



University Alumni Management System

UAMS_DesignDocument



High Level Design & Low Level Design

Design

Document Control :

Date	Version	Author	Approver Signature
26.10.2022	0.1 v	GROUP -3	

TEAM MEMBERS

EMP Number:	Name
46274556	Jigyanshu Ranjan Rautaray
46274795	Krishna Priyanka R
46275269	Gangula Vijitha
46274555	Ganta Supriya
46274776	Gummella Sowjanya
46275328	Boggula Krishna Sri

1. INTRODUCTION 6

1.1. INTENDED AUDIENCE	6
1.2. ACRONYMS/ABBREVIATIONS	6
1.3. PROJECT PURPOSE	6
1.4. KEY PROJECT OBJECTIVES	6
1.5. PROJECT SCOPE AND LIMITATION	7
1.5.1. In Scope	7
1.5.2. Out of scope	7
1.6. FUNCTIONAL OVERVIEW	7
1.7. ASSUMPTIONS, DEPENDENCIES & CONSTRAINTS	7
1.8. RISKS	7

2. DESIGN OVERVIEW 8

2.1. DESIGN OBJECTIVES	8
2.1.1. Recommended Architecture	8
2.2. ARCHITECTURAL STRATEGIES	8
2.2.1. Design Alternative	9
2.2.2. Reuse of Existing Common Services/Utilities	9
2.2.3. Creation of New Common Services/Utilities	9
2.2.4. User Interface Paradigms	9
2.2.5. System Interface Paradigms	9
2.2.6. Error Detection / Exceptional Handling	9
2.2.7. Memory Management	9
2.2.8. Performance	10
2.2.9. Security	10
2.2.10. Concurrency and Synchronization	10
2.2.11. Housekeeping and Maintenance	10

3. SYSTEM ARCHITECTURE	11
3.1. SYSTEM ARCHITECTURE DIAGRAM. (NOT NECESSARY)	11
3.2. SYSTEM USE-CASES	11
3.3. SUBSYSTEM ARCHITECTURE	12
3.4. SYSTEM INTERFACES	13
3.4.1. Internal Interfaces	13
3.4.2. External Interfaces	13
4. DETAILED SYSTEM DESIGN	
4.1. KEY ENTITIES	14
4.2. DETAILED-LEVEL DATABASE DESIGN	15
4.2.1. Data Mapping Information	16
4.2.2. Data Conversion	16
4.3. ARCHIVAL AND RETENTION REQUIREMENTS	16
4.4. DISASTER AND FAILURE RECOVERY	16
4.5. BUSINESS PROCESS WORKFLOW	17
4.6. BUSINESS PROCESS MODELING AND MANAGEMENT (AS APPLICABLE)	17
4.7. BUSINESS LOGIC	18
4.8. VARIABLES	18
4.9. ACTIVITY / CLASS DIAGRAMS (AS APPLICABLE)	19
4.10. DATA MIGRATION	21
4.10.1. Architectural Representation	3
4.10.2. Architectural Goals and Constraints	3
4.10.3. Logical View	3
4.10.4. Architecturally Significant Design Packages	3
4.10.5. Data model	3
4.10.6. Deployment View	3
5. ENVIRONMENT DESCRIPTION	
5.1. TIME ZONE SUPPORT	25

5.2. LANGUAGE SUPPORT	25
5.3. USER DESKTOP REQUIREMENTS	25
5.4. SERVER-SIDE REQUIREMENTS	25
5.4.1. Deployment Considerations	25
5.4.2. Application Server Disk Space	26
5.4.3. Database Server Disk Space	26
5.4.4. Integration Requirements	26
5.4.5. Jobs	26
5.4.6. Network	26
5.4.7. Others	26
5.5. CONFIGURATION	26
5.5.1. Operating System	26
5.5.2. Database	27
5.5.3. Network	27
5.5.4. Desktop	27

6. References

7. Appendix

Introduction

The Alumni Management System is a simple C project in which a university can keep track with its students and graduates. Students can view their profile and alumni can create job information. The aim of this project is to make connections between alumni and students. The project manages the fresh as well as old graduate students with their respective information in actively participating in registering, searching and managing the alumni information for job opportunities and resources.

1.1. Intended Audience

The intended audience of this project will be,

- Alumni – They are the passed out students. They can give their details and can update it.
- Student – They are the present students of that college. They can also give their details and can edit also.
- Admin – Admin can edit and delete alumni

Acronyms/Abbreviations

Not applicable – Have not used any abbreviations in this document.

