# High Level Design & Low Level Design

**Document Control :**

## Project Revision History

| Date | Version | Author | Brief Description of Changes | Approver Signature |
|---|---|---|---|---|
| 27.09.2022 | 1.0 | Group 7 | | |
| | | | | |
| | | | | |

# Index

# 1. Introduction: -

The introduction of the software requirement specification provides an overview of the entire software. The entire SRS with an overview description of the purpose, scope, tools used, and basic description. The aim of this document is to gather, analyze and give an in-depth insight into the complete University Alumni Management System by defining the problem statement in detail. The detailed requirements of the University Alumni Management System are provided in this document.

## 1.1 Intended Audience: -

This document is intended to be read by the Client.

## 1.2 Project Purpose: -

The purpose of this document is to show the requirements for the University Alumni Management System, which helps the college or university to keep track of the alumni students, by doing this it will help the students who are currently in that college or university to know about the job applications in the same field and to know about the technology which is in trend. This portal helps to connect students to alumni and helps to build their careers.

## 1.3 Key Project Objectives: -

a. Allow the University Students to add their data and login to the portal.
b. Allows the Alumni to enter the details and login to the portal.
c. Allows Alumni to create and update Job Posts.
d. Allows Students to view Job Posts and Alumni Details.
e. Allows Students to ask for Connect sessions.
f. Allows Admin to login to the portal and modify/ delete the records of Alumni.
g. Allows Admin to generate Connect session reports and delete that report.

h. Allows Admin to generate non placed student reports domain wise.

## 1.4 Project scope: -

This project aims to create the development of a University Alumni Management application in which administrators, students, and alumni will be able to continuously communicate with each other. This provides access to the student to access alumni information and posts, which will help students in their career.

## 1.5 Functional Overview: -

1.5.1 Following header files are included in the program:

a #include <iostream>
b #include <iomanip>
c #include <fstream>
d #include <string>
e #include <cctype>
f #include <pthread>
g #include <list>
h #include <map>
i #include <tuple>

1.5.2 Following functions are included in the program:

**Alumni Module**: An Alumni can first register by entering his details, once registered, alumni can log in and post the job-related contents for the other users so that they can access.

**1. alumni_get_data()**- It will register the new alumni using a unique university registration number. Alumni can enter details like name, address, telephone no., email id, department, company, years of experience, and salary.

**2. Alumni_Login ()** - After the registration is done th Alumni can login to the portal using their username and password.

**3. Edit_Details ()** - After entering username and password, alumni can update their details like Company name, salary, designation, technology, location and years of experience.

**4. Create_posts ()** - After login, alumni can create the job profile, technical advice/materials materials.

**5. view_posts ()** -Alumni can view the post for job profile, technical advice/materials.

**6. view_Profile ()** - Alumni can view his profile through this function.

**Student Module:** A student can register in the portal. If the student already has his login credentials then he can go into the portal and view all the information posted by alumni like job updates/technical advice/materials. Also can view their own department alumni details. Students can also edit their information like telephone number and email.

**7. student_get_data()** : The register function will register the new student into the portal by his unique university registration number.

**8. Student_Login()** : If a student has already registered into the portal, they can log in to the portal by giving their passcode and register number.

**9. view_profile()** : After the login is successful, the student can view his/her profile.

**10. Edit_Info()** : Students can also edit and update their details in the student database like telephone number, email id, etc. by using the edit info function.

**11. View_Alumni_Details** : Students can view the details of alumni.

**12. Job_Posts()** : Students can also see the current job offers, a list of which will be displayed.

**13. ask_for_connect()** : Students can ask for a connect with alumni.

**Admin Module:** Admin can log in using a password and can modify, delete, view records of Alumni database and as well student database too and prepare a report of students who are not placed. He can also enter a domain and see which alumni and students belong to that department.Generate the Report for Connect Session and also generate the Report for completed connect,once the connect is over it should be deleted .

**14. Edit_Alumni() :** Admin can login by giving the password and can edit the alumni details.

**15. Delete_Alumni_details() :** Admin can login by giving the password and can delete the student details.

**16. Report_Nonplaced_Student() :** Generate the Report for the students who are not placed in any company.

**17. Completed_Connect_Report() :** Generate a list of all connects already carried out.

**18. Delete_connect() :** After the connect is carried out delete that connect record from the file.

**19. list_of_connects():** It contains all the list of scheduled and the completed connects.

University Alumni Management comprises of three modules and following member functions:

| Name of the Module | Main menu |
|---|---|
| Handled by | Gayathri Sakthivel |
| Description | It consists of the main menu and three sub menus which will provide options to the user. |

| Name of the Module | Validations |
|---|---|
| Handled by | Riya Malik |
| Description | It consists of all the functions that validates all the inputs. |

| Name of the Module | Alumni_Register() |
|---|---|
| Handled by | Gayathri Sakthivel |
| Description | Register the new alumni using a unique university registration number. Alumni can enter details like name, address,telephone no., email id, department, company, years of experience, and salary. |

| Name of the Module | Alumni_login() |
|---|---|
| Handled by | Poojitha B |
| Description | Alumni can login to the portal and access various services using their username and password . \ |

| Name of the Module | Create_Posts() |
|---|---|
| Handled by | Poojitha B |
| Description | After login, alumni can create the job profile, technical advice/materials  materials. |

| | |
|---|---|
| Name of the Module | view_posts() |
| Handled by | Gayathri Sakthivel |
| Description | Alumni can view the post for job profile, technical advice/materials. |

