

Software Requirements Specification for ASTREON: AI Study Assistant and Personalized Learning Support

Version 1.0

**Prepared by
Alhambra, Berna Marie C.
Asinas, Alexander F.
Bertoso, Hans Kyle G.
Sereno, Micah T.**

SCAVES

September 20, 2024

Table of Contents

1. Introduction	1
2. Overall Description	2
3. System Features	3
4. External Interface Requirements	4
5. Other Nonfunctional Requirements	5
6. Other Requirements	5

Revision History

Name	Date	Reason For Changes	Version
Berna Alhambra	09/20/2024	initial draft	1.0 draft 1
Berna Alhambra	11/182024	Revision	1.1 draft 2

1. Introduction

1.1 Purpose

This SRS describes the software functional and non functional requirements for the release 1.0 of the ASTREON: AI Study Assistant and Personalized Learning Support (ASTREON). This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for release 1.0.

1.2 Document Conventions

This SRS document uses few font sizes for clear distinction. Main headings have bold and biggest fonts. Followed by subheadings that are also bold but smaller and lastly their body content which is a regular non italicized unless use for emphasis

1.3 Intended Audience and Reading Suggestions

This document is intended for users, documentation writers, developers and the project manager. The sequence of reading this document should be sequentially to get a good overview and details of the project.

1.4 Project Scope

The software system ASTREON being developed streamlines the processes of using learning tools and meeting learning objectives. Leveraging AI technology, we aim to help learners and teachers to have a tool that will meet their personalized learning needs and habits fast and easy.

1.5 References

Vincent, W. S. (2021). Django for beginners (3rd ed.). [Version 4.1]
Karl, W., & Joy, B. (2013). Software requirements (3rd ed.)

2. Overall Description

2.1 Product Perspective

Astreon is an innovative AI-powered study assistant designed to enhance learning and provide personalized support for students. Inspired by applications like Knowt and Gizmo, Astreon distinguishes itself by incorporating advanced features such as personalized study schedules, tailored quiz generation, and real-time feedback, all specifically catered to the needs of individual students.

2.2 Product Features

Feature	Description
Study Schedule	Allows for user to have a schedule for studying with suggested prerequisites topics that they can tackle
File Scanner	Allows for user to upload files with commonly used extensions (pdf, docx, jpg) and the system will be able to scan it
Quiz Generator	Allows for user to customize number of questions and generate it based on the scanned file
Notes Generator	Allow users to simplify and generate notes based on the uploaded file
AI Q&A Assistance	Provides AI-powered answers to user queries based on the subject or file references
Flashcard Generator	Creates flashcards from uploaded files for efficient self-study
Account Setup	Allows users to configure their accounts, including linking external accounts or managing preferences.
Account Creation	It enables new users to register and create an

	account to access all features.
Progress Tracking	Monitors user activity, including completed quizzes and sessions, and provides analytics for improvement.

2.3 User Classes and Characteristics

Guest functions:

- Generate quizzes, notes and flashcards: Guests can generate these study aids based on uploaded materials, but their outputs and usage history are not saved.
- Upload file: Guests can upload files in supported formats (e.g., PDF, DOCX, JPG) for temporary use during their session.
- Generate a study schedule: Guests can create a temporary study schedule tailored to their input preferences, but it cannot be saved for future use.
- Track Progress: Guests can view their progress within the current session only, as their data is not stored once they leave the system.

User functions:

- Generate quizzes, notes and flashcards: Users can generate these aids and save them for later review or editing.
- Upload file: Users can upload materials in supported formats, which are saved in their accounts for future reference and use.
- Generate a study schedule: Users can create, customize, and adjust study schedules, which are saved and synced across devices for seamless use.
- Track Progress: Users can track their academic performance over time, including completed quizzes, study schedules, and flashcard usage, with detailed analytics provided.

Admin functions:

- Manage user accounts: Administrators can create, update, suspend, or delete user accounts to maintain system integrity and compliance.
- Access user analytics and reports: Administrators can review platform usage statistics, user progress summaries, and other analytics to evaluate system performance and identify areas for improvement.

2.4 Operating Environment

Web/Mobile Browser

The system will operate primarily on modern web and mobile browsers. It is compatible with Google Chrome (version 76 or higher), Mozilla Firefox (version 70 or higher), and Microsoft Edge (version 79 or higher). The application ensures consistent functionality and usability across these browsers.

Operating System

The system will operate primarily on modern web and mobile browsers. It is compatible with Google Chrome (version 76 or higher), Mozilla Firefox (version 70 or higher), and Microsoft Edge (version 79 or higher). The application ensures consistent functionality and usability across these browsers.

2.5 Design and Implementation Constraints

- UI consistency - ensure that the design is consistent, intuitive and user-friendly
- File format support- ensures the wide range support of various file formats
- Security and privacy - implement strong authentication and adhere to rules in handling user's information
- Data Integrity and Redundancy - Ensure that all user data (quizzes, progress, etc.) is backed up and has redundancy to prevent data loss.
- AI Integration - Ensure proper AI integration to ensure smooth operation
- Cross-Browser and Cross-Device Compatibility -Ensure consistent functionality and appearance across supported browsers (Chrome, Firefox, Edge, Safari) and operating systems (Windows, Android)

2.6 User Documentation

The system shall provide an online hierarchical and cross-linked help system using sphinx that describes and illustrates all system functions. During a first time user accesses the app, it is expected to have a quick tutorial on how to navigate and use the application.

2.7 Assumptions and Dependencies

Assumptions

- All users have internet access - It is assumed that all users will have stable internet access to utilize the system effectively. The application's features, such as file uploads, AI-based quiz generation, and progress tracking, require an active internet connection.
- Users will upload files in specified formats - Users are expected to upload files only in the specified formats supported by the system, such as PDF, DOCX, JPG, and PNG, to ensure compatibility and smooth processing.
- The system will log your progress accurately - The system is designed to log user progress accurately, tracking activities like completed quizzes, generated flashcards, and study schedules, enabling users to monitor their academic performance reliably.
- The system will properly work on the operating environment mentioned - The system is expected to operate seamlessly on the specified operating environments, including modern web browsers (Google Chrome, Mozilla Firefox, and Microsoft Edge) and operating systems (Windows 10 or higher, Android, and macOS).

Dependencies

- File parsing libraries - The system relies on file parsing libraries to process uploaded documents and extract relevant content for generating quizzes, flashcards, and notes.
- Direct Data Ingestion - tools are necessary to enable efficient handling of large or structured datasets uploaded by users.
- Django framework - The backend of the application is developed using the Django framework, which provides robust support for building scalable web applications.
- Django rest framework - The Django REST framework facilitates the creation of APIs for seamless communication between the frontend and backend components.
- Frontend Framework (React) - The frontend of the application is built using React, chosen for its flexibility and efficiency in rendering dynamic user interfaces.
- Cloud hosting platforms (AWS, fly.io etc) - Cloud hosting platforms, such as AWS and Fly.io, provide scalable and reliable infrastructure for deploying and maintaining the application.
- PostgreSQL - used as the primary database management system, ensuring data consistency and supporting complex queries required by the system

3. System Features

3.1 Upload Lesson Files

3.1.1 Description and Priority

The system allows both students and teachers to upload lesson files in supported formats (e.g., PDF, DOCX, JPG). Once uploaded, these files serve as input for various system features, such as quiz generation, flashcard creation, and note summarization. This feature is critical for ensuring personalized learning experiences and is assigned a high priority as it serves as the foundational functionality for subsequent features.