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.

1.3. Key Project Objectives

The key objectives of the project is given below,

- This allows students to know each other and their current activities.
- To reduce manual work.
- To build a system that will be able to manage alumni data of a college and provide easy access.
- To provide a user-friendly interface.
- To allow old and new students at a university or college with each other.

1.4. Project Scope and Limitation

1.4.1. In Scope

The purpose of the application is to create a system which can be used to store, view and modify details of the student and the alumni. The admin

can also edit the details of alumni and students. The end goal is to create an Application that is easy to use and understand.

1.4.2. Out of scope

Not applicable for this project.

1.5. Functional Overview

It is an University Alumni Management portal which consists of three modules such as Admin, Student and Alumni. The Alumni can register the portal using the details like Name,Address,Phone number,Year of passing,Experience,Company,University Id,Username and Password. After registering on the portal, he/she can login using the Username and password and post the available new job updates ,technical advice and materials so that the students can know about the jobs related to their field and can easily access it. The Alumni can also modify, delete,view his/her Information.

UAM_01-> alumni_get_data() -> It will register the new alumni using a unique university registration number. Alumni can enter the details like name, address, telephone no., company, years of experience.

UAM_02-> Alumni_Login() -> After Registration is done Alumni can enter the username and password to login to the Alumni portal.

UAM_03-> view_Profile()-> Alumni can view their profile through this function.

UAM_04-> Edit_details()-> After entering username and password, alumni can update their details like Name,Address,Username ,Password,University number,Year of Passing,Mobile number,Company and Experience.

UAM_05-> Create_job_posts()-> After login, alumni can create the Job post, which will give information about the recruitment of a company .

The student can also Register in the portal by entering theUsername and password. If he is registered he can view all the details of his wishes for alumni and the job post that alumni had posted. The student can edit his own details like telephone number,Address,Username and Password in the Student Database and they can view their details.

UAM_06 -> student_get_data() : This function will register the new student into the portal using his/her Name,Address,Phone number,Department,Registration Number,Username and Password.

UAM_07 -> Student_Login() : If a student has already registered into the portal, they can login into the portal by giving their credentials like Username and password.

UAM_08 -> view_profile() : After the login is successful, the student can view his/her profile like Name, Registration number,Mobile number,Department and Address.

UAM_09 -> Edit_Info() : Students can also edit their details in the student data like Name, Mobile number, Department, Address, Username and Password.

UAM_10 -> View_Alumni_Details() : Students can search the registration number of Alumni then students can view the profile of alumni and the Job post which contains details like job role, date of Post, name of the company, Experience required and contact information.

Admin can login using a username and password so that they can modify , delete, view records of Alumni database.

UAM_11->Edit_Alumni() : Admin can edit the alumni details by entering the university no of that alumni to edit.

UAM_12->Delete_Alumni_Details() : Admin can delete the alumni details by entering the university no of that alumni to be deleted from the database.

1.6. Assumptions, Dependencies & Constraints

The assumptions and dependencies required are mentioned below:

- System should have Ubuntu Linux installed.
- The service is used preferably on a desktop or laptop.
- The operating system of the laptop or desktop should be Linux.
- The system should have either 4Gb or more RAM.

1.7. Risks

Not Applicable to this project.

Design Overview

UAMS is software which eases in managing the alumni and student data by keeping all the data in just one software. In this software Alumni and students can view and edit their profile. Admin can edit and delete the alumni details. Also we are providing an option for alumni to post a job recruitment and this can also be viewed by students as well as the alumni who have experienced and wanted a job.

