Course ID: CS-487

SOFTWARE ENGINEERING

Report title: FUTURE CAMPUS CONNECT

Team Final Report

Team name - I

Submitted by:-

Nagarjuna Bolla

Karthik Reddy Ereddy

Sailavanya Narthu

Tochi Ukegbu

Table of Content:

SECTION – 1:-

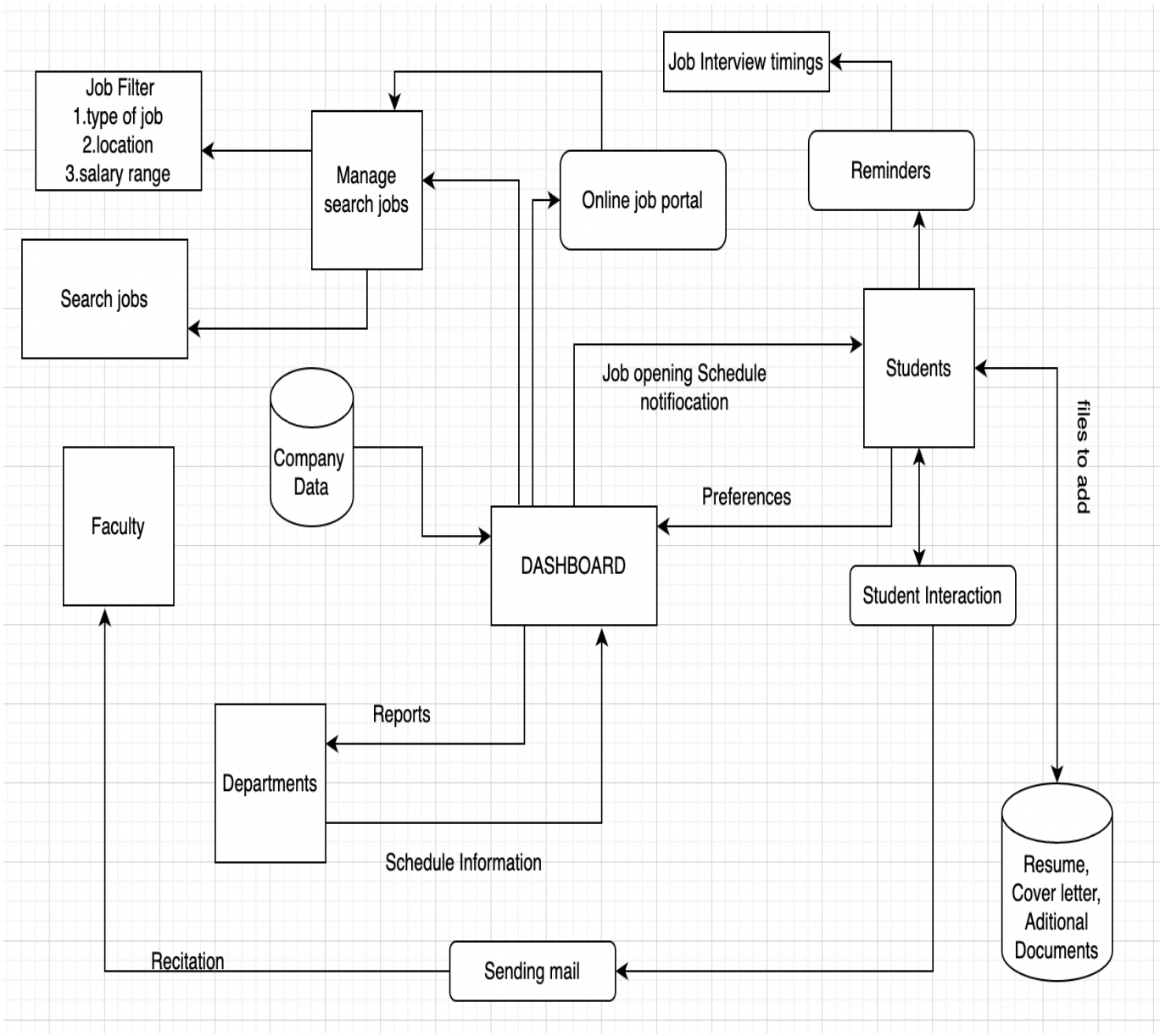
SUMMARY OF THE PROPOSED MODEL

The name of the application is **"FUTURE CAMPUS CONNECT"** which helps students to get their career started by providing various kinds of jobs related to their area of study and helps students to connect with different people who have the same interests. The app also gives a reminder every time a new job is posted and sends a reminder mail a day before the deadline for various job approaches. The application supports student users and various job portals that are linked with the app to list the jobs. The dashboard of the application has various information like personal information of the student which displays all the information like name, major area of study, Job preference whether looking for internships or full-time roles as per the interests etc.., provides all information regarding jobs and helps students to schedule their time also. The application also shows the history of currently studying in various colleges, previous student alumni from their college and from different universities.

The homepage is shown when a student starts the application, but in order to access all of its features, the user must either log in or register an account. First and last names, a UserID, a strong password, an email address, and a phone number must be provided by the user in order to register. The user will also have access to all features once registration is successful.To build database tables, entities that use those tables, establish entity model relationships, and carry out all CRUD actions on the Oracle database, we will be using MySQL Workbench as a middle layer in our application. Students can interact and share information online with their peers using this tool, which is a web-based portal. The objective is to support students in focusing on their careers and in finding better employment in the future. This software serves as a one-stop shop for all of the demands of the students.