3.1.2 Stimulus/Response Sequences

Stimulus 1: The user navigates to the "Upload Files" section of the application.

Response 1: The system displays an upload interface prompting the user to select a file from their device. Supported file formats are listed to guide the user.

Stimulus 2: The user selects a valid file and clicks the "Upload" button.

Response 2: The system validates the file format and size. If the file meets the criteria, the upload is processed successfully, and the user receives a confirmation message.

Stimulus 3: If the file format is invalid or the size exceeds the limit, the user attempts to upload the file.

Response 3: The system displays an error message, such as "Invalid file format. Please upload a PDF, DOCX, or JPG file," or "File size exceeds the 10MB limit. Please upload a smaller file."

Stimulus 4: After a successful upload, the user requests to use the uploaded file (e.g., for quiz generation or flashcard creation).

Response 4: The system processes the file content, extracts relevant information, and provides options for the user to proceed with their intended action, such as generating quizzes or notes.

3.1.3 Functional Requirements

Feature	Description
Upload.Validation	users can upload lesson files in supported formats (PDF, DOCX, JPG, PNG) for use in generating study aids
File.Upload.NotSupported	system validates the file's format and size before processing it. Unsupported files or large files will prompt an error message.
Upload.ProcessFile	system extracts relevant content from uploaded files to enable features like quiz generation and note creation.
Upload.ArchiveFile	uploaded files can be archived or deleted as needed by the user to manage their storage effectively.
Upload.DeleteFile	users can permanently delete uploaded files from their account.
Upload.ErrorMessage	displays appropriate error messages for issues such as unsupported formats or file size limits.

3.2 Generate Quizzes and Flashcards

3.2.1 Description and Priority

Referring to feature 3.1. using the files that the teacher or student uploaded, the user can use the application to generate quizzes and flashcards for both parties to use.

This feature enables users to generate quizzes and flashcards based on uploaded lesson files. The quizzes are customizable by question count and format, while flashcards are designed for quick self-study. Both tools enhance the learning experience by making uploaded content more interactive and accessible. This feature is assigned as a High = Priority for students and a Low = Priority for teachers, who can use the outputs as teaching aids

3.2.2 Stimulus/Response Sequences

Quizzes

Stimulus: the user will ask the app to generate a quiz

Response: the AI will ask for the file/lesson that the user will want the AI to take information from

Stimulus: user confirms file/lesson that they want the AI to make quizzes from

Response: creates a one-time quiz with a timer, saves the progress for the user to look back from

Flashcards

Stimulus: the user will ask the app to generate a Flashcard

Response: the AI will ask for the file/lesson that the user will want the AI to take information from

Stimulus: user confirms file/lesson that they want the AI to make quizzes from

3.2.3 Functional Requirements

Feature	Description
Generate.Quiz	generate quiz based on the files uploaded
Generate.Flashcard	generate flashcard based on the file uploaded
Generate.Quiz.Items	specifies the number of items the quiz contain
Generate.Quiz.History	user can access old quizzes taken and review them.

3.3 Generate Notes

3.3.1 Description and Priority

The user can use the file that they uploaded and ask the application to create notes from them. This will have the key points, timelines, and more that the user might need so they can study. Priority = High.

3.3.2 Stimulus/Response Sequences

stimulus: user ask for the app to make them their notes

response: AI asks which subject based on the files that the user uploaded

stimulus: user confirms file/subject

response: AI creates a word document that focuses on key points, concise explanation, and a more

understanding the version of the file that the user uploaded.

3.3.3 Functional Requirements

Feature	Description
File.Upload.GenerateNotes	Generate notes based on the file uploaded
GenerateNotes.SpecifyType	Generate notes based on the type the user wants (summarize, rewrite notes based on user's comfortability)

3.4 AI Assistance

3.4.1 Description and Priority

there will be a QnA option for the user to ask questions on their subject. This could be questions that are too broad for the AI to generate and too difficult for the student to decipher. Priority = High

3.4.2 Stimulus/Response Sequences

Stimulus: user asks question about something related to their subject of choice. they specify what lesson and what file they want the AI to reference their answer from

Response: the AI will answer based on the file that the user has given.

3.4.3 Functional Requirements

Feature	Description
Astreon.Question	the user can ask questions to the AI assistant
Astreon.Answer	AI assistant will answer the user based on what the user has specified
Astreon.Suggest	AI assistant can suggest subjects or

	lessons that they might need to get an in depth understanding of their question
--	---

3.5 Adjust Generated Schedule

3.5.1 Description and Priority

The application can generate a schedule for the user based on the time they are willing to devote to study and when they are free to study. this can be adjusted by the user at any time. Priority =Low

3.5.2 Stimulus/Response Sequences

stimulus 1: user creates an account

response 1: upon creation, AI will ask how many hours in a week they are willing to devote their time on studying and when they are free within the week

stimulus 2: user answers all question asked

response 2: AI creates the schedule, which can be adjusted at any time

3.5.3 Functional Requirements

Feature	Description
Schedule.Upload	Upload user's schedule
Schedule.Upload.Analyze	Analyzes the schedule based on the user upload
Schedule.Upload.Generate	Generate the schedule based on the analysis
Schedule.Modify	Allow for the modification of schedule

4. External Interface Requirements

4.1 User Interfaces

UI-1: The Astreon Study Buddy interface shall conform to the company's User Experience (UX) and Guidelines (TOS).

UI-2: The Astreon Study Buddy shall provide a first time tutorial how to navigate the interface, also a help Icon or command on every page that users can press/click or type to receive assistance with the A.I. chat systems functionalities.

UI-3: The Astreon Study Buddy A.I. chatbot conversation interface shall allow navigation via screen/keyboard and mouse for message entry, quick response, and navigation through message history.

UI-4: The Astreon Study Buddy A.I chat bot will accept attachment files of text files (Word, PDF,and PPT) to convert it for a reference/materials to be used as an assistance to help the user.

UI-5: The Astreon Study Buddy A.I. chat bot will also scan image files (.png, .jpeg, .svg, and .webp) to convert it for a reference/materials to be used as an assistance to help the user.

UI-6: The Astreon Study Buddy A.I. chat bot will display standard error messages in red text below the message input box when a user provides invalid input, such as "Sorry, I didn't understand that. Please try again."

UI-7: The Astreon Study Buddy A.I. chat bot will have the user interface shall include a persistent toolbar at the bottom, featuring buttons for "Restart Conversation," "End Chat," and "Feedback," accessible on all screens.

4.2 Hardware Interfaces

No hardware interfaces have been identified.

4.3 Software Interfaces

SI-1: User Authentication System

SI-1.1: The Astreon Study Buddy shall communicate with the User Authentication System (UAS) to verify user credentials (username password) through a secure entry via email.

SI-1.2: Upon successful authentication, the UAS shall transmit a session token to the Astreon Study Buddy, which will be used for the subsequent requests.

SI-1.3: If the user's session expires, the Astreon Study Buddy shall request a token refresh from the UAS.

SI-1.4: The UAS shall notify the Astreon Study Buddy when a user logs out or when an account is deactivated.