| | |
|---|---|
| Name of the Module | view_Profile() |
| Handled by | Poojitha B |
| Description | Alumni can view his profile through this function. |

| | |
|---|---|
| Name of the Module | student_get_data() |
| Handled by | Avula Shashi Sushmitha Reddy |
| Description | The register function will register the new student into the portal by his unique university registration number. |

| | |
|---|---|
| Name of the Module | Student_Login() |
| Handled by | Rehana Praveen S |
| Description | If a student has already registered into the portal, they can log in to the portal by giving their passcode and register number. |

| | |
|---|---|
| Name of the Module | view_profile() |
| Handled by | Avula Shashi Sushmitha Reddy |
| Description | After the login is successful, the student can view his/her profile. |

| | |
|---|---|
| Name of the Module | Edit_Info() |
| Handled by | Avula Shashi Sushmitha Reddy |
| Description | Students can also edit and update their details in the student database like telephone number, email id, etc. by using the edit info function. |

| | |
|---|---|
| Name of the Module | View_Alumni_Details() |
| Handled by | Riya Malik |
| Description | Students can view the details of alumni of the same departments. |

| | |
|---|---|
| Name of the Module | Job_Posts() |
| Handled by | Rehana Parveen S |
| Description | Students can also see the current job offers, a list of which will be displayed. |

| | |
|---|---|
| Name of the Module | ask_to_Connect() |
| Handled by | Riya Malik |
| Description | Students can ask for a connect with a alumni of his department. |

| | |
|---|---|
| Name of the Module | Edit_Alumni() |
| Handled by | Rehana Parveen S |
| Description | Admin can login by giving the password and can edit the alumni details. |

| | |
|---|---|
| Name of the Module | store_data() |
| Handled by | Riya Malik |
| Description | It will store the data from the alumni list to the alumni file. |

| | |
|---|---|
| Name of the Module | student_store_data() |
| Handled by | Poojitha |
| Description | It will store the data from the student list to the student file. |

| | |
|---|---|
| Name of the Module | student_fetch_data() |
| Handled by | Rehana Praveen S |
| Description | It will fetch the data from the student file to the student list. |

| | |
|---|---|
| Name of the Module | Report_Nonplaced_Student() |
| Handled by | Riya Malik |
| Description | Generate the Report for the students who are not placed in any company. |

| | |
|---|---|
| Name of the Module | fetch_data() |
| Handled by | Riya Malik |
| Description | It will fetch the data from the alumni file to the alumni list. |

| | |
|---|---|
| Name of the Module | Completed_Connect_Report() |
| Handled by | Avula Shashi Sushmitha Reddy |
| Description | Generate a list of all connects already carried out. |

| | |
|---|---|
| Name of the Module | Delete_connect() |
| Handled by | Gayathri Sakthivel |
| Description | After the connect is carried out, delete that connect record from the file. |

| | |
|---|---|
| Name of the Module | Delete_Alumni_details() |
| Handled by | Rehana Praveen S |
| Description | Admin can login by giving the password and can delete the student details. |

| | |
|---|---|
| Name of the Module | List_of_connects() |
| Handled by | Riya Malik |
| Description | It contains all the list of scheduled and the completed connects. |

## 2. Design Overview

### 2.1Design Objectives: -

- Alumni
  - ➢ Alumni get data
  - ➢ Alumni login
  - ➢ Edit details
  - ➢ View Posts
  - ➢ Create Posts
  - ➢ View Profile

- Student
  - ➢ Student get data
  - ➢ Student Login
  - ➢ View profile
  - ➢ Edit Info
  - ➢ View Alumni Details
  - ➢ Job posts
  - ➢ Ask for Connect

- Admin
  - ➢ Edit Alumni
  - ➢ Edit Student
  - ➢ Report Non placed Students
  - ➢ Connect Session
  - ➢ Completed Connect Report
  - ➢ Delete Connect

### 2.2 Design Alternative: -

We have used Standard Template Library (STL) to perform all the relevant operations in the particular file .

### 2.3 User Interface Paradigms: -

The project aims to create the development of a University Alumni Management application in which administrators, students, and alumni will be able to continuously communicate with each other. User is given an interface to add a new record in case he/she wants to add a record, update a record, view the details.