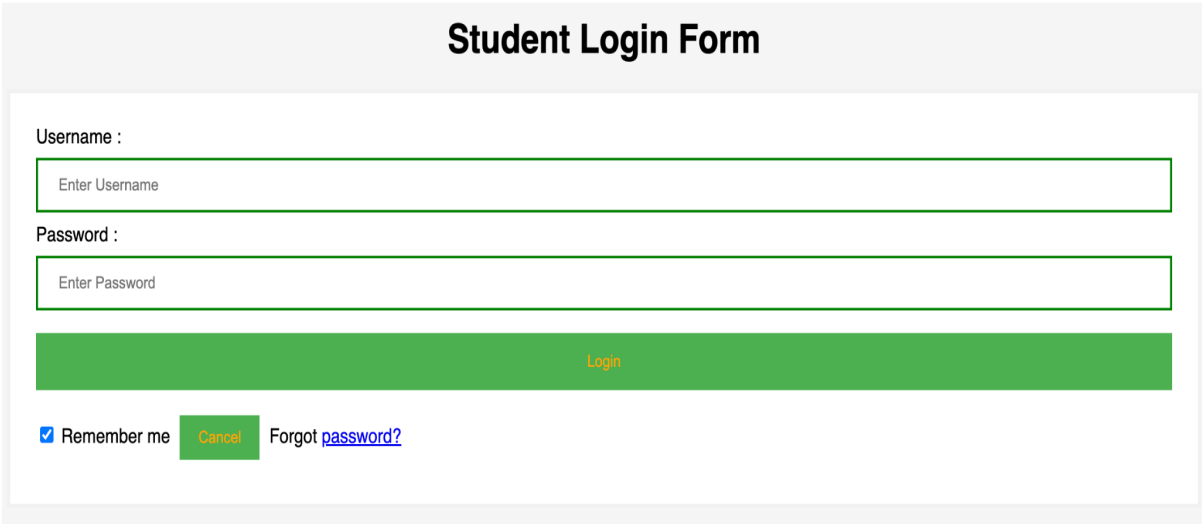
SECTION- 2:-

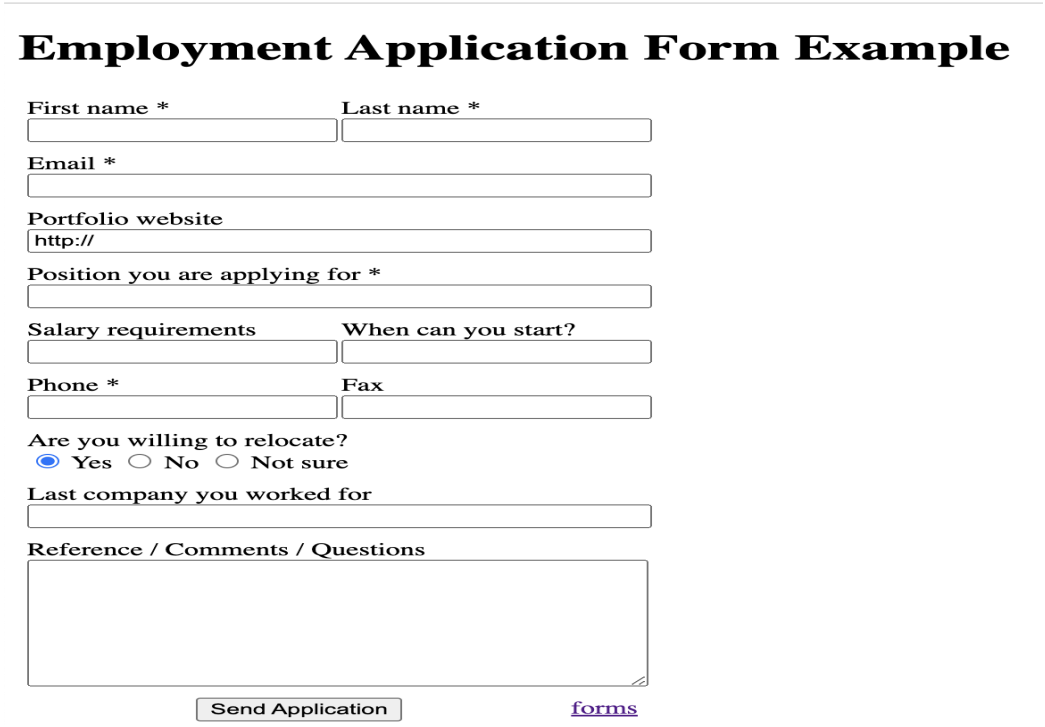
SYSTEM/CONTEXT MODEL

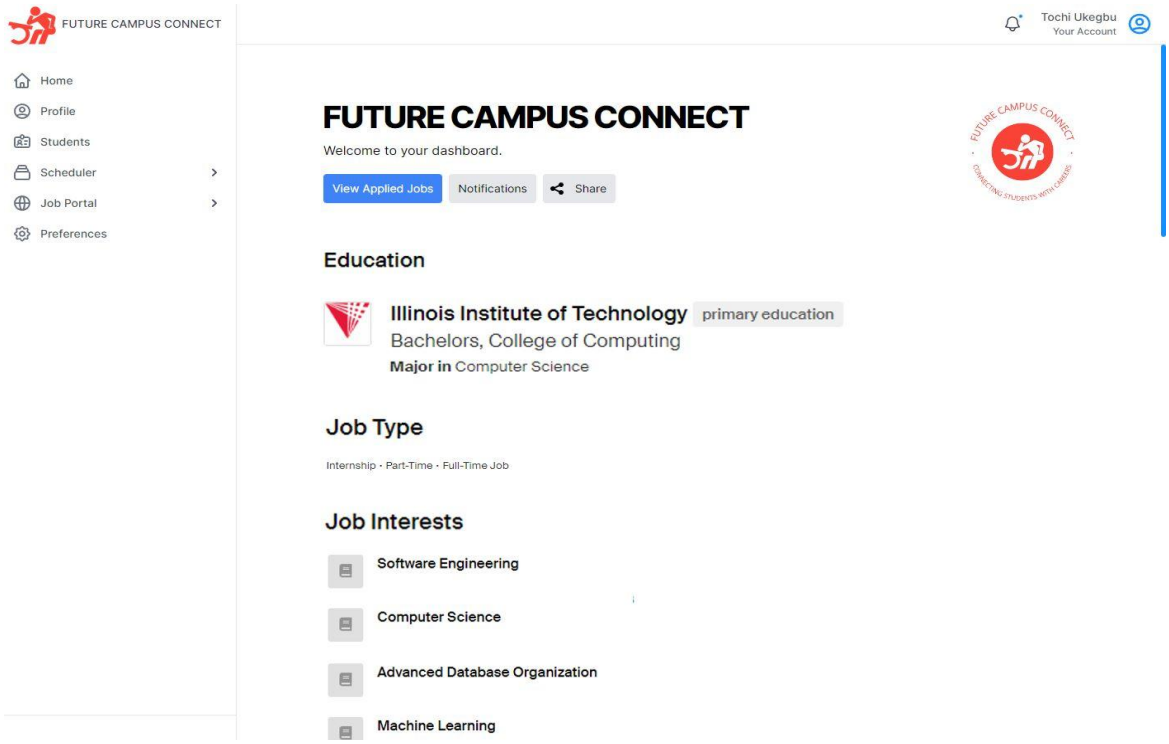


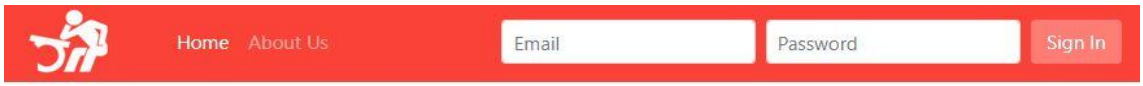
SECTION-3:-

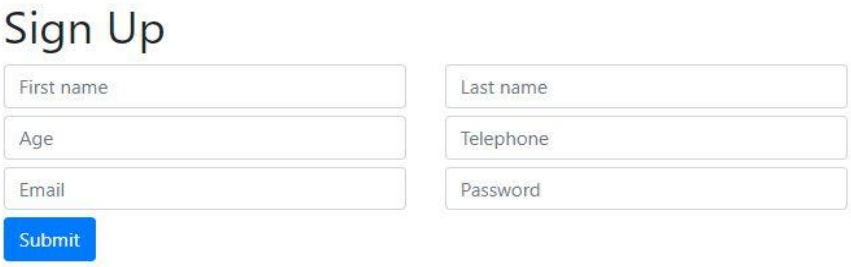
UI AND USER NAVIGATION:

