CSC 4330 Scrum 2 Report
March 9, 2020

Kareem Abdo
Christopher Chee
Derek Delahoussaye
Taylor Olinde
Quoc Than

Overall Architecture

- Multilayered (Client-Server) Architecture
  - Presentation Layer
  - Data Layer

Project Backlog

- Search engine to search for available positions and the ability to filter for traits such as salary, state, and level of education while excluding positions that have been filled.
- Database to keep track of active applications or status of applications.
- Message board for employees to companies they are working for.
- Database and web page to display / create a user profile for applicants with information such as email address and areas of interest.
- Database that keeps track of applicants, open positions, and other internal or external applications.
- Applicants can apply to each job only once and a maximum of 20 jobs.
- Jobs can accept up to 100 applications and only allow some applications by current employees.
- Statistics page to display the 5 most applied to companies and most applied to fields.

Sprint Requirements

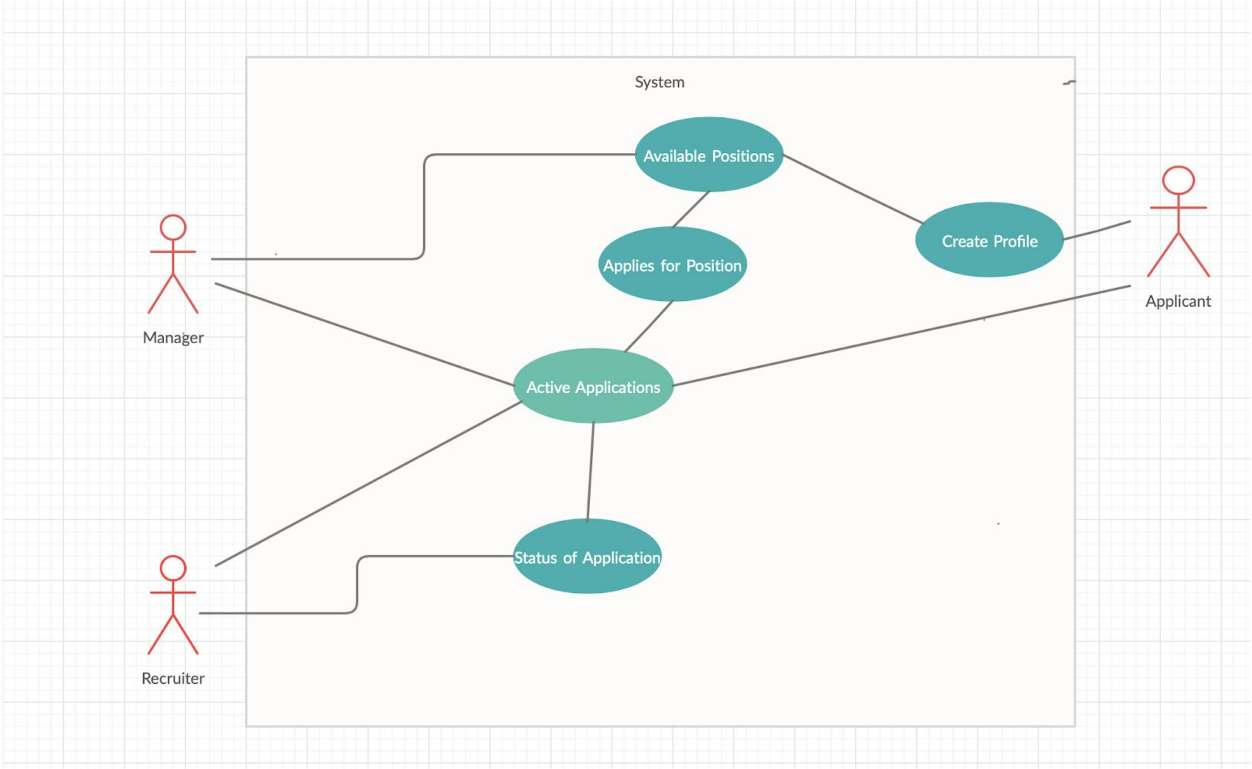
- Have a strategy for implementing front-end development code
- Consult with project manager on proper system requirements
- Setup database to keep track of active applications or status of applications.
- Setup database and create a webpage to display / create a user profile for applicants with information such as email address and areas of interest.
- Setup database that keeps track of applicants, open positions, and other internal or external applications.
- Write SQL queries to test database features

