

Study Aid Provider AI



Syed Asad Abbas Rizvi (BSE203026)

Muhammad Saad Tufail(BSE213070)

Javeria Babar (BSE213045)

Supervised by Dr. Aamer Nadeem

Fall 2024

BS Software Engineering

Capital University of Science and Technology, Islamabad

Table of Contents

Chapter 1	4
1. Introduction	4
1.1. Objectives.....	5
1.2. Existing Examples / Solutions	6
1.3. Proposed System.....	6
1.3.1. Seamless Voice Recognition and Transcription.....	6
1.3.2. Multilingual Support for Global Accessibility	7
1.3.3. Intelligent Note Categorization and Organization	7
1.3.4. Key Points Highlighting and Summarization	7
1.3.5. Additional Learning Resources Integration	7
1.3.6. Interactive and Dynamic Note-taking	8
1.3.7. Collaborative Learning Environment.....	8
1.3.8. Personalized Reminders for Academic Deadlines.....	8
1.3.9. Cross-Device Accessibility	9
1.3.10. Advanced Summarization using Large Language Models (LLMs).....	9
1.4. Business Scope	9
1.5. Useful Tools and Technologies.....	10
1.5.1. Flutter.....	10
1.5.2. Android Studio	10
1.5.3. VS Code	11
1.5.4. Python	12
1.5.5. LangChain.....	12
1.5.6. MySQL	13
1.6. Project Timeline	14
1.7. Project Work Breakdown	16
Chapter 2.....	17

2.	Requirement Specification and Analysis	17
2.1.	Functional Requirements	17
2.2.	Non-Functional Requirements	18
2.3.	System Use Case Modeling	20
2.4.	System Sequence Diagram.....	53
2.5.	Entity Relationship Diagram.....	83
2.6.	Domain Model	83
2.7.	User Interfaces	85

Chapter 1

1. Introduction

In an era where education and technology are merging more closely than ever, the need for efficient, personalized, and accessible study tools has become paramount. The rapid advancements in artificial intelligence, voice recognition, and interactive systems have opened up new avenues for innovation in the education sector. This project aims to develop a comprehensive **Study Aid Provider AI**, an application designed to revolutionize the way students engage with their learning material.

This AI-powered application leverages cutting-edge **Voice Recognition** and **Transcription** technology to allow students to convert their verbal notes into text effortlessly. Furthermore, by supporting **multiple languages**, it provides accessibility to a diverse range of users across different linguistic backgrounds, making it a globally adaptive tool.

One of the key features of the system is its ability to **automatically categorize and organize notes** by subject name, date, and topic, ensuring that students can easily retrieve and manage their notes. Moreover, the system uses **natural language processing** and **LLMs (Large Language Models)** to generate summaries of key points, thereby reducing cognitive load and helping users focus on essential material.

To enrich the learning experience, the AI integrates **additional resources** relevant to the topics being studied, allowing for a more comprehensive understanding of the material. Additionally, the app supports **Interactive Notes**, enabling students to interact dynamically with their content.

The collaboration aspect is another standout feature, enabling students to **share and work together on notes**, fostering a community-based learning environment. The system also includes a **reminder functionality** for upcoming assignments, quizzes, and exams, ensuring that students remain on top of their academic responsibilities. Accessibility is a critical focus,

with the application designed to be **cross-device compatible**, allowing students to use it seamlessly across smartphones, tablets, and PCs.

In summary, the **Study Aid Provider AI** is a holistic educational tool that not only helps students organize and engage with their learning materials but also enhances their overall experience through intelligent assistance, collaboration, and user-friendly design. This innovative approach has the potential to reshape the way students learn and interact with academic content, making it an essential companion for the modern learner.

1.1. Objectives

The main objective of the Study Aid Provider AI application is to create an innovative, intelligent tool that revolutionizes how students interact with their study materials. By integrating advanced technologies like voice recognition, multilingual support, automatic note organization, and large language models (LLMs) for summarization, the application aims to make learning more accessible, personalized, and efficient. It allows students to seamlessly convert spoken notes into text, organize their materials automatically, and focus on key points through AI-generated summaries, reducing the cognitive burden on learners. Additionally, the app fosters collaboration and engagement through interactive features like dynamic notetaking.

Before the development of this application, students often faced challenges with manual notetaking, scattered learning resources, and disorganized study materials. Traditional methods demanded significant effort to organize and retrieve information, and collaborative study options were limited. With the rise of digital education, many existing solutions lacked the personalization, linguistic inclusivity, and dynamic features that modern students require.

This application seeks to address these gaps by streamlining the study process, providing personalized and efficient solutions that save time and enhance engagement. By automating key tasks and fostering collaboration, the Study Aid Provider AI offers a comprehensive solution that not only makes studying easier but also more interactive and efficient, ultimately improving academic performance and user experience.

1.2. Existing Examples / Solutions

The following table demonstrates the comparison of our proposed system with the existing systems.

Facilities	Proposed System	Evernote	Onenote	Otter.ai	Google Keep	Notion
Voice Recognition	Yes	Basic	Basic	Yes	Yes	No
Transcription	Yes	No	No	Yes	No	No
Multi-language Support	Yes	Yes	Yes	Limited	Yes	Yes
Automatic Note Categorization	Yes	No	No	No	No	No
Note Organization	Yes	Yes	Yes	No	Yes	Yes
Highlighting Key Points	Yes	Yes	Yes	No	Yes	Yes
Automated Summary Creation	Yes	No	No	Yes	No	No
Additional Resource Support	Yes	Yes	Yes	No	Yes	Yes
Interactive Notes	Yes	Limited	Yes	No	No	Yes
Reminders	Yes	No	Limited	No	Yes	Yes
Accessibility	Yes	Yes	Yes	Yes	Yes	Yes
Collaboration with Other Users	Yes	Limited	Yes	Limited	Limited	Yes

Table 1 Existing Systems

1.3. Proposed System

The following proposed system outline the key functionalities and goals of the **Study Aid Provider AI** application:

1.3.1. Seamless Voice Recognition and Transcription

The primary objective is to integrate a highly accurate **Voice Recognition** and **Transcription** feature, allowing students to effortlessly convert their spoken words into written notes. This will significantly streamline the note-taking process,

particularly for those who prefer auditory learning or have difficulty with traditional methods.

1.3.2. Multilingual Support for Global Accessibility

To ensure inclusivity, the system will support **multiple languages**, making it accessible to users from various linguistic backgrounds. This will foster a more personalized and localized learning experience, catering to students around the world.

1.3.3. Intelligent Note Categorization and Organization

The application will implement an **automatic categorization** system that organizes notes by subject, date, and topic. This ensures that users can easily retrieve their notes in a structured and organized manner, improving their study efficiency.

1.3.4. Key Points Highlighting and Summarization

A core feature of the AI system is to automatically **highlight key points** in the notes and generate concise **summaries**. This will allow students to focus on the most important information without needing to manually sift through long-form notes.

1.3.5. Additional Learning Resources Integration

The application will offer **additional support** by providing users with curated learning resources related to the topics they are studying. This feature will help students deepen their understanding and enrich their study material with minimal effort.

1.3.6. Interactive and Dynamic Note-taking

An innovative feature of the system is **Interactive Notes**, where users can engage with their notes in a more dynamic way, such as through annotations, comments, or question-and-answer formats, which foster deeper engagement and learning.

1.3.7. Collaborative Learning Environment

The AI will allow for **collaboration**, enabling students to share notes and work together with classmates. This feature promotes a community-based learning experience and makes group study sessions more effective and organized.

1.3.8. Personalized Reminders for Academic Deadlines

The system will feature personalized **reminders for upcoming assignments, quizzes, and exams**. This ensures that students can manage their academic schedules efficiently without the risk of missing important deadlines.

1.3.9. Cross-Device Accessibility

To cater to students' need for flexibility, the application will be **accessible across multiple devices**, ensuring that users can access their study materials on smartphones, tablets, and desktop computers without any compatibility issues.

1.3.10. Advanced Summarization using Large Language Models (LLMs)

Leveraging the power of **LLMs (Large Language Models)**, the AI will generate tailored summaries of notes and materials according to the user's specific needs, providing concise and personalized study content at the click of a button.

1.4. Business Scope

The primary focus of this project is to develop a cutting-edge notetaking and study aid platform that addresses the growing need for efficient and interactive learning solutions. Students often face challenges in organizing their study materials, collaborating with peers, and managing deadlines. To solve these issues, our system leverages advanced **Voice Recognition and Transcription** technology, allowing users to easily record and transcribe notes in real-time, with support for **multiple languages**. The system goes beyond basic notetaking by automatically categorizing notes by **subject, date, and topic**, and using **LLMs** to generate **summaries** and highlight **key points**, making studying and revision more focused and effective.

To further enhance user engagement and productivity, the platform features **Interactive**

Notes that can be shared and collaboratively edited with peers, fostering a dynamic learning environment. **Reminders** for assignments, quizzes, and exams are integrated to help students manage their academic schedules effectively. The system is designed for **multidevice accessibility**, ensuring that users can access and edit their notes from anywhere. This project aims to revolutionize the study experience by combining the latest in AI and note-taking technologies.

1.5. Useful Tools and Technologies

1.5.1. Flutter

Flutter serves as the foundation for mobile app development by enabling the creation of a cross- platform application that works seamlessly on both iOS and Android devices. The framework's rich widget library simplifies UI design, while its robust performance and efficiency guarantee a responsive and user-centric mobile app for parking management.

This is the foundation for building the **cross-platform mobile application**, ensuring compatibility across both iOS and Android devices. It will manage the app's UI, navigation, and user experience. In Simple terms, we are using Flutter to create the mobile application.



Figure1 Flutter Logo

1.5.2. Android Studio

Android Studio is the primary integrated development environment (IDE) for our project's mobile app development as it provides a specialized environment for Android app creation while offering a range of features such as tools for code writing, debugging, and testing. Not only does it offer emulators and real device testing capabilities, enhancing the quality

and performance of the app but it is also integrated with the Android SDK and have access to Google's services making it the most suitable choice for developing our Android and iOS- compatible mobile app.

As the **IDE** for Android development, it will handle mobile app development, debugging, testing, and device emulation for Android. It integrates the Android SDK, making it useful for performance and quality testing.



Figure 2 Android Studio Logo

1.5.3. VS Code

VS Code will serve as an alternative integrated development environment IDE in case Android Studio encounters limitations in our project as it offers flexibility by supporting various programming languages, which can be advantageous for specific coding requirements.

As an **alternative IDE**, you might use VS Code for code editing, particularly when working with languages like Python or if you face limitations with Android Studio during development.



Figure 3 Visual Studio Code Logo

1.5.4. Python

Python is a general-purpose language, which means it's designed to be used in a range of applications, including data science, software and web development, automation, and generally getting stuff done.

Python will be used to integrate **data science** tasks, handle **back-end logic**, and work with **machine learning models** for tasks such as note summarization, data processing, and automation.



Figure 4 Python Logo

1.5.5. LangChain

LangChain is a software framework that helps facilitate the integration of large language models into applications. As a language model integration framework, LangChain's usecases largely overlap with those of language models in general, including document analysis and summarization, chatbots, and code analysis.

It will be used to integrate **LLMs (Large Language Models)** for advanced tasks like **document summarization, AI-driven notes categorization, and key points highlighting.**



Figure 5 LangChain Logo

1.5.6. MySQL

MySQL is the world's most popular open-source database. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo! and many more.

Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded applications.

As a **relational database**, it will store **user data, notes**, and manage **backend data storage**, ensuring scalability, data retrieval, and security for the application.



Figure 6 MySQL logo

1.6. Project Timeline

The timeline set for this project would be flexible.

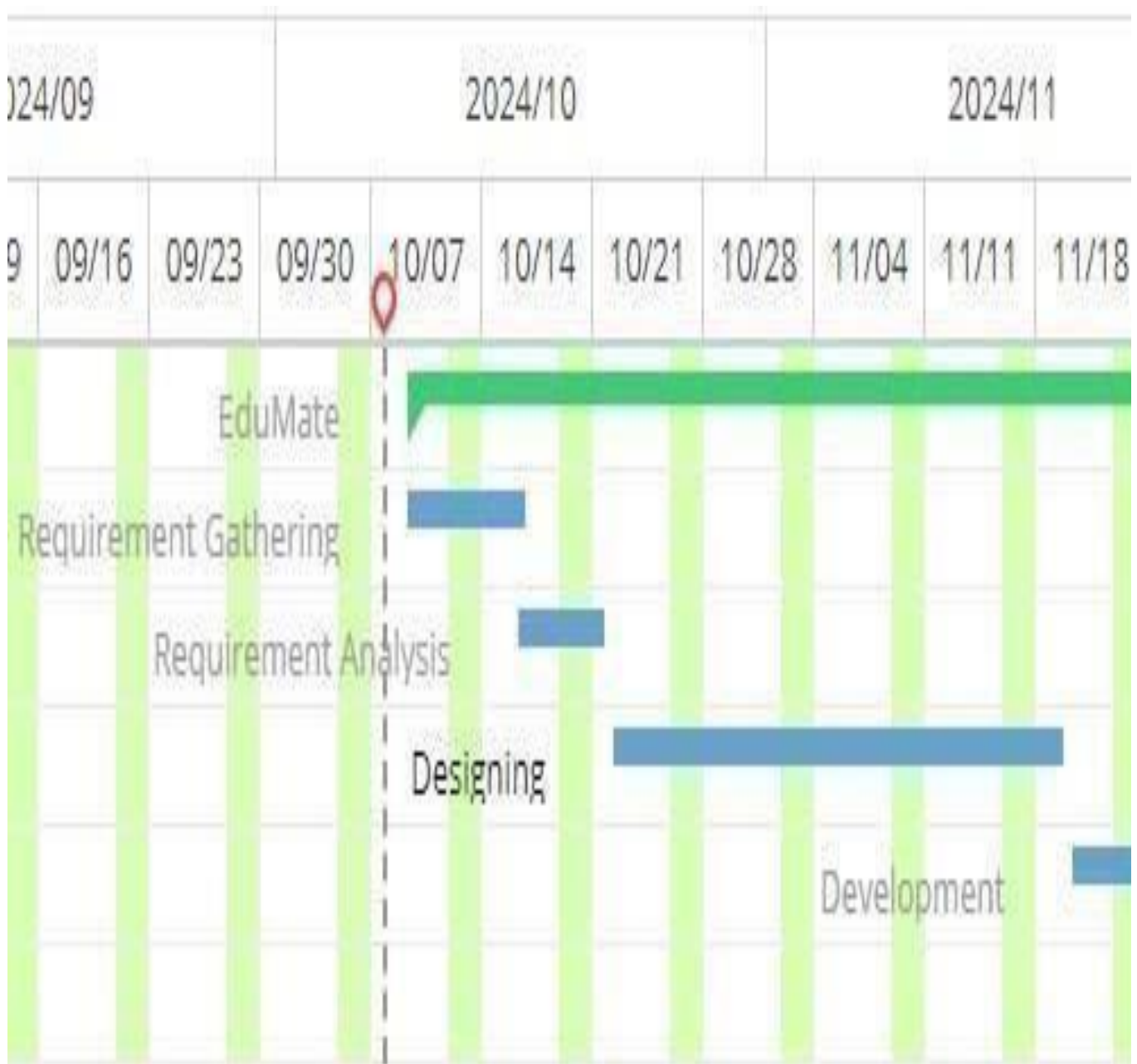


Figure 6 FYP-1 Timeline



Figure 7 FYP-2 Timeline

1.7. Project Work Breakdown

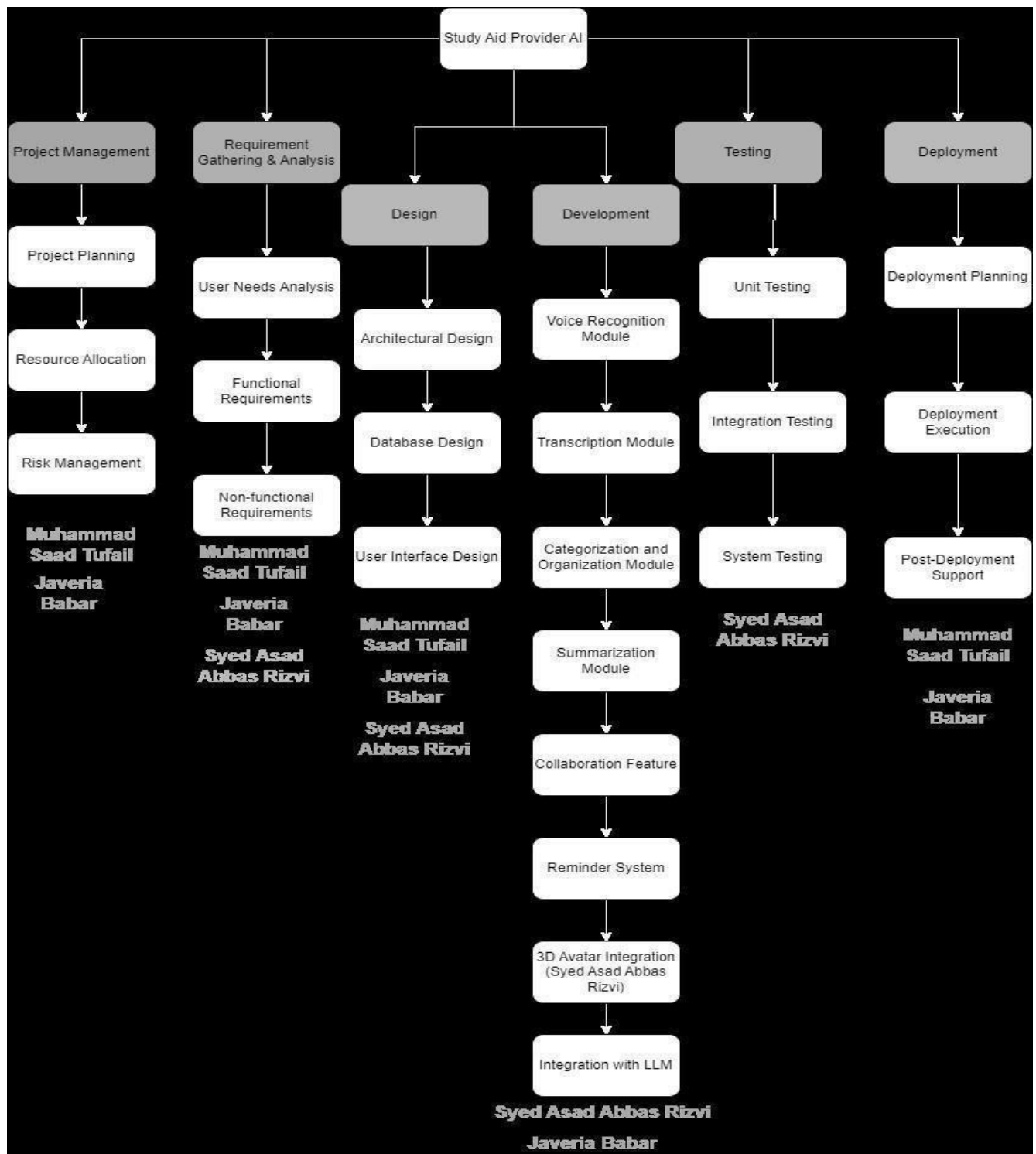


Figure 8 Project Work Breakdown

Chapter 2

2. Requirement Specification and Analysis

Requirement specification involves detailing what a software system must accomplish, covering functionalities, performance criteria, and constraints. This information is documented in a Requirements Specification, including functional and non-functional requirements. Requirement analysis follows, involving discussions, interviews, and validation to ensure clarity, completeness, and feasibility of the specified requirements. Analysts prioritize and categorize requirements, aiming to identify conflicts or gaps. Thorough requirement specification and analysis are critical in establishing a clear foundation for the software development life cycle, reducing the risk of misunderstandings and errors while aligning the system with stakeholder expectations. Effective communication and collaboration play pivotal roles in this process.