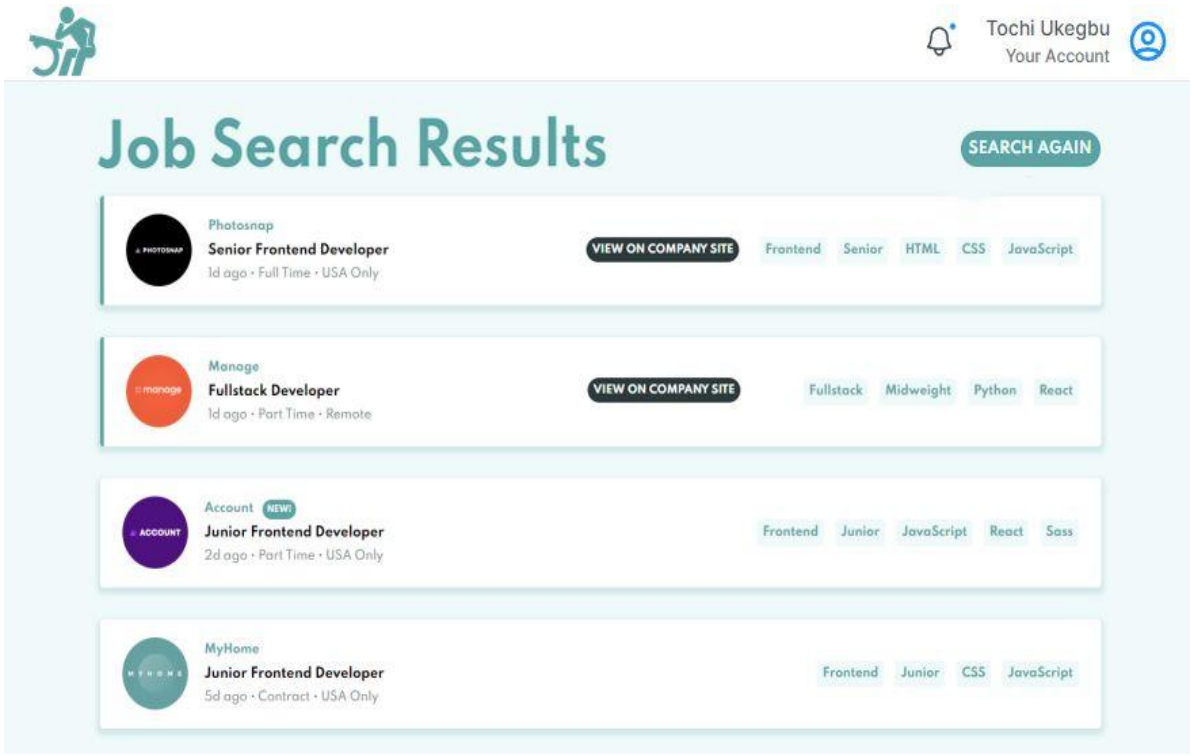


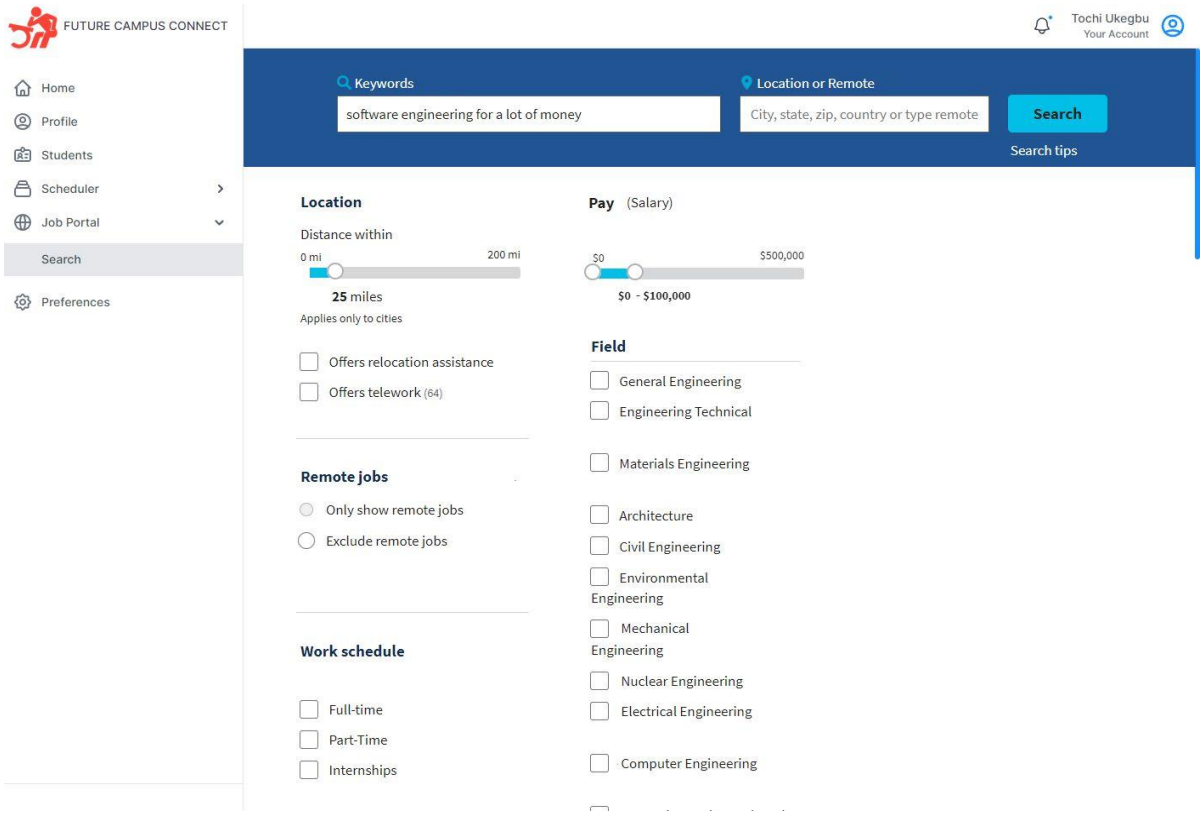



FUTURE CAMPUS CONNECT
Tochi Ukegbu
Your Account
Home
Profile
Students
Scheduler
Job Portal
Preferences
FUTURE CAMPUS CONNECT
Welcome to your dashboard.
View Applied Jobs
Notifications
Share
Education
Illinois Institute of Technology
primary education
Bachelors, College of Computing
Major in Computer Science
Job Type
Internship · Part-Time · Full-Time Job
Job Interests
Software Engineering
Computer Science
Advanced Database Organization
Machine Learning


Home
About Us
Email
Password
Sign In


Sign Up
First name
Last name
Age
Telephone
Email
Password
Submit


Tochi Ukegbu
Your Account
Job Search Results
SEARCH AGAIN
Photosnap
Senior Frontend Developer
1d ago · Full Time · USA Only
VIEW ON COMPANY SITE
Frontend
Senior
HTML
CSS
JavaScript
Manage
Fullstack Developer
1d ago · Part Time · Remote
VIEW ON COMPANY SITE
Fullstack
Midweight
Python
React
Account
NEW!
Junior Frontend Developer
2d ago · Part Time · USA Only
Frontend
Junior
JavaScript
React
Sass
MyHome
Junior Frontend Developer
5d ago · Contract · USA Only
Frontend
Junior
CSS
JavaScript