### 2.4 Error Detection / Exceptional Handling: -

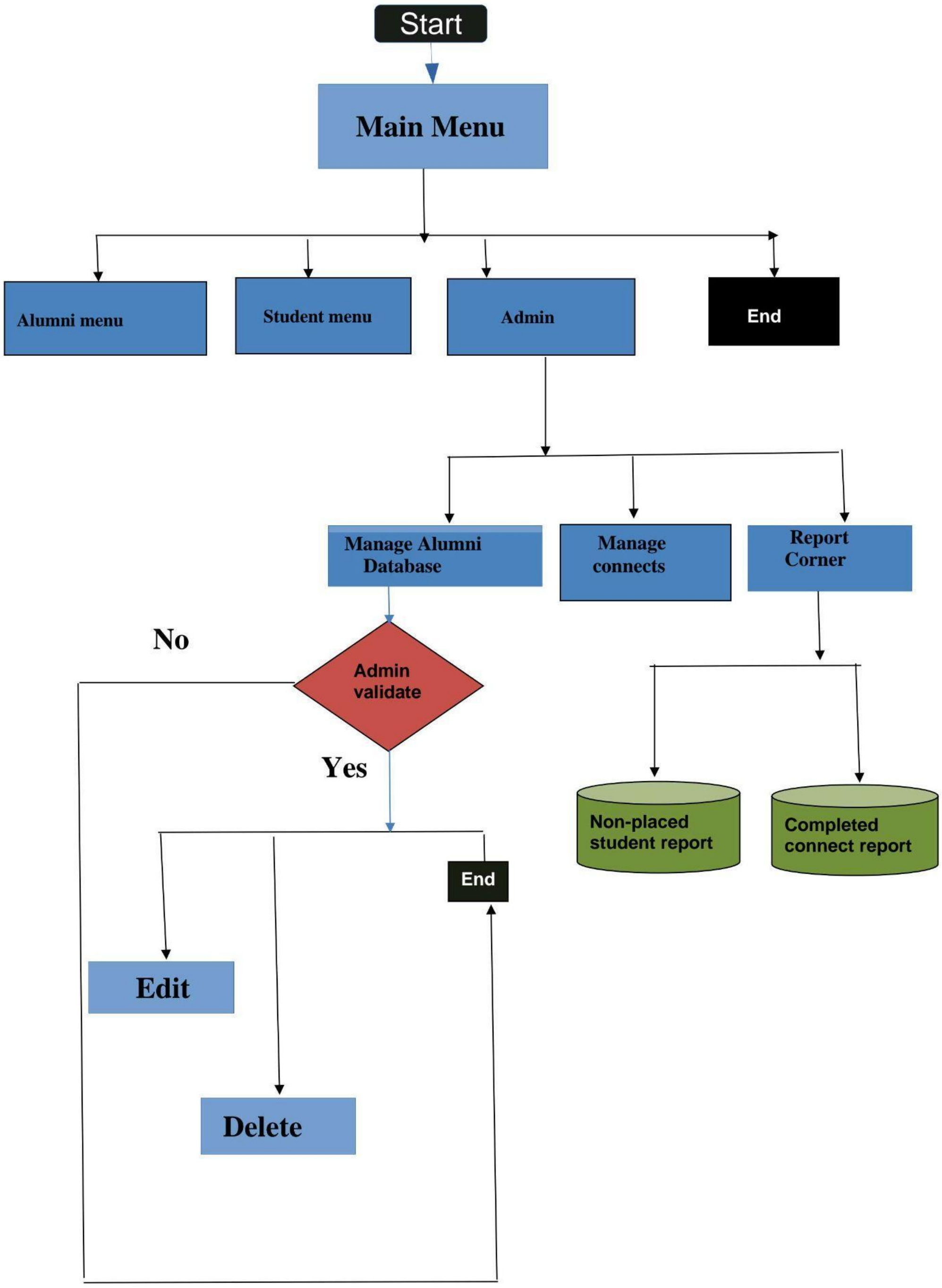
- User should first enter the details according to the condition and if the entered detail is not according to the condition specified, sometimes it displays the message that is entered and sometimes it returns with an error.

### 2.5 Validations

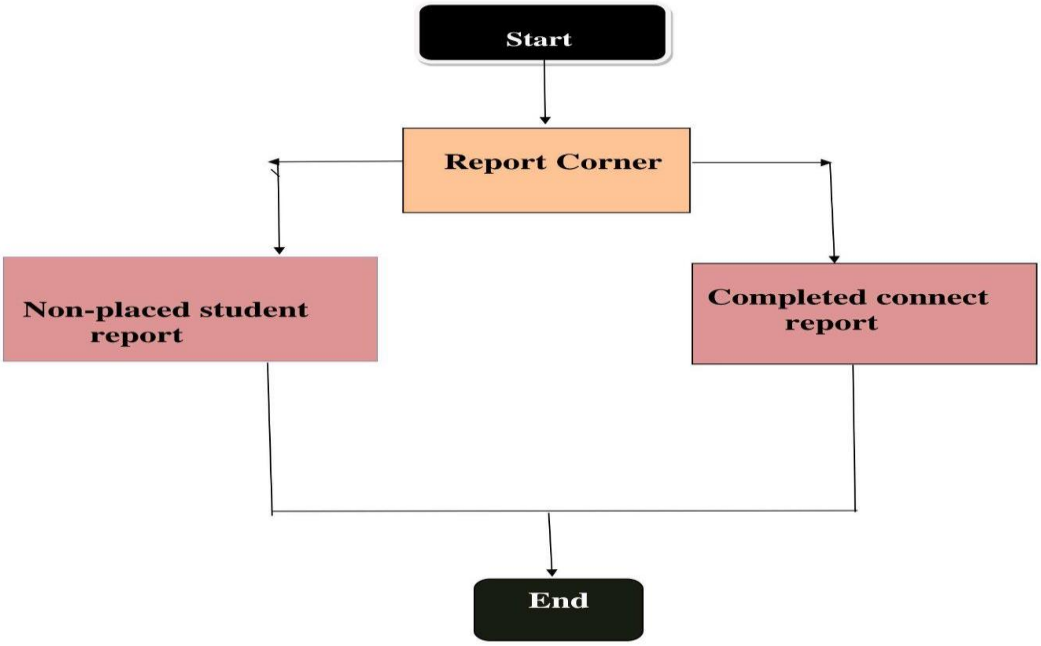
- The registration number should not be blank and it should be 6 letters long.
- The registration number for students should start with the US and for Alumni it should start with UA.
- The Telephone number length should be 10 and it should contain only Integers.
- The student and Alumni name should not be empty and it should contain only Alphabets.
- The Domain must be among CSE, ECE, EEE, MECH and CIVIL.