2.1. Functional Requirements

Sr. No	Functional Requirements	Type	Status
1	The user shall be able to sign-up for the application.	Core	Draft
2	The system shall be able to validate the email through an OTP.	Core	Draft
3	User shall be able to log-in to the application.	Core	Draft
4	User shall be able to log-out, without altering the personal data.	Core	Draft
5	The user shall be able to manage their profiles.	Core	Draft
6	The system shall allow users to login from multiple devices.	Core	Draft
7	Logging out from one device shall not affect the user's session on other devices unless manually chosen by the user.	Core	Draft
8	The user shall be able to create and customize study goals on the basis of their learning plans based on subject or topics.	Core	Draft
9	The system shall allow users to create study schedule with deadlines for assignments and exams.	Core	Draft
10	The system shall send reminders based on user's learning plans for upcoming assignments, quizzes, and exams.	Core	Draft
11	The system shall provide notifications to remind the users of upcoming reminders set priorly.	Core	Draft
12	The user shall be able to search through transcribed notes.	Core	Draft
13	The user shall be able to download the notes and summaries for offline access and sync updates once back online.	Core	Draft

14	The system shall allow users to edit and update their notes even when offline.	Core	Draft
15	The user shall be able to filter their notes and search results by using advanced filters.	Core	Draft
16	The system shall generate summaries of transcribed notes to provide a concise overview for the user.	Core	Draft
17	The system shall track users' progress.	Core	Draft
18	The system shall provide visual progress bars or graphs to show study completion percentages and pending topics.	Core	Draft
19	The system shall allow users to customize speech-to-text settings.	Core	Draft
20	The system shall automatically detect language of spoken input.	Core	Draft
21	The system shall transcribe audio inputs into text.	Core	Draft
22	The system shall provide tools to create visual learning aids.	Core	Draft
23	Flashcards with spaced repetition shall be automatically generated from the key concepts in the notes to assist with memorization.	Core	Draft
24	The system shall offer note-sharing capabilities via QR code, allowing quick sharing between the users.	Core	Draft
25	The user shall be able to generate customized quizzes.	Core	Draft
26	The system shall provide feedback for quizzes, including explanations for incorrect answers.	Core	Draft
27	The user shall be able to access quizzes generated from their notes to assess their understanding of various topics.	Core	Draft

Table 2 Functional Requirements

2.2. Non-Functional Requirements

Sr. No	Non-Functional Requirements	Category
1	The system shall provide real-time responsiveness for transcribing and organizing notes from audio input.	Performance
2	The application shall ensure all notes and data are synchronized across devices with minimal delay.	Portability
3	The application's interface shall be intuitive and easy to navigate for users of all technical and non-technical backgrounds.	Usability

4	The system shall allow users to access notes offline and synchronize changes when re-connected without losing any data.	Reliability
5	The application shall maintain high transcription accuracy even in noisy environments by using background noise filtering.	Reliability
6	The system shall handle up to 10,000 simultaneous users without degrading in performance or availability.	Scalability
7	The mobile application shall load operate smoothly, even during peak usage hours or with large note databases.	Scalability
8	The application shall ensure secure storage and handling of user's data, including study notes and personal information.	Security
9	The application shall be compatible with multiple devices and platforms, ensuring a seamless experience across all.	Compatibility
10	The application shall allow users to adjust various transcription settings (e.g. language, noise filtering) within seconds.	Usability

Table 3 Non-functional requirements

2.3. System Use Case Modeling

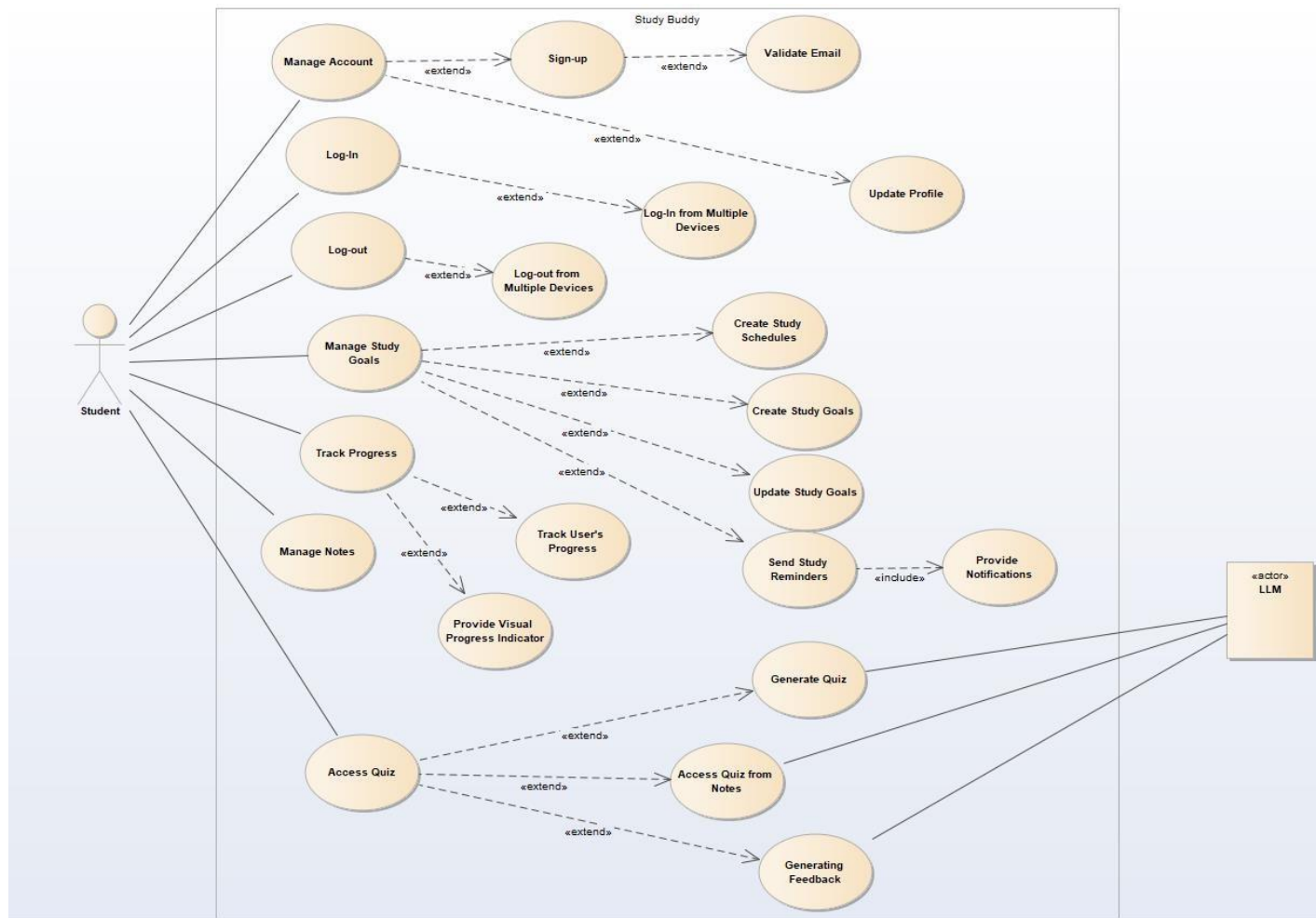


Figure 1 Study Buddy System

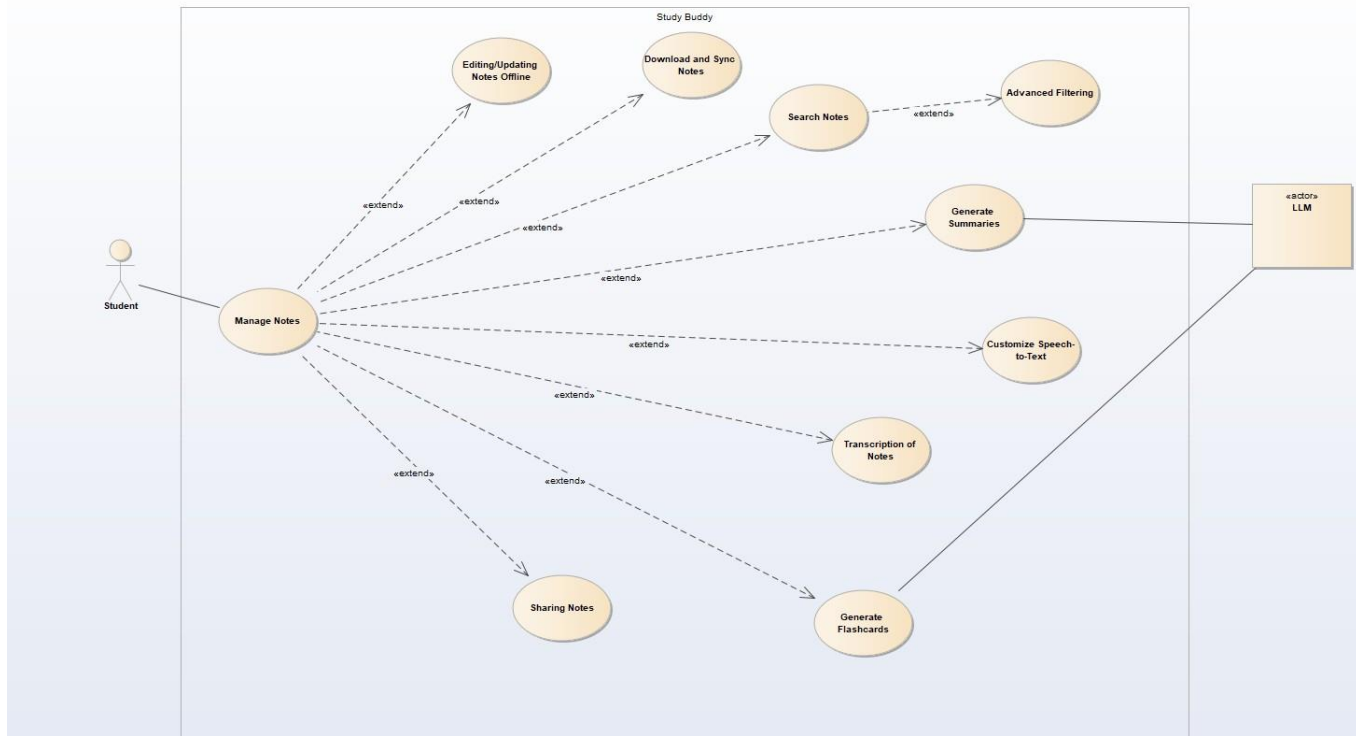


Figure 2 Study Buddy Continued

Table 1 Manage Account

Use Case ID:	UC-01		
Use Case Name:	Manage Account		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to manage and update their account settings, including profile details, password, and account security settings.		
Trigger:	The user accesses the “Account Settings” section in the application.		
Preconditions:	The user is logged into their account		
Post conditions:	The user’s account information is updated successfully.		
Normal Flow:	Actor	System	
	1. The user navigates to “Account Settings”. 3.The user updates desired information (e.g. profile details, password).	2. Validates the information and saves changes. 4.Confirms that the accountinformation has been successfully updated.	
Alternative Flows:	1. If the user tries to change their email address, the system may require re-verification with an OTP.		
Exceptions:	1. If there is an issue with saving the changes due to network problems, the system displays an error message.		

Table 2 Sign-up

Use Case ID:	UC-02		
Use Case Name:	Sign-up		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows a new user to create an account by providing necessary details and confirming their email address.		
Trigger:	The user selects the “Sign-up” option to register for the application.		
Preconditions:	The user must not have an existing account with the provided email address.		
Post conditions:	A new account is created, and the user can proceed to log in.		
Normal Flow:	Actor	System	
	1. The user selects “Sign-up” 2. The user enters personal information (e.g. name, email, password). 4. The user enters the OTP for verification.	3. Sends an OTP to the provided email address. 5.Verifies the OTP and confirms the creation of the account.	
Alternative Flows:	1. If the OTP is in-correct, the system prompts the user to re-enter or request a new OTP.		
Exceptions:	1. If the email is already registered, the system notifies the user. 2. If the password is not strong enough, then the user is prompted to make the password strong. 3. If the user enters the wrong information, then the user is prompted by the system.		

Table 3 Validate Email

Use Case ID:	UC-03		
Use Case Name:	Validate Email		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	System		
Description:	Confirms the user’s email address during the sign-up process by sending an OTP for verification.		
Trigger:	The user initiates the sign-up process and submits their email.		
Preconditions:	The has provided a valid, unregistered email address.		
Post conditions:	The email address is verified, and the user can proceed with registration.		
Normal Flow:	Actor	System	
	2. The user enters the OTP in the application.	1. Sends an OTP to the user’s email address. 3. Verifies the OTP and confirms the email validation.	
Alternative Flows:	If the OTP entry is in-correct, the user is prompted to enter it again or request a new OTP.		
Exceptions:	If the OTP cannot be sent due to network issues, the system displays an error message.		

Table 4 Update Profile

Use Case ID:	UC-04		
Use Case Name:	Update Profile		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to update their profile information, such as name, profile picture, and personal details.		
Trigger:	The user accesses the “Update Profile” option within account settings.		
Preconditions:	The user is logged into their account.		
Post conditions:	The user’s profile information is updated and saved.		
Normal Flow:	Actor	System	
	1. The user navigates to “Update Profile”. 2. The user modifies profile information.	3. Validates new information and saves the updates. 4. Confirms that the profile has been successfully updated.	
Alternative Flows:	1. If the user attempts to change sensitive information, the system may prompt additional verification.		
Exceptions:	1. If the update fails due to server issues, the system displays an error message.		

Table 5 Log-In

Use Case ID:	UC-05		
Use Case Name:	Log-In		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Enables the user to login to their account by entering their credentials.		
Trigger:	The user selects the “Login” option and provides login credentials.		
Preconditions:	The user has an active account and has entered valid credentials.		
Post conditions:	The user is logged into the application		
Normal Flow:	Actor	System	
	1. The user selects “Login” and enters their email and password.	2. Validates the credentials. 3. Initiates a session if the credentials are correct. 4. Confirms that the user is successfully logged in.	
Alternative Flows:	1. If the user forgets their password, they can select “Forgot Password” to reset it.		
Exceptions:	1. If the credentials are in-correct, the system notifies the user and prompts them to try again.		

Table 6 Login from Multiple Devices

Use Case ID:	UC-06		
Use Case Name:	Login from Multiple Devices		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to log in from multiple devices simultaneously without affecting sessions on other devices.		
Trigger:	The user logs in on a new device.		
Preconditions:	The user is already logged into their account on at least one other device.		
Post conditions:	The user is logged into the application on multiple devices simultaneously.		
Normal Flow:	Actor	System	
	1. The user log in on a new device by entering their credentials.	2. Authenticates the user’s credentials. 3. Initiates a new session on the current device while maintaining sessions on other devices. 4. Confirms that the user is successfully logged in on the new device.	
Alternative Flows:	1. If the user chooses to log-out from all devices, all active sessions will be terminates.		
Exceptions:	1. If multiple logins are detected from different locations, the system may prompt the user to confirm their identity.		

Table 7 Log-out

Use Case ID:	UC-07		
Use Case Name:	Log-out		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to securely log out from their account without affecting personal data or active sessions on other devices.		
Trigger:	The user selects the “Log-out” option in the application.		
Preconditions:	The user is currently logged into the application.		
Post conditions:	The user is logged out from the current devices, but sessions on other devices remain active.		
Normal Flow:	Actor	System	
	1. The user selects the “Log-out” option.	2. Ends the session on the current device. 3. Confirms that the user has been successfully logged out from this device.	
Alternative Flows:	1. If the user selects “Log-out from All Devices”, all sessions are terminated.		
Exceptions:	1. If there is a network error during log-out, the system would remain active or logged in.		

Table 8 Logout from Multiple Devices

Use Case ID:	UC-08		
Use Case Name:	Logout from Multiple Devices		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to log out from all the active sessions on multiple devices.		
Trigger:	The user selects “Log-out from All Devices” from the account settings.		
Preconditions:	The user is logged into the application on at least two devices.		
Post conditions:	All active sessions on all devices are terminated		
Normal Flow:	Actor	System	
	1. The user selects “Log out from All Devices”	2. Ends all active sessions across all devices. 3. Confirms that the user has been successfully logged out from all devices.	
Alternative Flows:	1. If the user only wishes to log out from specific device, they can view and select individual sessions to terminate		
Exceptions:	1. If there is a network issue, the system may not immediately log out from all devices and will display a warning message.		

Table 9 Manage Study Goals

Use Case ID:	UC-09		
Use Case Name:	Manage Study Goals		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Enables the user to create, customize, and track study goals based on their learning plans and specific subjects or topics.		
Trigger:	The user accesses the “Study Goals” section in the application.		
Preconditions:	The user is logged into their account and has access to study goal management features.		
Post conditions:	The user’s study goals are saved, updated, or deleted as requested.		
Normal Flow:	Actor	System	
	1. The user navigates to “Manage Study Goals”. 2. The user creates new study goals or updates existing ones.	3. Validates the information provided. 4. Saves the new or updated goals. 5. Confirms that the goals have been successfully saved or updated.	
Alternative Flows:	1. If the user wants to delete a study goal, they can select the goal and choose “Delete”		
Exceptions:			

Table 10 Create Study Schedules

Use Case ID:	UC-10		
Use Case Name:	Create Study Schedules		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Enables the user to create study schedules with deadlines for assignments, quizzes, and exams to manage time and study effectively.		
Trigger:	The user accesses the “Create Study Schedule” option within the application.		
Preconditions:	The user is logged into the application.		
Post conditions:	A study schedule with deadlines is created an saved in the system		
Normal Flow:	Actor	System	
	1. The user navigates to “Create Study Schedule”. 2. The user enters details, such as subjects, topics, and deadlines.	3. Validates the information provided. 4. Saves the study schedule and deadlines. 5. Confirms that the study schedules has been successfully created.	
Alternative Flows:	1. If the user wants to edit a schedule, they can navigate to the schedule and select “Edit”.		
Exceptions:			

Table 11 Create Study Goals

Use Case ID:	UC-11		
Use Case Name:	Create Study Goals		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to create specific study goals based on their learning objectives, focusing on particular subjects or topics.		
Trigger:	The user selects the option to create study goals.		
Preconditions:	The user is logged in and has access to goal-setting features.		
Post conditions:	A new study goal is created and saved in the system.		
Normal Flow:	Actor		System
	<ol style="list-style-type: none"> 1. The user navigates to “Create Study Goals”. 2. The user specifies the study goals, including subjects and desired outcomes. 		<ol style="list-style-type: none"> 3. Validates the provided details. 4. Saves the study goals. 5. Confirms that the study goal has been successfully created.
Alternative Flows:	<ol style="list-style-type: none"> 1. If the user wants to modify a goal, they can access the goal and choose “Edit”. 		
Exceptions:	<ol style="list-style-type: none"> 1. If the provided information is invalid then the system prompts the user to provide the correct information. 		

Table 12 Update Study Goals

Use Case ID:	UC-12		
Use Case Name:	Update Study Goals		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to update existing study goals based on progress or changes in learning objectives.		
Trigger:	The user selects an existing goal to update.		
Preconditions:	The user is logged in and has created at least one study goal.		
Post conditions:	The study goal is updated and saved with the new details.		
Normal Flow:	Actor		System
	1. The user navigates to an existing study goal. 2. The user modifies the goal as desired.		3. Validates the updated information. 4. Saves the changes to the study goal. 5. Confirms that the study goal has been successfully updated.
Alternative Flows:	1. If the user decides to delete the goal, they can select “Delete to remove it.		
Exceptions:			

Table 13 Send Study Reminders

Use Case ID:	UC-13		
Use Case Name:	Send Study Reminders		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	System		
Description:	The system sends reminders to the user based on their study schedule and upcoming assignments, quizzes, or exams.		
Trigger:	The system detects that a scheduled study event is approaching based on the user’s study schedule.		
Preconditions:	The user has created a study schedule with deadlines, and notifications are enabled.		
Post conditions:	The user receives a reminder notification for an upcoming study event.		
Normal Flow:	Actor	System	
		<div>1. Checks the study schedule for upcoming deadlines.</div> <div>2. Generates a reminder notification.</div> <div>3. Sends the notification to the user</div>	
Alternative Flows:	If the notifications are disabled, the system skips sending reminders.		
Exceptions:			

Table 14 Provide Notifications

Use Case ID:	UC-14		
Use Case Name:	Provide Notifications		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	System		
Description:	The system provides notifications to the user regarding reminders, deadlines, and updates in the application.		
Trigger:	A scheduled reminder, update, or system message is available for the user.		
Preconditions:	The user has enabled notifications within the application.		
Post conditions:	The user receive a relevant notification.		
Normal Flow:	Actor	System	
		<div>1. Checks for any reminders, deadlines, or important updates.</div> <div>2. Generates notification.</div> <div>3. Sends the notification to the user.</div>	
Alternative Flows:	1. If the user has opted out of specific notifications, the system filters out those messages.		
Exceptions:			

Table 15 Track Progress

Use Case ID:	UC-15		
Use Case Name:	Track Progress		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student, System		
Description:	Tracks the user’s study progress by logging completed study goals, schedules, and updates on study materials.		
Trigger:	The user marks tasks as complete, or the system periodically updates progress based on tracked data.		
Preconditions:	The user has set study goals and schedules.		
Post conditions:	The user is updated and displayed to the user.		
Normal Flow:	Actor	System	
	1. Marks tasks or goals as completed.	2. Logs the completion status. 3. Updates the completion status. 4. Displays the updated progress to the user.	
Alternative Flows:	1. The user can view progress without marking tasks as complete.		
Exceptions:	1. If there is a system error while tracking, progress may not update, and an error message is shown.		

Table 16 Track User's Progress

Use Case ID:	UC-16		
Use Case Name:	Track User’s Progress		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student, System		
Description:	This use case allows the system to track the user’s progress by logging completed study goals, schedules, and updates on study materials. It provides an overview of completed tasks and remaining tasks, assisting the user in monitoring their progress.		
Trigger:	The user marks tasks, goals, or schedule items as completed, or the system automatically updates progress based on tracked data.		
Preconditions:	The user has set study goals or schedules, and progress tracking is enabled in the system.		
Post conditions:	The user’s progress is updated in the system, and the latest progress overview is available for the user to view.		
Normal Flow:	Actor	System	
	1. Marks tasks, goals, or schedules items as completed in the application.	2. Logs the completion status of each marked item. 3. Updates the user’s progress data by calculating the percentage of completed and pending tasks. 4. Displays the updated progress overview to the user.	
Alternative Flows:	1. The user may choose to view progress for specific subjects or goals.		

	2. The system may be able to filter the progress overview based on selected subjects or goals and displays it to the user.
Exceptions:	1. IF the system encounters an error while updating the progress, the update may not be complete, and an error message is displayed.

Table 17 Provide Visual Progress Indicator

Use Case ID:	UC-17		
Use Case Name:	Provide Visual Progress Indicator		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	System		
Description:	Displays visual indicators, such as progress bars or graphs, to show the user's study progress, completed goals, and pending topics.		
Trigger:	The user navigates to the progress section, or the system periodically updates the visual indicator.		
Preconditions:	The user has completed some study goals or schedules.		
Post conditions:	The visual progress indicator is updated and shown to the user.		
Normal Flow:	Actor	System	
		<ol style="list-style-type: none"> 1. Calculates the user's progress based on completed goals and schedules. 2. Updates the visual progress indicator 3. Displays the updated indicator to the user. 	

Alternative Flows:	1. The user may choose to view progress by specific subjects or goals.
Exceptions:	1. If the progress data cannot be retrieved due to a system error, an error message is displayed instead of a progress indicator.

Table 18 Access Quiz

Use Case ID:	UC-18		
Use Case Name:	Access Quiz		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to access quizzes that have been created, enabling them to test their understanding of various topics and study materials.		
Trigger:	The user selects the “Access Quiz” option from the menu in the application.		
Preconditions:	The user has at least one quiz available in the system.		
Post conditions:	The user can begin taking the quiz		
Normal Flow:	Actor	System	
	1. Navigates to the “Access Quiz” section in the application. 2. Selects a quiz to begin.	3. Loads the selected quiz and present the first question to the user.	
Alternative Flows:	1. If the user has no available quizzes, the system prompts them to generate one.		
Exceptions:			

Table 19 Generate Quiz

Use Case ID:	UC-19		
Use Case Name:	Generate Quiz		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student, System		
Description:	The system allows the user to generate a quiz based on their notes or study materials. The user can customize the quiz based on specific topics, difficulty level, and question types.		
Trigger:	The user selects the “Generate Quiz” option from the application’s menu.		
Preconditions:	The user has study notes or material available in the system.		
Post conditions:	The quiz is generated based on the selected criteria and made available to the user.		
Normal Flow:	Actor		System
	<ol style="list-style-type: none"> 1. Navigates to the “Generate Quiz” section. 2. Selects the topics or materials to include in the quiz. 3. Choose the quiz customization options. 		<ol style="list-style-type: none"> 4. Generates the quiz based on the user’s selections. 5. Saves the quiz and present it to the user for access.
Alternative Flows:	<ol style="list-style-type: none"> 1. If the user selects no topics or material, the system prompts them to choose at least one. 		
Exceptions:	<ol style="list-style-type: none"> 1. If the system is unable to generate the quiz due to in-efficient content or system error, the user is notified. 		

Table 20 Access Quiz from Notes

Use Case ID:	UC-20		
Use Case Name:	Access Quiz from Notes		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	The system allows the user to access a quiz directly from their notes, enabling them to generate and take quizzes based on specific notes or specific material.		
Trigger:	The user selects a note or study material and chooses the option to generate a quiz.		
Preconditions:	The user has notes or study material available in the system.		
Post conditions:	The user is able to access a quiz generated from their notes for self-assessment.		
Normal Flow:	Actor		System
	1. Navigates to a note or study material. 2. Selects the option to generate or access a quiz from the material.		3. Create a quiz based on the selected material or notes. 4. Present the quiz to the user for taking.
Alternative Flows:	1. If the selected notes are insufficient for generating a quiz, the system prompts the user to add more content.		
Exceptions:	1. If the quiz cannot be generated due to system error or insufficient content, the user is notified.		