FUTURE CAMPUS CONNECT
Home
Profile
Students
Scheduler
Job Portal
Search
Preferences
Tochi Ukegbu
Your Account
Keywords
software engineering for a lot of money
Location or Remote
City, state, zip, country or type remote
Search
Search tips
Location
Distance within
0 mi
200 mi
25 miles
Applies only to cities
Offers relocation assistance
Offers telework (64)
Remote jobs
Only show remote jobs
Exclude remote jobs
Work schedule
Full-time
Part-Time
Internships
Pay (Salary)
$0
$500,000
$0 - $100,000
Field
General Engineering
Engineering Technical
Materials Engineering
Architecture
Civil Engineering
Environmental Engineering
Mechanical Engineering
Nuclear Engineering
Electrical Engineering
Computer Engineering

SECTION- 4:-

Pseudo code and key functionalities:

Login Page:

Student login:-

function authenticate(token,){

- Get user information like user id, and password from token
- Validate if given information is correct

if “is_valid”:

get(student)
Generate auth token with a expiration and send back to UI
else:
send “Invalid User” message
}

Signup:

Create a form on UI and take student information and save to the database.

Student:

Section Profile:

.

User Personal Information
function save_student information (parameters, token){
Get user information from UI and save to DB.
}
function get_student_information (student id, token){
Get user info from DB and send back to UI
}

Section Job:

function get jobs (different websites, filters, sort, token){
if filters == NAN:
list of latest jobs based in chronological order by default.
else:
filter based input and send a list of latest jobs based on type of filters.
}

```
function apply_jobs (user_id, filters, sort, token){
  List the jobs related to filter
  If(applied job == not expired):
        Add the job to applied job section
  else:
        Display “job is expired.”
  }


 function withdraw_jobs (user_id, filters, sort, token){
        List the applied jobs
        If (selected job == applied jobs):
        Withdraw from the applied job
        Update the applied jobs list.
  else:
         Display “job is already withdrawn.”
  }
```

Section time management:

```
function student notification(student) {
        Display the new jobs with last dates to apply
        Send reminder to the registered college mail id.
  }
```

