**COMP 3059 – Capstone Project I**

**Software Requirements Analysis and Design Assignment**

This assignment is an overview to gather the software needs with requirements analysis and help to proceed with the design.

The requirements analysis helps to break down functional and non-functional requirements to a basic design view to provide a clear system development process framework. It involves various entities, including business, stakeholders and technology requirements.

The design is the activity following requirements specification and before programming. Software design usually involves problem solving and planning a software solution.

To work on this assignment, you could use the references and a sample template given below. The sample template can be customised to suit the nature of your project.

Reference Readings/Example:

<http://www.uacg.bg/filebank/acadstaff/userfiles/publ_bg_397_SDP_activities_and_steps.pdf>

[www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc](http://www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc)

Reference template:

[www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc](http://www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc)

# 1.0 Introduction

## 테이블이(가) 표시된 사진 자동 생성된 설명Scope

## Purpose

To meet the individual needs of the enterprise to the greatest extent, according to the characteristics of the enterprise and the management needs and user needs. The customized software is clearly organized to meet the user's characteristics and habits. Users only need to have basic knowledge to operate. Adhere to the basic principle of serving users and provide enterprises with comprehensive services. In the initial stage of use, users can be directly contacted to inquire about usage and experience and feedback, collect suggestions and adjust maximize user experience.

## Scope

The most difficult part of developing a software system is specifying exactly what to develop. The most difficult conceptual task is to write detailed technical requirements, which include all user-oriented, machine-oriented and other software system interfaces. At the same time, this is the part that will eventually cause great damage to the system once it is done wrong, and it is extremely difficult to modify it later.

Organizations that implement effective requirements engineering management can gain many benefits. The biggest advantage is that the redo work in the later development stage and the entire maintenance stage is greatly reduced. The correct demand process emphasizes the full cooperation in product development, including the active efforts of multiple stakeholders throughout the project.

# System Overview

We will use the latest HTML web design elements to enrich our website. The front end of the site will use CSS and HTML to design. On the back end of the server, we will use SQL to store the database. To ensure that users are aware of the latest information, we will design and implement a web crawler. It will help users have a better experience when using the site.

## Project Perspective

The system we are going to do is not unique. There are already many social networking sites or software widely used today. Our team focuses on making our products unique. We will make our own social recruitment website by referring to and absorbing the advantages and disadvantages of surrounding software. It is worth mentioning that we are not a social platform and users cannot chat or add friends on the site. What we want is for the user to just browse the information, and for the system administrator to do all the rest. So it will be a new-self contained system.

## System Context

-JavaScript:：Our system will use JavaScript to implement the specific functions and design of the website, and with CSS to make the website beautiful, the server will be implemented with Node.js and NPM.

-SQL database： System database storage we use SQL to achieve, because SQL is more convenient and more suitable for small and medium-sized website data storage and operation

-Git ：Depending on the requirements of the team, we can also use the form of Git to publish our own system if necessary.

## General Constraints

To make our system concrete, we need to design a lot of code and relate it. So, there are a lot of procedural problems during design, such as code errors or logic errors, but our team will discuss them together and solve them. As in previous teamwork, we have good cooperation.

## Assumptions and Dependencies

Because given the team presentation at that time, we learned a lot about concerns and problems. We've made a lot of assumptions about these issues, and one of them is that nationality leads to confusion and racial discrimination. Our solution is to set the user to choose their current country of residence and prohibit the user from registering for an account if it doesn't match the IP address. In addition, the system designed by us will only be open in Canada at present. Whether to internationalize it is to be considered.

## 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 Requirements.

### 3.1.1 Use Case: Add Promotion.

|  |  |
| --- | --- |
| **Use Case Name** | Add Promotion |
| **Trigger** | **Admin** selects to add a new promotion to the database |
| **Basic Path** | 1. The system shows a blank form to enter the promotion information. 2. Admin fill out the form and submits the form 3. The system validates fields of the promotion form if any fields are not blank, insert them into the database. Either any fields are blank, shows the validation messages |
| **Output** | A new promotion has been added to database. |

### Use Case: Edit Promotion.

|  |  |
| --- | --- |
| **Use Case Name** | Edit Promotion |
| **Trigger** | **Admin** selects to edit a promotion to the database |
| **Basic Path** | 1. The system shows a filled form from selected promotion to edit the promotion information. 2. Admin edit the promotion fields and submits the form 3. The system validates fields of the promotion form if any fields are not blank, update them to the database. Either any fields are blank, shows the validation messages |
| **Output** | An edited promotion has been updated to database. |