Table 21 Generate Feedback

Use Case ID:	UC-21		
Use Case Name:	Generate Feedback		
Created By:	Muhammad Saad Tufail	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	System		
Description:	After the user completes a quiz, the system generates feedback, including explanations for incorrect answer and insights into the user’s strengths and areas for improvement.		
Trigger:	The user completes a quiz.		
Preconditions:	The user has completed a quiz, and answer data is available.		
Post conditions:	The user receives feedback on their quiz performance.		
Normal Flow:	Actor	System	
		<div>1. Evaluates the user’s answers and compares them with the correct answers.</div> <div>2. Generates feedback for each question, highlighting incorrect responses and providing explanations.</div> <div>3. Presents the feedback to the user, including suggestions for improvement.</div>	
Alternative Flows:	1. If the user requests more detailed feedback, the system provides additional insights or resources.		
Exceptions:	1. If there is an error in processing the quiz results, the system notifies the user and prompts a retry.		

Table 22 Manage Notes

Use Case ID:	UC-22		
Use Case Name:	Manage Notes		
Created By:	Muhammad Saad Tufail	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to manage their notes, including creating, updating, deleting, and organizing notes within the application.		
Trigger:	The user selects the “Manage Notes” option from the menu.		
Preconditions:	The user has notes already created in the system.		
Post conditions:	The user’s notes are successfully updated or organized as per their actions.		
Normal Flow:	Actor	System	
	1. Selects the “Manage Notes” option from the application 2. Choose an action (create, update, delete, or organize) for their notes.	3. Saves the user’s changes and displays the updated list of notes.	
Alternative Flows:	1. If the user wants to categorize notes by subject or date, the system allows filtering and sorting options.		
Exceptions:	1. If there is an issue saving changes, the system prompts user to retry.		

Table 23 Edit Notes Offline

Use Case ID:	UC-23		
Use Case Name:	Editing Notes Offline		
Created By:	Muhammad Saad Tufail	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to edit their notes even when they are offline, with automatic syncing once an internet connection is restored		
Trigger:	The user opens a note while offline and begins editing it.		
Preconditions:	The user opens has the notes stored locally on their device.		
Post conditions:	The edited notes are saved locally and synced with the cloud once the connection is restored.		
Normal Flow:	Actor	System	
	<div>1. Opens a note for editing while offline.</div> <div>2. Make changes to the note and save them locally.</div>	<div>3. Stores the change locally until the device is back online.</div> <div>4. Synchronizes the changes to the cloud when a connection is re-established.</div>	
Alternative Flows:	1. If the user wishes to manually sync the notes, they can trigger a sync action once the connection is restored.		
Exceptions:	1. If the system encounters an error while syncing, it alerts the user and prompts them to try again later.		

Table 24 Download and Sync Notes

Use Case ID:	UC-24		
Use Case Name:	Download and Sync Notes		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to download their notes for offline access and sync them with the cloud when they are back online, ensuring data consistency across devices.		
Trigger:	The user selects the option to download or sync notes.		
Preconditions:	The user has notes available in the system, and their device has internet connection for syncing.		
Post conditions:	The notes are downloaded for offline use and synchronized with the cloud once the device is connected to the internet.		
Normal Flow:	Actor		System
	<ol style="list-style-type: none"> 1. Selects the “Download Notes” option to save notes for offline use. 2. Once back online, selects the “Sync Notes” option to sync the local changes to the cloud. 		<ol style="list-style-type: none"> 3. Download the selected notes to the device. 4. Syncs any locally saved changes with the cloud server.
Alternative Flows:	<ol style="list-style-type: none"> 1. If the device is already online, the system automatically syncs the notes without requiring the user to select the sync option. 2. If the user’s device loses internet connection during syncing, the system retries syncing once the connection is restored. 		
Exceptions:			

Table 25 Search Notes

Use Case ID:	UC—25		
Use Case Name:	Search Notes		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to search for specific notes by keywords, subjects or other relevant parameter within the application to quickly locate desired content.		
Trigger:	The user enters a search query into the search bar.		
Preconditions:	The user has notes saved within the system.		
Post conditions:	The system displays a list of notes matching the search criteria.		
Normal Flow:	Actor	System	
	1. Types a keyword or query into the search bar.	2. Searches the notes for the matching content. 3. Displays a list of notes or sections matching the search query.	
Alternative Flows:	1. The user may filter the result based on specific criteria such as date or topic.		
Exceptions:	1. If no notes match the search query, the system notifies the user that no results were found.		

Table 26 Advanced Filtering

Use Case ID:	UC-26		
Use Case Name:	Advanced Filtering		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to apply advanced filters to narrow down search results or view notes based on various criteria such as tags, or dates.		
Trigger:	The user selects advanced filtering options from the search interface.		
Preconditions:	The user has notes with metadata such as tags, or dates.		
Post conditions:	The system displays notes that match the selected filtering criteria.		
Normal Flow:	Actor	System	
	1. Clicks on the “Advanced Filters” option within the search menu. 2. Selects various filters options such as date range, subject.	3. Filter the notes according to the user’s selected criteria. 4. Displays the filtered list of notes to the user.	
Alternative Flows:	1. The user may combine multiple filter to refine their search results further.		
Exceptions:	1. If the selected filter criteria return no results, the system notifies the user that no notes match the selected filters.		

Table 27 Generate Summaries

Use Case ID:	UC-27		
Use Case Name:	Generate Summaries		
Created By:	Muhammad Saad Tufail	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student, LLM		
Description:	Allows the user to generate the summary of their notes saved to quickly review key points of the notes.		
Trigger:	The user initiate the “Generate Summary” option		
Preconditions:	The user has existing notes saved in the system.		
Post conditions:	The summarized version of the selected notes is available for the users.		
Normal Flow:	Actor	System	
	1. The user selects the notes to summarize.	2. The system sends the notes to the LLM for processing. 3. The LLM generate a summary of the notes. 4. The system displays the summary to the users.	
Alternative Flows:	1. If the LLM fails to generate a summary, the system would notify the user and suggest trying again later.		
Exceptions:	1. Network error occurs during summary generation. The system prompts the student to check their connection and try again.		

Table 28 Customize Speech-to-Text

Use Case ID:	UC-28		
Use Case Name:	Customize Speech-to-Text		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to customize the settings of the speech-to-text functions, such as language, punctuation preferences, or specific terms.		
Trigger:	The user opens the speech-to-text customization settings.		
Preconditions:	The user has enabled the voice input feature in the app.		
Post conditions:	Customized settings are saved and applied to future transcriptions.		
Normal Flow:	Actor		System
	1. The user accesses customization settings. 2. The user adjusts settings as desired.		3. The system saves the customization settings. 4. The system applies these settings in future transcriptions.
Alternative Flows:	1. If the settings fail to save, the system prompts the user to try again.		
Exceptions:	1. If a setting is not supported, the system shows an error message and uses the default options.		

Table 29 Transcribe Notes

Use Case ID:	UC-29		
Use Case Name:	Transcribe Notes		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	This usecase involves converting spoken notes into text using the speech-to-text options, which allows the user to capture notes verbally.		
Trigger:	The user starts the transcription feature.		
Preconditions:	The user has enabled voice input and configured speech-to-text settings if needed.		
Post conditions:	The spoken notes are converted to text and saved in the notes section.		
Normal Flow:	Actor	System	
	1. The user initiates the transcription feature and starts collecting notes.	2. The system captures the audio inputs and processes it. 3. The system transcribes the audio to text. 4. The system saves the transcribed notes.	
Alternative Flows:	1. If the audio is unclear, the system prompts the user to speak more clearly.		
Exceptions:	1. Network issues disrupt transcription; the system alerts the user to check connectivity.		

Table 30 Generate Flashcards

Use Case ID:	UC-30		
Use Case Name:	Generate Flashcards		
Created By:	Syed Asad Abbas Rizvi	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student, LLM		
Description:	Automatically generates flashcards from existing notes to help the users with spaced repetition learning.		
Trigger:	The user selects notes for flashcard generation.		
Preconditions:	The user has saved notes in the system.		
Post conditions:	Flashcards are generated and saved for the system.		
Normal Flow:	Actor	System	
	1. The user selects notes for flashcard creation.	2. The system sends notes to the LLM. 3. The LLM identifies key points and organizes them into flashcard format. 4. The system saves the flashcards for future review.	
Alternative Flows:	1. If the LLM cannot identify key points, the system notifies the user and suggests manual inputs.		
Exceptions:	1. If flashcard generation fails due to system issue, the system displays an error message and logs the incident.		

Table 31 Share Notes

Use Case ID:	UC-31		
Use Case Name:	Share Notes		
Created By:	Muhammad Saad Tufail	Last Updated By:	Muhammad Saad Tufail
Date Created:	12/11/2024	Last Revision Date:	
Actors:	Student		
Description:	Allows the user to share selected notes with peers for collaborative study or review.		
Trigger:	The user selects the “Share” option for specific notes.		
Preconditions:	The user has notes saved in the system and sharing permission are active.		
Post conditions:	Selected notes are successfully shared with the specific peers.		
Normal Flow:	Actor	System	
	<div>1. The user selects the notes to share.</div> <div>2. The user provides the QR-code to the recipient for giving access.</div> <div>3. The recipient scans the code to gain access.</div>	<div>4. The system notifies recipient of the shared notes.</div>	
Alternative Flows:	1. If the selected recipient is not in the system, the user is prompted to add them first.		
Exceptions:	1. Note Sharing fails due to permission issues, and the system alerts the student to review their settings.		