SI-2: Attachment File

SI-2.1: The Astreon Study Buddy shall pull course materials, reference, and quizzes from the attachment file to customize study plans for the user.

SI-2.2: The attachment file shall provide real-time updates to the Astreon Study Buddy when new content is added or modified.

SI-2.3: The Astreon Study Buddy shall send study progress track back to the user, including completed quizzes and practice tests.

SI-2.4: The Astreon Study Buddy shall query the data to retrieve performance analytics for the user, enabling personalized learning recommendations

SI-3: Calendar Integration

SI-4.1: The Astreon Study Buddy shall integrate with the LMS/ personal calendar (Google Calendar, Outlook, etc.) via a programmatic API to retrieve and display scheduled study sessions, exams, and assignment deadlines.

SI-4.2: The Astreon Study Buddy shall push reminders and notifications to the user's calendar for upcoming study sessions and assignments.

SI-4.3: Users shall be able to adjust their study plan directly from the Astreon Study Buddy, which will then update the calendar accordingly through the same API.

SI-4: Knowledge Base System

SI-5.1: The Astreon Study Buddy shall access a cloud-based Knowledge Base (KB) through a query based API to retrieve relevant information for answering user questions during study sessions.

SI-5.2: The KB system shall send update notifications to the Astreon Study Buddy when new articles or resources become available, ensuring the AI uses the most up-to-date information.

SI-5.3: The Astreon Study Buddy shall log user queries and their results in the KB for future analysis and optimization of the AI's recommendation engine.

4.4 Communications Interfaces

CI-1: The Astreon Study Buddy shall send an email verification to confirm the validation of the account of the user.

CI-2: The Astreon Study Buddy shall send an email and or a notification to the user to be reminded of its studying streak, daily logins, or schedule input dates.

CI- 3 Websocket: The application will communicate with the backend server by establishing a web socket connection

CI-4: JSON: All communication between the client and server will utilize JSON for data interchange

CI-5 Restful APIs: the backend will expose various RESTful APIs for various functionalities (user auth, file uploads and quiz generation)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Response Time

The system must generate quiz questions in under ≤ 20 secs under normal load condition

Throughput

The system must support up to 10 concurrent users without noticeable decrease in the performance

Load Time

The system must be fully load in ≤ 3 secs in both mobile and web environment

Availability

The system must maintain 90% of uptime for students to use the application

5.2 Safety Requirements

Data Safety

The system must protect user data from unauthorized access, modification, or deletion.

User Authentication and Authorization

Implement secure user authentication methods to prevent unauthorized access.

Backup and Recovery

Regular backups should be scheduled, and data recovery procedures must be in place to avoid data loss in case of system failures.

User Safety

Ensure the system does not cause harm to the end-users, especially in educational contexts

Monitoring and Alerts

Regular monitoring and alerts to notify users on any suspicious activity

5.3 Security Requirements

Authentication and Authorization

Ensure that only authenticated have access and authorized user will have access to certain data

Data Encryption

Encrypt data for protection while being transmitted

Vulnerability Protection

Protect against common vulnerabilities such as SQL Injection, cross site forgery attacks

Session Handling

Safeguard active sessions to prevent unwanted attacks

5.4 Software Quality Attributes

Portability

The system shall be able to run on modern browsers (Firefox,edge,chrome) and is compatible with android OS

Reusability

The system's code and components should be reusable to reduce the development time in the future

Interoperability

The system shall integrate with thor-party API that ensure seamless data exchange

Testability

The system shall be at least 80% covered by automated tests, with all critical features (login, data storage, quiz generation) tested through both unit and integration testing

6. Other Requirements

Database Requirements

The system must use PostgreSQL as the primary database.

The database schema must be designed to conform to Third Normal Form (3NF)

Audit and Logging Requirements

All user actions must be logged, including login attempts, file uploads, and profile updates.

AI Session logs should be stored for at most 3 months

Appendix A: Glossary

1. **HTTP (Hypertext Transfer Protocol):** A protocol used for transmitting data over the internet.
2. **HTTPS (Hypertext Transfer Protocol Secure):** A secure version of HTTP, where data transferred between the browser and server is encrypted using SSL/TLS, ensuring privacy and data integrity.
3. **AI (Artificial Intelligence):** The simulation of human intelligence by machines, particularly computers.
4. **API (Application Programming Interface):** A set of rules and protocols for building and interacting with software applications.
5. **LMS (Learning Management System):** A software application or platform used for the administration, documentation, tracking, and delivery of educational courses or training programs.
6. **UI (User Interface):** The means by which a user interacts with a system, typically involving elements like buttons, icons, and screens. It focuses on the layout and design of these elements.
7. **UX (User Experience):** The overall experience a user has when interacting with a product or system.
8. **JSON (JavaScript Object Notation):** A lightweight data-interchange format that is easy for humans to read and write, and easy for machines to parse and generate.
9. **REST (Representational State Transfer):**
A set of constraints for creating scalable web services. REST-compliant web services allow requests for data to be made over the internet in a stateless manner, usually with JSON or XML.
10. **RESTful (Representational State Transfer):** An architectural style for designing networked applications, where systems interact through stateless operations and standard HTTP methods like GET, POST, PUT, and DELETE.
11. **KBS (Knowledge-Based System):** A system that uses a knowledge base of human expertise to solve complex problems, often involving reasoning and learning components.

Appendix B: Analysis Models

System Architecture

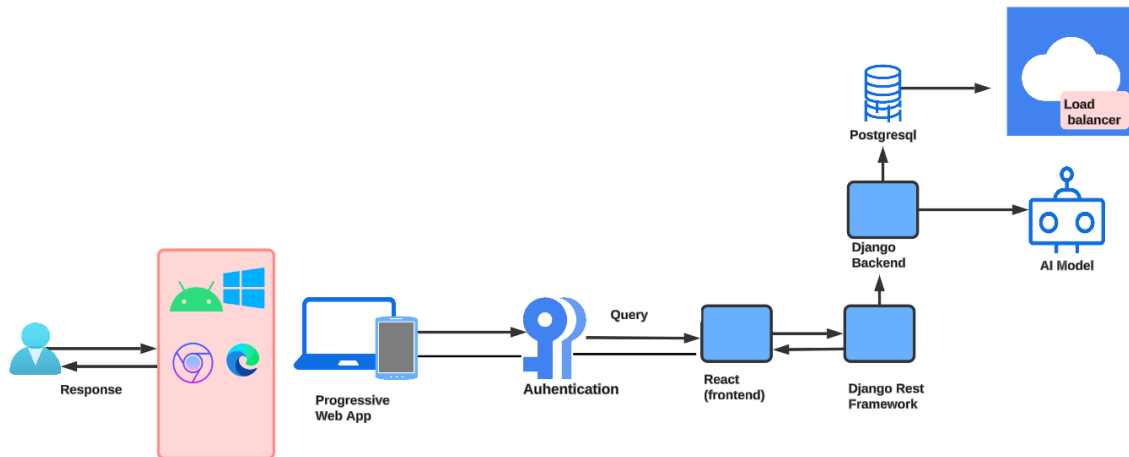


Fig 1 System Architecture

At the user interface level, individuals interact with the system through a Progressive Web App (PWA) accessible via various browsers and mobile platforms, ensuring consistent cross-platform accessibility. The React Frontend serves as the user interface layer, communicating with the backend through APIs to enable core functionalities like course enrollment, flashcard generation, and interaction with learning tools.