Job Websites Linking :-

Section Job:

```
 function home_page (linked websites){
        Display the listed_jobs
        Display the expired_jobs
```

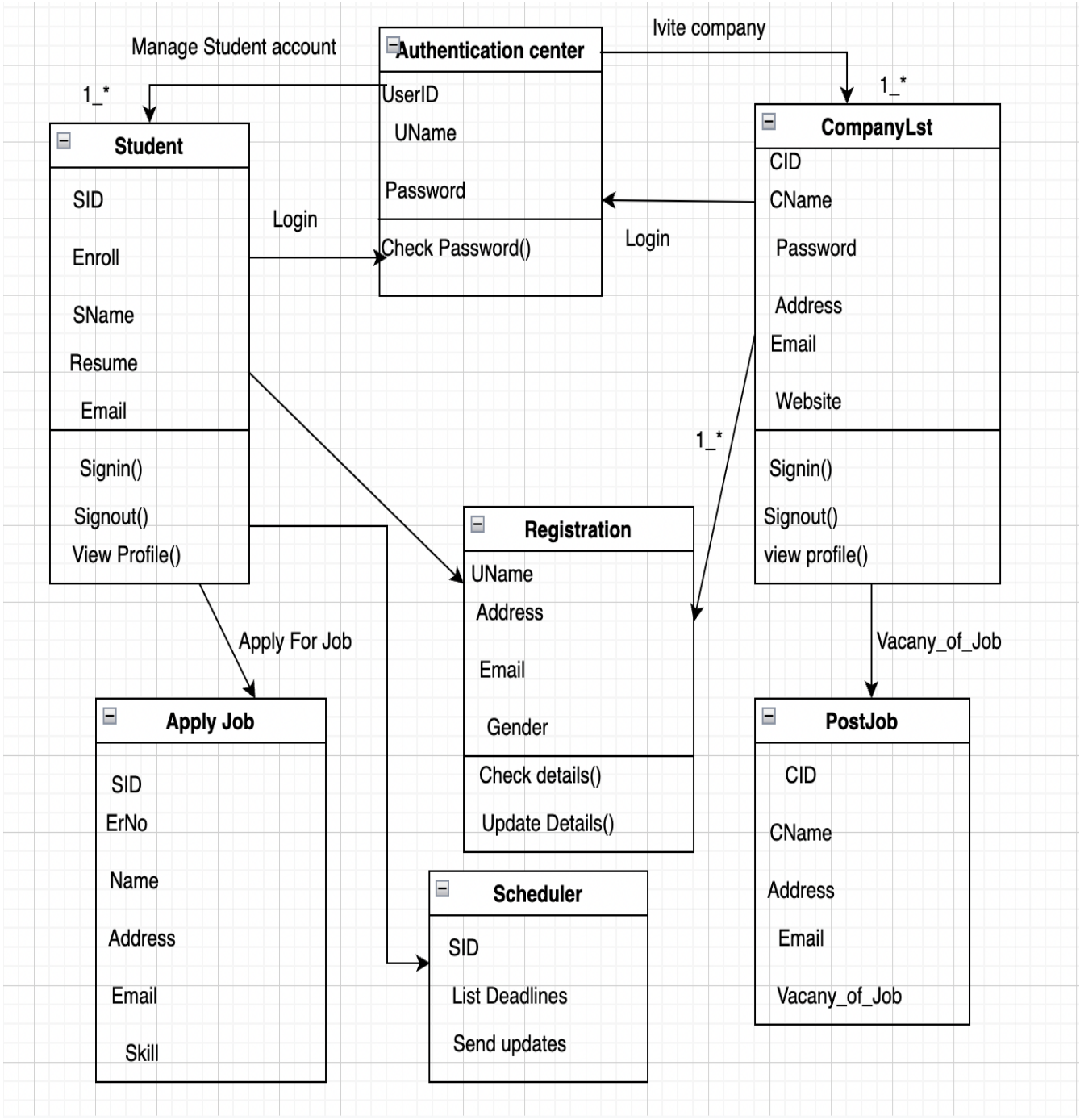
```
    }
 function get_a_job(linked job websites){
  Display the listed_jobs
  If(job_posted exists in listed_jobs):
                              Display “showing job is already posted.”
  else:
                          get the job
                           Update the job in listed_job.
  }
function remove_a_job(linked job websites){
  Display the listed_jobs
          If(job_selected exists in listed_jobs):
                              Remove the job.
                               Update in listed_jobs
          else:
               Display “job is already expired and removed from the DB.”
  }


function edit_a_job(linked job websites){
                          Display the listed_jobs
                    If(job_selected exists in listed_jobs):

                                              Update about job in listed_jobs.
               else:
                                              Display “job doesn’t exist.”
                }
```