Design Object

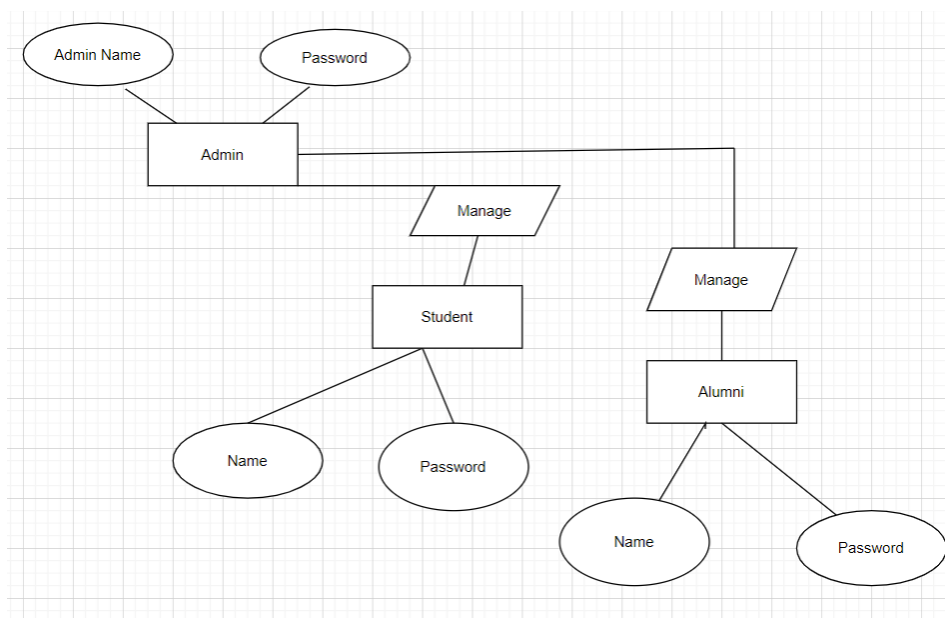
There are three major modules there.

Alumni: Here alumni can login and view/edit and create job posts.

Student: Here students can login and edit/view information and can also view job posts.

Admin: Can able to Edit and Delete alumni details'

1.7.1. Recommended Architecture



Architectural Strategies :

ADMIN

1. Admin Login
2. Add/Delete Details
3. Search Alumni
4. Add Alumni
5. Registration/Update Alumni Details

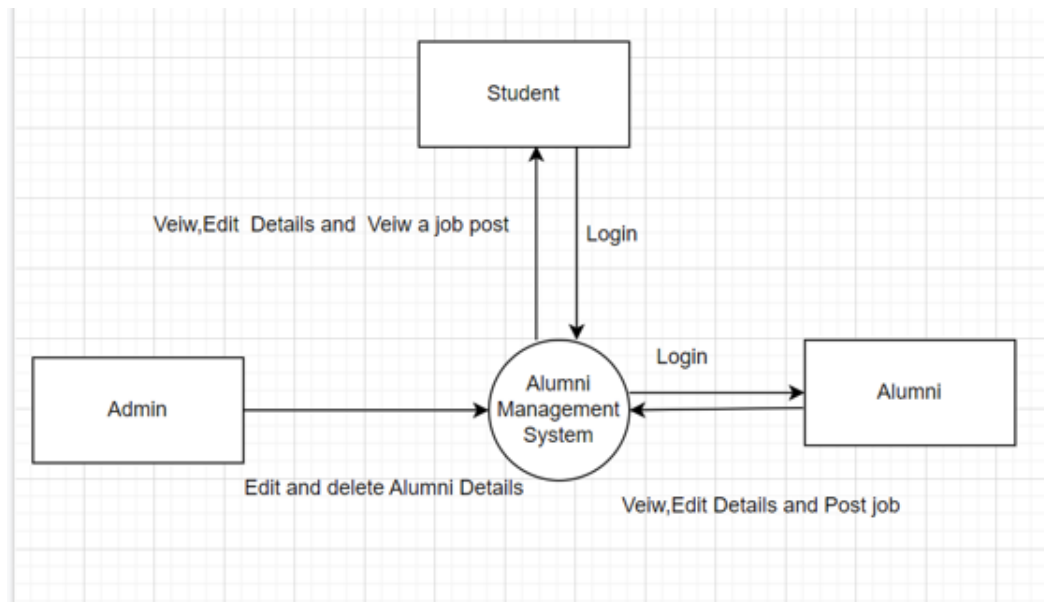
ALUMNI

1. Alumni Registration
2. Login
3. Update Profile
4. View Details

STUDENT

1. Student Login
2. Add/Edit Details
3. View job Details
4. View Alumni Details

1.7.2. Design Alternative



The alternative design of the project is shown above.