### 3.1.3 Use Case: Delete Promotion

|  |  |
| --- | --- |
| **Use Case Name** | Delete Promotion |
| **Trigger** | **Admin** selects to delete the promotion to the database |
| **Basic Path** | 1. The system accesses the database to get all promoted information and shows it on the screen. 2. Admin selects a promote and click on the delete button 3. The system shows a delete confirmation window. 4. A selected promote record has been deleted in the database after the admin clicked on confirm button. |
| **Output** | A selected promotion has been deleted to database. |

### 3.1.4 Use Case: View Job Posting, Application and Employer/Job Seeker profile.

|  |  |
| --- | --- |
| **Use Case Name** | View Job Posting, Application and Employer/Job Seeker profile. |
| **Trigger** | **Admin** enters the value to search, selects the Job Posting, Application or Employer/Job Seeker profile from category, clicks on it to view. |
| **Basic Path** | 1. Admin can search by Job posting/Employer/Job Seeker then type the information in the search field. 2. The system accesses the database to get the information record related to the input value and shows it on the screen. 3. Admin clicks on the title of Job Posting, Application and Employer/Job Seeker profile to view. |
| **Output** | Admin could view the Job posting and Employer/ Job Seeker profile. |

### 3.1.5 Use Case: Search for Job Posting, Application and Employer/Job Seeker profile.

|  |  |
| --- | --- |
| **Use Case Name** | Search for Job Posting, Application and Employer/Job Seeker profile. |
| **Trigger** | **Admin** enters the value to search, selects the Job Posting, Application or Employer/Job Seeker profile. |
| **Basic Path** | 1. Admin can search by Job posting/Employer/Job Seeker then type the information in the search field. 2. The system accesses the database to get the information record related to the input value and shows it on the screen. |
| **Output** | Admin found the Job posting and Employer/ Job Seeker profile. |

### 3.1.6 Use Case: Accept/Reject Job Posting.

|  |  |
| --- | --- |
| **Use Case Name** | Accept/ Reject Job Posting |
| **Trigger** | **Admin** selects the new job posting from the notification or from the New Job posting to accept or reject it. |
| **Basic Path** | 1. Admin go to New Job Posting or click on notification to view it. 2. The system accesses the database to get the selected record to show it on the screen. 3. Admin can accept or reject the job posting by click on the button. If admin clicks on the accept button, that job posting status will set accepted. Otherwise, that job posting status will set rejected. |
| **Output** | Admin could view the Job Posting and then accept or reject it. |

### 3.1.7 Use Case: Delete Job Posting and Employer/Job Seeker account.

|  |  |
| --- | --- |
| **Use Case Name** | Delete Job Posting and Employer/Job Seeker account. |
| **Trigger** | **Admin** selects to delete the Job Posting and Employer/Job Seeker account to the database. |
| **Basic Path** | 1. Admin can look for the Job posting/Employer/Job Seeker to delete by searching it by categories. 2. The system accesses the database to get the record related to the input value and shows it on the screen. 3. Admin clicks on the record of Job Posting and Employer/Job Seeker profile to delete. 4. The system shows a delete confirmation window. 5. A selected record has been deleted in the database after the admin clicked on confirm button. |
| **Output** | A selected record has been deleted to database. |

### 3.1.8 Use Case: Send the financial report to Employer.

|  |  |
| --- | --- |
| **Use Case Name** | Send the financial report to Employer. |
| **Trigger** | **Accountant** searches the Employer to send the report to Employer. |
| **Basic Path** | 1. Accountant can type the value in the search field to look for the Employer. 2. Select the Employer, click on send the report and select the file to send it, then submit it. 3. The systems will store it in database. |
| **Output** | Accountant could send the report to Employer. |

### 3.1.9 Use Case: Register Account.

|  |  |
| --- | --- |
| **Use Case Name** | Register Account |
| **Trigger** | **Employer/Job Seeker** select register to sign up a new account. |
| **Basic Path** | 1. The Employer/ Job Seeker select register, then fill out the fields and select the role (Employer/Job Seeker), submit the form. 2. The system validates fields of the register form if any fields are not blank, insert them into the database. Either any fields are blank, shows the validation messages |
| **Output** | A new account has been added to the database. |