2.4. System Sequence Diagram



Figure 1 Manage Account SSD



Figure 2 Sign-up SSD



Figure 3 Validate Email SSD



Figure 4 Update Profile SSD

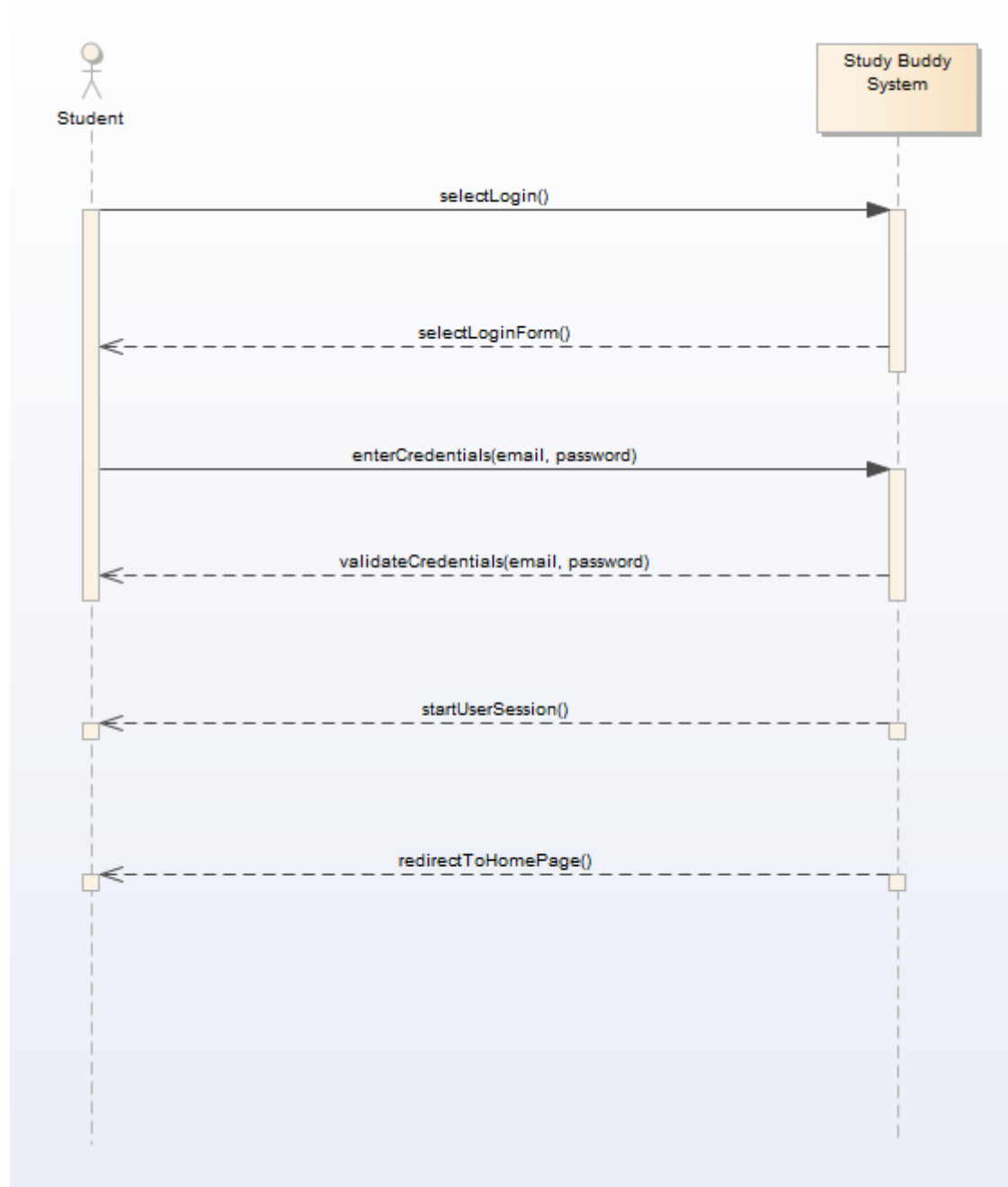


Figure 5 Login SSD

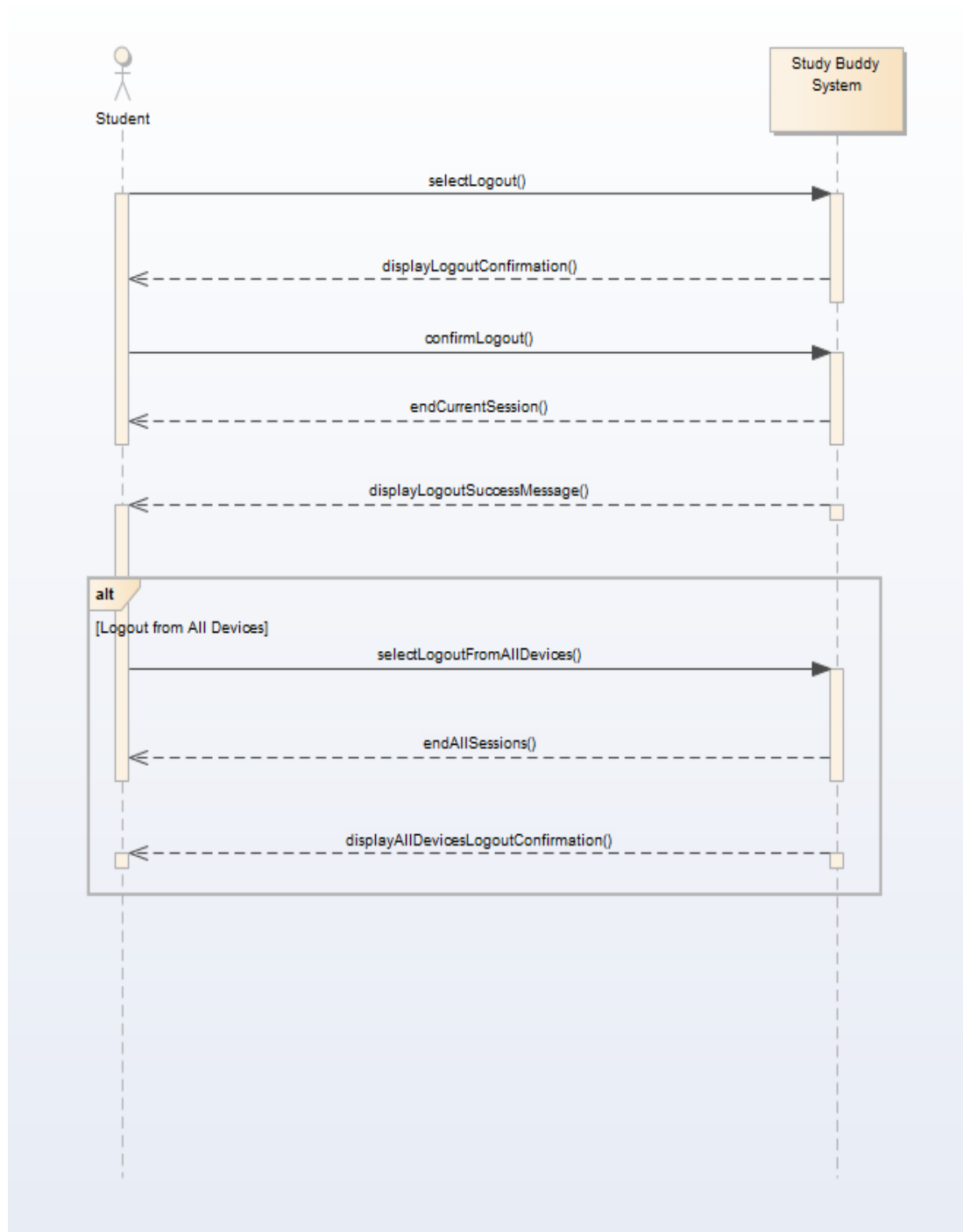


Figure 6 Log-out SSD

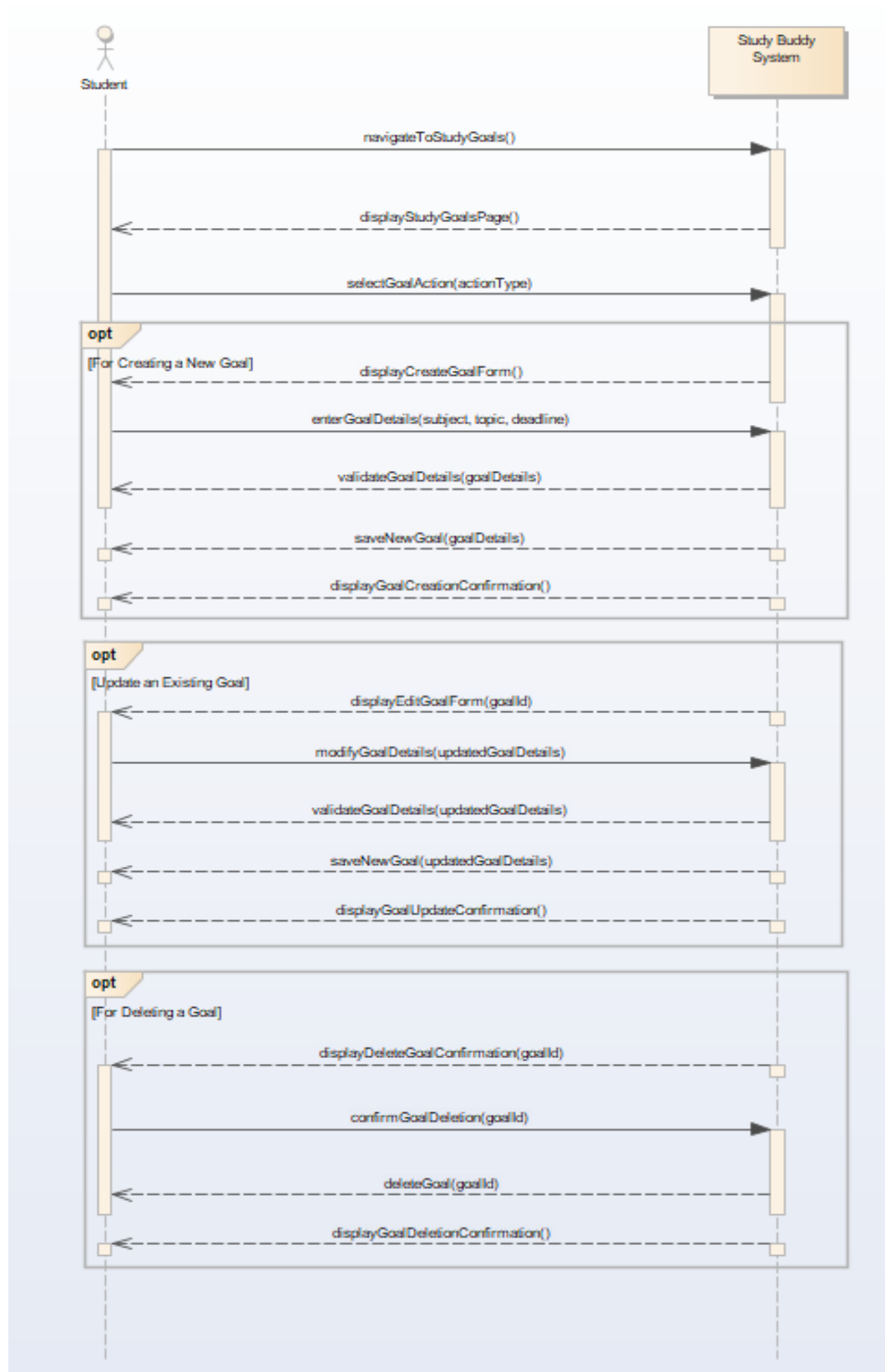


Figure 7 Manage Study Goals SSD



Figure 8 Create Study Schedules SSD

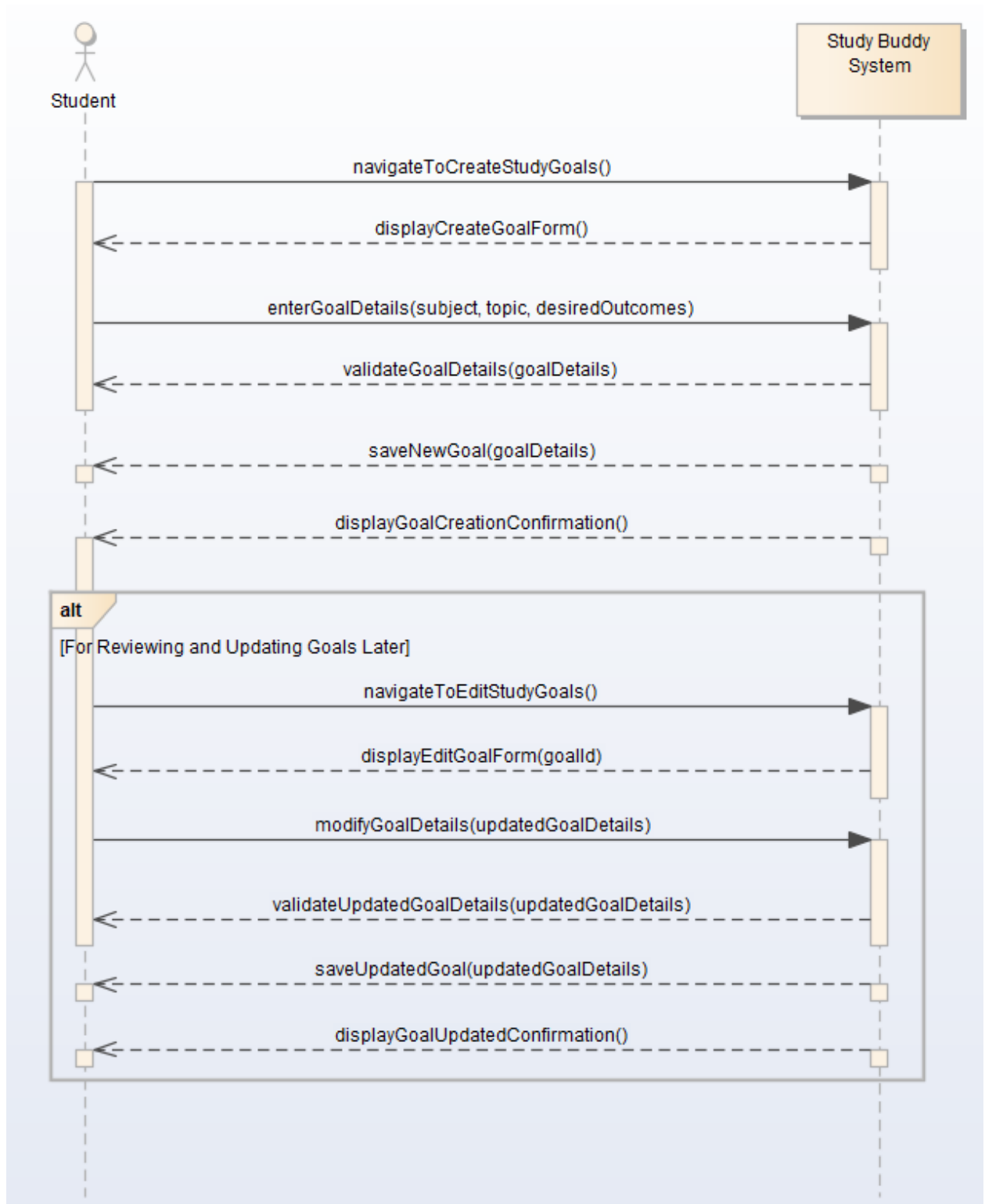


Figure 9 Create Study Goals SSD

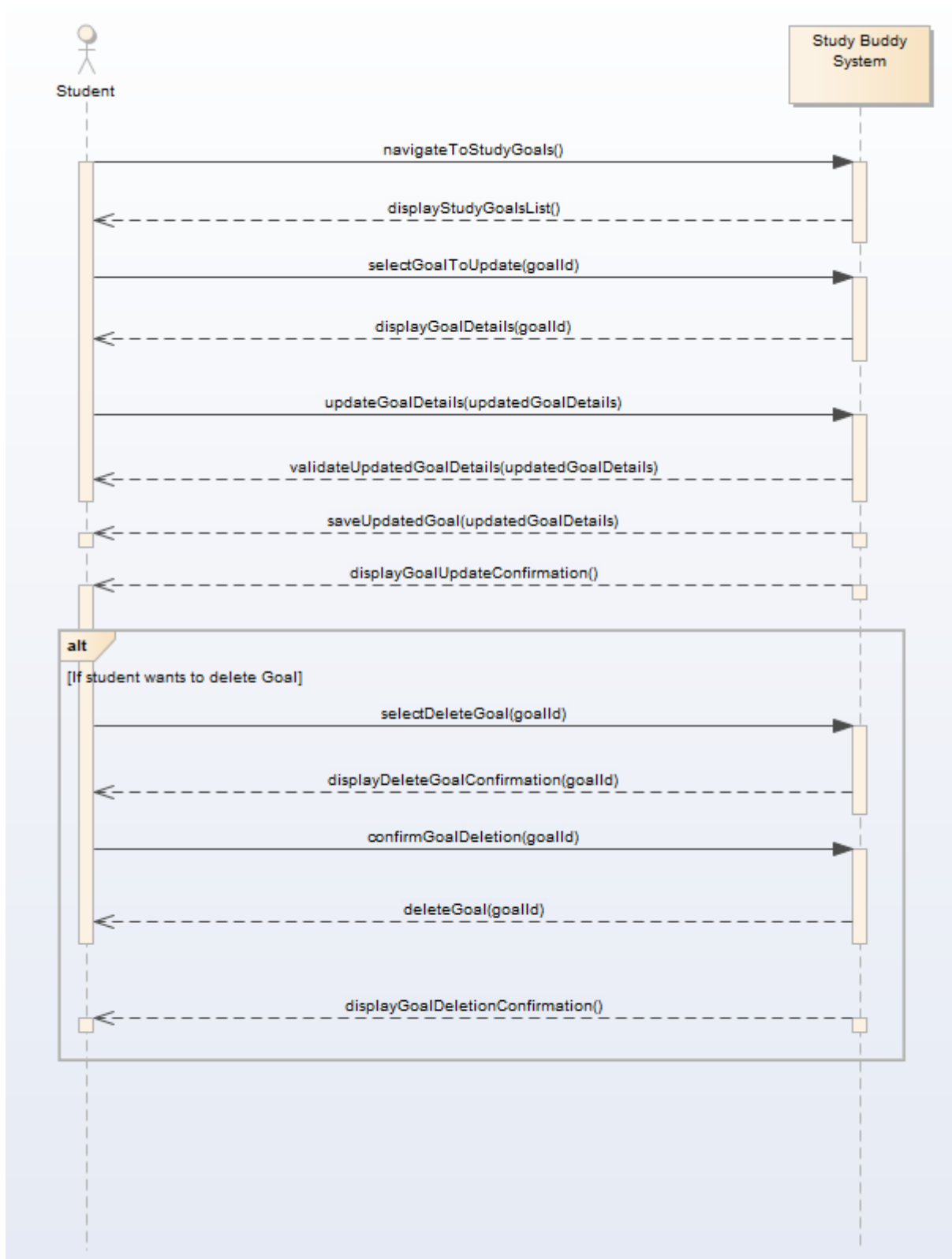


Figure 10 Update Study Goals SSD

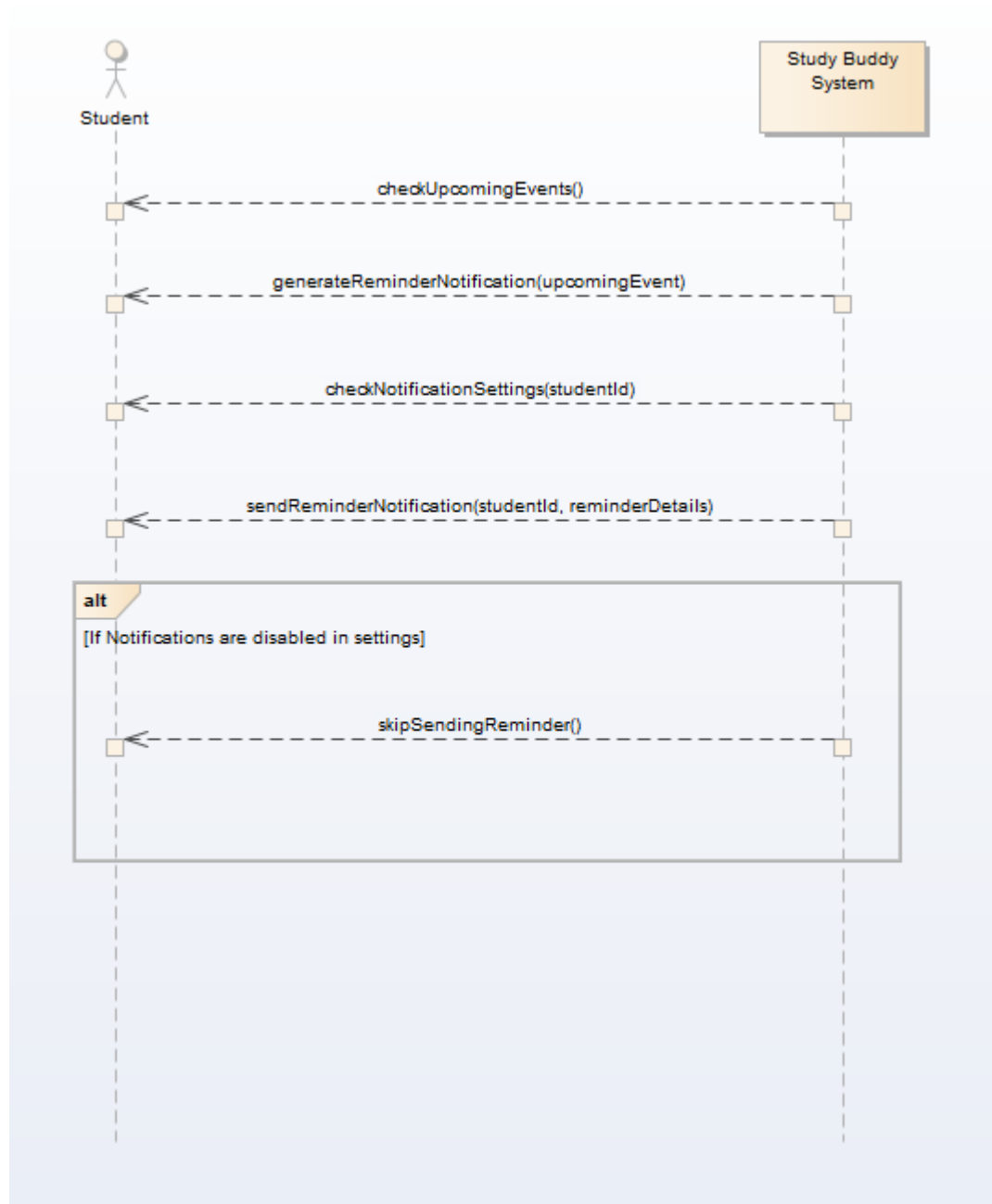


Figure 11 Send Study Reminders SSD



Figure 12 Provide Notifications SSD

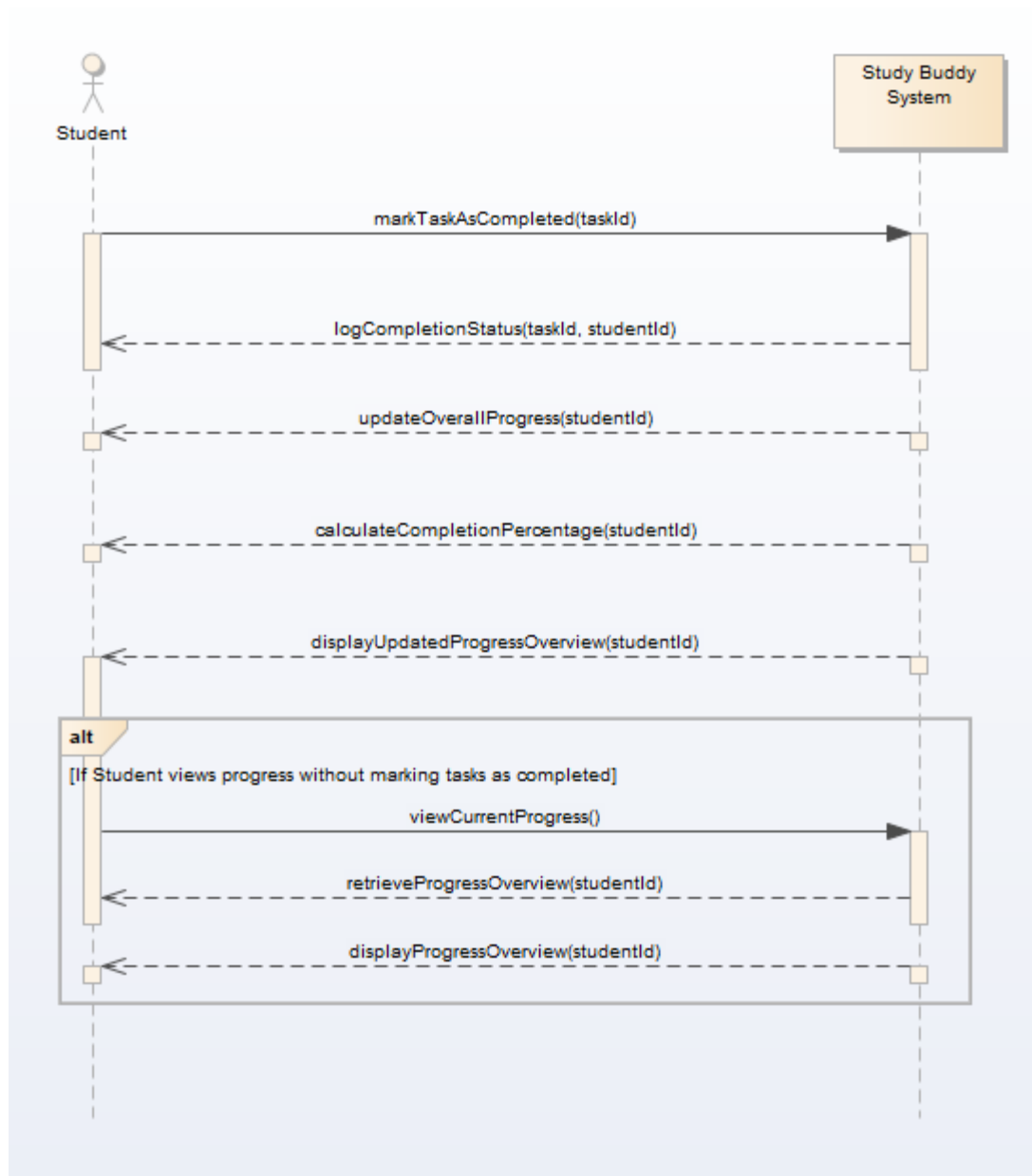


Figure 13 Track Progress SSD

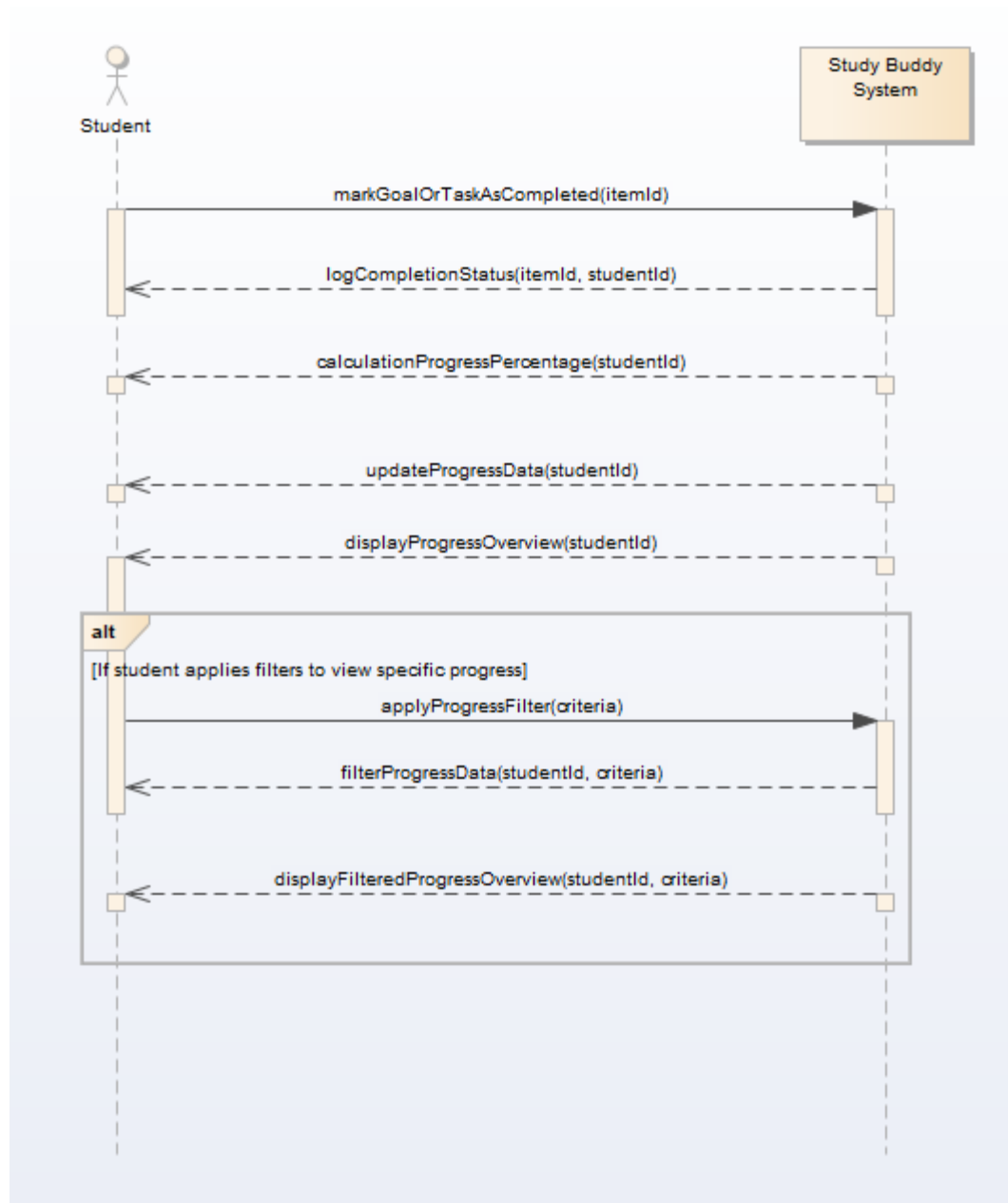


Figure 14 Track User's Progress SSD

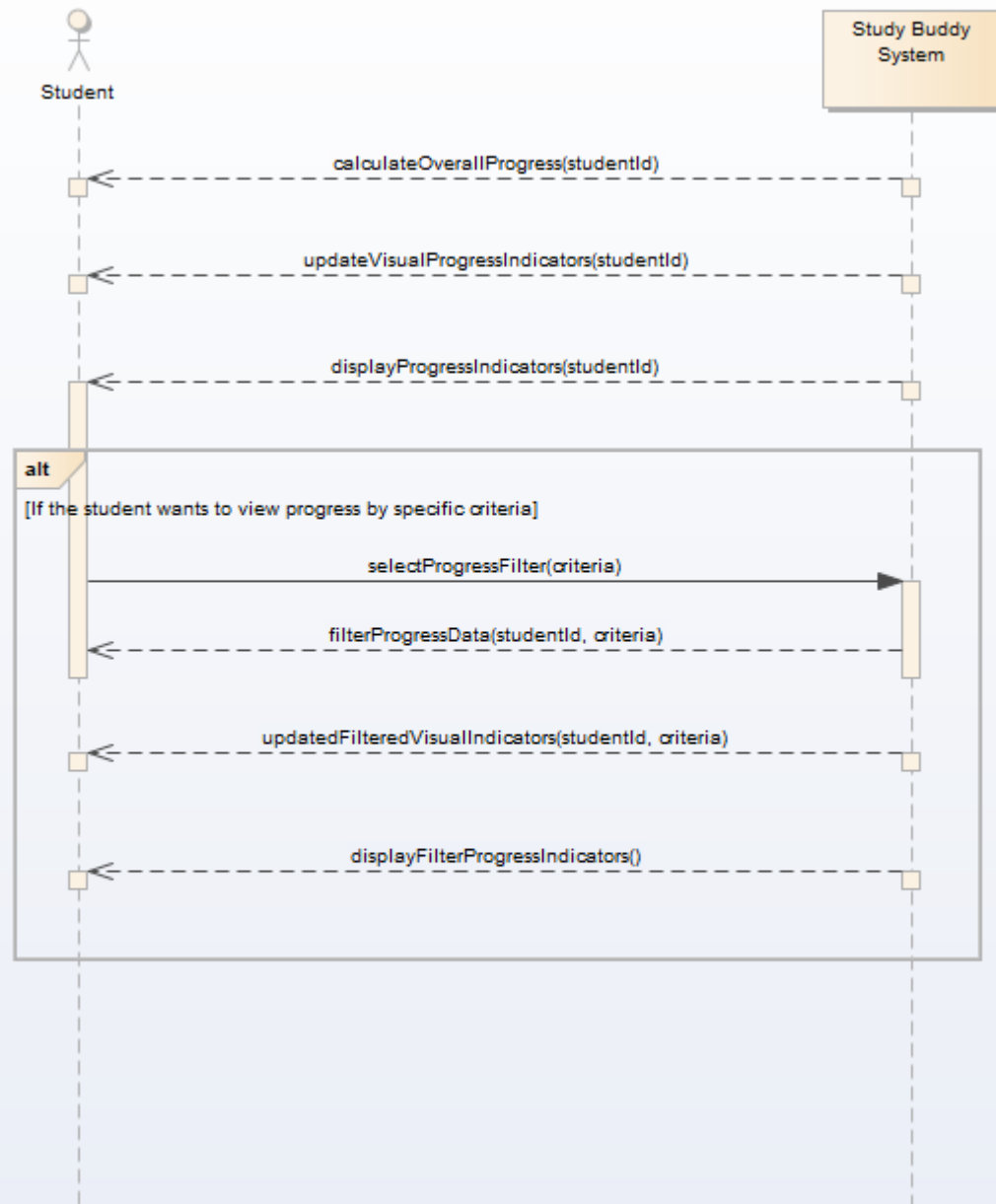


Figure 15 Provide Visual Progress Indicator SSD

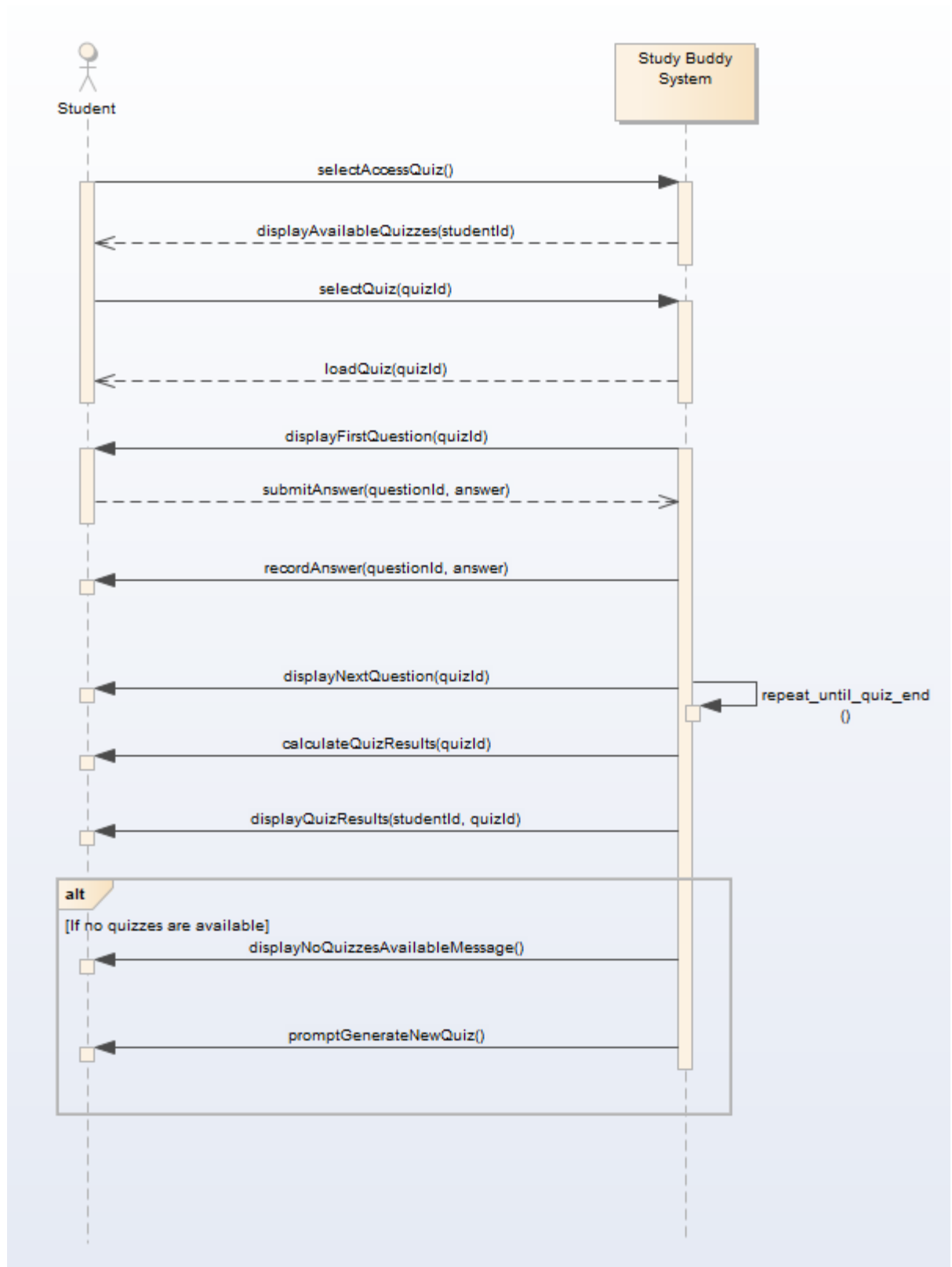


Figure 16 Access Quiz SSD



Figure 17 Generate Quiz SSD



Figure 18 Access Quiz from Notes SSD



Figure 19 Generate Feedback SSD

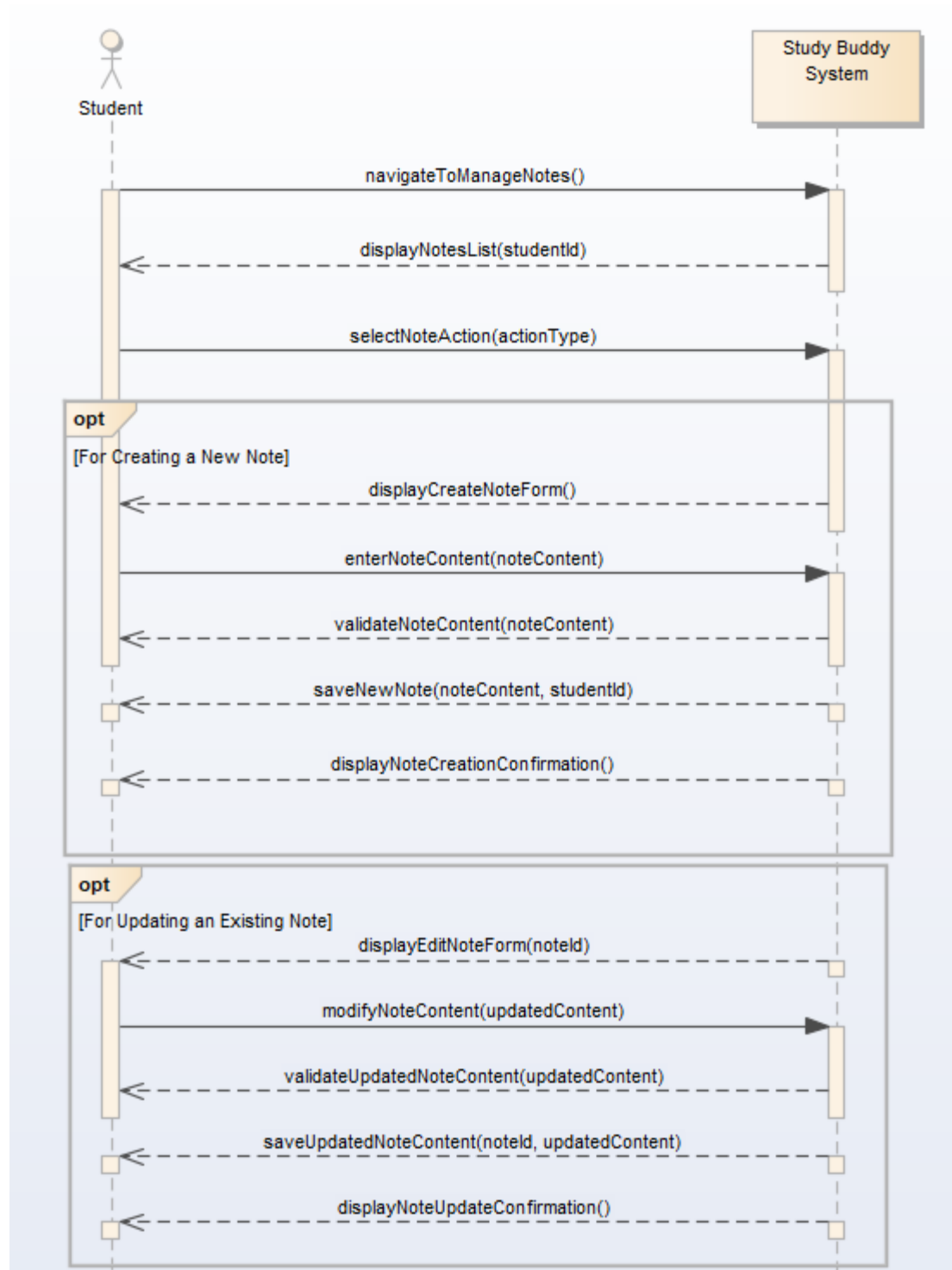


Figure 20 Manage Notes SSD



Figure 21 Manage Notes (continue) SSD

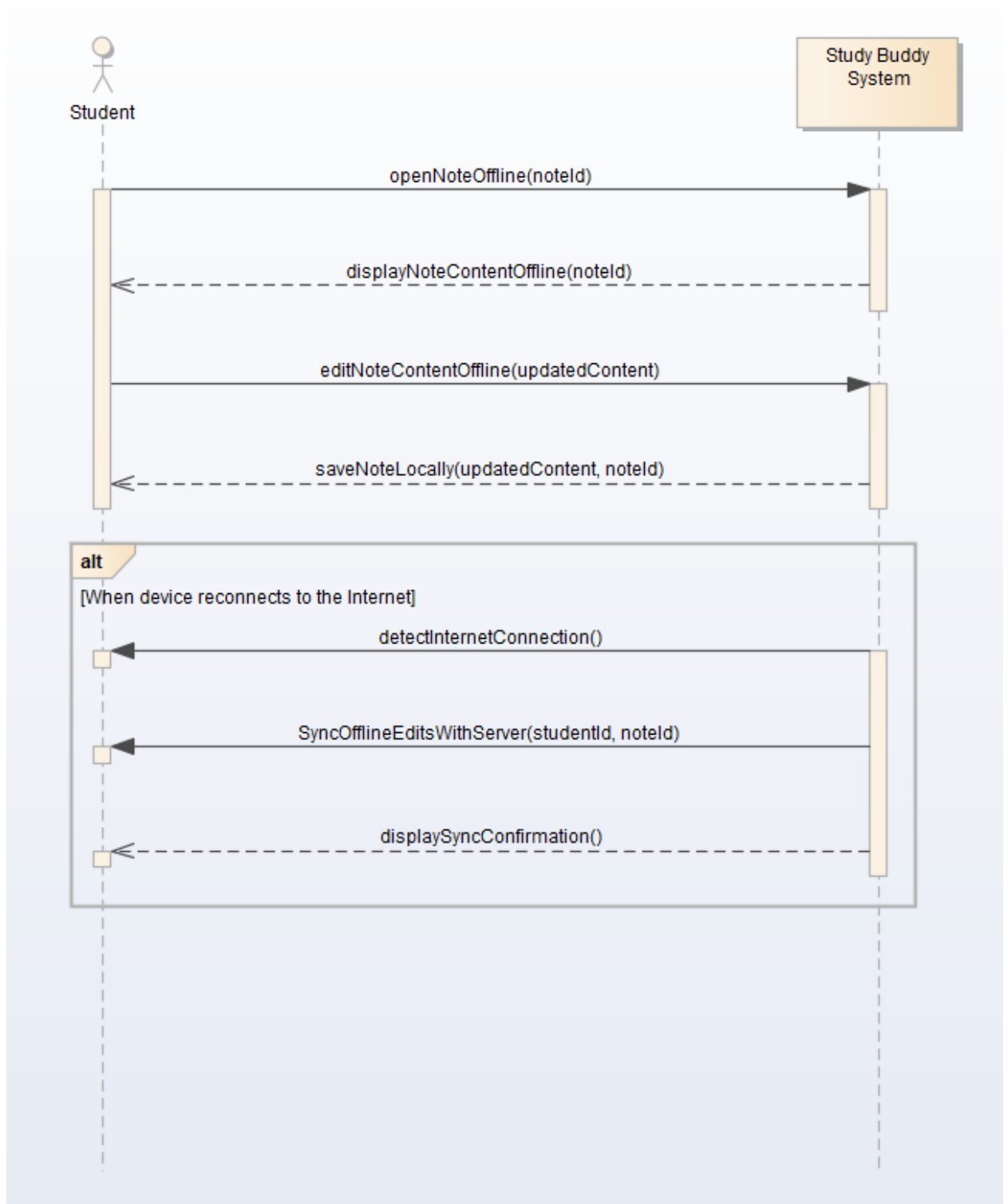


Figure 22 Editing Notes Offline SSD

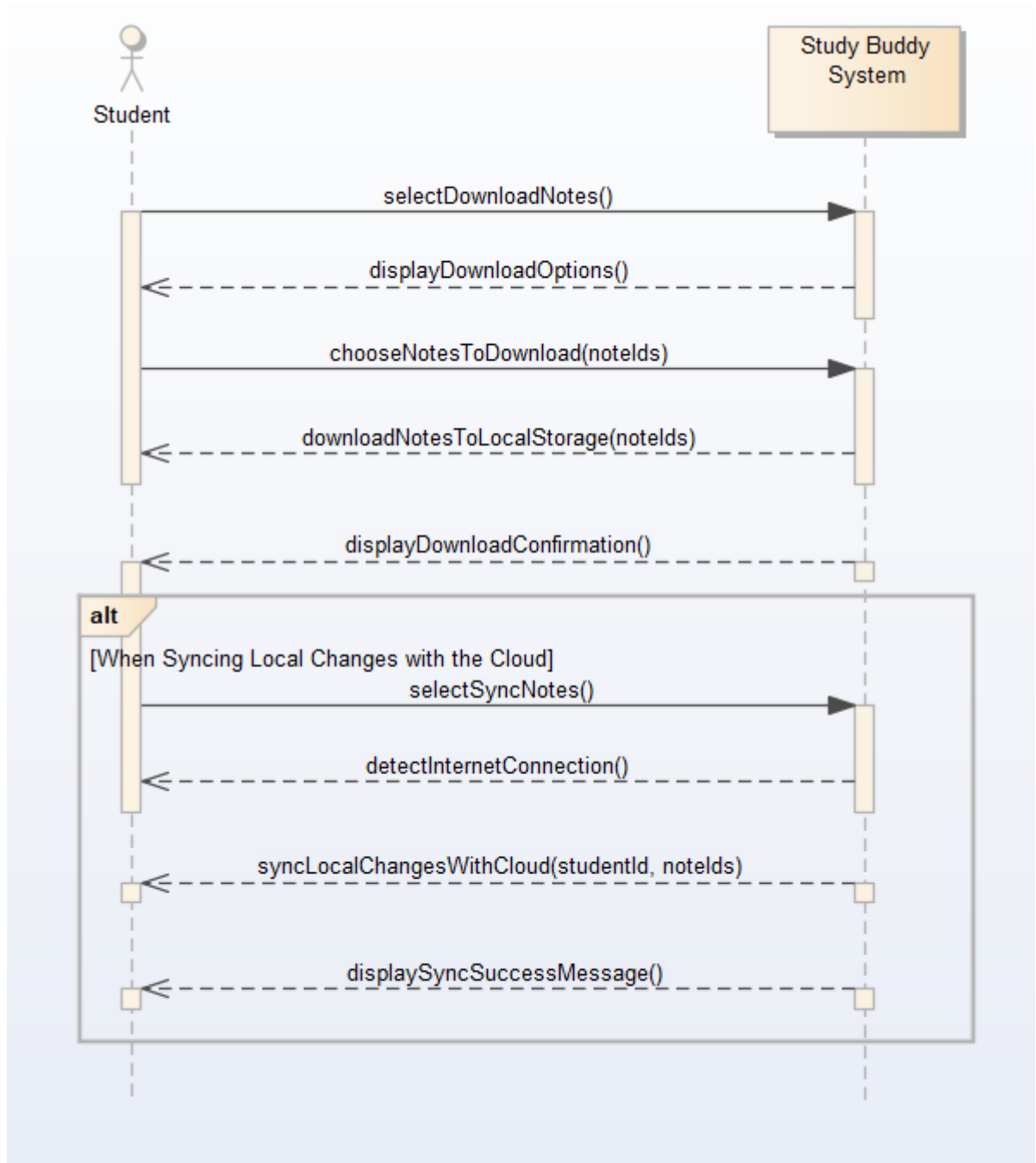


Figure 23 Download and Sync Notes SSD



Figure 24 Search Notes SSD

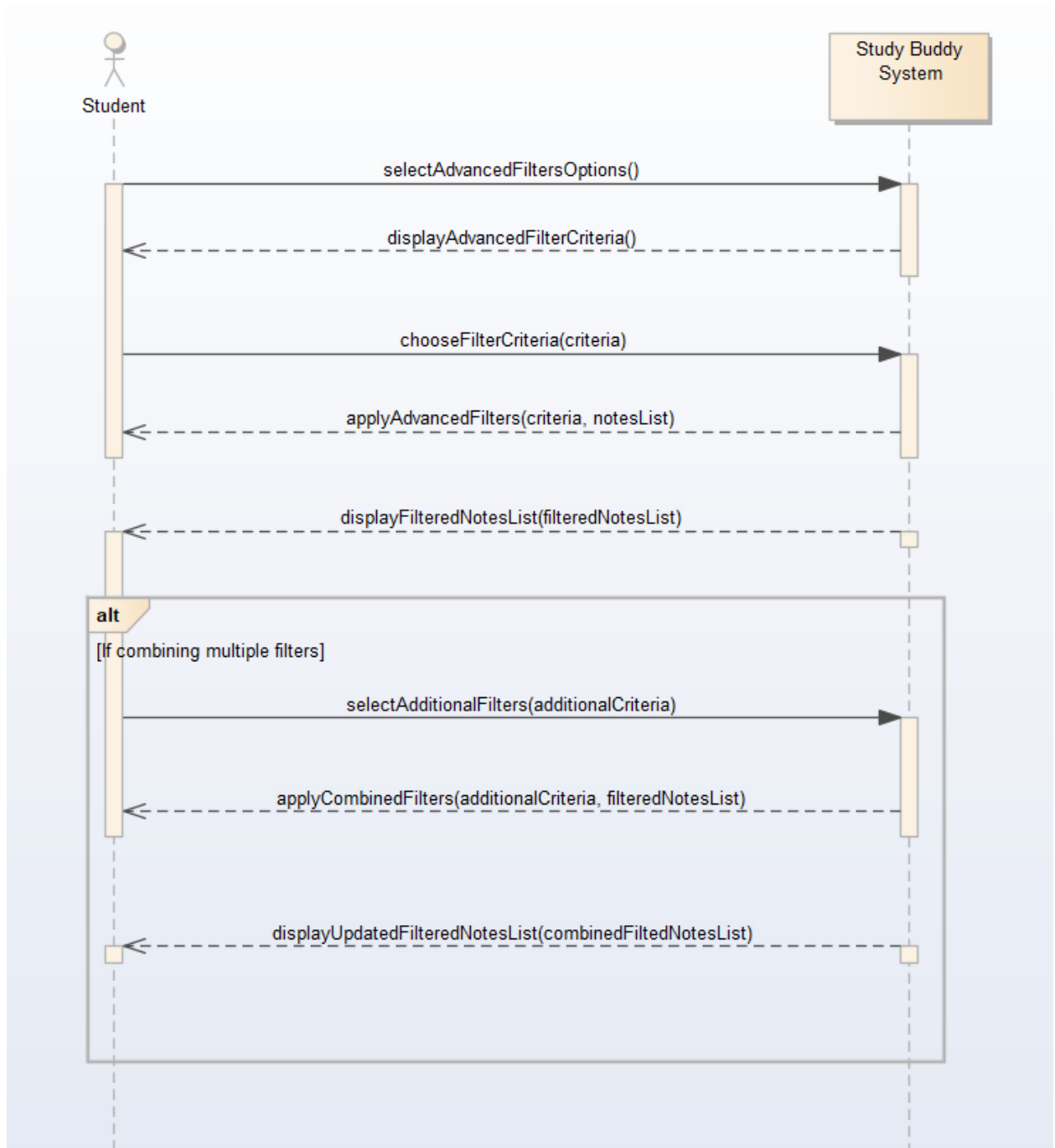


Figure 25 Advanced Filtering SSD

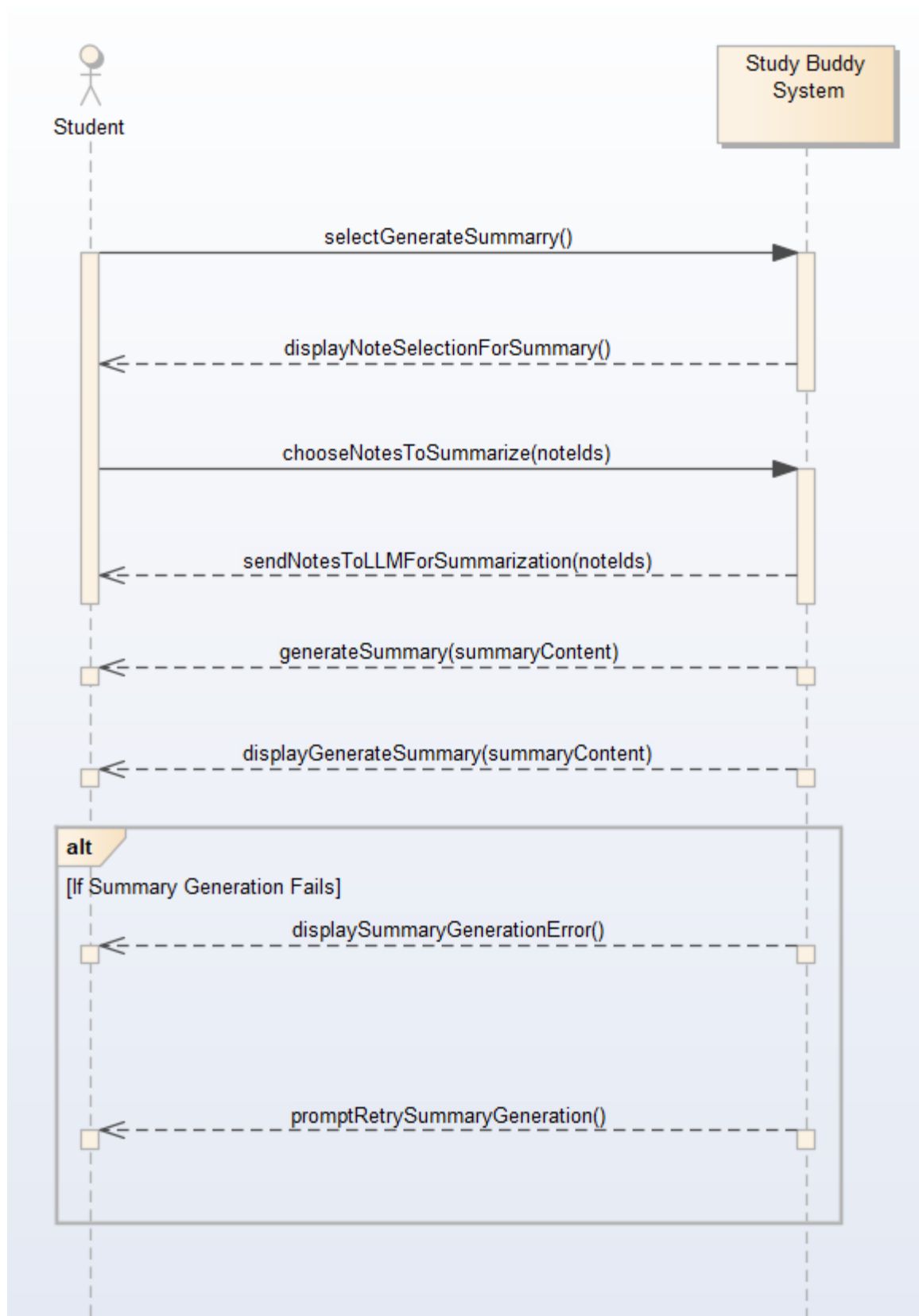


Figure 26 Generate Summaries SSD

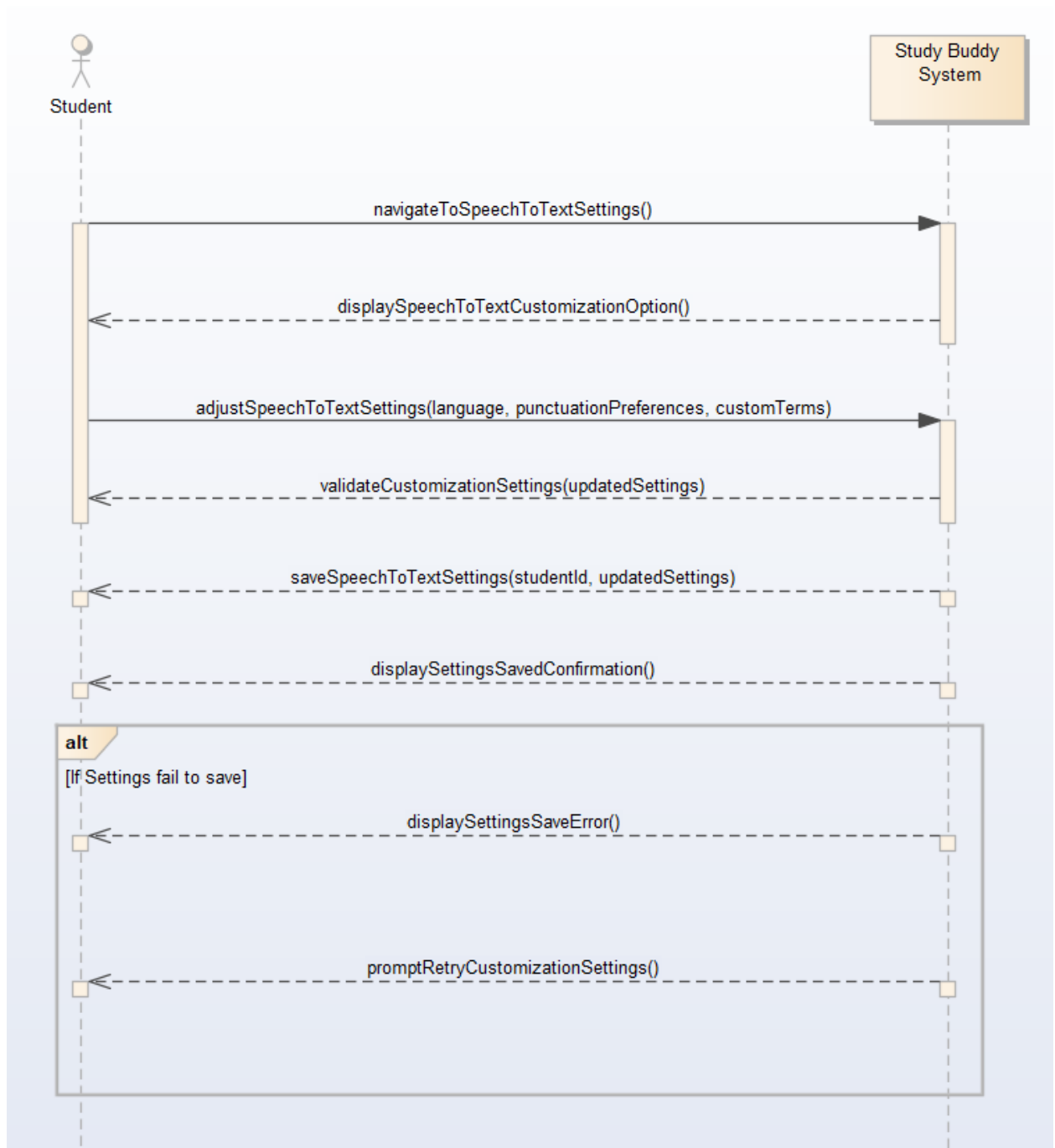


Figure 27 Customize Speech-to-Text SSD



Figure 28 Transcribe Notes SSD



Figure 29 Generate Flashcards SSD



Figure 30 Share Notes SSD

2.5. Entity Relationship Diagram

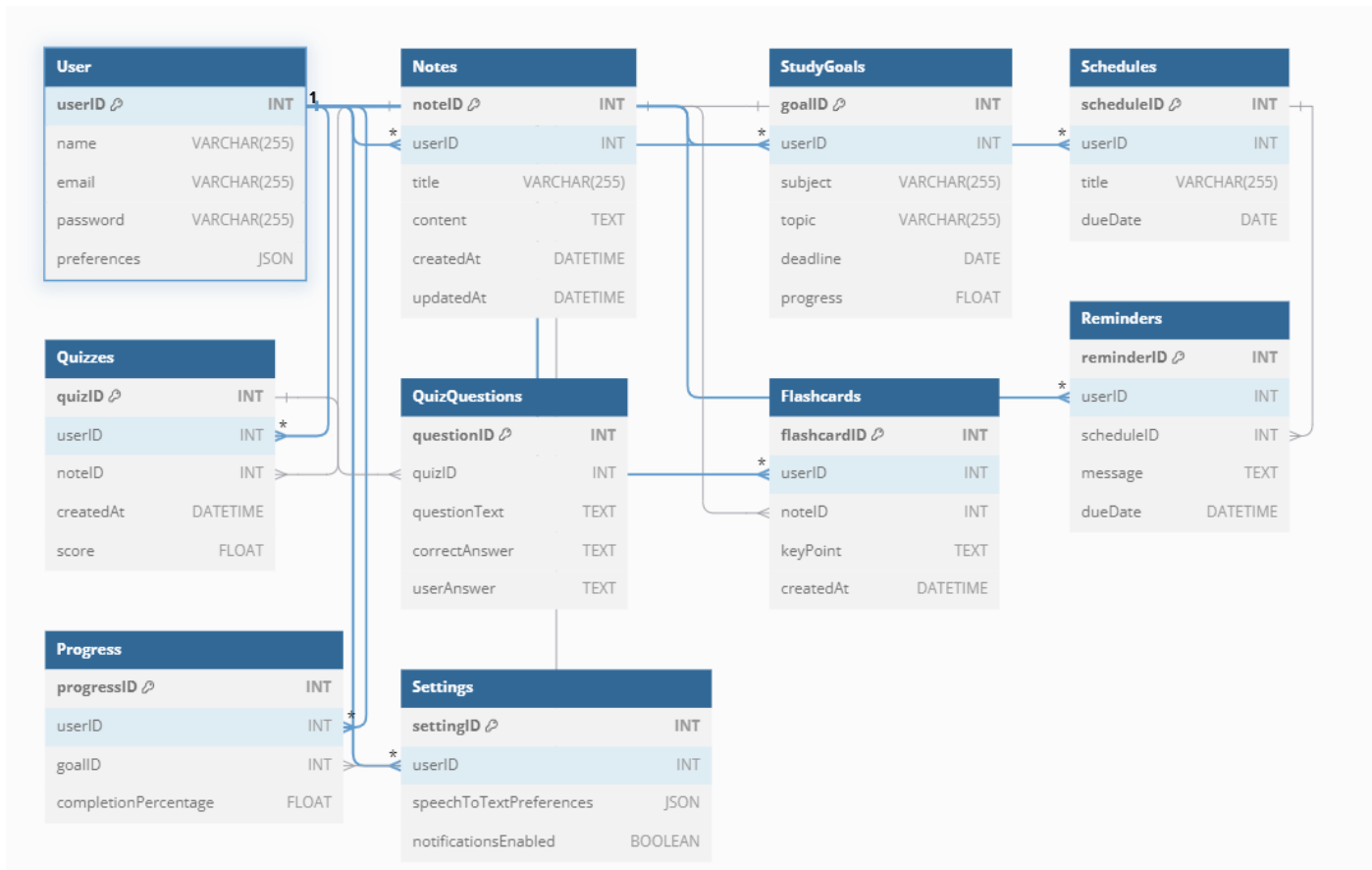


Figure 31 System ERD

2.6. Domain Model

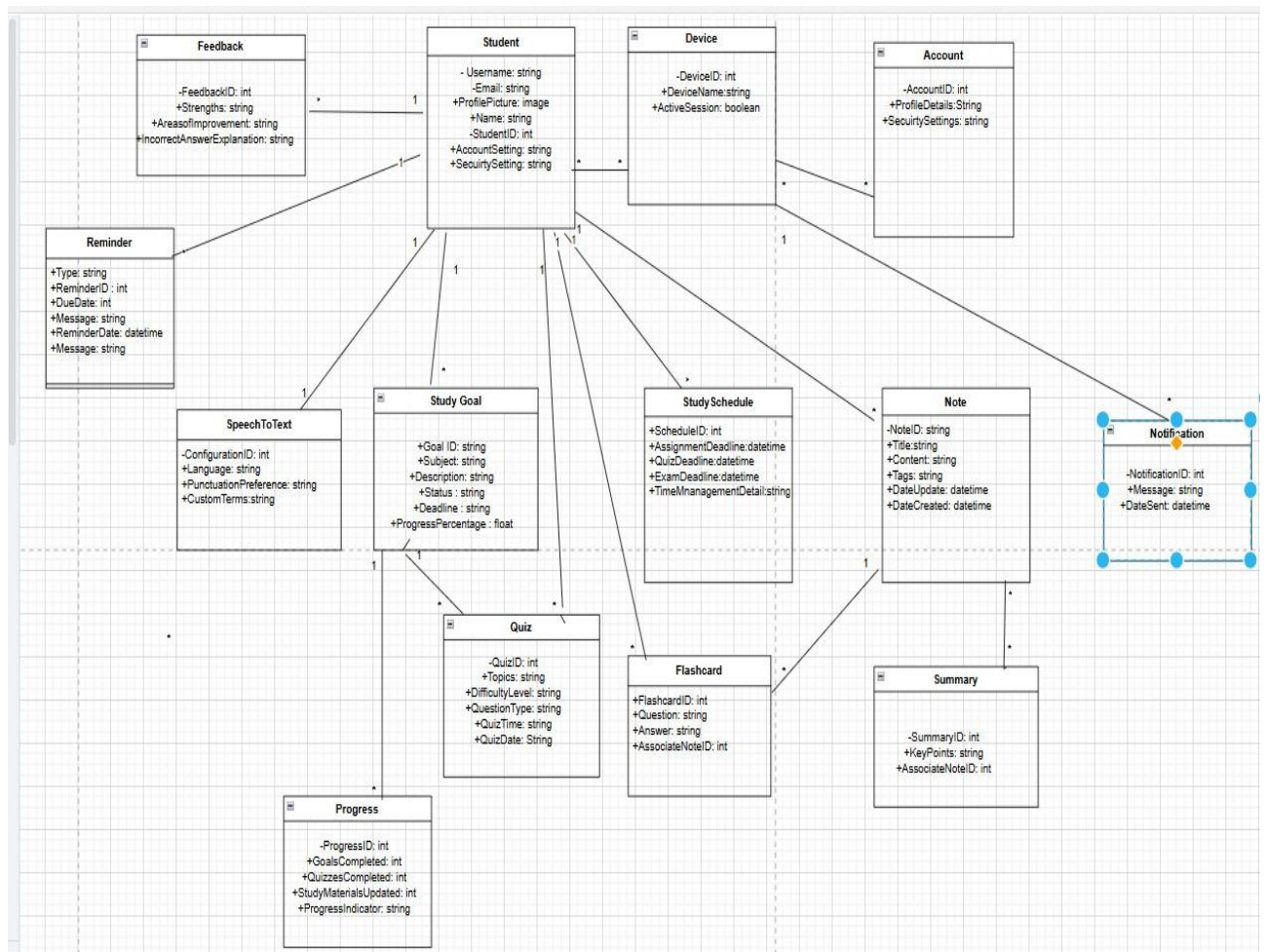


Figure 32 Domain Model

2.7. User Interfaces Design (Prototypes)

When the user opens the app, they will be greeted with the Study Buddy cover page. The interface features a subtle gradient background in soft purple tones, giving it a calming and approachable look. At the center of the screen, the app's mascot, a stylized owl wearing a graduation cap, is prominently displayed in green and black, symbolizing knowledge and learning. Below the mascot, the app's name, 'STUDY BUDDY,' is boldly written in black uppercase letters, with the tagline 'Your Pocket Tutor' elegantly placed beneath in a smaller font. A sleek horizontal progress bar is positioned near the bottom of the screen, indicating the app's loading status.