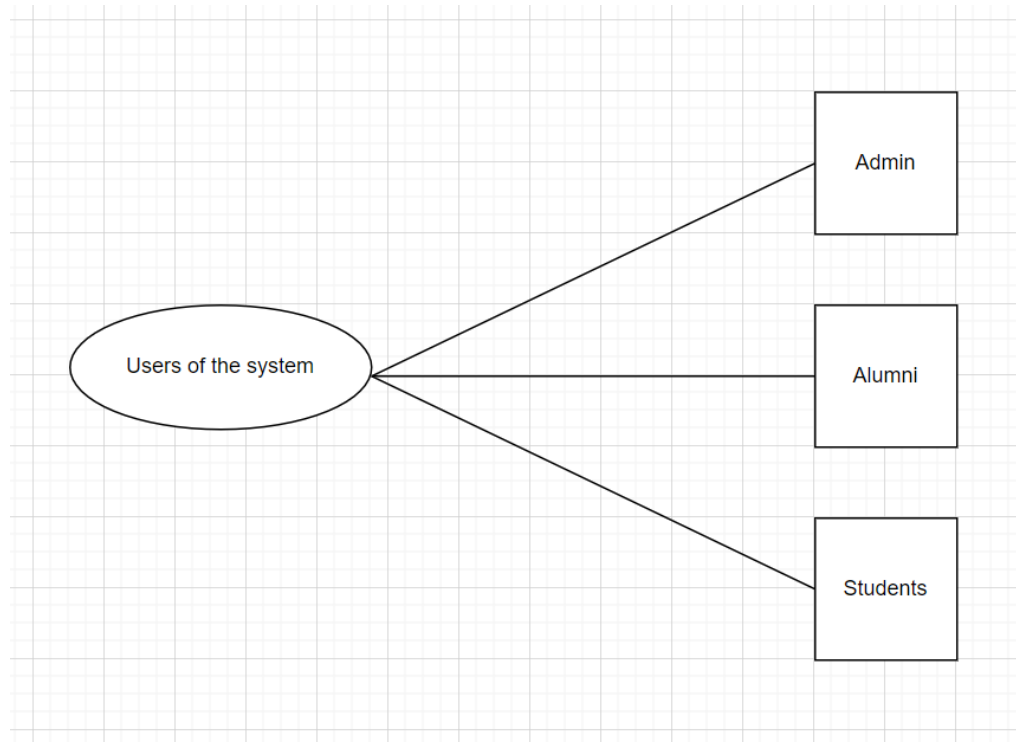
1.7.3. Reuse of Existing Common Services/Utilities

Not applicable – Since we have not used any existing common services and Utilities.

1.7.4. Creation of New Common Services/Utilities

Not applicable – Since we have not created any common services and Utilities.

1.7.5. System Interface Paradigms



1.7.6. Error Detection / Exceptional Handling

Not Applicable for this project.

1.7.7. Memory Management

- In this Alumni Management system data are stored in System safe and in a secure manner.
- Memory is always used to store the data collected from the alumni and the student.
- Once a data is deleted by the admin, it will be deleted from memory also.

- We use file management for storing the datas of students and alumni.

Performance

- The Alumni Management System is reliable.
- This project will be very responsive and fast.
- This will not take much space of the system.
- This project is optimized in every aspect thus works finely.
- This platform enables organizations to handle passwords secure to login and other users can't access personal details of other alumni or students .

Security

This system designed with good security principles ensures Integrity of the system and prevents attacks and data leakage.

Concurrency and Synchronization

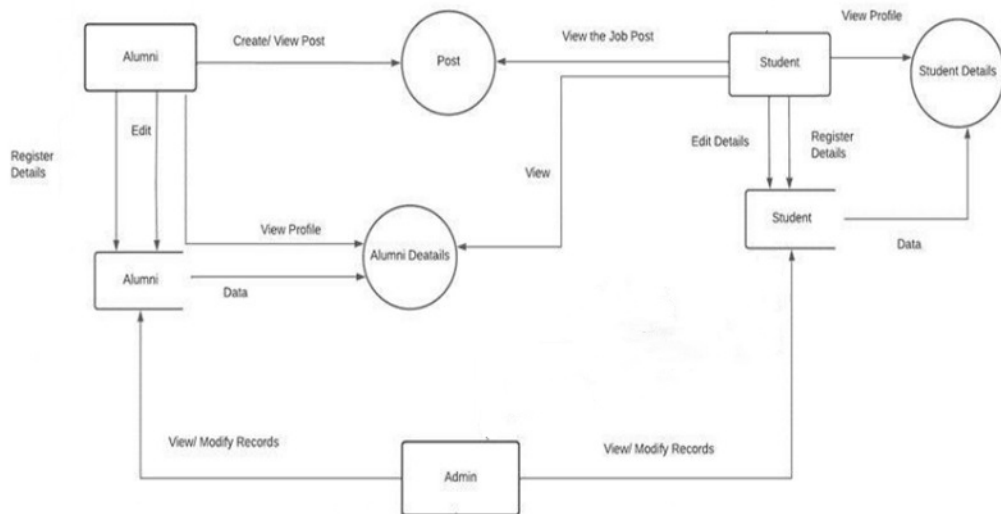
Not applicable for this project.