# 3.DETAILED SYSTEM DESIGN

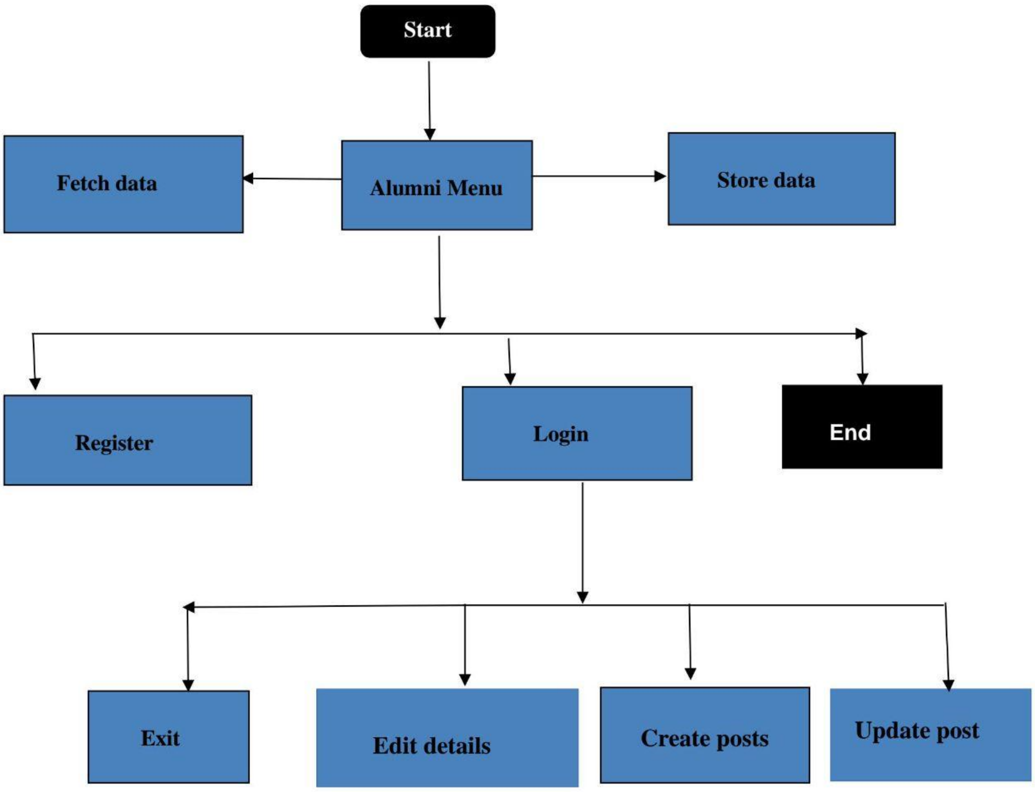
## 3.1 Flow chart of main menu



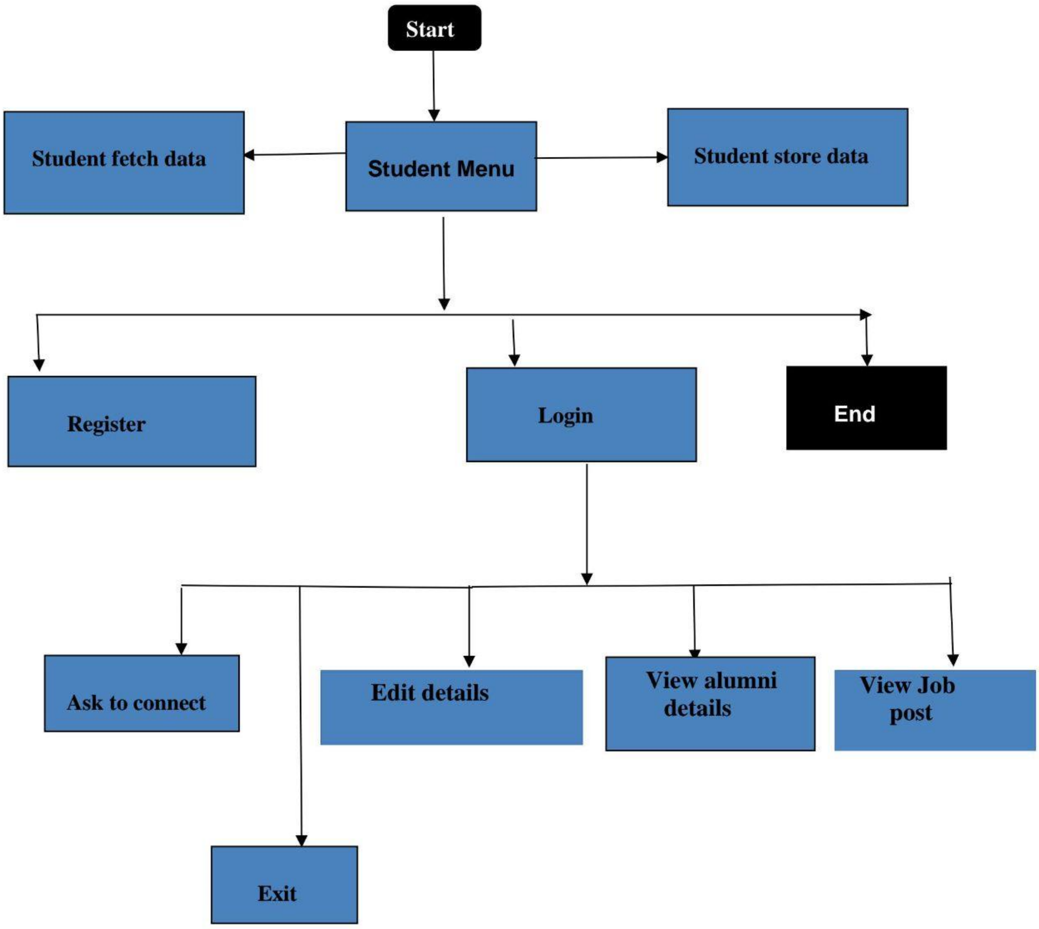
## 3.2 Flow chart for Reports



### 3.3 Flow chat for Alumni menu

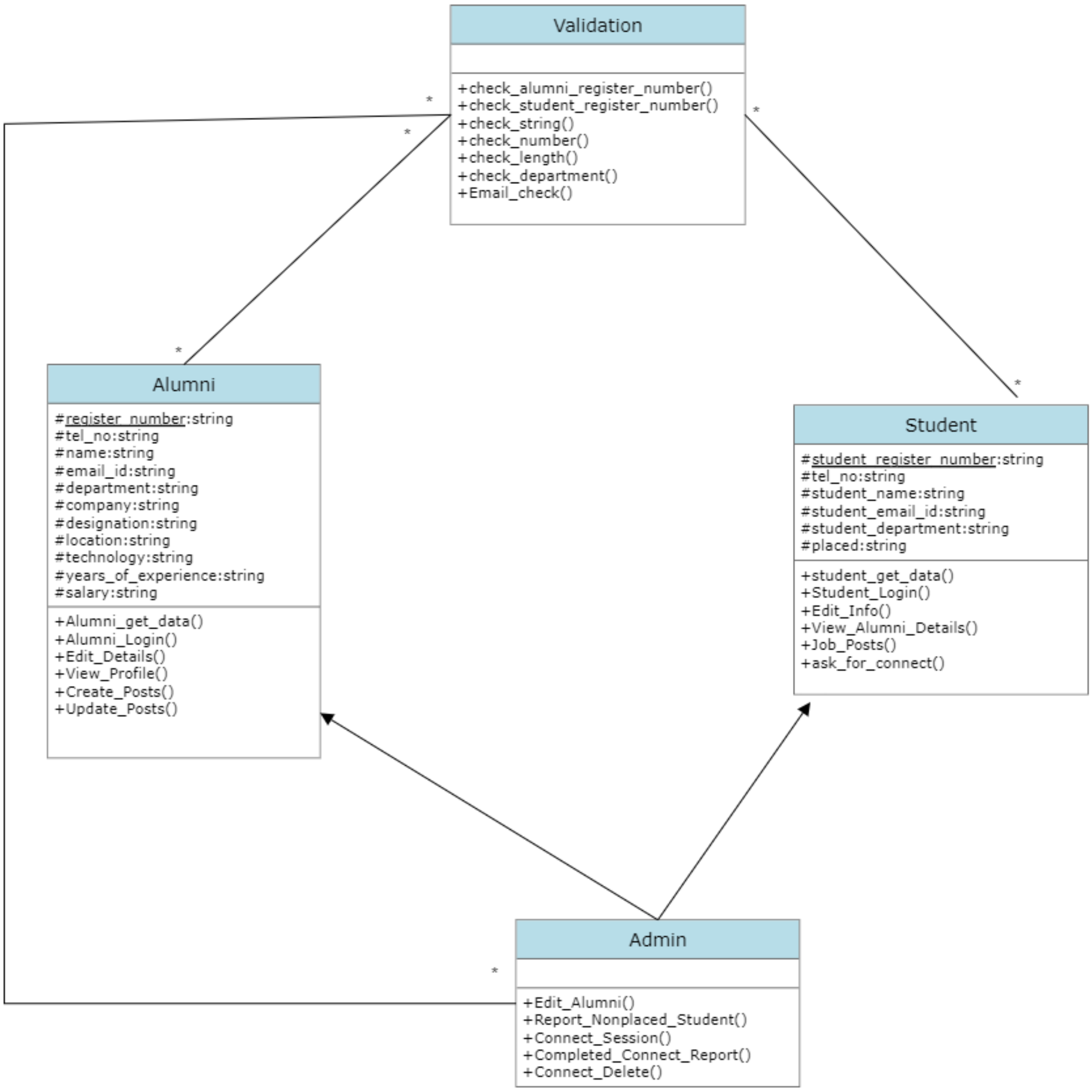


### 3.4 Flow chart for Student menu

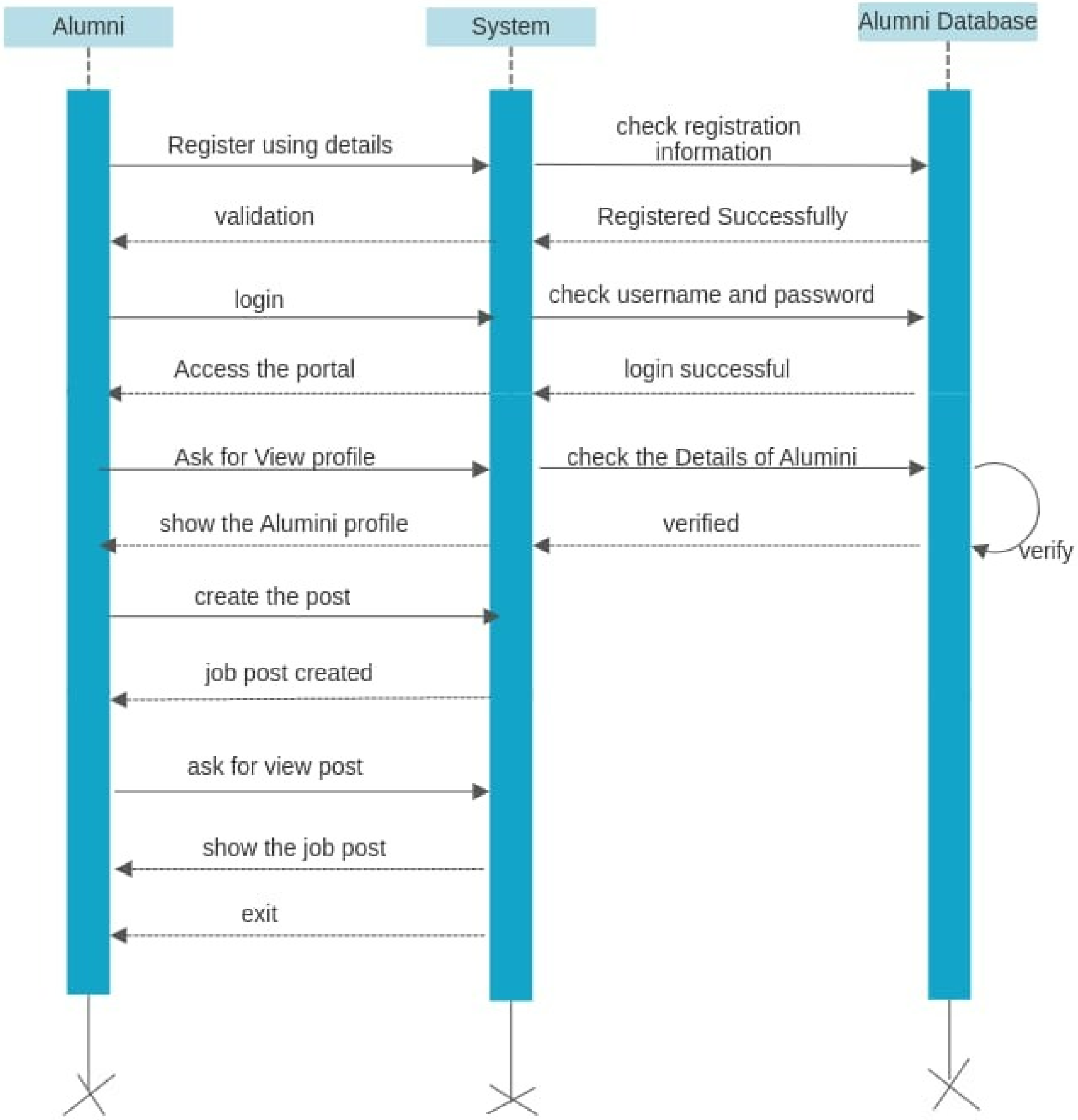