Table of Responsibilities

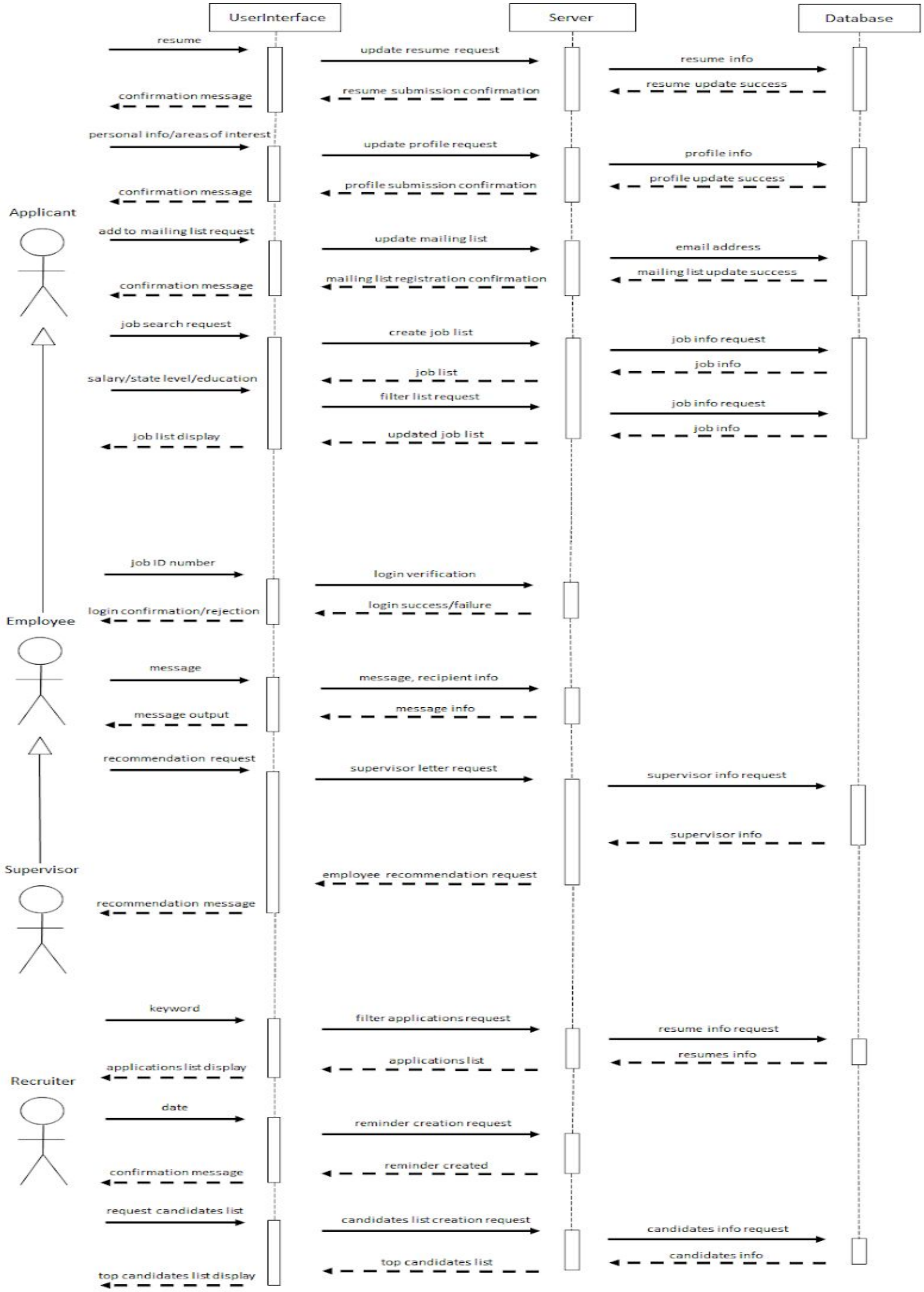
Database Management	Taylor Olinde, Kareem Abdo
Development (Front-end)	Christopher Chee, Derek Delahoussaye (JS), Quoc Ai Than
Github	Everyone