Housekeeping and Maintenance

Not applicable for this project.

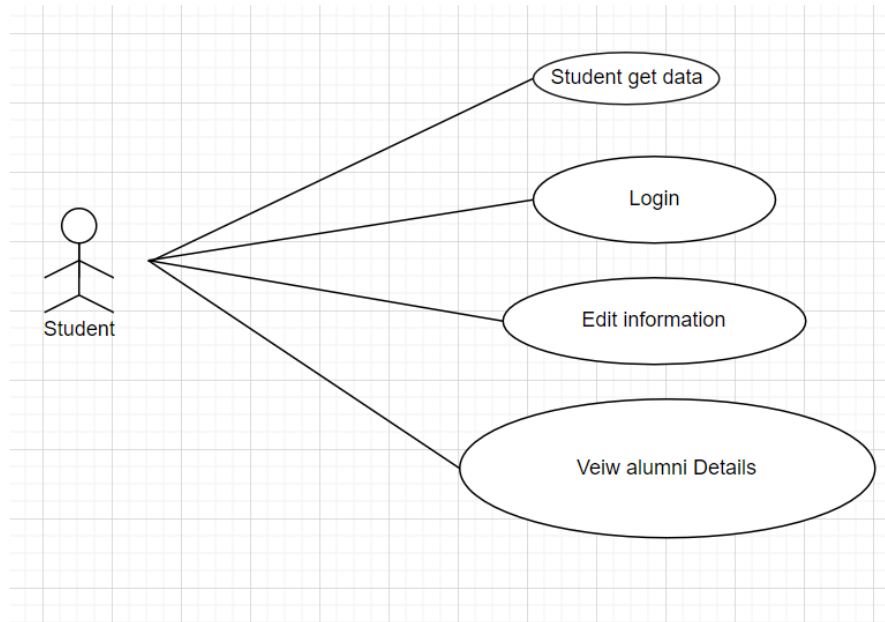
2. System Architecture

2.1. System Architecture Diagram.

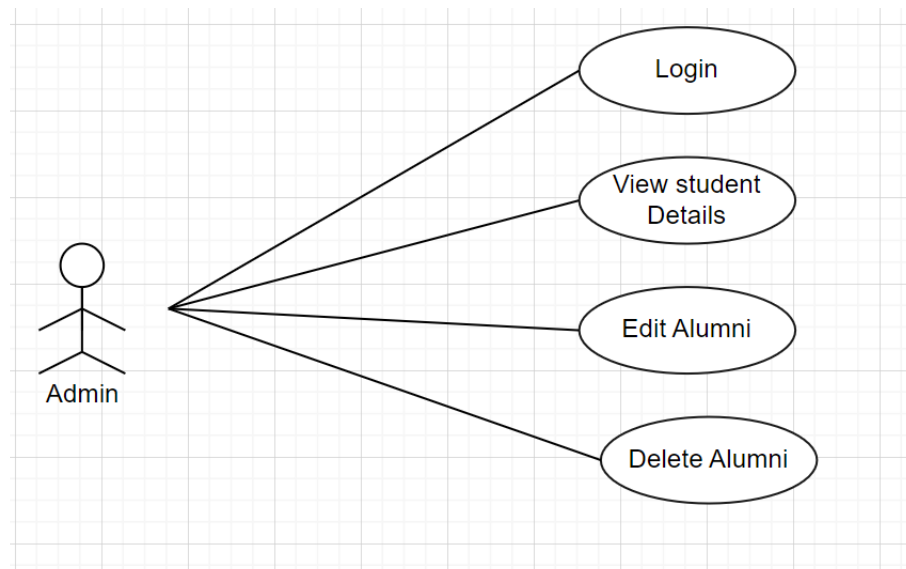


2.2. System Use-Cases

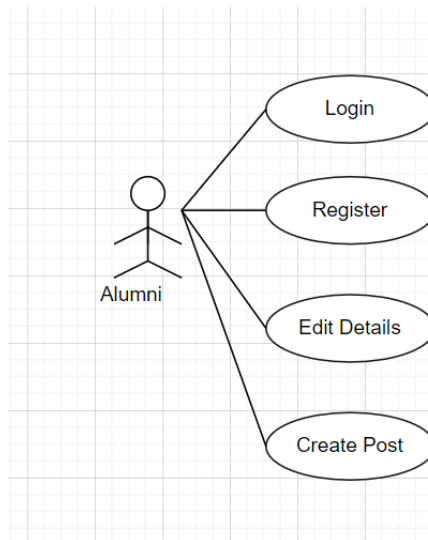
Student Use-Case



Administrator Use-Case



Alumni Use-Case



2.3. Subsystem Architecture

