COMSATS University Islamabad,
COMSATS Road, off GT Road, Sahiwal, Pakistan

SOFTWARE DESIGN DESCRIPTION

(SDD DOCUMENT)

for

Batch Advisory

Version 1.0

By

BILAL AHMAD	**CIIT/SP21-BSE-004/SWL**
SAAD ALI	**CIIT/SP21-BSE-029/SWL**
JAVID IRSHAD	**CIIT/SP21-BSE-049/SWL**

Supervisor

Ms. Mubeen Javed

Bachelor of Science in Computer Science (2021-2025)

Table of Contents

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee) *include the ones given at scope time both in doc and presentation	**Action taken**

Supervised by

Ms. Mubeen Javed

Signature__________

1 Introduction:

This document outlines a comprehensive approach to automate the batch advisory system in a university. Manual processes in course recommendation and application handling are time-consuming and error prone. Our software intends to solve these issues by automating processes, improving communication, and integrating real-time features. The main modules we use in this we application are:

1.1 Module 1: Course Recommendation:

Generates course recommendations based on prerequisites and Scheme of Study (SOS) of students.

1.2 Module 2: Application System:

Enables students to submit applications online and communicate with batch advisors.

2 Design Methodology and Software Process Model

Considering the nature of batch processing and the need for iterative development and flexibility the design methodology and software process model we use is:

2.1 Service-Oriented Architecture (SOA):

- SOA enables you to design the system as a collection of loosely coupled services, which aligns well with the modular nature of batch advisory systems.
- It allows for easier scalability and maintenance as you can independently develop, deploy, and scale individual services.

2.2 Agile Software Development:

- Agile methodologies like Scrum or Kanban promote flexibility, adaptability, and collaboration.
- Batch advisory systems often involve evolving requirements and frequent feedback loops, making Agile a suitable choice for iterative development.

3 System Overview

This system will be used by students and batch advisors in universities or collages. The system aims to develop a batch advisory system related to the needs of academic institutions. This system addresses the challenges of manual course allocation and application processing by automating these tasks. For designing and developing we use React.js for building a responsive and interactive user interface, providing students and administrators with an intuitive experience. It supports modular component-based design, allowing for easy customization and maintenance of frontend elements. For Backend we use Node.js for building a robust and scalable backend server to handle application logic, data processing, and communication with databases or external systems.

4 Architectural design

In batch advisory system we use two main modules first is allocation of courses according to certain criteria and pre-requisites of desired course and the second module is application submission where students submit their application to their department and check the status of that application. So, the relationship between these modules and its core components are:

- The Batch Advisory System module serves as the central controller and interacts with both the Allocation of Courses module and the Application Submission module.
- During the course allocation process, the Batch Advisory System module interacts closely with the Allocation of Courses module to obtain student preferences and criteria for allocation.
- The Application Submission module interacts with the Batch Advisory System module to receive course allocation results and updates, ensuring that students have accurate information when submitting their applications.