Project Deliverables and Reports	Everyone
Use Case Diagram	Derek
Sequence Diagram	Chris
Class Diagram/Database Schema	Taylor
Set up MySQL Workbench database	Derek

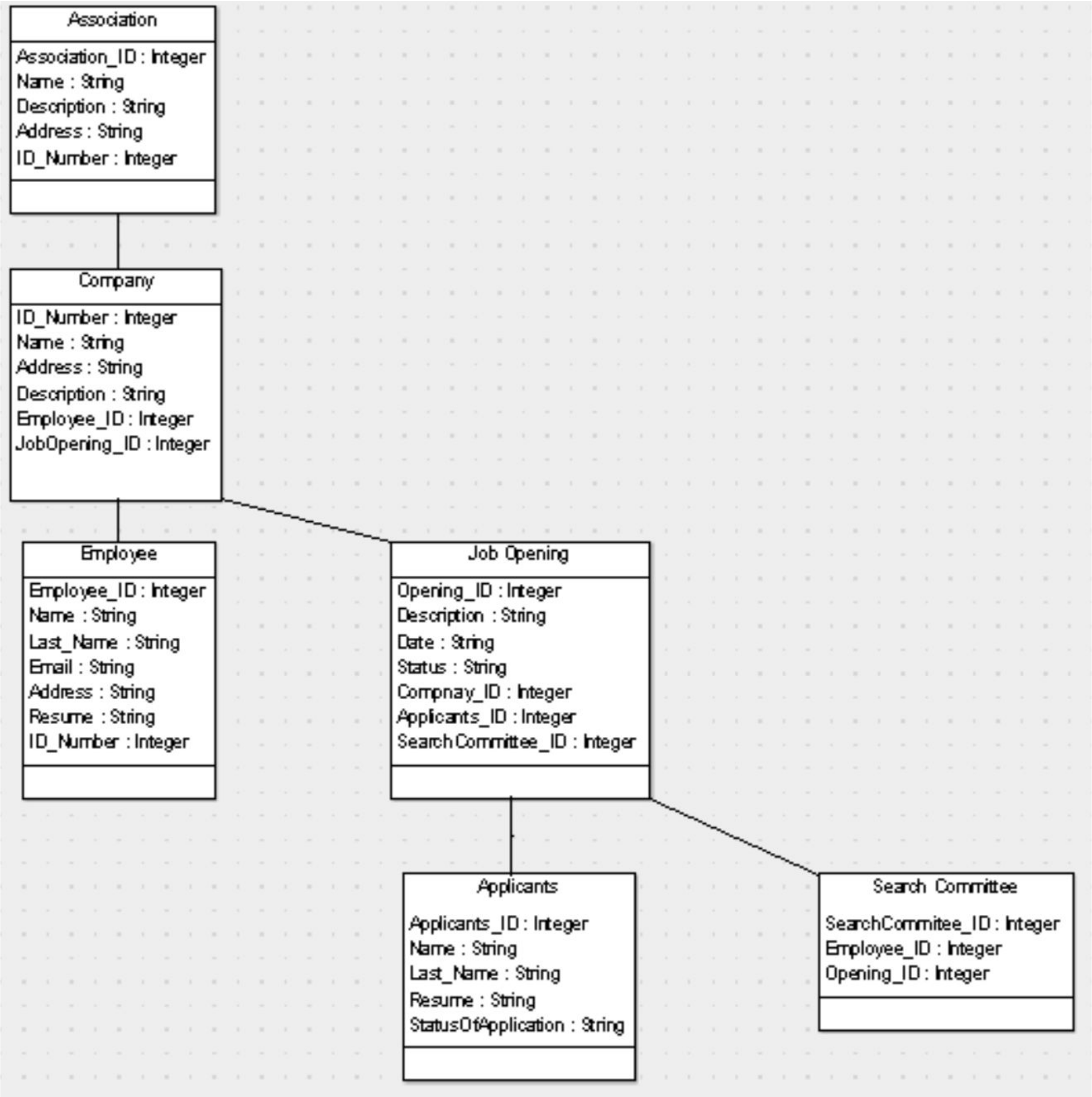
Use Case Diagram



Sequence Diagram



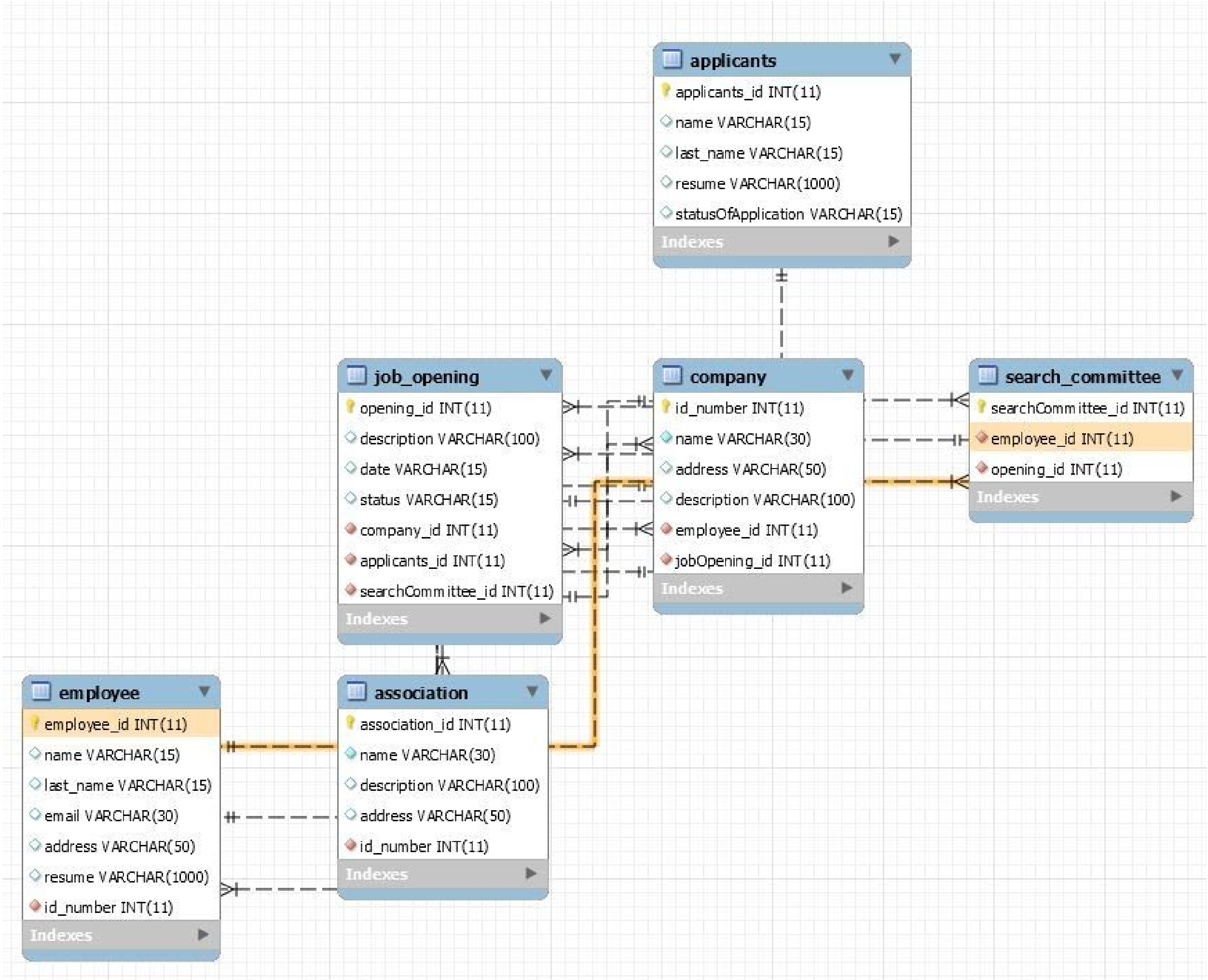
Class Diagram



Database Table Creation Statements

```
CREATE DATABASE csc4330project;

USE csc4330project;

CREATE TABLE IF NOT EXISTS association(
		association_id INT NOT NULL,
    name VARCHAR(30) NOT NULL,
   description VARCHAR(100),
   address VARCHAR(50),
   id_number INT NOT NULL,
   CONSTRAINT association_pk PRIMARY KEY (association_id)
);

CREATE TABLE IF NOT EXISTS company(
        id_number INT NOT NULL,
   name VARCHAR(30) NOT NULL,
   address VARCHAR(50),
   description VARCHAR(100),
   employee_id INT NOT NULL,
   jobOpening_id INT NOT NULL,
   CONSTRAINT company_pk PRIMARY KEY (id_number)
);

CREATE TABLE IF NOT EXISTS employee(
        employee_id INT NOT NULL,
   name VARCHAR(15),
   last_name VARCHAR(15),
   email VARCHAR(30),
   address VARCHAR(50),
   resume VARCHAR(1000),
   id_number INT NOT NULL,
   CONSTRAINT employee_pk PRIMARY KEY (employee_id)
);

CREATE TABLE IF NOT EXISTS job_opening(
        opening_id INT NOT NULL,
   description VARCHAR(100),
   date VARCHAR(15),