### 3.1.10 Use Case: Add Job Posting.

|  |  |
| --- | --- |
| **Use Case Name** | Add Job Posting |
| **Trigger** | **Employer** selects to add a new Job Posting to the database |
| **Basic Path** | 1. The system shows a blank form. 2. Admin fill out all the fields and submits the form 3. The system validates fields of the form if any fields are not blank, insert them into the database. Either any fields are blank, shows the validation messages |
| **Output** | A new job posting has been added into the database. |

### 3.1.11 Use Case: Edit Profile.

|  |  |
| --- | --- |
| **Use Case Name** | Edit Profile |
| **Trigger** | **Employer/Job Seeker** selects to edit the profile to the database |
| **Basic Path** | 1. The system shows a filled form of your profile record. 2. Employer/Job Seeker edit the fields and submits the form 3. The system validates fields of the form if any fields are not blank, update them to the database. Otherwise, any fields are blank, shows the validation messages. |
| **Output** | An edited profile has been updated to database. |

### 3.1.12 Use Case: Delete Job posting.

|  |  |
| --- | --- |
| **Use Case Name** | Delete Job Posting. |
| **Trigger** | **Employer** selects to delete the Job Posting to database |
| **Basic Path** | 1. The employer can look for the Job posting to delete by searching 2. The system accesses the database to get the record related to the input value and shows it on the screen. 3. Employer clicks on the record of Job Posting to delete. 4. The system shows a delete confirmation window. 5. A selected record has been deleted in the database after the admin clicked on confirm button. |
| **Output** | A selected record has been deleted to database. |

### 3.1.13 Use Case: Search for Job Seeker profile/Job Posting.

|  |  |
| --- | --- |
| **Use Case Name** | Search for Job Posting/Job Seeker profile. |
| **Trigger** | **Employer** enters the value to search, selects the Job Posting/Job Seeker profile. |
| **Basic Path** | 1. Employer can search by Job posting/Job Seeker then type the value in the search field. 2. The system accesses the database to get the record related to the input value and shows it on the screen. |
| **Output** | Employer found the Job posting and Job Seeker profile. |

### 3.1.14 Use Case: View Job Posting/Application/Job Seeker profile.

|  |  |
| --- | --- |
| **Use Case Name** | View Job Posting/Application/Job Seeker profile. |
| **Trigger** | **Employer** enters the value to search, selects the Job Posting or Job Seeker profile from category, clicks on it to view. |
| **Basic Path** | 1. Employer can search by Job posting/Application/Job Seeker then type the value in the search field. 2. The system accesses the database to get the information record related to the input value and shows it on the screen. 3. Employer clicks on the record of Job Posting/Application/Job Seeker profile to view. 4. Download the application or job posting. |
| **Output** | Employer could view the Job posting, Application and Job Seeker profile. |

### 3.1.15 Use Case: Apply for job.

|  |  |
| --- | --- |
| **Use Case Name** | Apply for job |
| **Trigger** | **Job Seeker** enters the value to search the job posting, selects the Job Posting, clicks on it to view. |
| **Basic Path** | 1. Job Seeker select search, then types the value in the search field, and select Job posting. 2. The system accesses the database to get the information record related to the input value and shows it on the screen. 3. Job Seeker clicks on the record of Job Posting to view. 4. Submit the application, then it stores in the database |
| **Output** | Job Seeker could apply the job |

### 3.1.16 Use Case: View Application Status.

|  |  |
| --- | --- |
| **Use Case Name** | View Application Status |
| **Trigger** | **Job Seeker** clicks on application status to view. |
| **Basic Path** | 1. Job Seekercan search by Job posting/Employer, then type the value in the search field. 2. The system accesses the database to get the information record related to the input value and shows it on the screen. 3. Job Seeker clicks on the record to view the status. |
| **Output** | Job Seeker could view their application status. |

### 3.1.17 Use Case: View Job Posting/Employer.

|  |  |
| --- | --- |
| **Use Case Name** | View Job Posting/Employer. |
| **Trigger** | **Job Seeker** enters the value to search, selects the Job Posting, Employer, Profile from category, clicks on it to view. |
| **Basic Path** | 1. Job Seekercan search by Job posting/Employer then type the value in the search field. 2. The system accesses the database to get the information record related to the input value and shows it on the screen. 3. Job Seekerclicks on the record to view it. |
| **Output** | Job Seeker could view the Job Posting/Employer. |

