



CSE 300 Software Engineering

Coding and Integration Standard

Project title: AUVisionX - Ahmedabad
University Integrated Resource System.

Faculty: Prof. Khushru Doctor

Student Details

Roll No.	Name of the Student
AU2040152	Devyash Shah
AU2040252	Sagar Bajaj
AU2040137	Kush Patel
AU2040196	Vandan Shah
AU2040123	Manthan Patel

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
2.7 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
4. System Features	6
4.1 User Registration and Authentication	6
4.2 Course Search and Selection	6
4.3 Course Registration and Enrollment	7
5. Other Nonfunctional Requirements	4
5.1 Performance Requirements	4
5.2 Safety Requirements	5
5.3 Security Requirements	5
5.4 Software Quality Attributes	5
5.5 Business Rules	5
6. Other Requirements	5
Appendix A: Glossary	5
Appendix B: Analysis Models	5
Appendix C: To Be Determined List	6

1. Introduction

1.1 Purpose

The purpose of AUVisionX is to revolutionize university management and student engagement by providing a comprehensive software solution. This software system aims to streamline and enhance various aspects of university operations, primarily focusing on the core components. It is designed to offer a 'Seamless and Efficient Course Registration System' that simplifies course enrollment for students of Ahmedabad University. Additionally, it includes an 'Alumni Student Discord platform' to foster alumni and student collaboration, a 'Secure Student Login Page' to ensure data protection, and efforts to reduce overall system latency for an improved user experience. This product aspires to modernize and optimize university operations and community engagement.

1.2 Document Conventions

The AUVisionX Software Requirements Specification (SRS) adheres to standard documentation conventions, maintaining clarity and consistency. For readability, Times New Roman font in 12-point font size for the body and larger font sizes for the headings are utilized throughout the document to create a clear visual hierarchy. Higher-level requirements' priorities cascade down to detailed requirements, ensuring that each requirement's criticality is well-defined and aligns with the project's goals. Each requirement statement is assigned a unique priority, enhancing clarity and ease of reference throughout the document.

1.3 Intended Audience and Reading Suggestions

The AUVisionX SRS is intended for various stakeholders, including developers, project managers, testers, and documentation writers. It provides an overview of the product, external interface requirements, system features, non-functional requirements, use case diagrams, sequential flow diagrams, data flow diagrams, and a conclusion. To optimize readability, a suggested reading sequence begins with the overview sections to provide context. Developers should delve into system features and interface requirements, while project managers may focus on non-functional requirements for resource planning. Testers can explore use cases and sequential flow diagrams for test scenario understanding, and documentation writers may find the conclusion and overview sections useful for creating user documentation.

1.4 Product Scope

AUVisionX's primary objective is to enhance university operations and student engagement. It focuses on reducing website latency and improving the student course registration system, resulting

in a smoother user experience. In addition, AUVisionX plans to incorporate supplementary features, including an Alumni Discord platform, fortified student data security, and increased reliability of the events calendar. By achieving these goals, AUVisionX aligns with the university's corporate objectives of offering a technologically advanced and user-friendly environment, fostering alumni-student interactions, ensuring data security, and streamlining administrative processes for a more efficient educational experience.

2. Overall Description

2.1 Product Perspective

The university student portal represents an innovative and self-contained software product tailored to elevate the digital experience for students, faculty, and administrators. It is not an extension or replacement of any existing product family or system. The SRS establishes this software as a distinct and independent solution, designed to meet the evolving needs of the university community. Its introduction brings a fresh, comprehensive approach which even in times of load especially during course registration works appropriately improving the experience and performance for the user.

2.2 Product Functions

The major functions of the university student portal include:

1. Admin management
2. Course registration
3. Chatbot
4. Course management

2.3 User Classes and Characteristics

3. Students:

- Different educational levels and academic experiences.
- Regular users of the portal for course registration, academic information, and collaboration.

4. Faculty (Teachers):

- Extensive academic experience.
- Primarily responsible for grading.

5. **Administrators:**

- Technical expertise.
- Responsible for managing user accounts, courses, and system-wide settings.

5.1 **Operating Environment**

- **Hardware Platform:** The platform uses modern, industry-standard hardware infrastructure to ensure reliability and performance.
- **Operating System:** It is compatible with a variety of operating systems, including Windows and Linux.
- **Software Components:** The portal interacts with web browsers and standard web technologies.