The system's authentication service manages user login, registration, and authentication, maintaining secure communication between frontend and backend components. At the heart of the system, the Django Backend handles the core business logic, processing frontend requests and managing features like AI-powered study resource generation, progress tracking, and comprehensive data management for courses and learning plans. The Django REST Framework facilitates seamless API communication, handling data serialization and deserialization effectively.

The AI Model integration, hosted on an Amazon EC2 instance, enables advanced features such as personalized study plans and automated quiz generation. Data storage is handled by a PostgreSQL database, which maintains user data, course information, and progress tracking details. An Amazon EC2 load balancer ensures high availability and optimal traffic distribution across servers. The system completes its workflow by sending processed data back to the frontend for display in the PWA.

This architecture supports key features for both students and teachers. Students benefit from personalized learning tools including quizzes, flashcards, and schedules, while also tracking their progress through streaks and mastery metrics. Teachers can create courses, manage students, and utilize AI-powered content generation. The system leverages real-time updates and scalability through Amazon EC2, while maintaining cross-platform compatibility through its PWA architecture.

ER Diagram (Database)

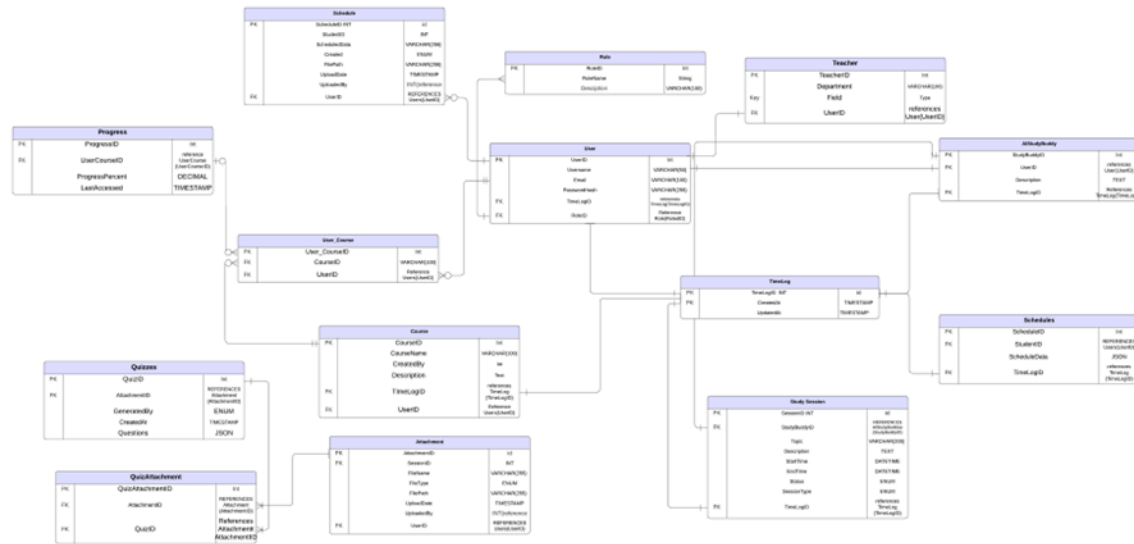


Fig 1.1 ER Diagram

The ER diagram provides a detailed representation of the database structure for the Astreon Study Buddy system, comprising multiple interconnected entities and their attributes. The User entity forms the foundation, storing information about all system users including students, teachers, and admins, with attributes such as UserID (Primary Key), Username, Email, PasswordHash, and RoleID (Foreign Key) linking to the Role table. The Role entity defines user types with attributes including RoleID, RoleName, and Description, while the Teacher entity contains specific attributes like TeacherID, Department, and Field, connected to the User table via UserID.

The Course entity represents available courses in the system, containing CourseID, CourseName, Description, and foreign keys linking to creators and time logs. The User_Course entity serves as an associative entity connecting users and courses, establishing many-to-many relationships. Progress tracking is handled by the Progress entity, which monitors user activity within courses through attributes like ProgressID, ProgressPercent, and LastAccessed, linked to specific users and courses via UserCourseID.

The system includes several supporting entities: Quizzes stores quiz details with attributes like QuizID, GeneratedBy (ENUM), and Questions (JSON); QuizAttachment tracks quiz-related attachments; and the Attachment entity stores file-related metadata. The TimeLog entity maintains creation and update timestamps, while Schedule manages student-specific scheduling details through ScheduleID and ScheduleData (JSON). Study Session tracks individual learning sessions with attributes such as SessionID, StartTime, and EndTime, and the AIStudyBuddy entity manages AI interactions and content descriptions.

The relationships between these entities are structured in various ways:

Many-to-One relationships exist between User and Role (where a user has one role, but a role can be assigned to many users) and between Course and User (where a course has one creator but multiple associated users).

Many-to-Many relationships are established between User and Course through the User_Course entity, and

One-to-Many relationships connect Quizzes to Attachments and Schedule to User. This comprehensive diagram serves to visualize the system's data model, demonstrating how users, courses, quizzes, and schedules interact while maintaining system integrity through primary and foreign key constraints.

Data Flow Diagram (Context Diagram)