   status VARCHAR(15),
   company_id INT NOT NULL,
   applicants_id INT NOT NULL,
   searchCommittee_id INT NOT NULL,
   CONSTRAINT job_opening_pk PRIMARY KEY (opening_id)
);

CREATE TABLE IF NOT EXISTS applicants(
        applicants_id INT NOT NULL,
   name VARCHAR(15),
   last_name VARCHAR(15),
   resume VARCHAR(1000),
   statusOfApplication VARCHAR(15),
   CONSTRAINT applicants_pk PRIMARY KEY (applicants_id)
);

CREATE TABLE IF NOT EXISTS search_committee(
          searchCommittee_id INT NOT NULL,
   employee_id INT NOT NULL,
   opening_id INT NOT NULL,
   CONSTRAINT search_committee_pk PRIMARY KEY (searchCommittee_id)
);

ALTER TABLE association ADD CONSTRAINT association_company_fk FOREIGN KEY
(id_number) REFERENCES company (id_number);
ALTER TABLE company ADD CONSTRAINT company_employee_fk FOREIGN KEY (employee_id)
REFERENCES employee (employee_id);
ALTER TABLE company ADD CONSTRAINT company_job_opening_fk FOREIGN KEY
(jobOpening_id) REFERENCES job_opening (opening_id);
ALTER TABLE employee ADD CONSTRAINT employee_company_fk FOREIGN KEY (id_number)
REFERENCES company (id_number);
ALTER TABLE job_opening ADD CONSTRAINT job_opening_company_fk FOREIGN KEY
(company_id) REFERENCES company (id_number);
ALTER TABLE job_opening ADD CONSTRAINT job_opening_applicants_fk FOREIGN KEY