Not applicable for this project – Since everything is mentioned above.

2.4. System Interfaces

Already mentioned above for this project.

2.4.1. Internal Interfaces

Not applicable for this project.

2.4.2. External Interfaces

Not applicable for this project.

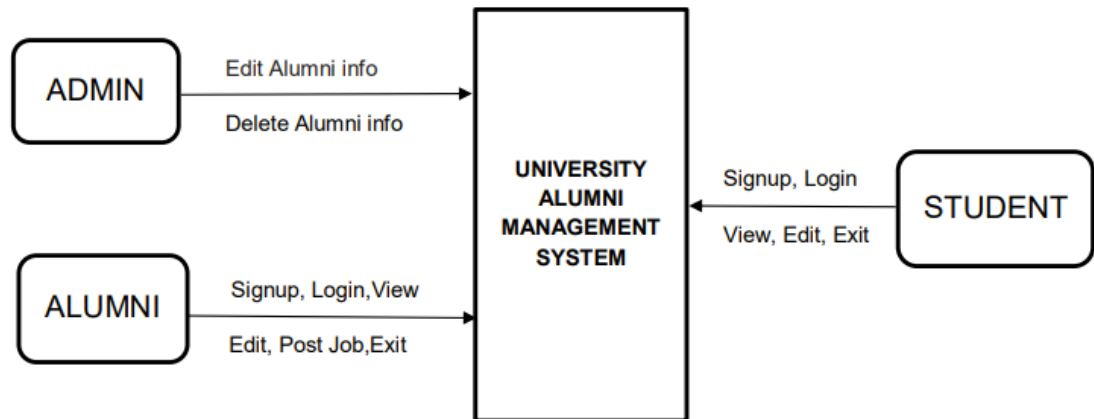
3. Detailed System Design

2.5. Key Entities

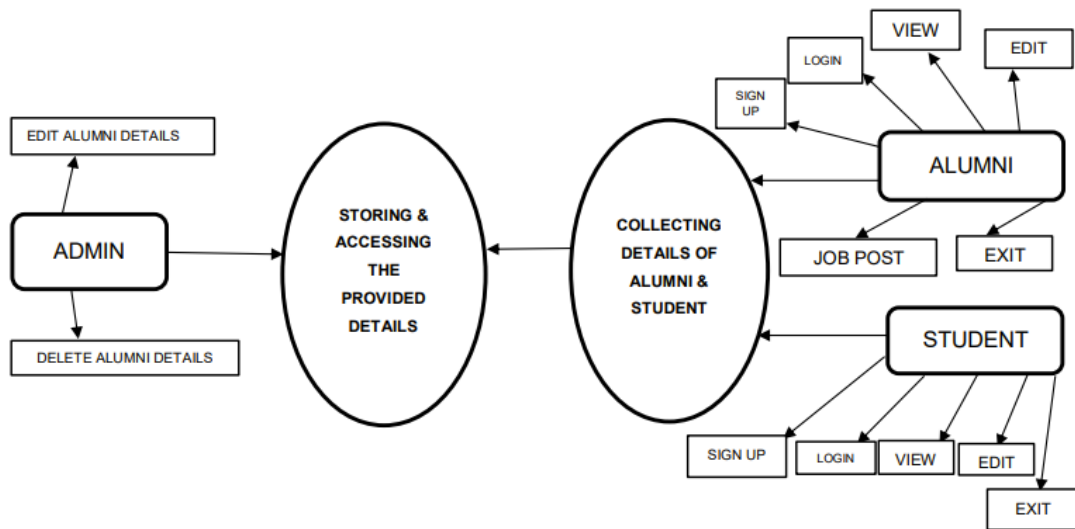
Already mentioned in the document.