Figure 33 Splash Screen

The series of screens showcases a seamless user registration and login flow for the Study Buddy app:

1. Welcome Screen:

The first screen introduces the app with its logo and tagline, "Your Pocket Tutor." It provides social login options with Google and Facebook, placed prominently for quick access. Below, a purple "Sign up" button invites new users, while a text link allows existing users to navigate to the login page. Terms of service and privacy policy links are displayed at the bottom for transparency.

2. Registration Form:

The second screen captures essential user details, including "Full Name," "Email," and "Password." Each input field is outlined with a white border, maintaining a clean and modern appearance. Users must agree to the terms and privacy policy before signing up, with a checkbox included for this purpose. The "Sign up" button is prominently styled in purple, encouraging action. A link for existing users to log in is also provided for convenience.

3. OTP Verification Screen:

The next step involves verifying the email address through an OTP. Users enter the code received via email into a four-digit input box styled in white for clarity. A "Resend OTP" option ensures users can request another code if needed. The "Verify Me" button is purple and bold, signaling the final step of verification.

4. Successful Verification Screen:

This screen confirms that the OTP was successfully verified. It includes an indication of success with pre-filled user details and a prompt to set up the password. Users can then proceed with registration through the "Sign up" button.

5. Login Screen:

The login page allows existing users to enter their "Email" and "Password." A "Forget Password?" link is available for recovery options. Social login buttons (Google and Facebook) are presented again for convenience. The purple "Login" button completes the design, making it simple and efficient for users to access their accounts.

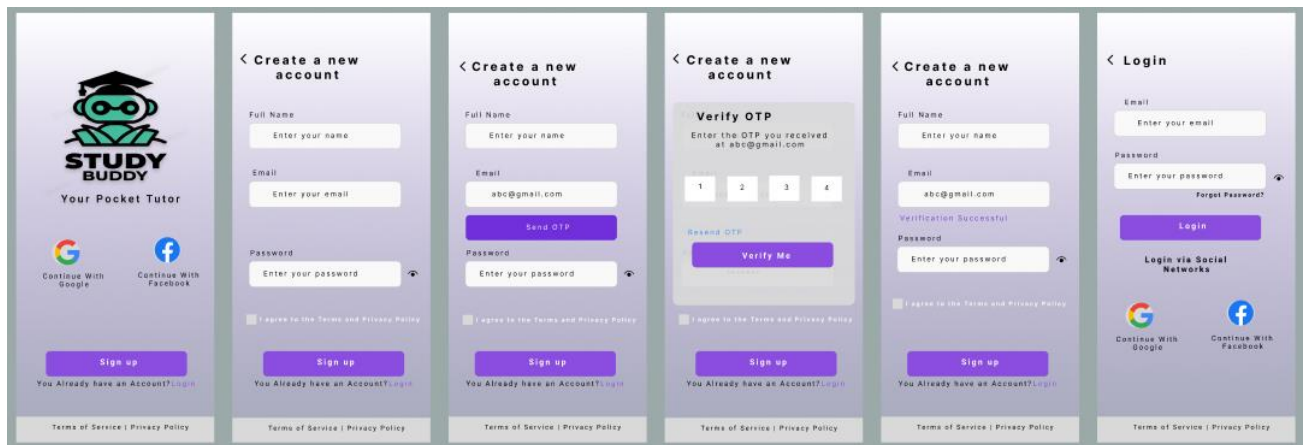


Figure 34 Welcome, Registration, and Login Screens

These dashboard screens of the Study Buddy app provide transcription functionality for users:

1. **Initial Dashboard:**