5.2 **Design and Implementation Constraints**

- **Hardware Limitations:** The portal should operate efficiently within defined timing and memory requirements.
- **Interfaces:** It should integrate seamlessly with other university applications and systems.
- **Security Considerations:** Robust security measures must be implemented to protect user data and privacy.

5.3 **User Documentation**

User documentation components that will be delivered with the software include user manuals, online help, and tutorials. The documentation will be available in standard formats for ease of use.

5.4 **Assumptions and Dependencies**

- The student won't be having a credit extension of more than 25 credits.

-
- The Course credit requirements are not considered and should be checked by students.
 - Reattempting for a course is not included.
 - Registration for any course beyond the deadline must be done through an administrator.
 - If there is a prerequisite, the user is expected to complete it before enrolling the course.

6. External Interface Requirements

6.1 User Interfaces

The user interface for AUVisionX should offer an intuitive and user-friendly experience, enabling easy navigation and interaction for all users.

The design of the interface should follow established GUI (Graphical User Interface) standards, ensuring a consistent and visually appealing layout.

Accessibility guidelines should be implemented to guarantee that individuals with disabilities can use the interface effectively.

The user interface must be responsive, adapting to various screen sizes, including desktop and mobile devices.

6.2 Hardware Interfaces

AUVisionX is designed to seamlessly interact with a range of hardware components to ensure efficient functioning and user convenience.

Supported device types encompass desktop computers, laptops, smartphones, tablets, and other common computing devices.

The software product must establish smooth data and control interactions with these devices for effective data exchange, user interaction, and system control.

6.3 Software Interfaces

Data sharing across software components is facilitated through standardized API protocols, ensuring secure and consistent data exchange.

Implementation constraints may specify the use of specific data formats (e.g., JSON, XML) to maintain data consistency and security during exchanges.

The software is designed for compatibility with various operating systems, ensuring accessibility across different user environments.

This interface directs messages to the operating system, guaranteeing seamless operation on diverse platforms for a wide user base.

7. System Features

7.1 User Registration and Authentication:- This feature allows users to create an account, log in, and securely authenticate themselves on the course registration website. User registration typically involves collecting personal information, such as name, email address, and password, and verifying user identity through email confirmation or other secure methods.

Stimulus/Response Sequences:

- Stimulus: User accesses the website and selects the "Register" option.
- Response: The registration form is presented, and the user provides their personal information.
- Stimulus: User submits the registration form.
- Response: The system verifies the information, and activates the user's account.

Functional Requirements:

- Users can register with a unique email address and password.
- The system should validate email addresses to ensure they are not already in use.
- Users can log in securely using their registered email and password.
- The system must securely store user credentials, protecting them against unauthorized access.

7.2 Course Search and Selection:- This feature enables users to search for available courses, view course details, and select courses they want to register for. Users can filter courses based on criteria like department, course number, instructor, and availability. It also provides information on course descriptions, schedules, and prerequisites.

Stimulus/Response Sequences:

- Stimulus: User logs in and accesses the course catalog.
- Response: The system displays a list of available courses and search filters.
- Stimulus: User applies search filters and selects a course.
- Response: The system provides detailed information about the selected course.

Functional Requirements:

- Users can search for courses by various criteria (e.g., department, course number, instructor).

-
- The system should display course details, including title, description, schedule, and prerequisites.
 - Users can add courses to their registration cart.
 - The system should prevent users from registering for conflicting courses.
 - Users can remove courses from their registration cart.

7.3 Course Registration and Enrollment:- This feature allows registered users to add courses to their schedule and finalize the registration process. Users will be able to view their selected courses, check for any registration conflicts, and receive confirmation once the registration is successful.

Stimulus/Response Sequences:

- Stimulus: User logs in, selects courses from their cart, and initiates registration.
- Response: The system checks for course conflicts and availability.
- Stimulus: User confirms the registration.
- Response: The system displays a registration confirmation and updates the user's schedule.

Functional Requirements:

- Users can view and edit the courses in their registration cart before finalizing registration.
- The system should prevent registration for courses with time conflicts.
- Users receive immediate feedback on successful or failed registrations.
- Users can view their course schedule after successful registration.
- The system should send email notifications with course registration details.