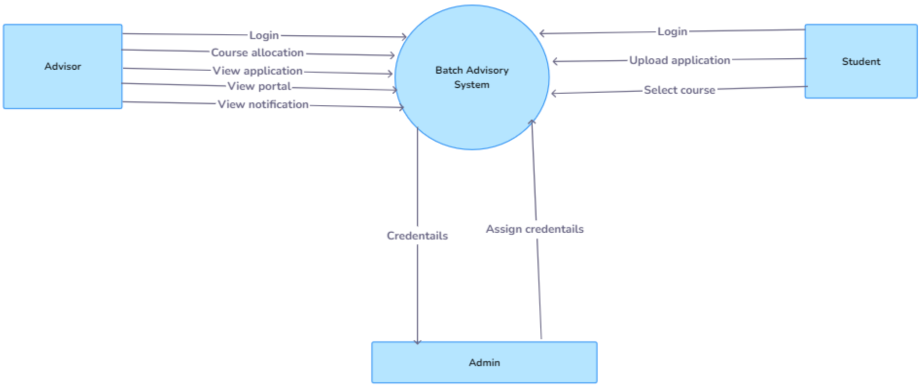


Figure 1: Architectural Diagram

4.1 Flow Chart Representation

4.1.1 Registration

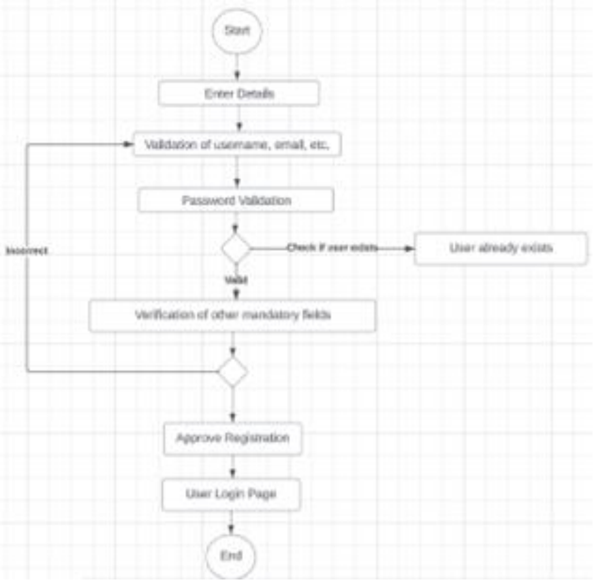


Figure 2: Registration flow

4.1.2 Course Allocation and Application Submission

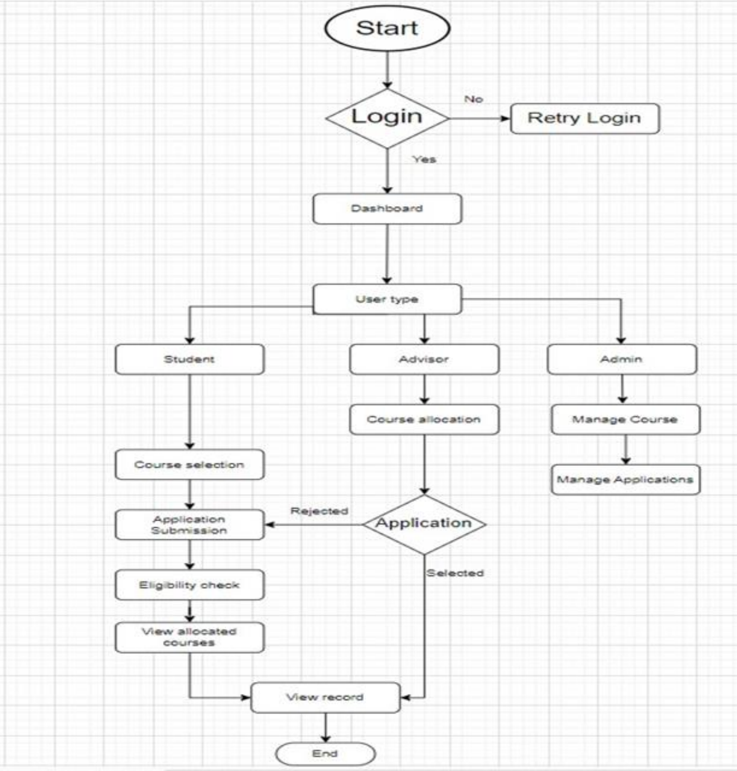


Figure 3: Modules Flow

4.2 Class Diagram

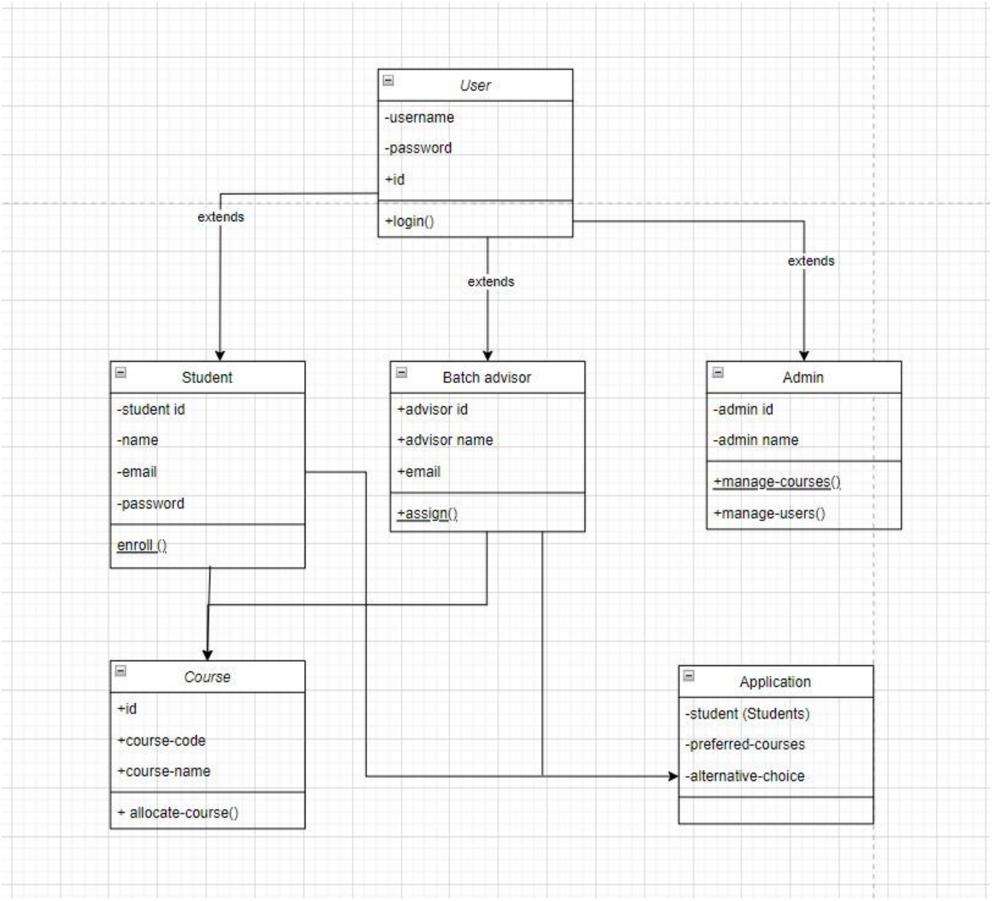


Figure 4: Class Diagram

4.3 Dataflow Diagram

4.3.1 DFD Level 0

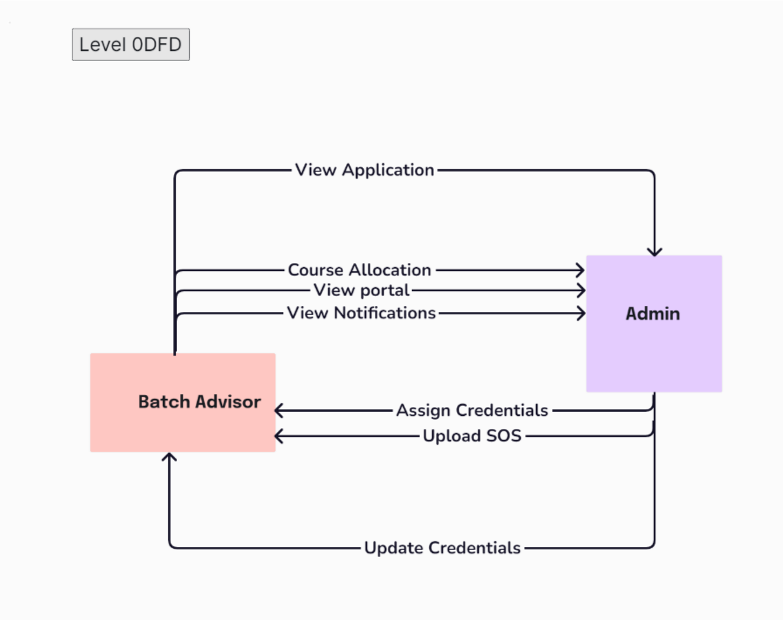


Figure 5 : DFD Level 0

4.3.2 DFD Level 1

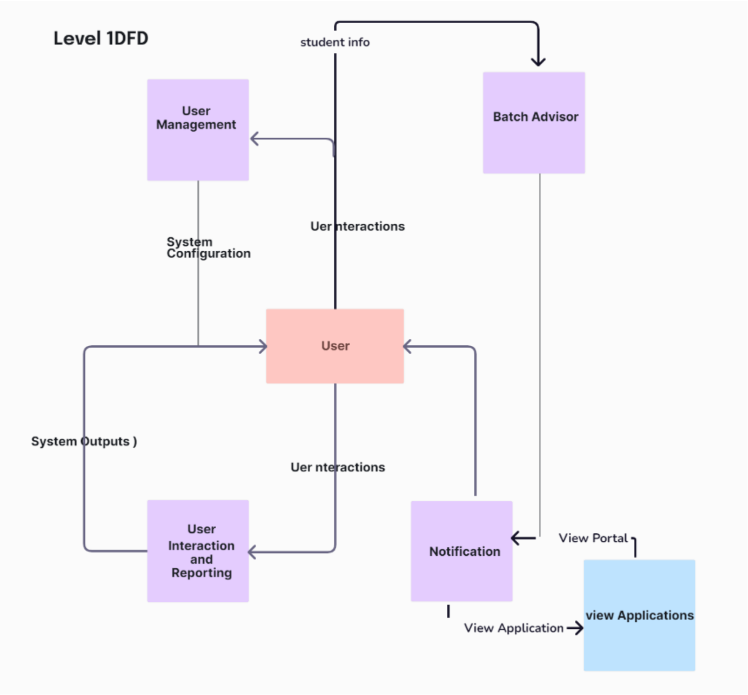


Figure 6: DFD Level 1

4.4 Sequence Diagram

4.4.1 Student View

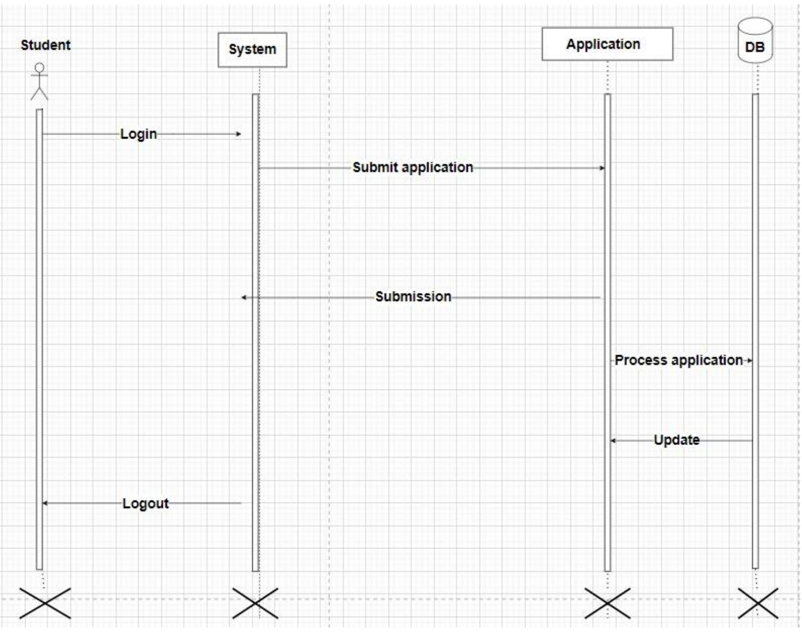


Figure 7: Sequence Diagram

4.4.2 Advisor View

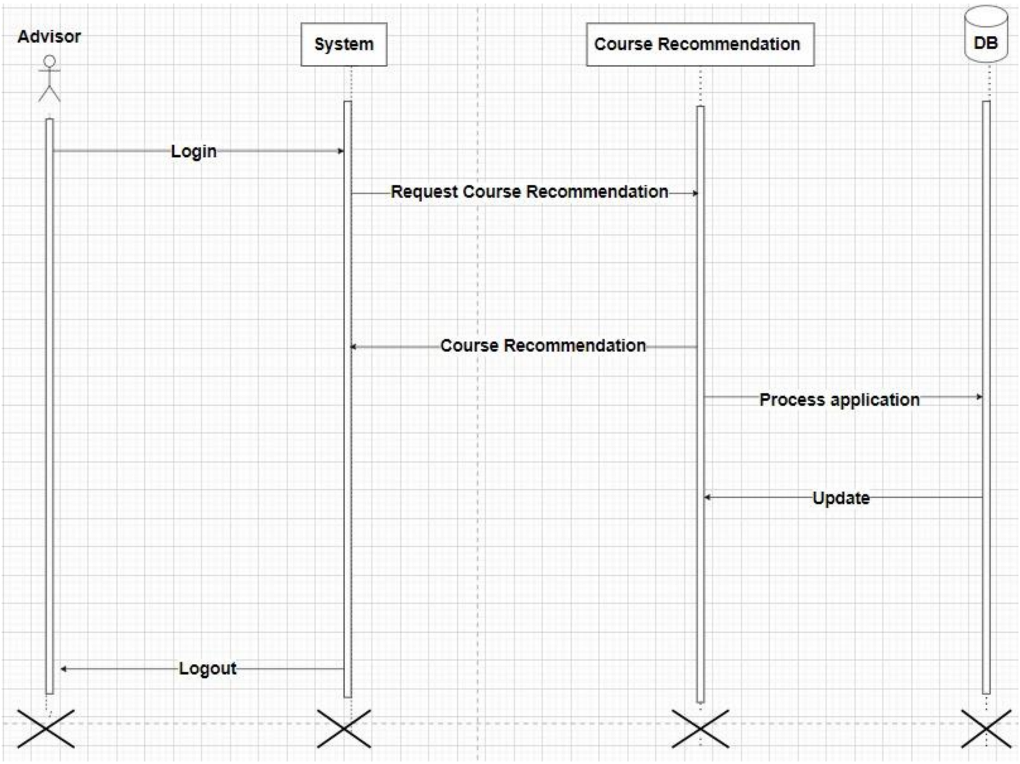


Figure 3.1: Sequence Diagram

5 Data Design

The information about courses, students, faculty, and timetable is stored in a relational database management system (RDBMS) such as MySQL or PostgreSQL. Each application submitted by a student