SECTION-5

Data Model:



SECTION - 6 :-

Design Approach:

Security:

This is achieved by requesting a student registered email id and password from the students in order to access the application.

Security can be set up by asking college specific mail ids which end with some indications when they sign up, which will then check to see if it already exists in our database.

Performance:

The information is kept in a database.

SQL statements execute database queries to access data. The way SQL statements are constructed makes it simple to access data.

SQL optimization tools were employed to examine the access routes that the DB chooses to perform SQL statements in order to identify any potential issues and enhance performance.

Increased performance while using stored procedures.

The application has been designed to be simple to use for students.

Ease of use:

The program is made accessible to both administrators and users.

Students can easily find jobs and submit job applications without any difficulties.

It is simple to find information on employment to look at the schedules and due dates.

Test Result:

Test Case:Collaboration I

Initial State: User 1 and User 2 are logged into the application and are part of the same course

Input: User 1 posts a question

Input: User 2 views posted questions

Expected End State: User 2 sees the question posted by User 1

Test Case: Collaboration II

Initial State: User 1 has posted a question

Input: User 2 replies to a question

Expected End State: User 1 sees the reply posted by User 2

Test Case: Time Management

Initial State: User 1 is logged into the application

Input: User sets a reminder on a school activity and the scheduled date arrives

Expected End State: User is notified of school activity

Test Case: Access to Course Material

Initial State: User is logged into the application

Input: User registers course as in-progress on the application

Expected End State: App recommends books relevant to the course, either in the course syllabus or of the same topic as the course

Conclusion:

People sometimes hesitate to interact with a stranger and ask their queries in a new environment.This web application provides a solution to the problem. So, this proposed interactive and intelligent connection app fosters the students to explore a lot and socialize with the fellow students and company recruiters regarding the jobs.

References:

- Environment. Technology. Resources: Proceedings of the 11th International Scientific and Practical Conference, vol. 2, pp. 24-29, 2017. H. Gorskis, L. Aleksejeva, and I. Polaka, "Ontology-Based System Development for Medical Database Access."
- Artificial Intelligence, vol. 262, no. 2018, pp. 52–95, M. Benedikt, B. Cuenca Grau, and E. V. Kostylev, "Logical underpinnings of information disclosure in ontology-based data integration."
- Information Technology and Management Science, vol. 20, no. 1, pp. 69–73, 2017. H. Gorskis, L. Aleksejeva, and I. Polaka, "Database Concepts in a Domain Ontology."
- The Essentials of Programming Languages was published in 1992 by Friedman, Wand, and Haynes. Cambridge, MA: MIT Press and McGraw-Hill.
- The second version of Practical Unix & Internet Security was published in 1996 by S. Garfinkel and G. Spafford. California's Sebastopol-based O'Reilly & Associates
- (PDF) Software Engineering Reference Framework - ResearchGate. (n.d.). Retrieved December4,2022,from-https://www.researchgate.net/publication/254857111_Software_engineering_reference_framework
- *(PDF) Software Engineering Reference Framework - ResearchGate*. (n.d.). Retrieved December4,2022,from-https://www.researchgate.net/publication/254857111_Software_engineering_reference_framework
- *HTMLreference*.Dofactory.(n.d.).RetrievedDecember3,2022,from-https://www.dofactory.com/html/ref