
Yes
No
Return to posting page

2. Normalized Data Model Diagram

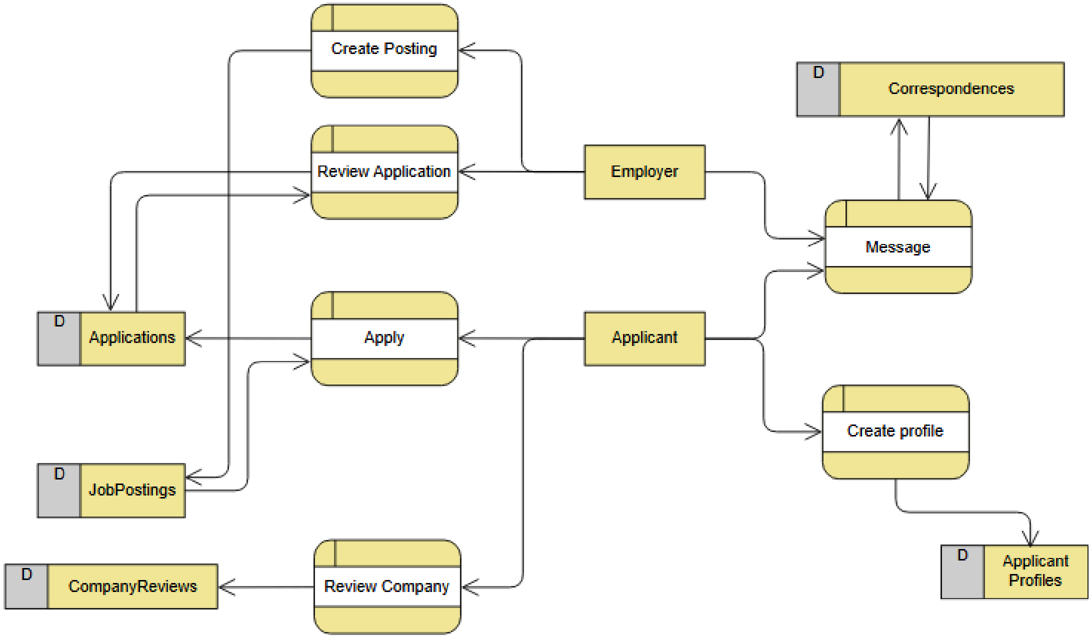


3. UML Class Diagram



3.4 Process Modelling

Data Flow Diagram:



4.0 Non-Functional Requirements

- Intuitive Design
  - The platform will have a user-friendly design to ensure that even first-time users can easily navigate the interface
- Performance
  - The system should execute create/read/update/delete/list (CRUDL) operations in under 1 second, as there are no complex data operations, providing users with a smooth and fast experience.
- Security
  - The platform must be secure due to the storage of personal data, prioritizing user privacy and data protection.
- Maintainability
  - The system should be easy to maintain, allowing for efficient updates and improvements over time.
- Scalability
  - The design will be scalable, for example the potential to expand into a mobile application to enhance accessibility and user engagement.

5.0 Logical Database Requirements

A NOSQL (MongoDB) database will be used to store user information, job postings, applications, reviews, company information, and message histories. Login information will need to be stored securely using multiple methods such as hash encryption, injection protection, and carefully constructed logic to ensure that sensitive data is not accessed unless completely necessary. Another logical requirement that must be closely monitored is database size – file uploads such as images or resumes will NOT be stored in the database to avoid bloat, rather path URLs to their location in the server will be used. Data will need to be heavily sanitized prior to being stored to ensure data integrity. Data relationships will need to be strictly defined as well, such that the integrity of the base is consistent and bulletproof.

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc?

6.0 Approval

The signatures below indicate their approval of the contents of this document.

Project Role	Name	Signature	Date
Front-End Engeneer	Leonid Serebryannikov	L.S.	2024-11-08
Back End Engineer	Michael Mocioiu	M.B.M	2024-11-08