is represented as a data structure containing information such as student ID, preferred courses, alternative choices, special requests, etc. Similar to the course allocation module, the information related to applications, application status, and notifications is stored in the same RDBMS. Relational Database Management System (RDBMS) is used to store structured data related to courses, students, faculty, applications, and other entities.

6 Data dictionary

A. Course Allocation Module:

Courses

- Type: Entity
- Description: Represents the courses offered by the institution.

Faculty

- Type: Entity
- Description: Represents faculty members who teach courses.

Students

- Type: Entity
- Description: Represents students enrolled in the institution.

B. Application Submission Module:

Applications

- Type: Entity
- Description: Represents applications submitted by students for course enrollment.

Functional Description:

process_application(application)

- Parameters: application (Application)
- Description: Processes the submitted application and updates its status accordingly.

Object-Oriented Description (OO):

A. Course Allocation Module:

1. **Courses**
    - Attributes:
        - course_code: str
        - title: str
        - description: str
2. **Faculty**
    - Attributes:
        - faculty_id: str
        - name: str
        - expertise: List[str]

3. **Students**
    - Attributes:
        - student_id: str
        - name: str
        - program: str
        - preferences: List[Course]

B. Application Submission Module:

1. **Applications**
    - Attributes:
        - application_id: str
        - student: Student
        - preferred_courses: List[Course]
        - alternative_choices: List[Course]
        - special_requests: str
2. **Application Status**
    - Attributes:
        - application: Application
        - status: str
        - comments: str

7 Algorithm & Implementation

Course Allocation Module:

1. **process_application(application)**

PDL Summary:

1. Check the availability of seats in the preferred courses mentioned in the application.

2. If seats are available:

a. Update the application status to "Accepted".

b. Allocate the student to the preferred courses.

c. Notify the student of the acceptance.

3. If seats are not available:

a. Update the application status to "Rejected".

b. Notify the student of the rejection.

4. Return the updated application.

2. send_notification(student, message)

PDL Summary:

1. Create a notification with the provided message.

2. Send the notification to the specified student.

3. Log the sent notification for reference.

Application Submission Module:

1. **process_application(application)**

PDL Summary:

1. Check the eligibility of the application based on prerequisites and available seats.

2. If eligible:

a. Assign the application a status of "Pending".

b. Log the application for review.

c. Notify the student of the pending status.

3. If not eligible:

a. Assign the application a status of "Rejected".