8. Other Nonfunctional Requirements

8.1 Performance Requirements

1. The university student portal shall respond to user actions within a maximum of 2 seconds under normal load conditions.
2. The system should be capable of scaling horizontally to handle an increased number of concurrent users without a significant degradation in performance.

-
3. The system should achieve an availability rate of 99.9% over any given calendar month, excluding scheduled maintenance.
 4. During high load events like Course registration, performance should be appropriate so that every student does not miss opportunity due to system lags for <1000 concurrent users.

8.2 Safety Requirements

1. The system must implement robust data encryption and access control mechanisms to safeguard sensitive information such as user personal data, academic records, and financial transactions.
2. The system must provide adequate user training to ensure that students, faculty, and staff understand how to use the system safely and effectively.

8.3 Security Requirements

1. The system portal must implement strong encryption mechanisms to protect sensitive data during transmission and storage. Data at rest and in transit must be encrypted.
2. The system must enforce access controls to restrict user access to data and features based on their roles and permissions. Users should only have access to the information and actions appropriate to their roles.
3. Regular security testing, including vulnerability assessments and penetration testing, must be conducted to identify and address potential security weaknesses.

8.4 Software Quality Attributes

1. The user interface of the university student portal should be intuitive, user-friendly, and easy to navigate.
2. The system should be highly reliable, with minimal downtime or disruptions to services.
3. The system should gracefully handle errors, recover from failures, and provide informative error messages.

8.5 Platform

1. The platform should be compatible with various web browsers and devices to ensure that users can access the system from different platforms.

9. Other Requirements

Appendix A: Glossary

SRS: Software Requirements Specification, a detailed document outlining software requirements.

AUVisionX: Project title, Ahmedabad University Integrated Resource System.

API: Application Programming Interface, enabling software communication.

SSL: Secure Sockets Layer, encrypts data during transmission.

CDN: Content Delivery Network, speeds up content delivery.

Load Testing: Measures system performance under load.

Redundancy: Duplication for system reliability.

API Key: Auth code for API access.

UI/UX: User Interface/User Experience, design and satisfaction.

Access Control System: Manages resource access.

Peripheral Devices: External hardware like printers.

SSO: Single Sign-On, one credential for multiple apps.

API Documentation: Detailed API information.

Load Balancing: Distributes network traffic.

Data Backup: Copies data for protection.

Disaster Recovery: Plan for catastrophic failure.