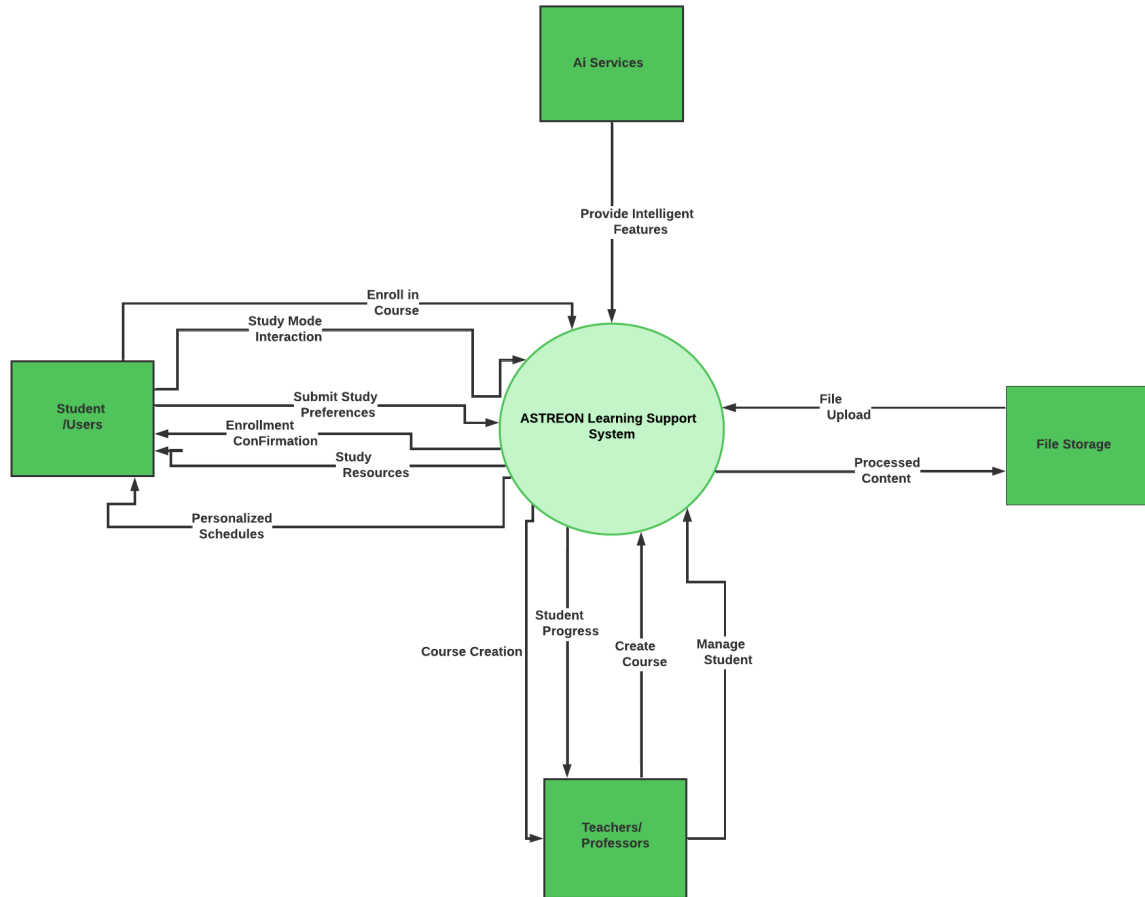


Fig 1.2 Data Flow Diagram (Context Diagram)

The 0-Level DFD (Context Diagram) illustrates the high-level data flow and interactions between the main components of the Astreon Learning Support System (ALSS) and its external entities. At the center of the diagram is the ALSS, represented by a green circle, which functions as the core system responsible for processing, storing, and managing all user-related data and activities.

The system interacts with several external entities, beginning with the Users (Students) who engage with the system through multiple channels. Students can enroll in courses, upload files for conversion into study resources or assignment submission, access AI-generated learning materials like flashcards and quizzes, and maintain continuous

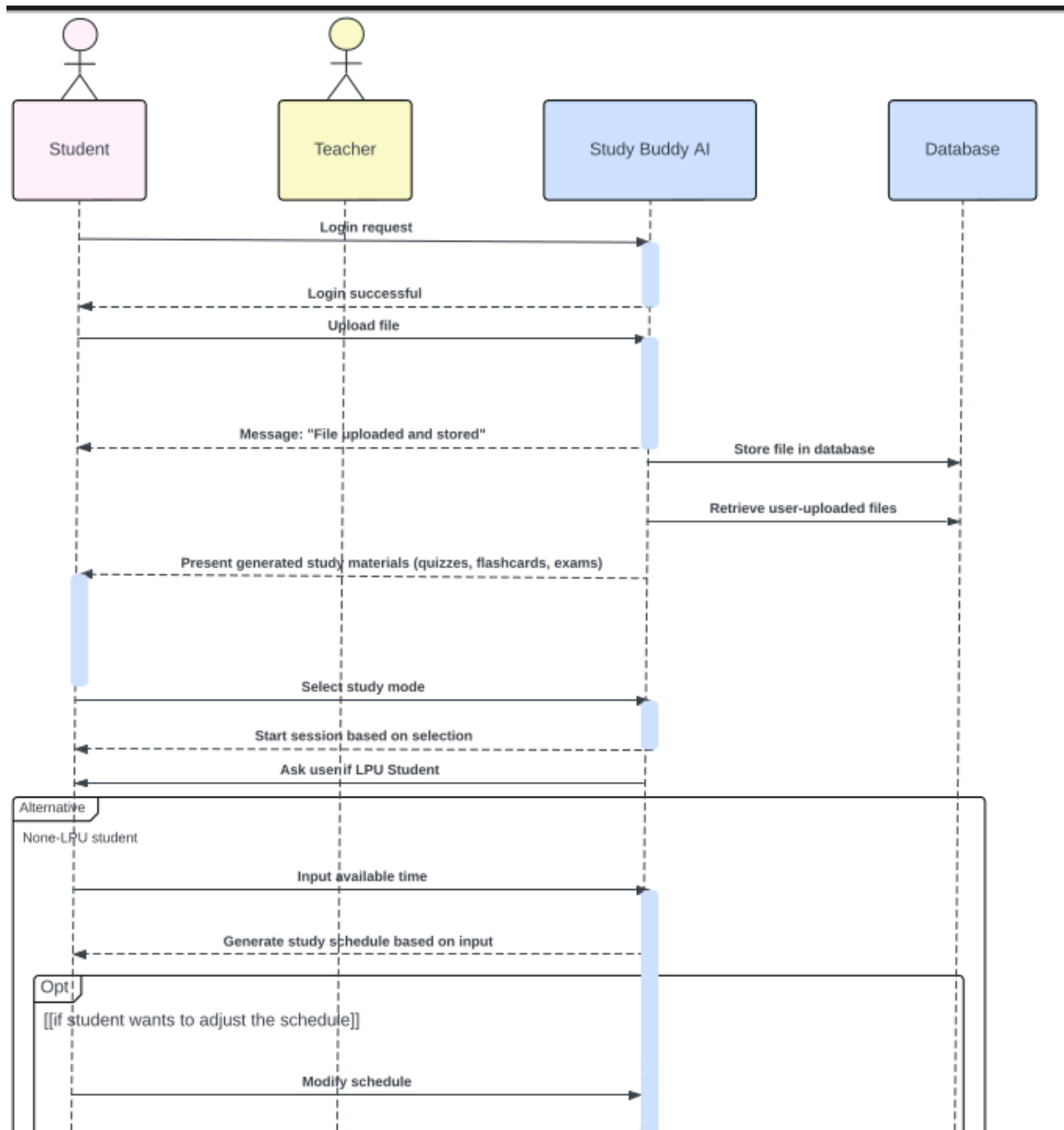
interaction with the system through various learning modes. The system provides enrollment confirmation back to students upon successful registration.