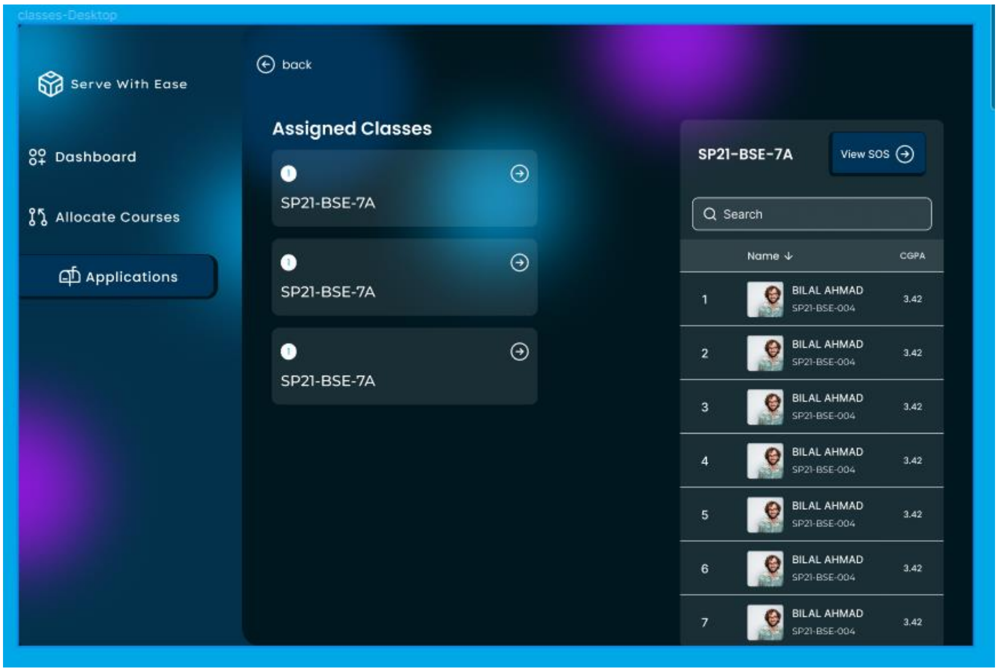
b. Notify the student of the rejection.

4. Return the updated application status.

8 Human interface design

There will be a user-friendly and responsive user interface. A user maybe a student, facility or admin of the system. Students have separate panel that is student panel. A user can get access after login. Then he/she can process and generate and check status of applications and allocation of courses

8.1 Screen Images



Assigned Classes

SP21-BSE-7A

SP21-BSE-7A

SP21-BSE-7A

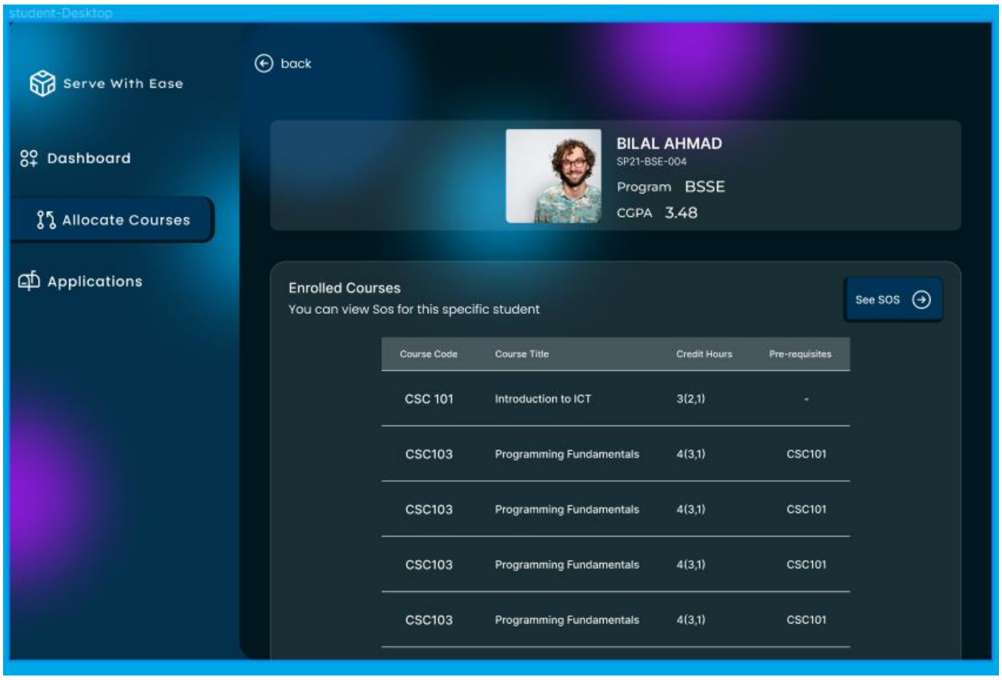
SP21-BSE-7A

View SOS

Search

	Name ↓	CGPA
1	BILAL AHMAD SP21-BSE-004	3.42
2	BILAL AHMAD SP21-BSE-004	3.42
3	BILAL AHMAD SP21-BSE-004	3.42
4	BILAL AHMAD SP21-BSE-004	3.42
5	BILAL AHMAD SP21-BSE-004	3.42
6	BILAL AHMAD SP21-BSE-004	3.42
7	BILAL AHMAD SP21-BSE-004	3.42