The screen greets the user with "Welcome back, [User Name]!" and features a large placeholder box labeled "Transcription will appear here." Below, a purple "Start Recording" button is displayed, enabling users to begin audio recording.

2. **Recording In Progress:**

When recording starts, the "Start Recording" button changes to "Stop Recording," with a square stop icon for clarity. The transcription box remains empty, awaiting further actions.

3. **Audio File Saved:**

After stopping the recording, the recorded audio file (e.g., "RecordedAudioFile.wav") is displayed beneath the transcription box, with a play button for playback. The user can choose to "Start Transcribing" using the adjacent button.

4. **Transcription in Progress:**

When transcription begins, the "Start Transcribing" button changes to "Stop Transcribing," indicating the ongoing process. The placeholder text in the transcription box remains visible until the process is completed.

5. **Completed Transcription:**

The transcription box is updated with the converted text from the audio file. The user can now review or interact with the transcription.

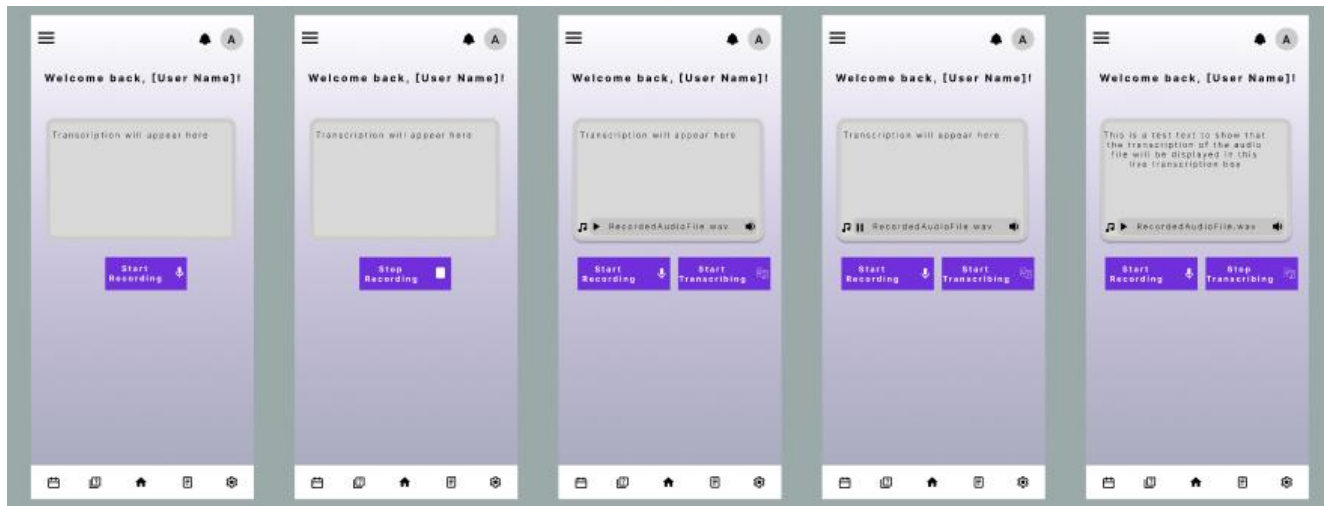


Figure 35 Dashboard Screens

These screens depict a user dashboard with a side navigation panel expanded. The panel contains options like:

1. **Reminders**
2. **Notes**
3. **Quizzes**
4. **Summaries**
5. **Study Schedules**
6. **Goals**
7. **Progress Overview**
8. **About App**
9. **Help & Support**
10. **Log-out**

The user's profile picture, name, and email appear at the top of the panel, with a settings icon at the bottom. Each screen seems to have the same layout, differing only slightly in context or background content.

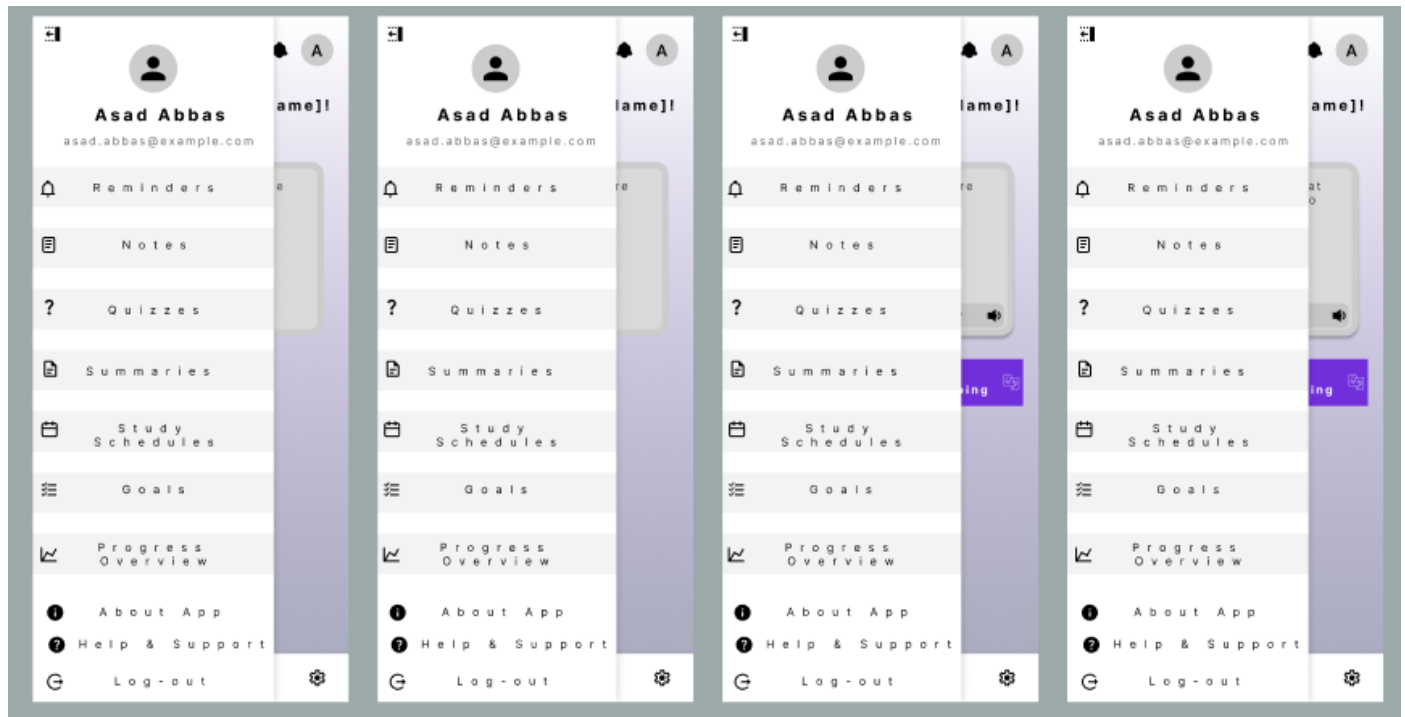


Figure 36 Dashboard Side Panel Screens

These screens show the dashboard with a **notification drop-down panel** open. The panel displays two reminders:

1. **Complete Your Quiz by 3 PM**
2. **Assignment Due by Tomorrow**

The layout remains consistent, with the notification icon in the top bar triggering the drop-down, and the bottom navigation bar includes icons for various features (e.g., calendar, home, and settings). The background content is slightly visible behind the drop-down. All screens are nearly identical.

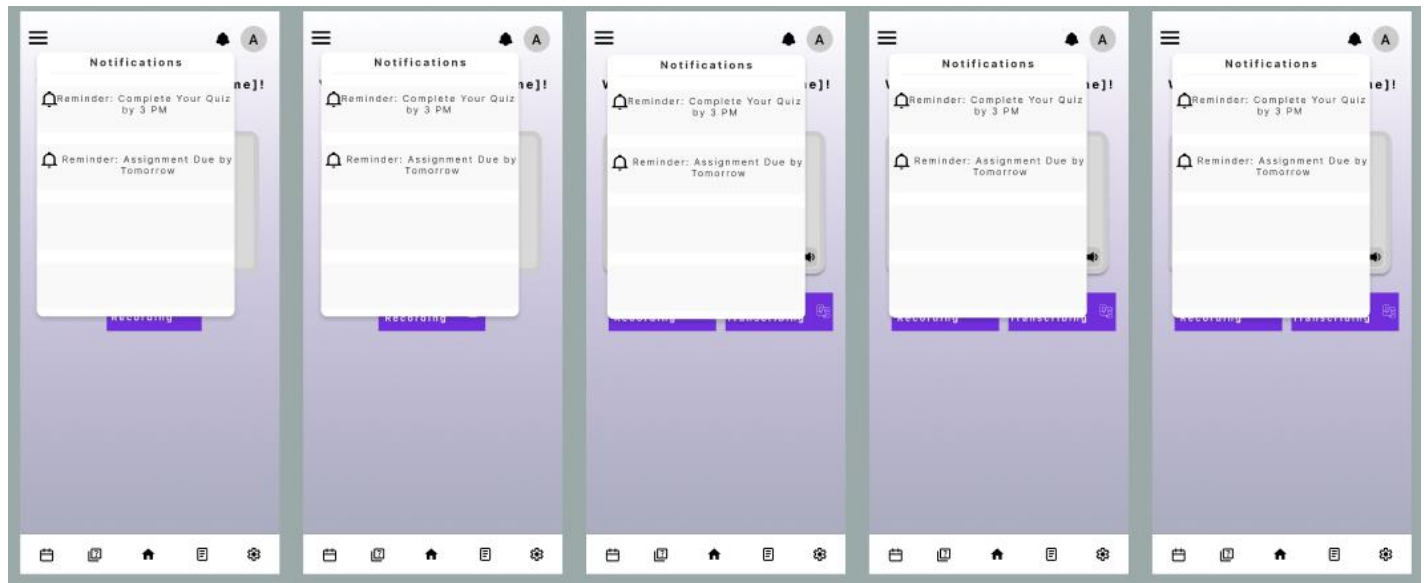


Figure 37 Dashboard Notification Drop-down Panel Screen

The drop-down panel provides access to user settings and information. It includes options for profile settings, general settings, activity log, saved items, help and support, and logout.