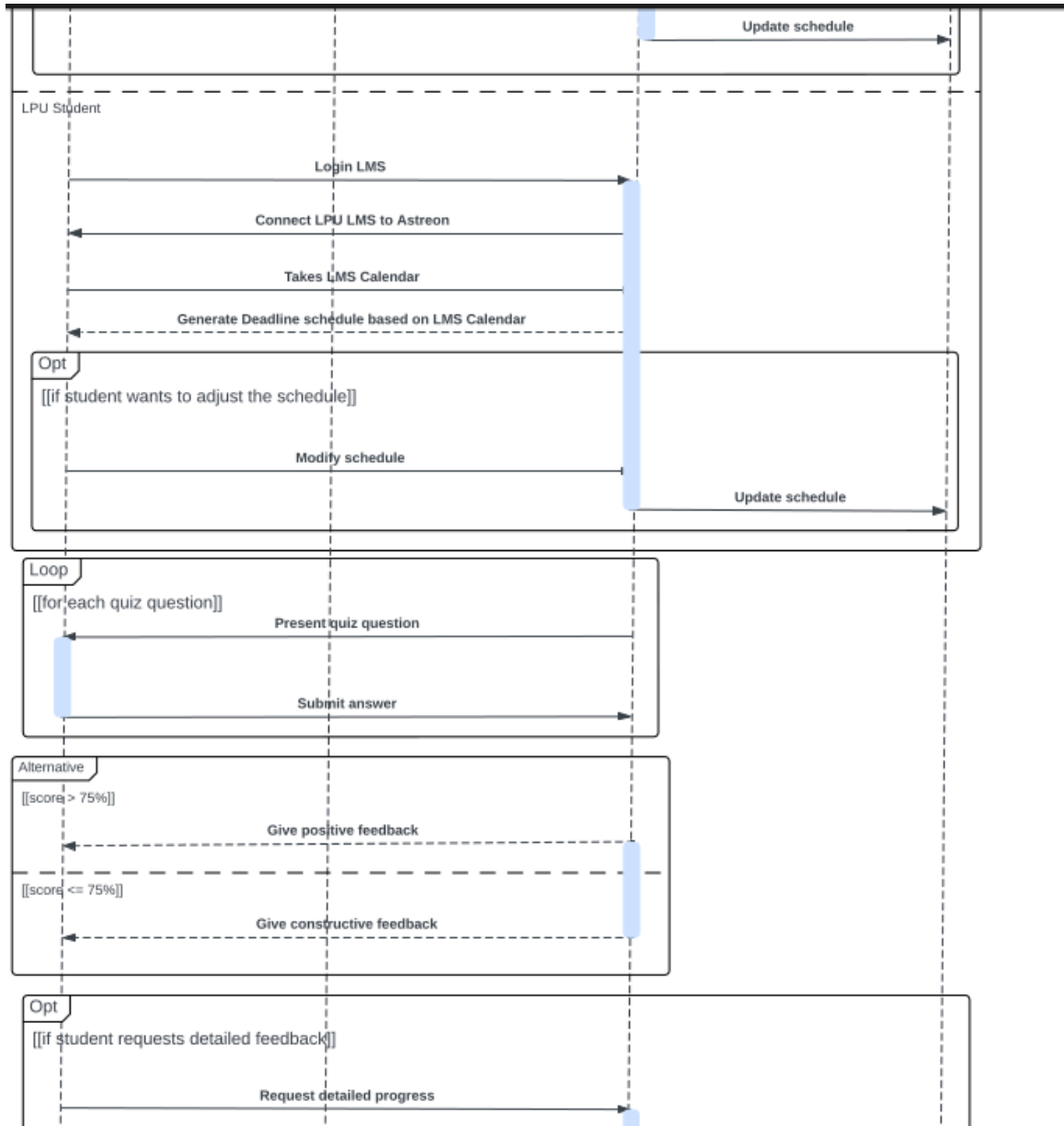
Teachers and Professors represent another key external entity, with specific functionalities designed for educational management. They can create and manage courses, oversee student progress, review submissions, and distribute activities. The system provides them with analytical tools to monitor student performance and content mastery. AI Services form another crucial component, providing intelligent features such as generating quizzes, flashcards, and personalized schedules, while also processing and optimizing user inputs for enhanced learning outcomes.

The File Storage component handles various file-related operations, including storing user-uploaded files, retrieving content when requested, and maintaining processed content such as OCR-processed documents or AI-generated study materials. The data flows within the system following specific patterns: students send enrollment requests and files while receiving study resources and confirmations; teachers manage courses and track progress; AI services process requests and return results; and the ALSS continuously interacts with file storage for content management.

This context diagram effectively demonstrates how the Astreon Learning Support System serves as a central hub connecting students, teachers, AI services, and file storage. It clearly shows the primary inputs and outputs flowing through the system, aligning with the goal of delivering personalized learning experiences through efficient data integration and processing.

Sequence Diagram





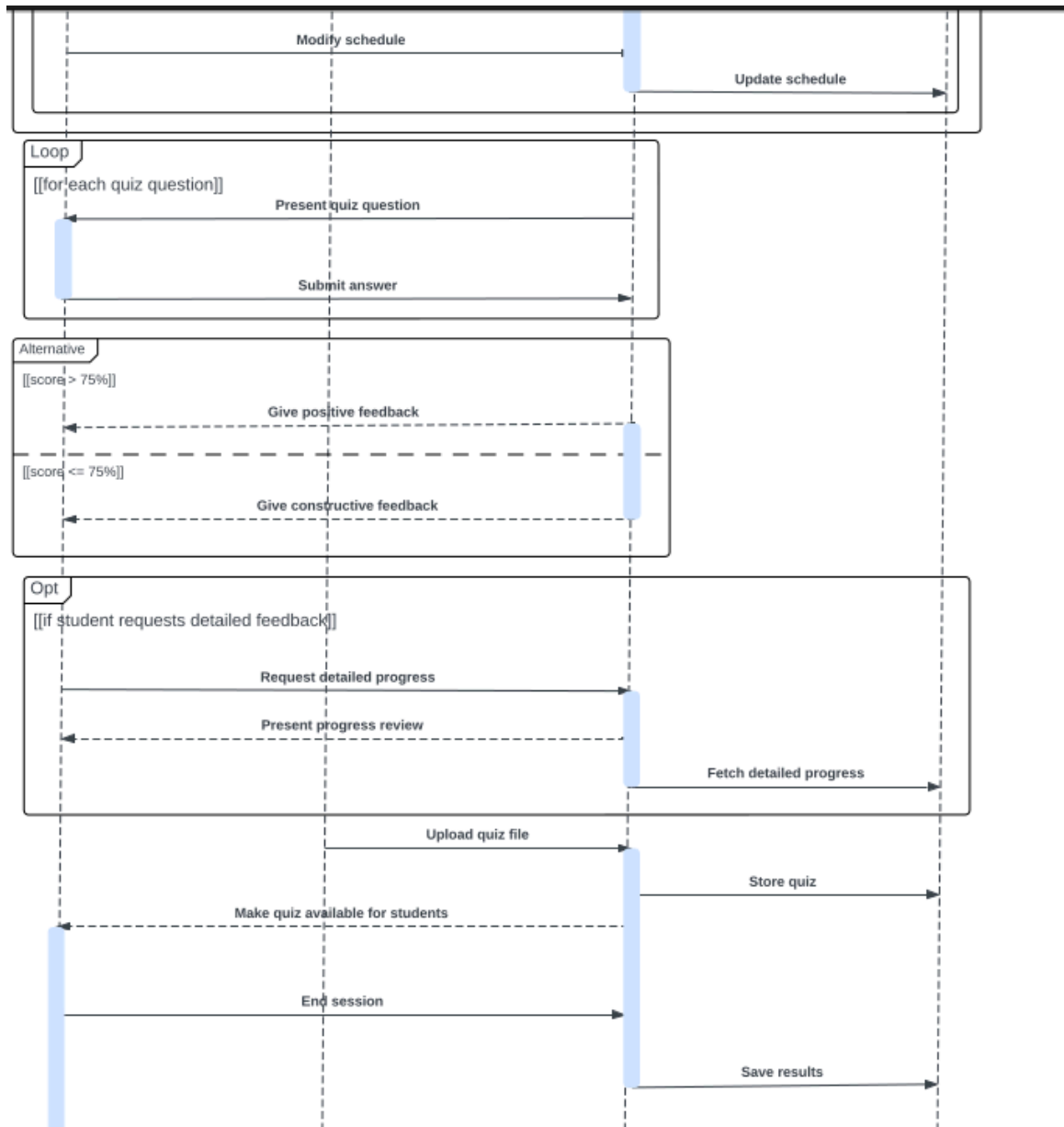


Fig 1.3 Sequence Diagram

The sequence diagram illustrates the detailed flow of interactions between various entities in the Study Buddy system. The diagram involves four key actors: the Student (a user accessing course content), the Teacher (an educator managing educational material), the