### 3.1.18 Use Case: Admin.

|  |  |
| --- | --- |
| **Use Case Name** | Search for Job Posting/Employer. |
| **Trigger** | **Job Seeker** enters the value to search, selects the Job Posting/Employer. |
| **Basic Path** | 1. Job Seeker can search by Job posting/Employer then type the value in the search field. 2. The system accesses the database to get the record related to the input value and shows it on the screen. |
| **Output** | Employer found the Job posting and Employer. |

## 3.2 Use Cases

### 3.2.1 Use Case: Admin.

Diagram

Description automatically generated

### 3.2.2 Use Case: Accountant.

Diagram

Description automatically generated

### 3.2.3 Use Case: Employer.

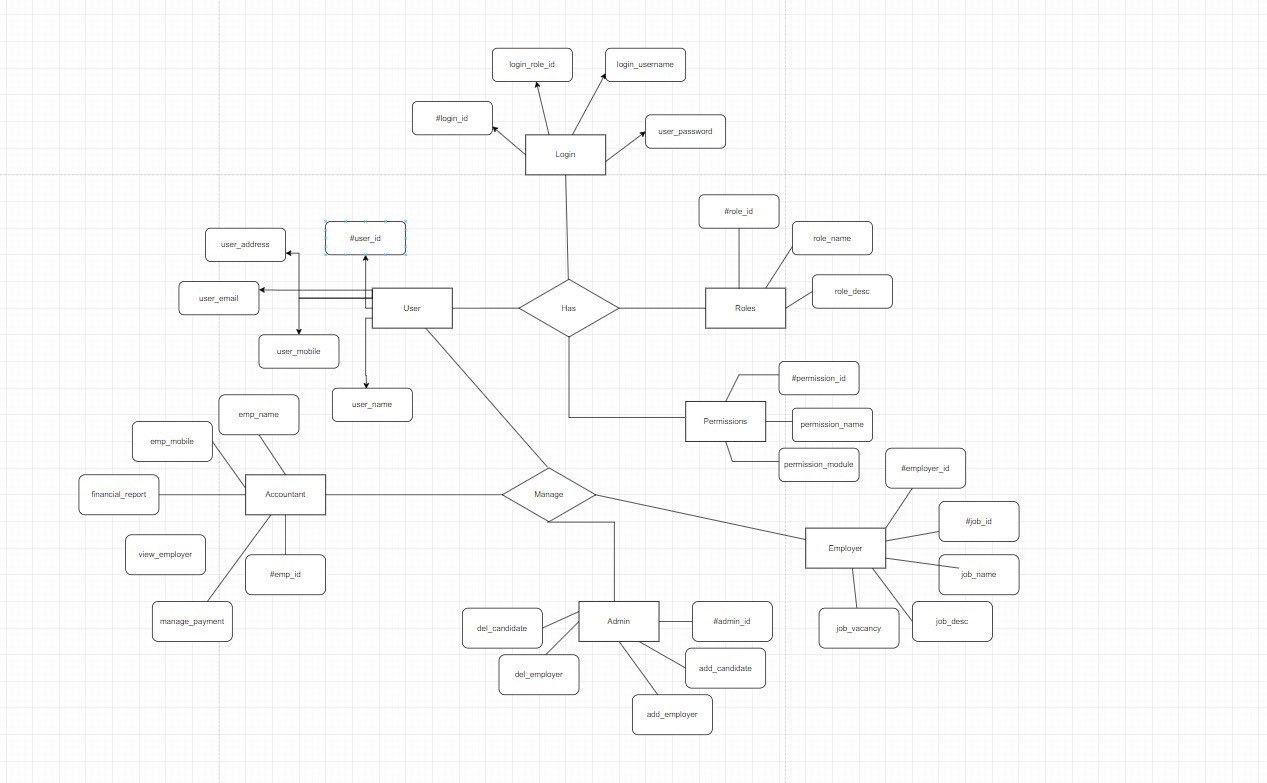
Diagram

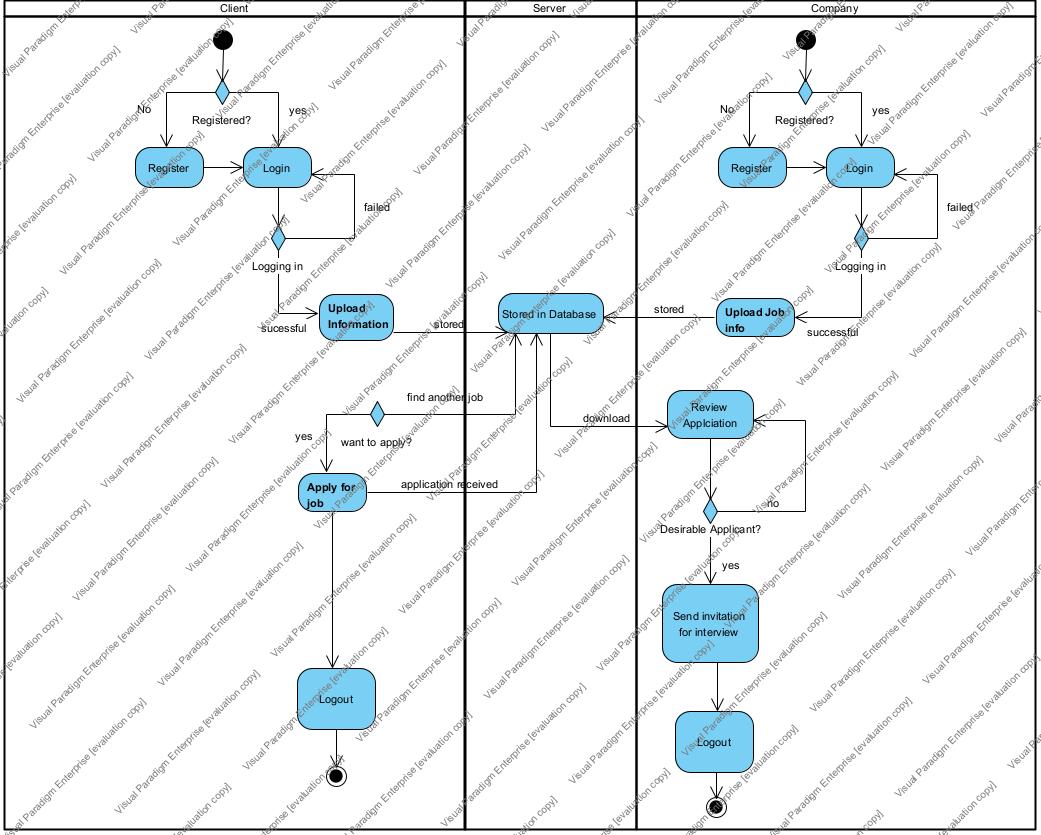
Description automatically generated

### 3.2.2 Use Case: Job Seeker.

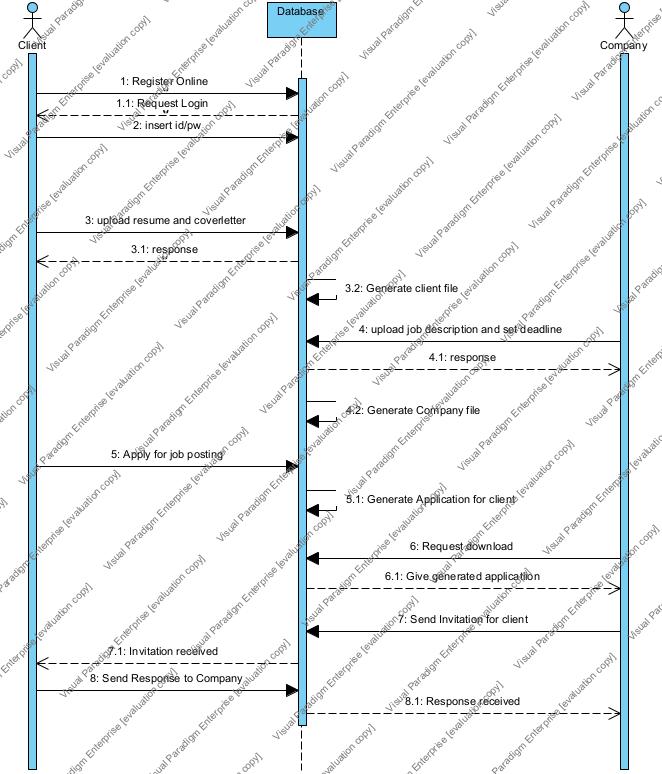
### Diagram Description automatically generated