(applicants_id) REFERENCES applicants (applicants_id);
ALTER TABLE job_opening ADD CONSTRAINT job_opening_search_committee_fk FOREIGN KEY
(searchCommittee_id) REFERENCES search_committee (searchCommittee_id);
ALTER TABLE search_committee ADD CONSTRAINT search_committee_employee_fk FOREIGN
KEY (employee_id) REFERENCES employee (employee_id);
ALTER TABLE search_committee ADD CONSTRAINT search_committee_job_opening_fk FOREIGN
KEY (opening_id) REFERENCES job_opening (opening_id);
```

ER Diagram


applicants
applicants_id INT(11)
name VARCHAR(15)
last_name VARCHAR(15)
resume VARCHAR(1000)
statusOfApplication VARCHAR(15)
Indexes
job_opening
opening_id INT(11)
description VARCHAR(100)
date VARCHAR(15)
status VARCHAR(15)
company_id INT(11)
applicants_id INT(11)
searchCommittee_id INT(11)
Indexes
company
id_number INT(11)
name VARCHAR(30)
address VARCHAR(50)
description VARCHAR(100)
employee_id INT(11)
jobOpening_id INT(11)
Indexes
search_committee
searchCommittee_id INT(11)
employee_id INT(11)
opening_id INT(11)
Indexes
employee
employee_id INT(11)
name VARCHAR(15)
last_name VARCHAR(15)
email VARCHAR(30)
address VARCHAR(50)
resume VARCHAR(1000)
id_number INT(11)
Indexes
association
association_id INT(11)
name VARCHAR(30)
description VARCHAR(100)
address VARCHAR(50)
id_number INT(11)
Indexes