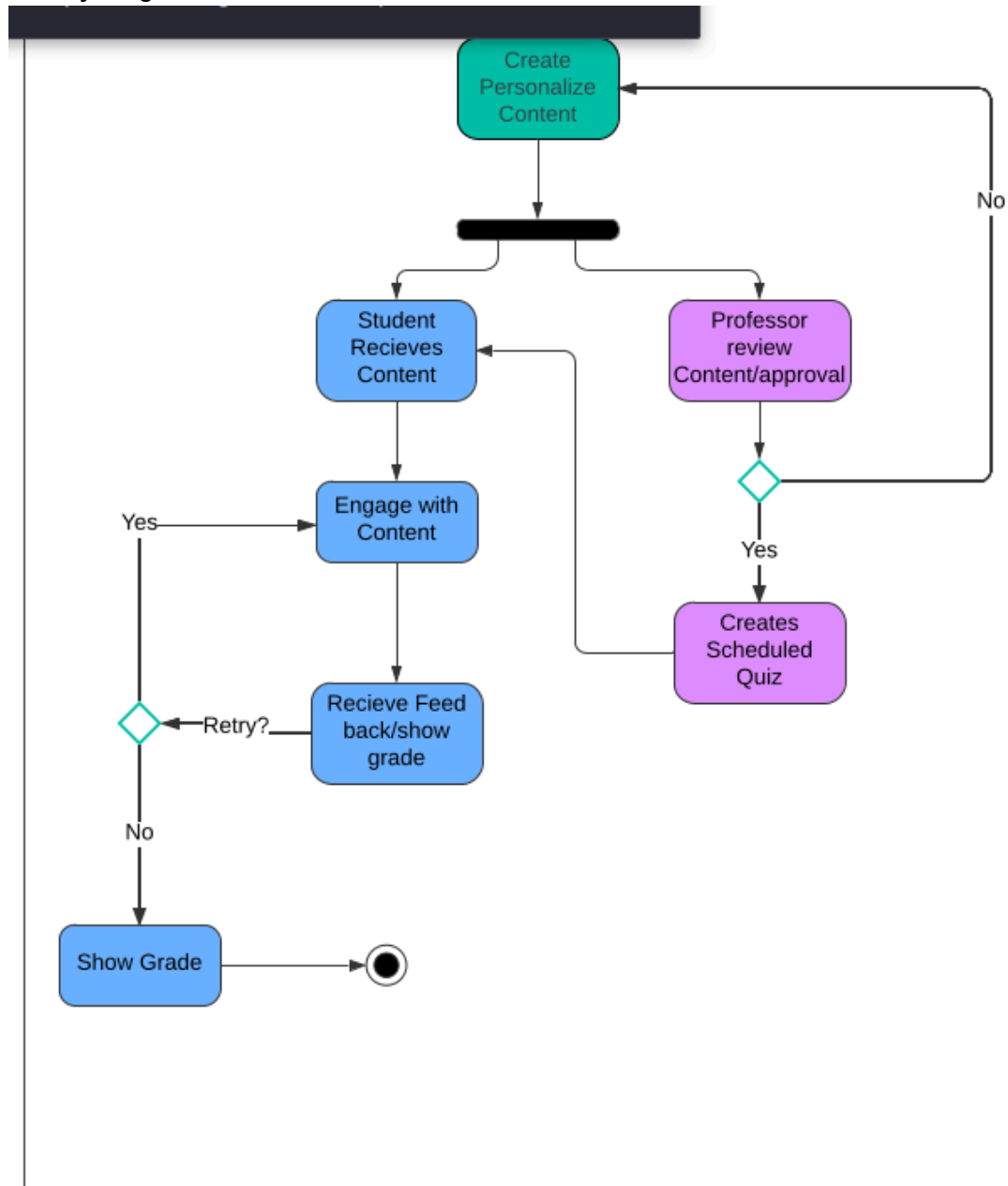
Study Buddy (the primary system facilitating user actions), and the Database (storing user data, schedules, files, quiz results, and progress).

The interaction flow begins with the login process, where students or teachers submit login credentials. Study Buddy verifies these credentials with the database, resulting in either successful access or an unsuccessful login notification. Once logged in, users can create schedules such as study timetables or quiz schedules, which the system stores in the database along with associated course information. The file upload process allows users, primarily teachers, to upload course materials or resources, with the system storing these files and returning either success or failure messages.

Content creation represents another key interaction, where users can create or request content like quizzes or personalized learning material, which is stored in the database and made available for use. Students then interact with this content through quizzes or flashcards, with questions presented in a loop until completion and responses stored in the database. After completing activities, the system presents results by retrieving stored data, providing either positive feedback for scores above a threshold (e.g., 75%) or constructive feedback for improvement when scores fall below the threshold.

The final major interaction involves progress tracking, where students can request to view their grades and quiz history, with Study Buddy fetching and displaying this data from the database. Notable observations from the diagram include its emphasis on feedback-driven learning with continuous improvement through quizzes and personalized feedback, while the teacher's role focuses primarily on resource provision as students engage with content and monitor their progress.