# 3. UML Diagrams

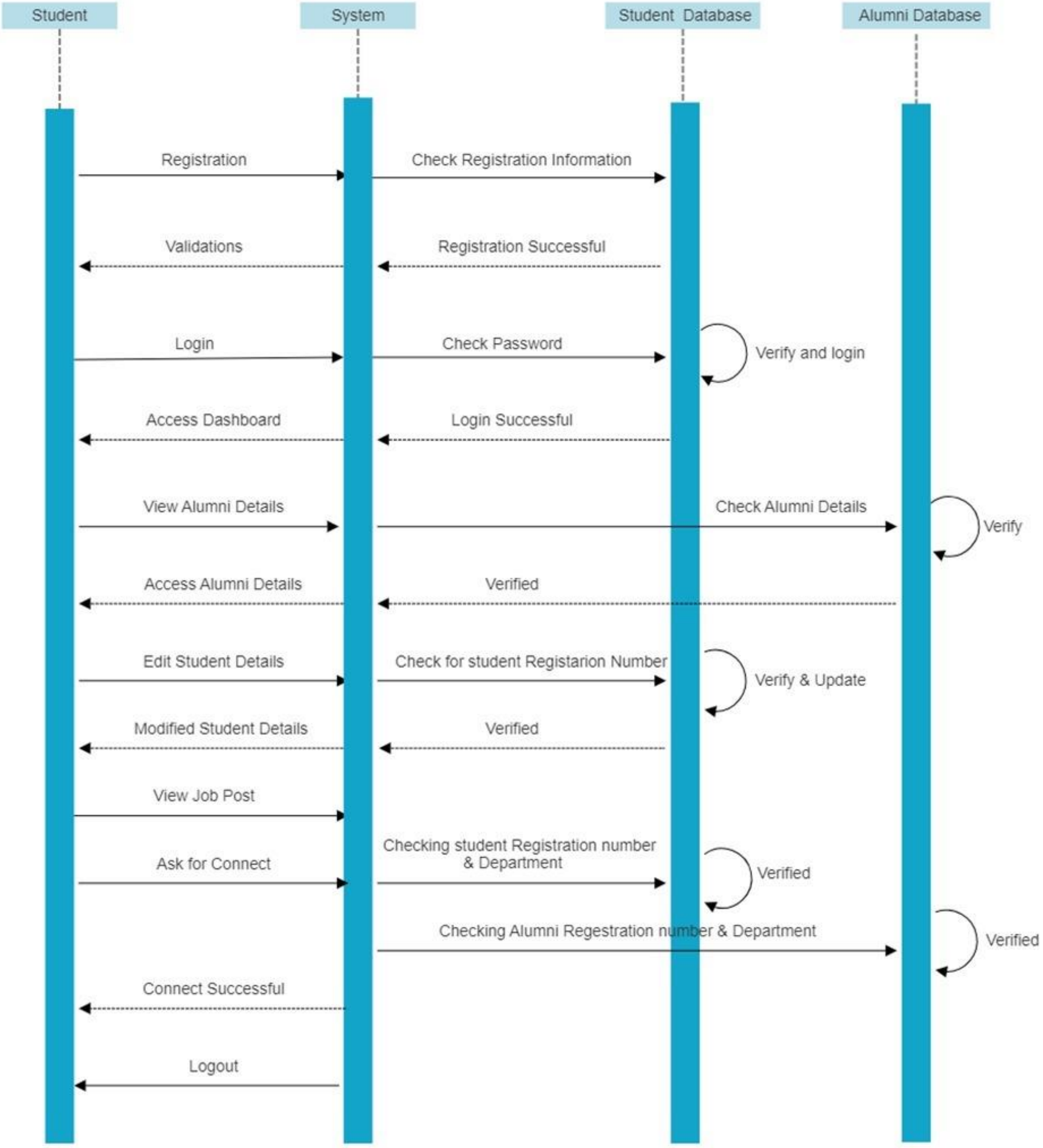
## 3.1 Class Diagram



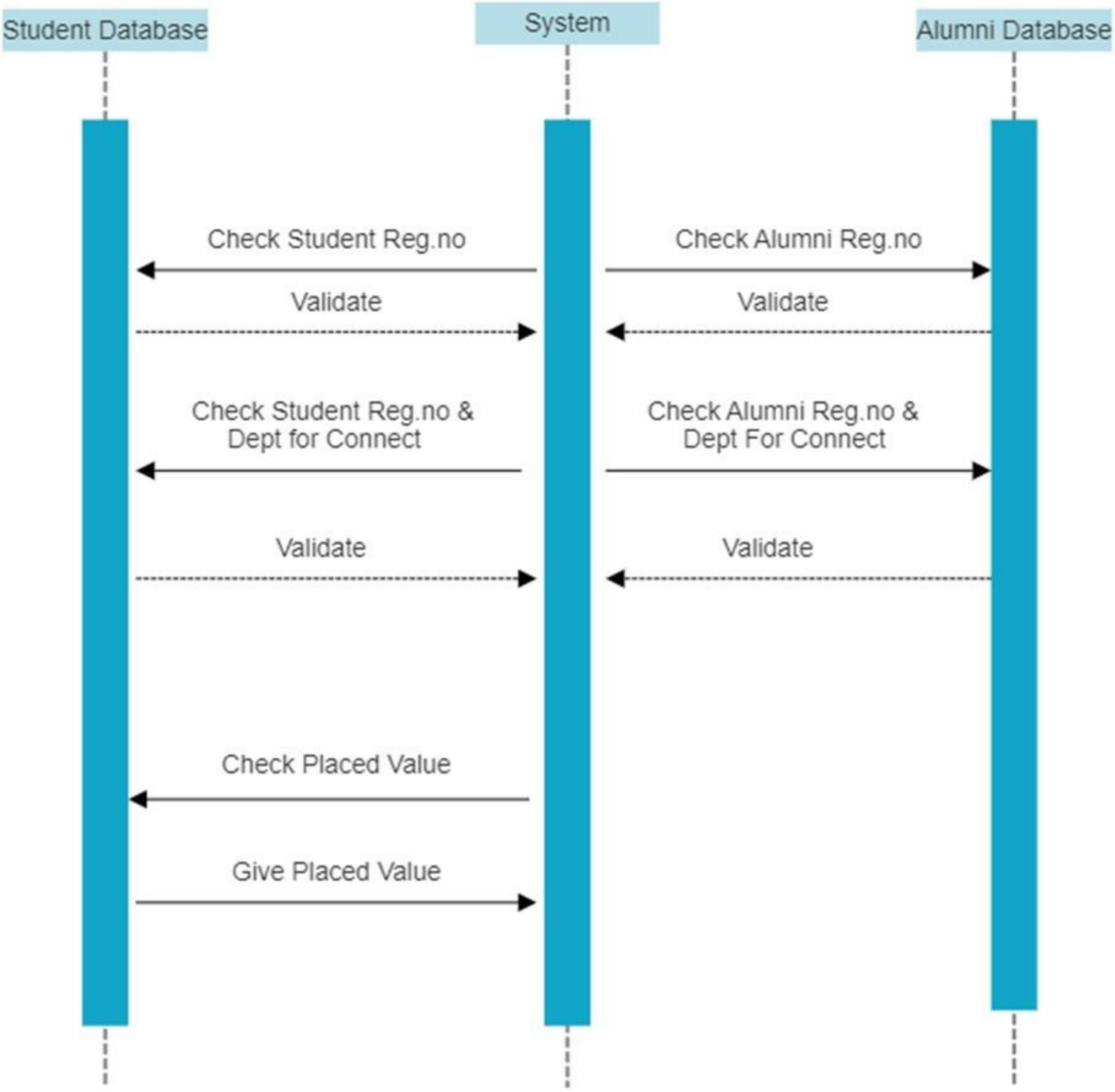
3.2 **Sequence Diagram for Alumni**



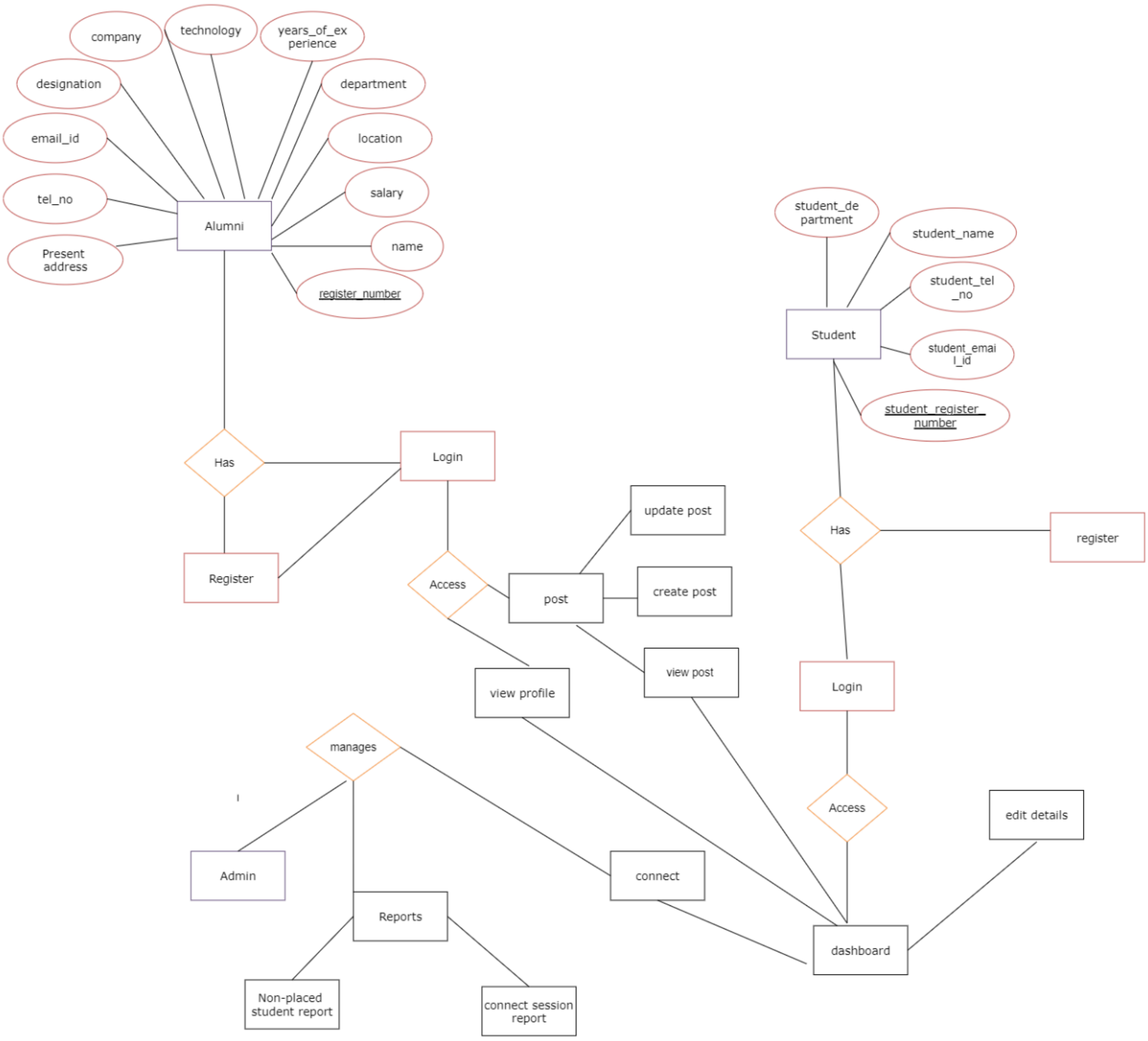
### 3.3 Sequence Diagram for Student



### 3.4 Sequence diagram for Admin



### 3.5 ER Diagram



## 5. Environment Description: -

**5.1 Time Zone Support: -** IST

**5.2 Language Support: -** English

**5.3 User Desktop Requirements: -**

a. 64-bit processor, 1 GHz or faster
b. At least 2 GB free hard drive space
c. At least 1 GB RAM

**5.4.integration Requirements: -**

a. Language: - C++
b. Tools: - Gcov, Valgrind, Makefile ,Cppunit,Gprof,CPPcheck
c. Compiler: - g++
d. Linux Environment

**5.5 Network**: - End to End

**5.6 Configuration: -**

5.6.1 Operating System: - Linux environment

## 6. Reference: -

**https://www.geeksforgeeks.org/list-cpp-stl/**

**https://www.geeksforgeeks.org/thread-functions-in-c-c/**

**https://www.geeksforgeeks.org/file-handling-c-classes**

**https://gcc.gnu.org/onlinedocs/gcc/Gcov.html**

**https://www.tutorialspoint.com/unix_commands/gprof.htm#:~:text=Gprof%20reads%20the%20given%20object,in%20the%20given%20profile%20files.**

**https://www.tutorialspoint.com/makefile/index.htm#:~:text=Makefile%20is%20a%20program%20building,help%20of%20user%2Ddefined%20makefiles.**