2.6. Detailed-Level Database Design

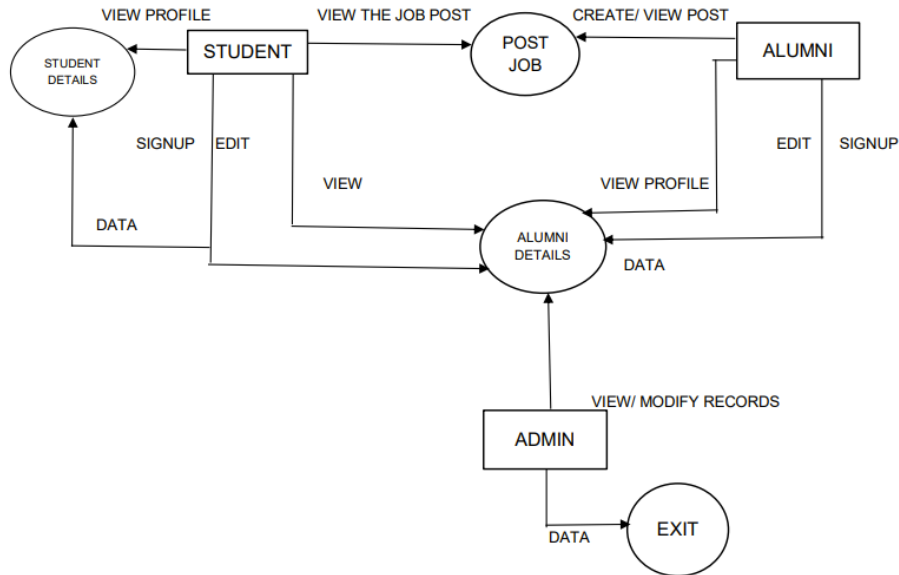
DFD Level 0 :



DFD Level 1



DFD Level 2 (Flowchart)



2.6.1. Data Mapping Information

Not applicable for this project.

2.6.2. Data Conversion

Not applicable for this project.

2.7. Archival and retention requirements

Not applicable for this project.

2.8. Disaster and Failure Recovery

Not applicable for this project.

2.9. Business Process workflow

Not applicable for this project.

2.10. Business Process Modeling and Management (as applicable)

Not applicable for this project.

2.11. Business Logic

Not applicable for this project.

2.12. Variables

All the naming functions of the variables and functions are followed by the rules. And those can be easily understandable for the new project developer. Thus the naming is universally done.

2.13. Activity / Class Diagrams (as applicable)

Not applicable for this project.

2.14. Data Migration

Not applicable for this project.