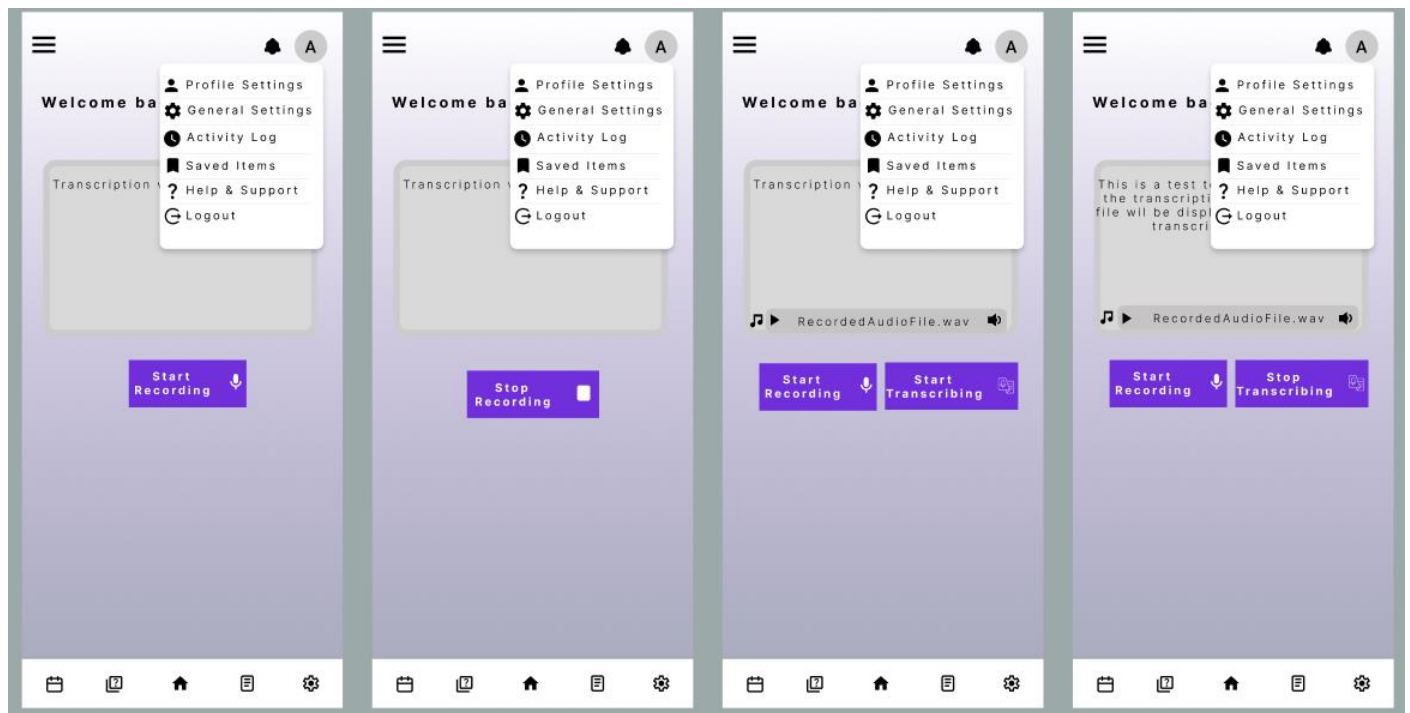


Figure 38 Profile icon drop-down panel screen.

1. Reminder Creation:

- The user taps the "+" button to initiate the creation process.
- A form appears with fields for Title, Description, Date, and Time.
- The user enters the necessary details.
- The user taps "Save" to confirm the reminder.

2. Reminder Display:

- Created reminders are displayed in a list view.
- Each reminder shows its Title, Description, Date, and Time.
- The list can be sorted by date or time.

3. Reminder Editing:

- The user taps on a reminder to open its details.
- They can edit the Title, Description, Date, and Time.
- The user taps "Save" to update the reminder.

4. Reminder Deletion:

- The user swipes left on a reminder to reveal a "Delete" button.
- Tapping "Delete" removes the reminder from the list.



Figure 39 Reminder Screens

These screens showcase the Reminders app with the side panel open. The side panel provides quick access to various features and settings within the app, such as Home, Notes, Quizzes, Summaries, Study Schedules, Goals, Progress Overview, About App, Help & Support, and Log-out.

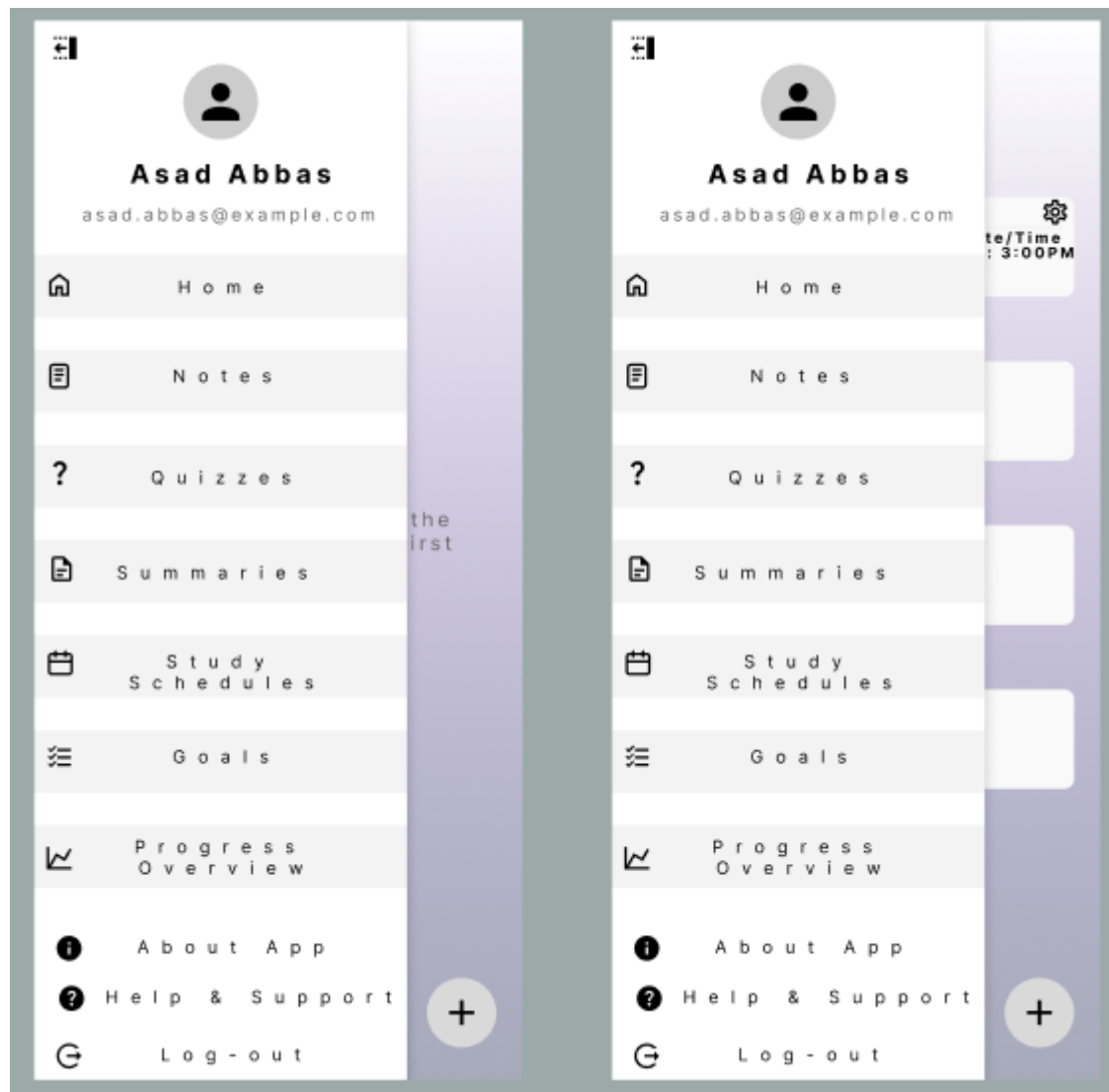


Figure 40 Reminders Side Panel Screens

Creating a New Note:

1. **Tap the "+" button:** This action initiates the note creation process.
2. **Enter Note Title and Content:** The user is prompted to provide a title and content for the note.
3. **Set Course Name (Optional):** The user can optionally assign a course name to the note.
4. **Add Tags (Optional):** Tags can be added to categorize the note for easier searching.
5. **Save:** Once the details are filled in, the user taps "Save" to create the note.

Viewing and Managing Notes:

- **Note List:** The main screen displays a list of created notes, showing their titles and course names.
- **Search:** The search bar allows users to filter notes by title, content, or tags.
- **Advanced Filters:** The advanced filter options provide more granular control over the search results, including filtering by course name, date range, and note type.

- **Side Panel:** The side panel provides access to other app features like reminders, quizzes, summaries, study schedules, goals, progress overview, app information, and help & support.

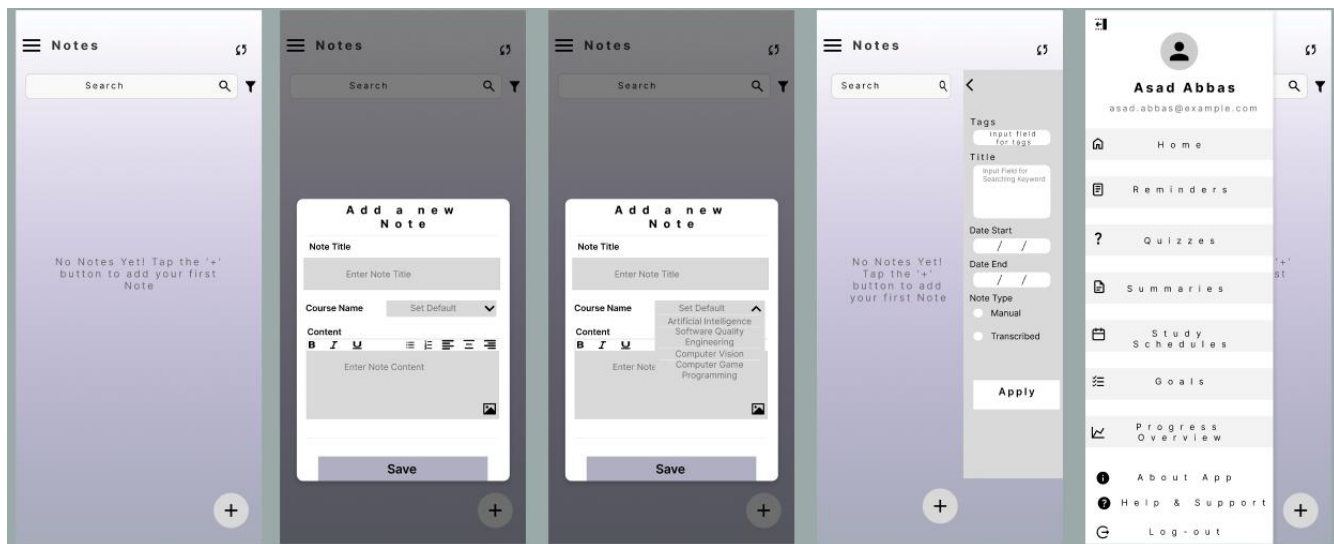


Figure 41 Notes Screens

Creating a New Note:

1. **Tap the "+" button:** This action initiates the note creation process.
2. **Enter Note Title and Content:** The user is prompted to provide a title and content for the note.
3. **Set Course Name (Optional):** The user can optionally assign a course name to the note.
4. **Add Tags (Optional):** Tags can be added to categorize the note for easier searching.
5. **Save:** Once the details are filled in, the user taps "Save" to create the note.

Viewing and Managing Notes:

- **Note List:** The main screen displays a list of created notes, showing their titles and course names.
- **Search:** The search bar allows users to filter notes by title, content, or tags.
- **Gear Button (Advanced Filters):** Clicking the gear button opens a panel with advanced filter options. These options allow for more granular filtering by course name, date range, and note type.
- **Side Panel:** The side panel provides access to other app features like reminders, quizzes, summaries, study schedules, goals, progress overview, app information, and help & support.

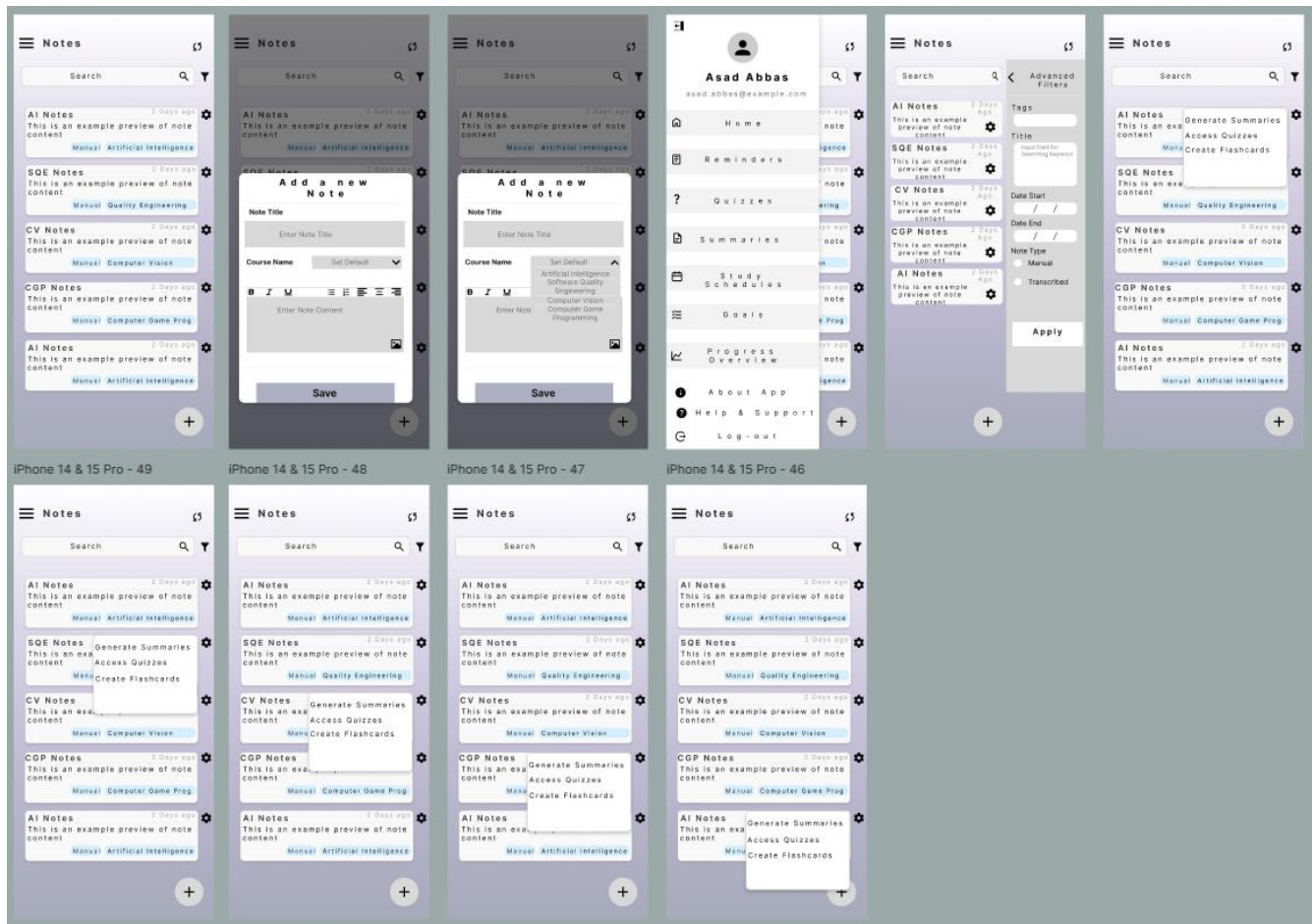


Figure 42 Populated Notes Screen

Note Details View:

- When a user taps on a note from the list, the app displays the detailed view of that note.
- The details view shows the note's title, content, course name, and tags.
- A small drop-down panel is present in the top right corner of the screen.

Drop-down Panel:

- The drop-down panel provides options to:
 - **Edit Note:** This allows the user to modify the title, content, course name, and tags of the note.
 - **Delete Note:** This permanently removes the note from the app.

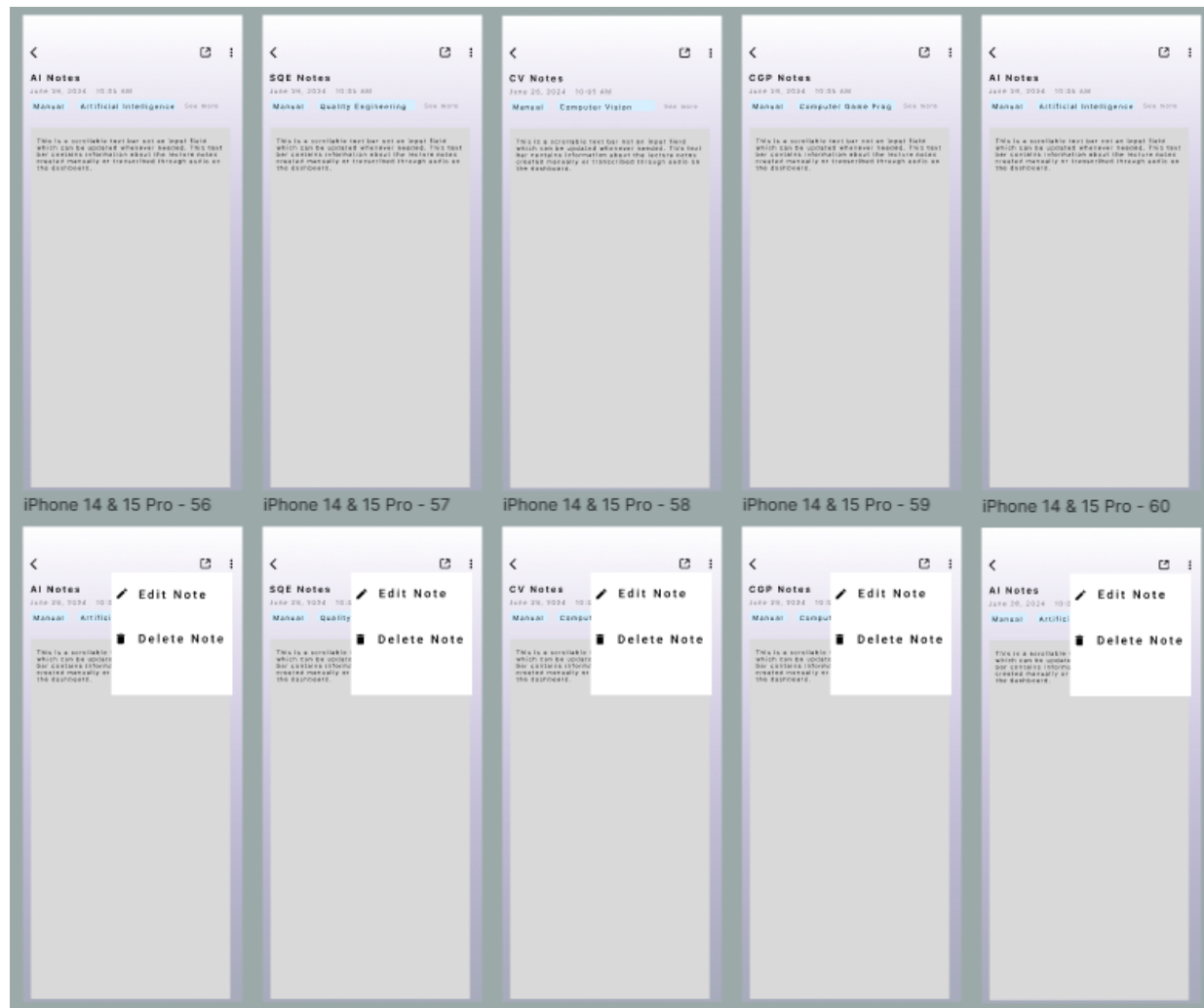


Figure 43 Detailed Notes Screen

These screens show the confirmation and editing processes for notes within the app.

Confirmation:

- When a user chooses to delete a note, a confirmation dialog appears asking them to confirm their action.
- The dialog provides "Delete" and "Cancel" buttons for the user to choose from.

Editing:

- When a user selects the "Edit Note" option, the note's title, content, course name, and tags become editable.
- The user can make changes to these fields as needed.
- Once the changes are complete, the user taps the "Save" button to update the note.

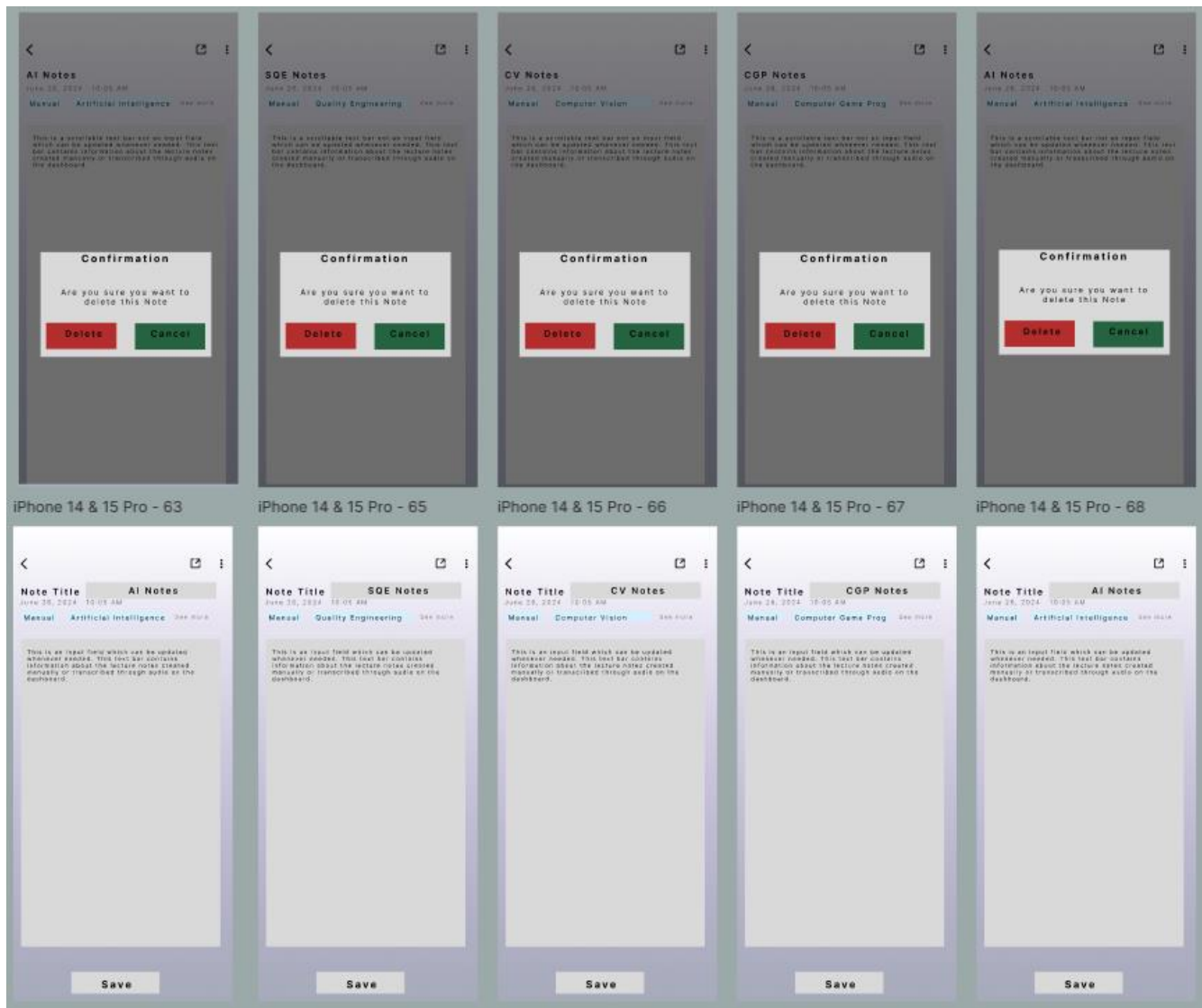


Figure 44 Deletion Confirmation and Note Editing Screens

These screens show the updated note list after a note has been deleted. The deleted note is no longer visible in the list, and the remaining notes are displayed in the same format as before. The user can continue to create, edit, and delete notes as needed.