**3.3 Data Modelling and Analysis**

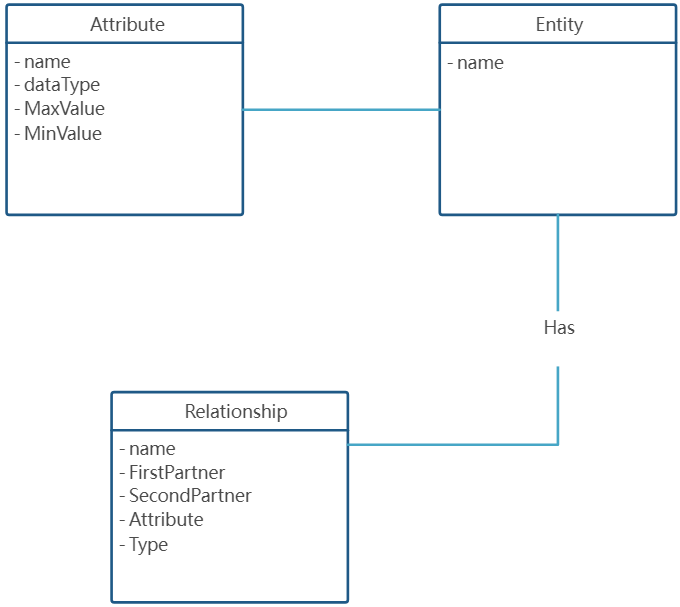
* Normalized Data Model Diagram
* Activity Diagrams



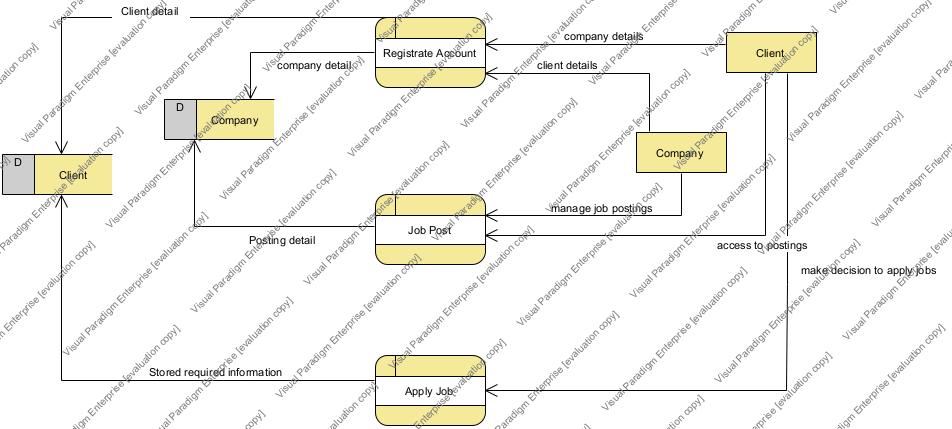
* Sequence Diagrams



* UML Class Diagram

**3.4 Process Modelling**

* Data Flow Diagram



## 4.0 Non-Functional Requirements

We also consider the stability of the system performance because the quality of the database directly affects the data storage speed and front-end stability. So, we decided to make the database independent, in other words, the front end can be used as usual during system maintenance without affecting users. By the time we made the update and decided to upload it, unless the front-end design was updated or optimized, the maintenance could be done within half an hour. The daily maintenance does not exceed 2 minutes. The system is maintained, and data is processed briefly at night when no one is using it.

## 5.0 Logical Database Requirements

This project involves a database to store all the information from clients and companies. Most of the data will be entered by users therefore, logical database is only needed to give analytics or trends on how companies are recruiting to the management team to set up a marketing strategy.

## 6.0 Other Requirements

Because the system is a for-profit site, unlike social sites. So, we need to recruit advertisers and ask for investment fees. However, as the system is in its initial stage, our team decided to attract investment after attracting many customers and popularity.

**7.0 Approval**

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

|  |  |  |  |
| --- | --- | --- | --- |
| Project Role | Name | Signature | Date |
| Project manager | Yoo Kyung Baek | Yoo Kyung Baek | Nov 11, 2021 |
|  |  |  |  |
|  |  |  |  |