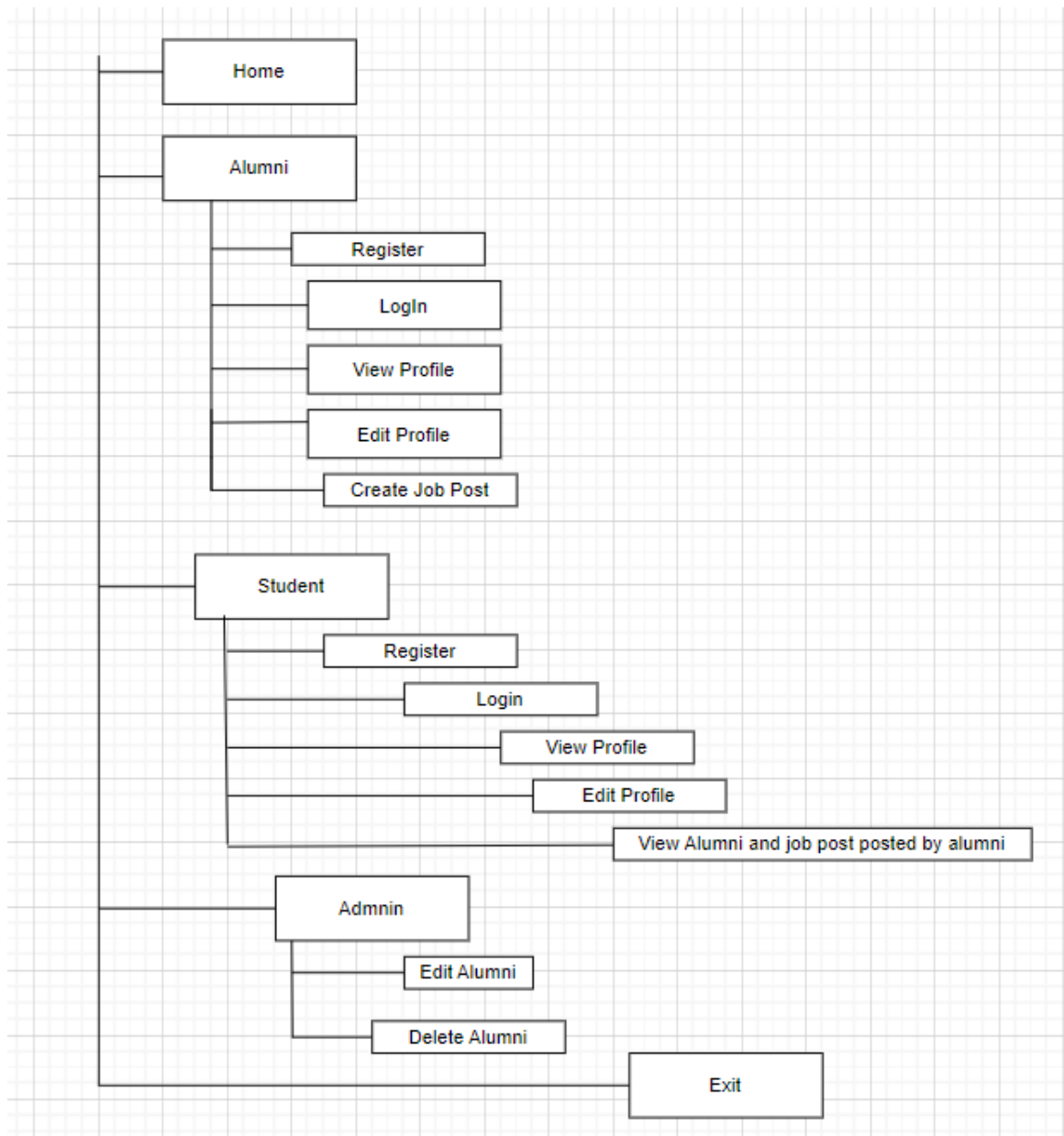
2.14.1. Architectural Representation

Already mentioned in the document.

2.14.2. Architectural Goals and Constraints

Already mentioned in the document.

2.14.3. Logical View



2.14.4. Architecturally Significant Design Packages

Not applicable for this project.

2.14.5. Data model

Not applicable for this project.

2.14.6. Deployment View

Not applicable for this project.

3. Environment Description

3.1. Time Zone Support

This project will support India Time zone (GMT+5:30).

3.2. Language Support

English is the only language supported by this project.

3.3. User Desktop Requirements

Already mentioned in the document.

3.4. Server-Side Requirements

Not applicable for this project.

3.4.1. Deployment Considerations

Not applicable for this project.

3.4.2. Application Server Disk Space

Not applicable for this project.

3.4.3. Database Server Disk Space

Not applicable for this project.

3.4.4. Integration Requirements

Not applicable for this project.

3.4.5. Jobs

Not applicable for this project.

3.4.6. Network

Not applicable for this project.

3.4.7. Others

NIL - Everything is covered in the above topics.

3.5. Configuration

Already mentioned in the document.

3.5.1. Operating System

- The operating system of the laptop or desktop should be Linux.
- System should have either 4Gb or more RAM.

3.5.2. Database

Not applicable for this project.

3.5.3. Network

Not applicable for this project.

3.5.4. Desktop

Already mentioned in the document.

4. References

- <https://projectsgeek.com/2015/11/alumni-management-system-java-project.html>
- <https://www.sourcecodester.com/php/14524/alumni-management-system-using-phpmysql-source-code.html>
- <https://www.slideshare.net/snehaindulkar2/alumni-management-system-248322228>
- <https://studentprojectguide.com/php/alumni-management-system/>

5. Appendix

- http://www.junikhyatjournal.in/no_1_Online_21/65.pdf
- <https://www.coursehero.com/file/59128235/Alumni-Management-System-Reportdoc/>
- http://103.47.12.35/bitstream/handle/1/1814/1713203017_HARSH%20PATEL_FinalProjectReport%20-%20harsh%20patel.pdf?sequence=1&isAllowed=y

Change Log

QMS Template Version Control (Maintained by QA)

Date	Version	Author	Description