Activity Diagram



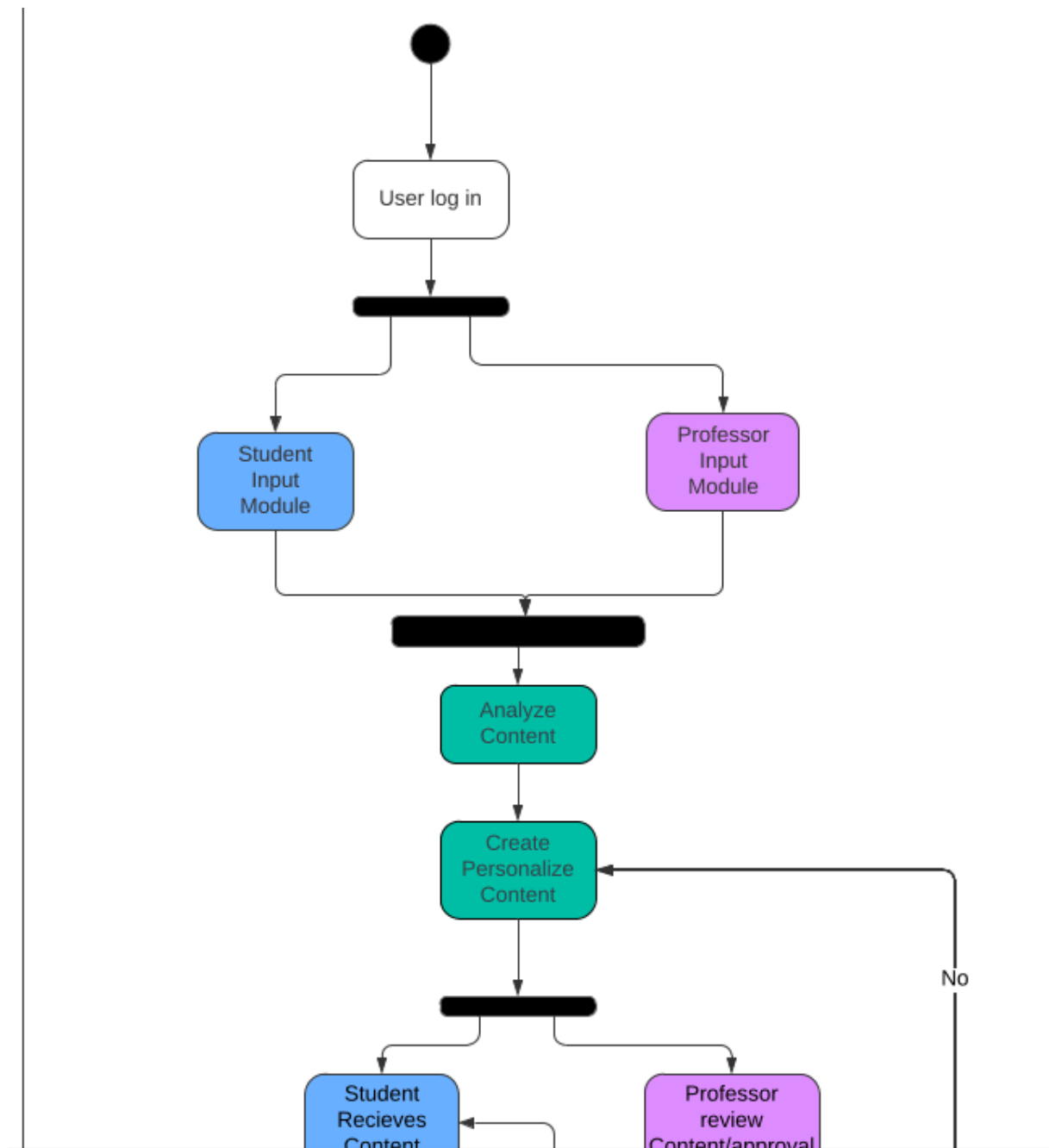


Fig 1.4 Activity Diagram

The activity diagram for the Astreon project illustrates the interaction flow between students, professors, and the system, with a focus on personalized content delivery and evaluation. The process initiates when a user logs into the system, after which the flow

diverges into two distinct paths based on the user's role: students are directed to the Student Input Module for system interaction, while professors access the Professor Input Module for content management and input provision.

Once user input is received, the system analyzes the content by gathering and processing data to understand user needs and actions. This analysis leads to the creation of personalized content tailored to individual student requirements. The generated content then undergoes a review and approval process by professors to ensure quality and relevance. If approved, the content moves forward in the process; if not, it returns to the content creation stage for refinement.

After approval, the content is delivered to students for engagement, allowing them to study and complete exercises. The system provides immediate feedback or grades based on student performance. If students are unsatisfied with their results or fail to meet requirements, the system offers options to retry or re-engage with the content. Additionally, professors can create scheduled quizzes for further student evaluation. The process culminates with the display of final grades upon completion of all activities.

This diagram for Astreon, demonstrating an educational platform designed to deliver personalized learning experiences, facilitate continuous assessment and feedback, and enable professors to curate and review learning materials. The flow ensures a comprehensive learning experience while maintaining quality through professor oversight and allowing for student improvement through multiple attempts.

Use-case Diagram

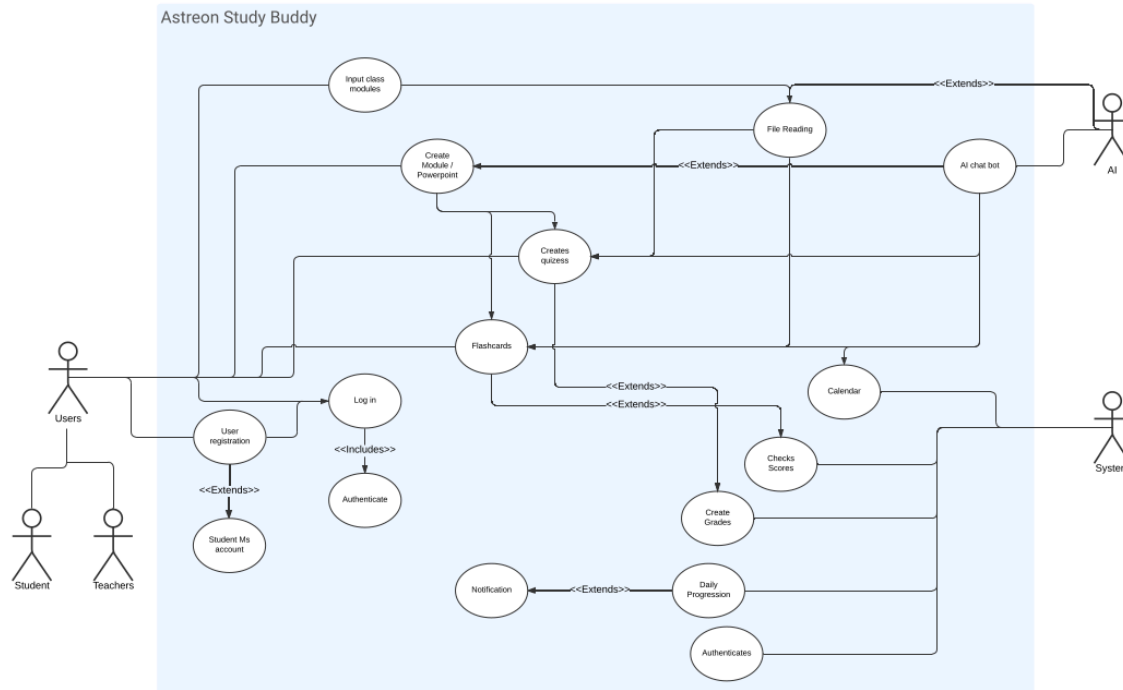


Fig 1.5 Use Case Diagram

This diagram provides a visual representation of interactions between users (actors) and system functionalities (use cases), organized in multiple layers to show both core and optional system operations. The system involves several key actors: Users, which include students who can log in, use study tools, and track progress, and teachers who assist with content creation and progress monitoring. Both groups participate in creating modules, quizzes, and flashcards. Additional actors include AI, which assists with file reading and powers the chatbot feature; the System, which handles notifications, scoring, and other operations; and Admin, who manages content, users, system monitoring, and analytics.

The primary use cases encompass several key functionalities. User Registration & Login allows students and teachers to register and access the system through authentication. Content Creation enables users to create modules, such as PowerPoints or lessons, and generate quizzes and flashcards for learning reinforcement. AI Features include file reading for enhanced content processing and an AI chatbot for interactive learning experiences. The system also includes Calendar & Notifications for tracking daily progression and sending reminders, as well as Scores and Progression features for monitoring daily progress and maintaining motivation.

The diagram utilizes two types of relationships: "Includes" relationships indicate mandatory features required to complete a function (such as Login including Authentication), while "Extends" relationships show optional or conditional functionalities (such as notifications extending from progress tracking). Overall, the Astreon Study Buddy Use Case Diagram demonstrates a comprehensive educational platform that integrates content creation, learning progress tracking, and AI interaction, with students and teachers as primary users and admin oversight of backend operations.

Appendix C: Issues List

The dynamic list of open requirements issues, including TBDs, pending decisions, and conflicts, is actively tracked in the Jira project management tool. This list is updated in real-time by the project team.