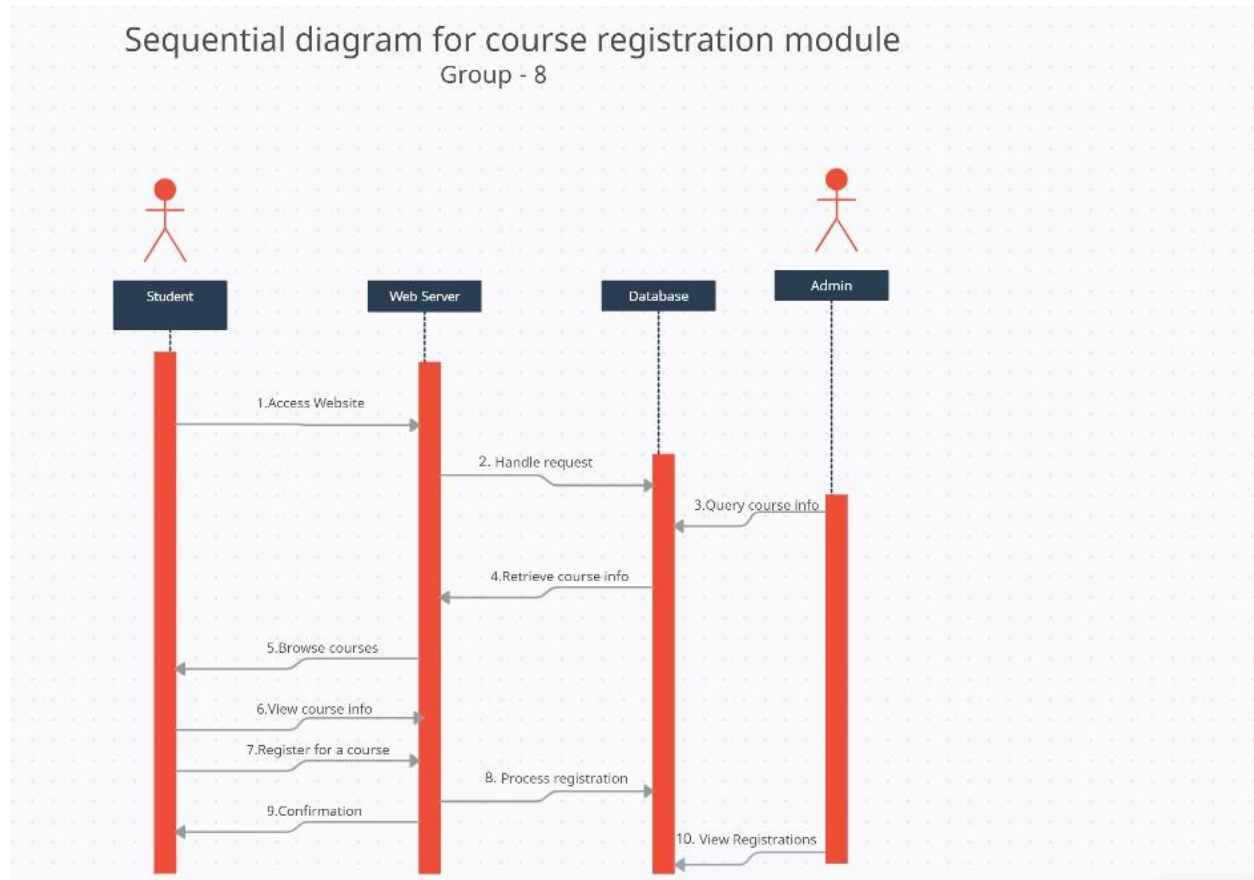
Caching: Stores frequently used data for speed.

Appendix B: Analysis Models

Use Case Diagram:

,

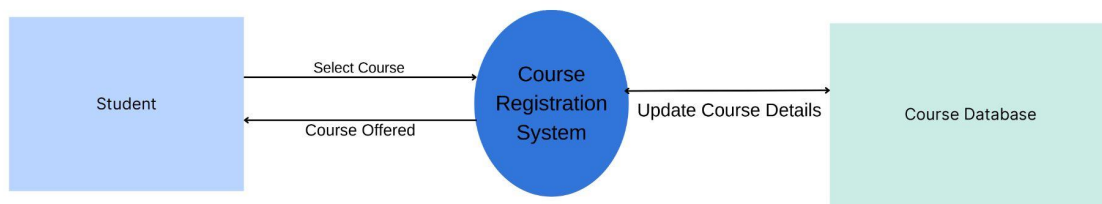
Sequence Flow Diagram:



Data Flow Diagram:

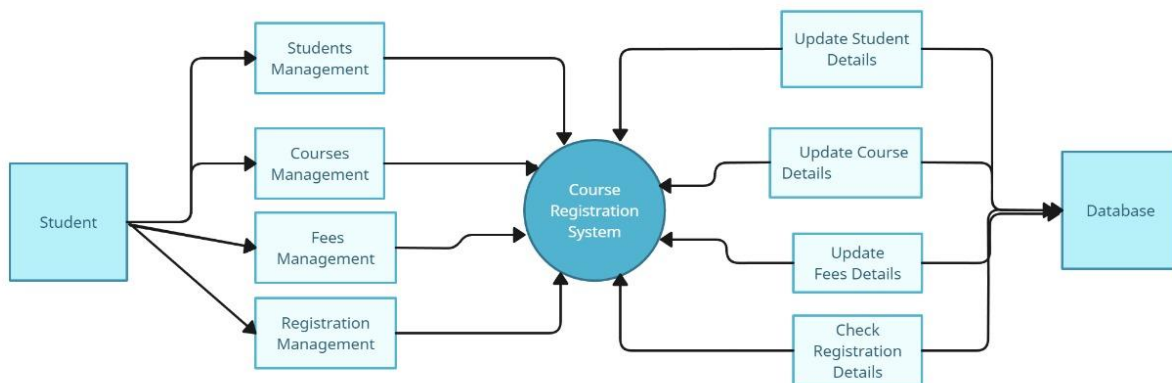
Level 0

Level 0 Dataflow Diagram

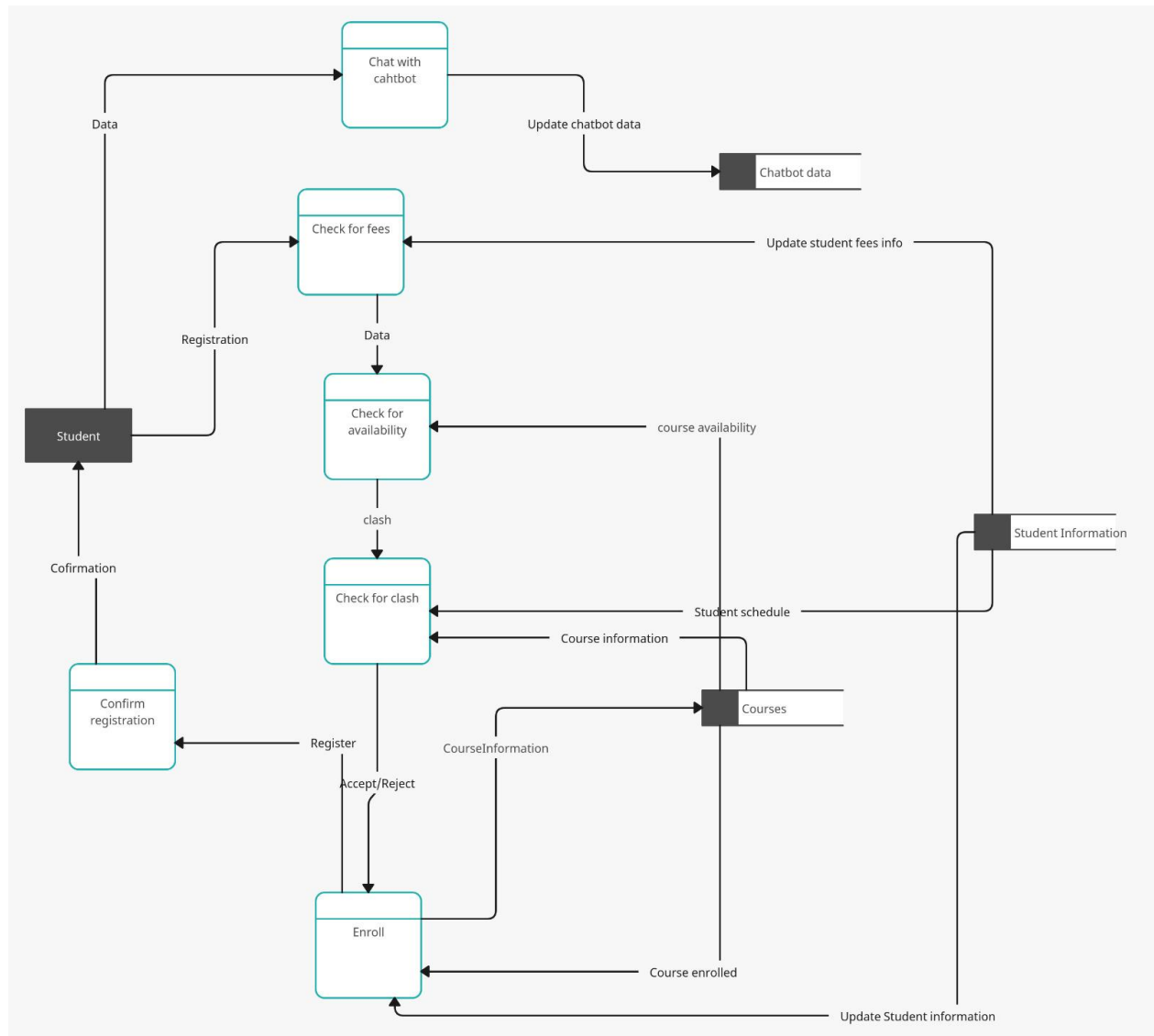


Level 1

Level 1 DataFlow Diagram



Level 2



Appendix C: To Be Determined List

TBD (to be determined)