Figure 8: Screen Image 1



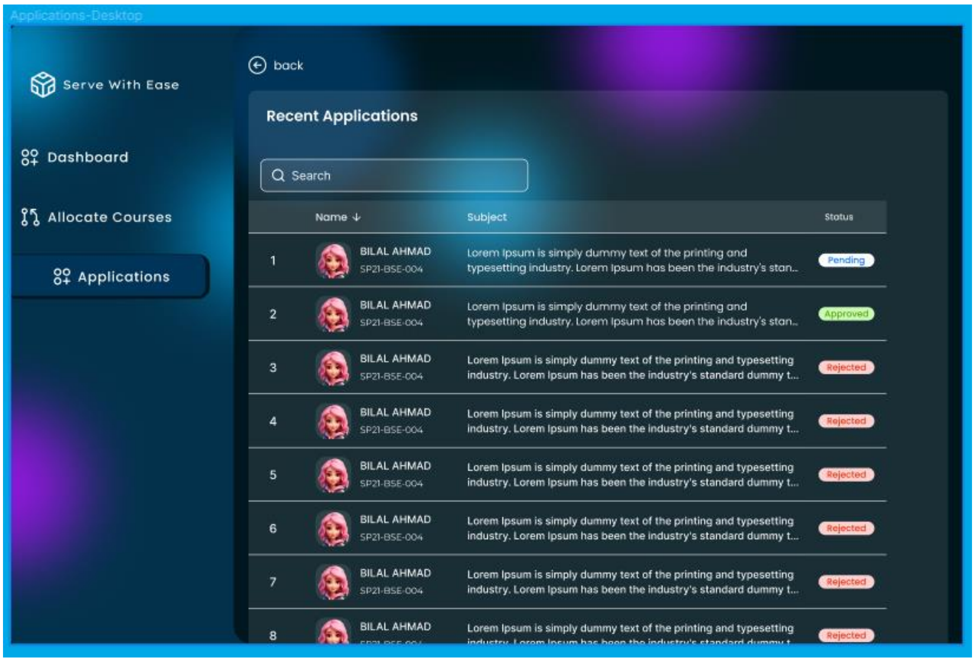
BILAL AHMAD
SP21-BSE-004
Program BSSE
CGPA 3.48

Enrolled Courses
You can view Sos for this specific student

See SOS

Course Code	Course Title	Credit Hours	Pre-requisites
CSC 101	Introduction to ICT	3(2,1)	-
CSC103	Programming Fundamentals	4(3,1)	CSC101
CSC103	Programming Fundamentals	4(3,1)	CSC101
CSC103	Programming Fundamentals	4(3,1)	CSC101
CSC103	Programming Fundamentals	4(3,1)	CSC101

Figure 9: Screen 2



Recent Applications

	Name ↓		Subject	Status
1	BILAL AHMAD SP21-BSE-004		Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's stan...	Pending
2	BILAL AHMAD SP21-BSE-004		Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's stan...	Approved
3	BILAL AHMAD SP21-BSE-004		Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy t...	Rejected
4	BILAL AHMAD SP21-BSE-004		Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy t...	Rejected
5	BILAL AHMAD SP21-BSE-004		Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy t...	Rejected
6	BILAL AHMAD SP21-BSE-004		Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy t...	Rejected
7	BILAL AHMAD SP21-BSE-004		Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy t...	Rejected
8	BILAL AHMAD		Lorem Ipsum is simply dummy text of the printing and typesetting	Rejected

Figure 10: Screen 3

8.2 Screen objects and actions.

Screen objects or images contains student panel where student can check assigned classes and see his SOS according to his class and checks the total CGPA. In student panel the allocate courses page allow student to see the enrolled courses according to SOS of his/her class. In Application page of student panel student can check the status of his/her applications.