Figure 45 Updated Notes Screen

These screens display the QR code generated for sharing a specific note. By scanning this QR code, another user can access the shared note. This feature allows for easy sharing of notes among students or colleagues.

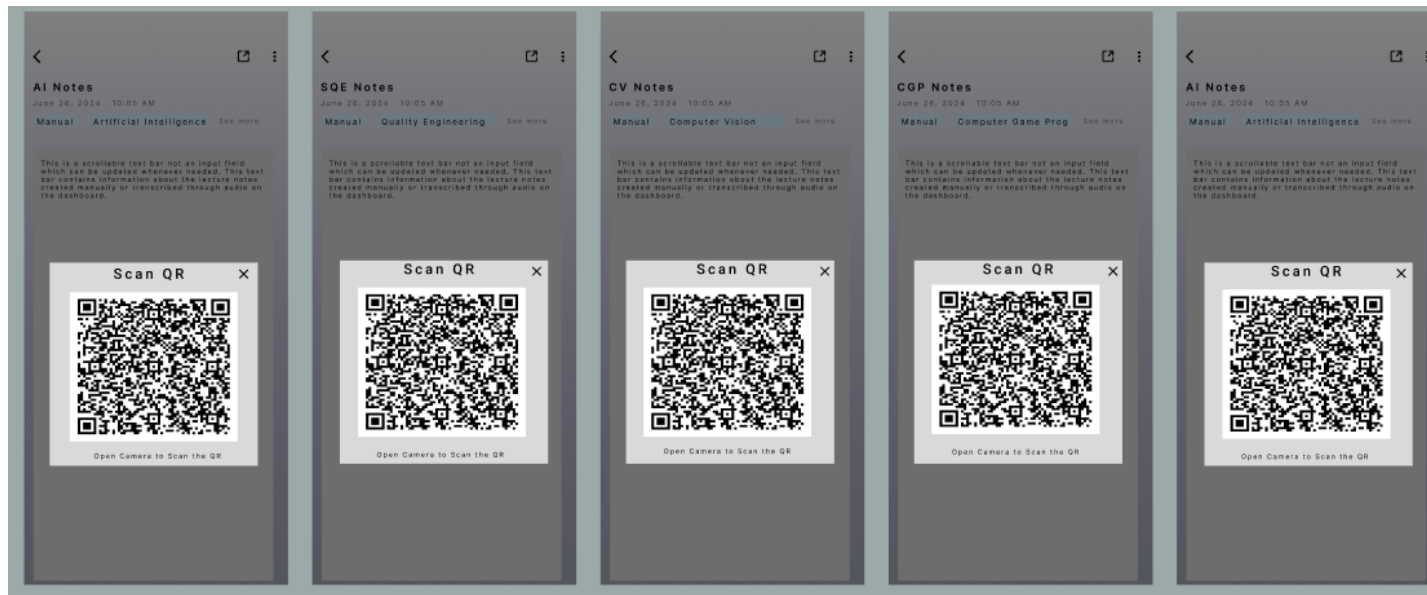


Figure 46 Share Notes Screens

Initial Summary Screen:

- This screen displays an empty list of summaries, indicating that no summaries have been created yet.

Advanced Filters Side Panel:

- This panel provides options for filtering summaries based on specific criteria, such as tags, date range, and note type.

Side Panel:

- The side panel provides access to other app features like reminders, notes, quizzes, study schedules, goals, progress overview, app information, and help & support.

Filled Summary Screen:

- This screen displays a list of created summaries. Each summary shows its title, description, and other relevant details.

Advanced Filters on Filled Summary Screen:

- When the advanced filters are applied, the list of summaries is filtered based on the specified criteria, displaying only the relevant summaries.

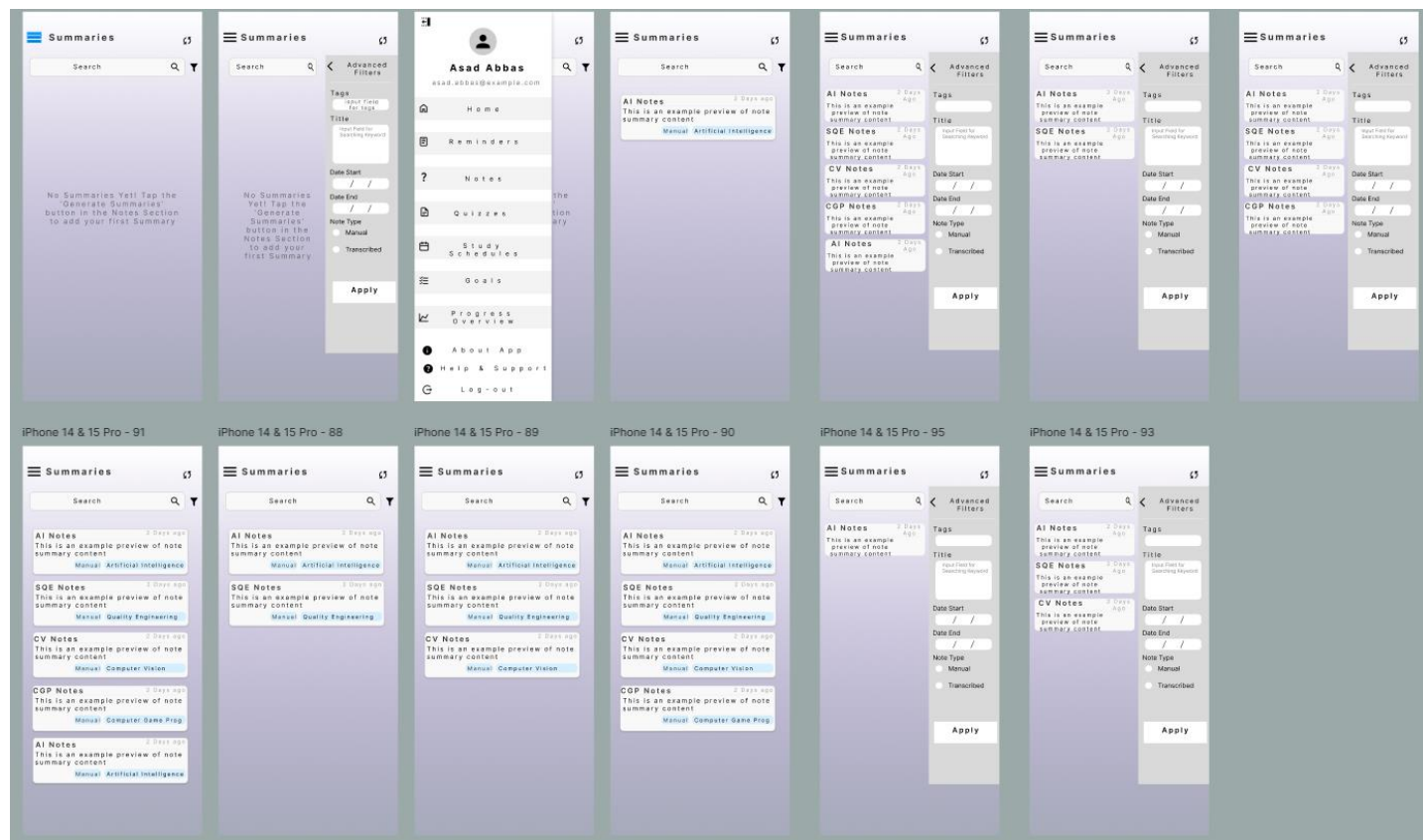


Figure 47 Summary Screens

Side Panel:

- The side panel provides access to other app features like reminders, notes, quizzes, study schedules, goals, progress overview, app information, and help & support.

Summary Details:

- The detailed view displays the title, description, and other relevant information about the selected summary.
- This view provides a clear and concise overview of the summary's content.

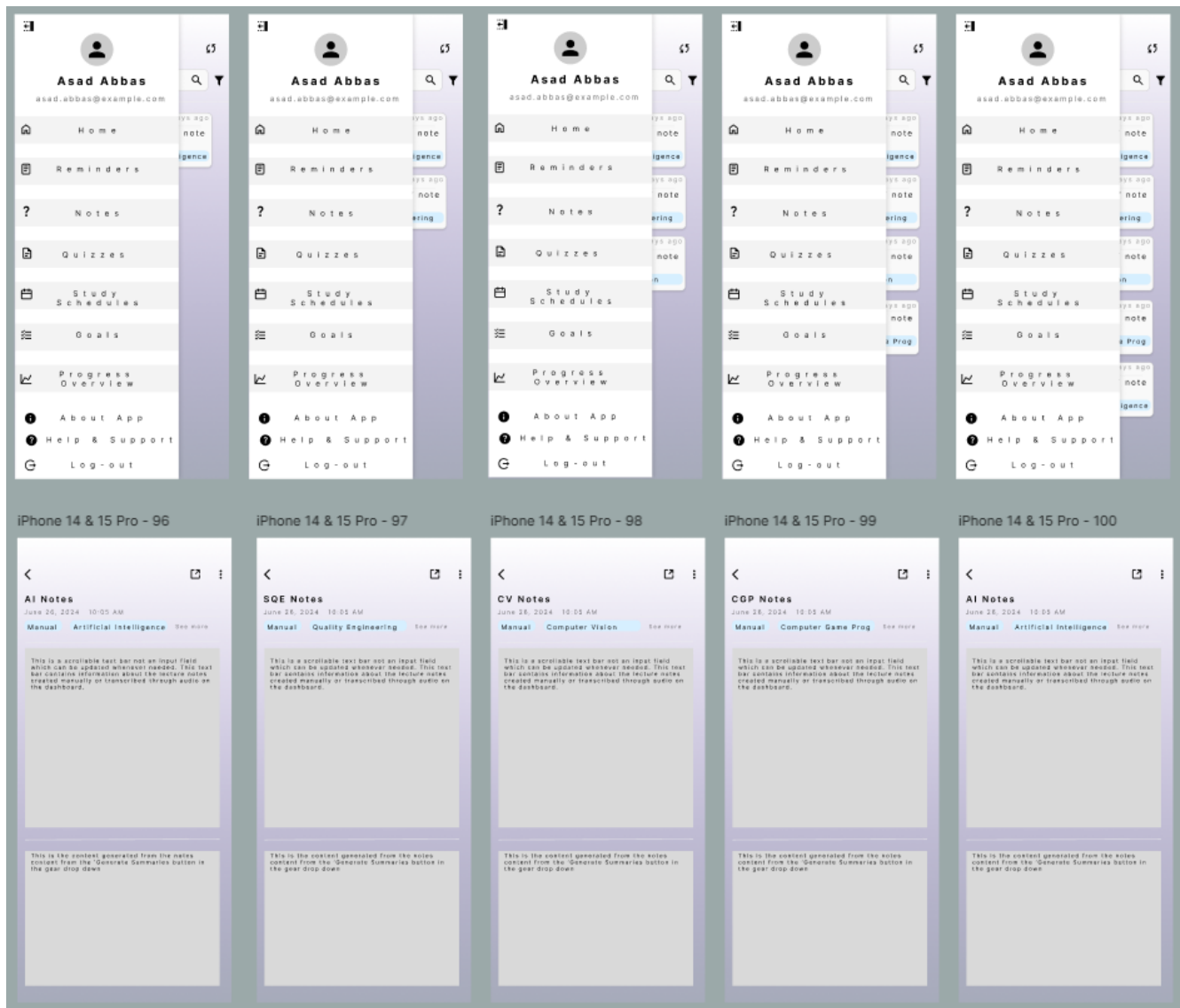


Figure 48 Detailed Summaries and Summaries Side Panel Screens

Initial Goal Screen:

- This screen displays an empty list of goals, indicating that no goals have been created yet.

Side Panel:

- The side panel provides access to other app features like reminders, notes, quizzes, study schedules, summaries, progress overview, app information, and help & support.

Adding a New Goal Screen:

- This screen allows users to create a new goal by entering the following information:
 - Goal Title: A brief description of the goal.
 - Subject/Topic: The relevant subject or topic area.
 - Deadline: The target date for achieving the goal.
 - Progress (%): The current progress towards the goal.

Populated Goal Screen:

- This screen displays a list of created goals. Each goal shows its title, subject/topic, deadline, and current progress.

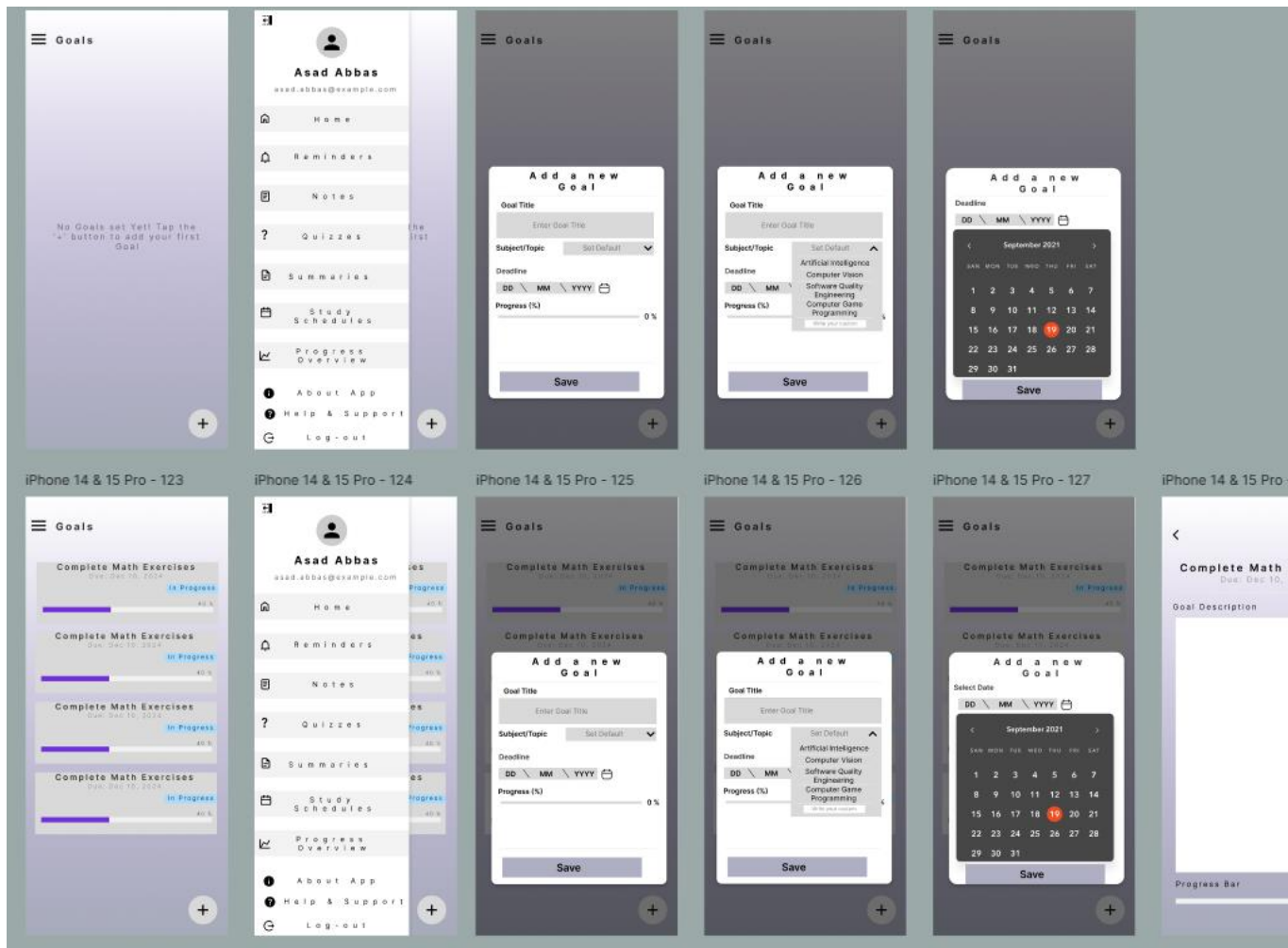


Figure 49 Goal Screens

Initial Quiz Screen:

- This screen displays a list of available quizzes.
- Users can tap on a quiz to start it.

Side Panel:

- The side panel provides access to other app features like reminders, notes, summaries, study schedules, goals, progress overview, app information, and help & support.

Populated Quiz Screen:

- This screen shows a list of available quizzes, with additional information such as the quiz title and a brief description.

Quiz Starting Point:

- This screen displays the first question of the quiz, along with a timer.
- Users can answer the question and proceed to the next one.

Multiple Types of Quiz Questions:

- The app supports different types of quiz questions, including true/false, multiple-choice, and explanatory questions.
- Users can select the appropriate answer or type their response for explanatory questions.

Quiz Result Screen:

- This screen displays the user's score and a breakdown of their performance on each question.
- It also shows the correct answers for each question.

Completed Quiz Screen:

- This screen indicates that the quiz has been completed.
- It may display a summary of the user's performance or provide feedback.

Completed Quiz Screen with Side Panel:

- This screen shows the completed quiz screen with the side panel open, allowing users to access other app features.

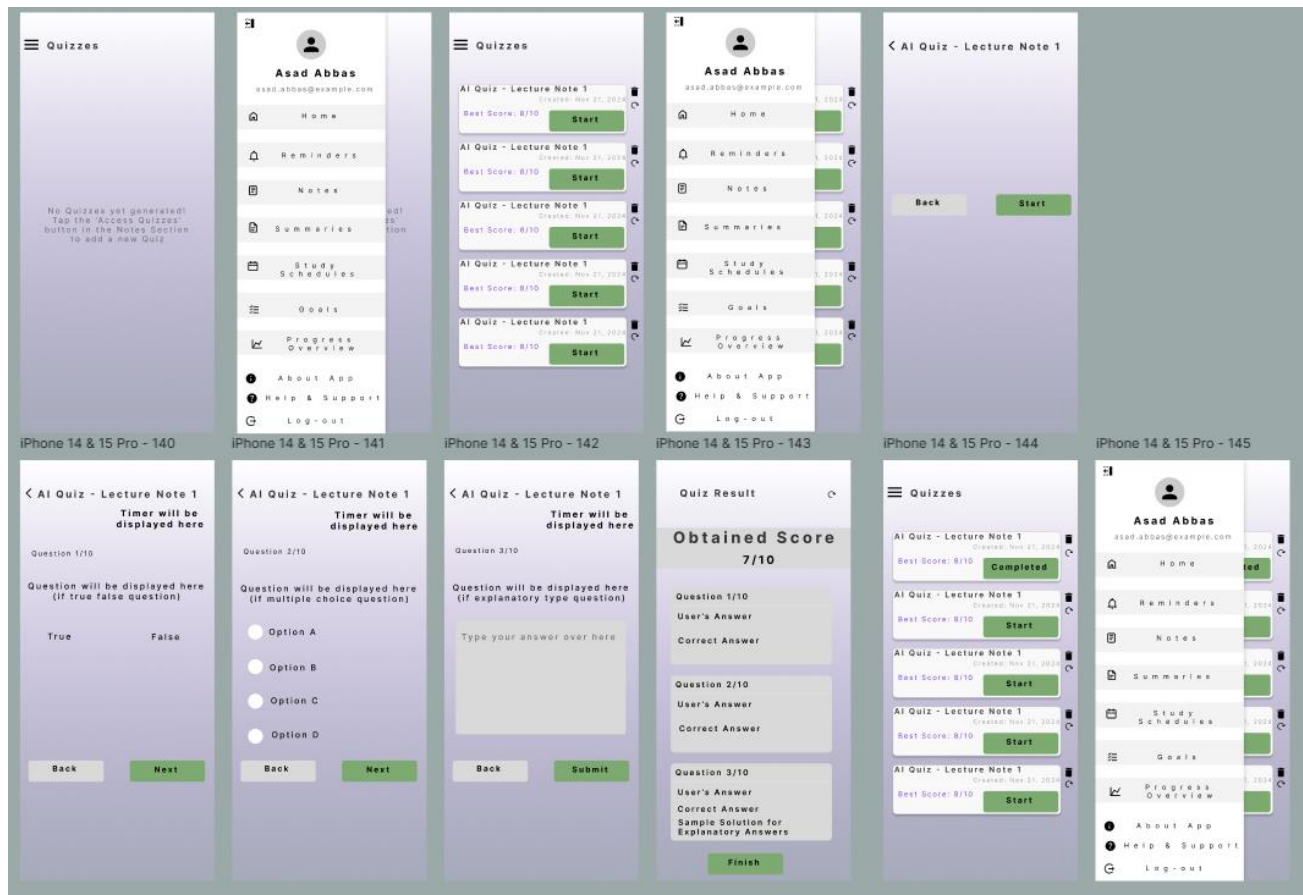


Figure 50 Quiz Screens

Initial Progress Overview Screen:

- This screen displays an empty overview, indicating that no progress has been tracked yet.
- It encourages users to set goals and complete quizzes to start tracking their progress.

Advanced Filters Side Panel:

- This panel allows users to filter the progress data based on different time periods, such as the last 7 days, 30 days, or all time.

Side Panel:

- The side panel provides access to other app features like reminders, notes, quizzes, study schedules, goals, summaries, app information, and help & support.

Populated Progress Overview:

- This screen displays various progress metrics in a visual format:

- **Goals Achieved:** A pie chart showing the percentage of completed goals.
- **Quizzes Completed:** A bar chart displaying the number of quizzes completed over time.
- **Study Sessions Completed:** A line chart illustrating the duration of study sessions over time.

Advanced Filters on Populated Progress Overview:

- When the advanced filters are applied, the progress data is filtered to display information for the selected time period.

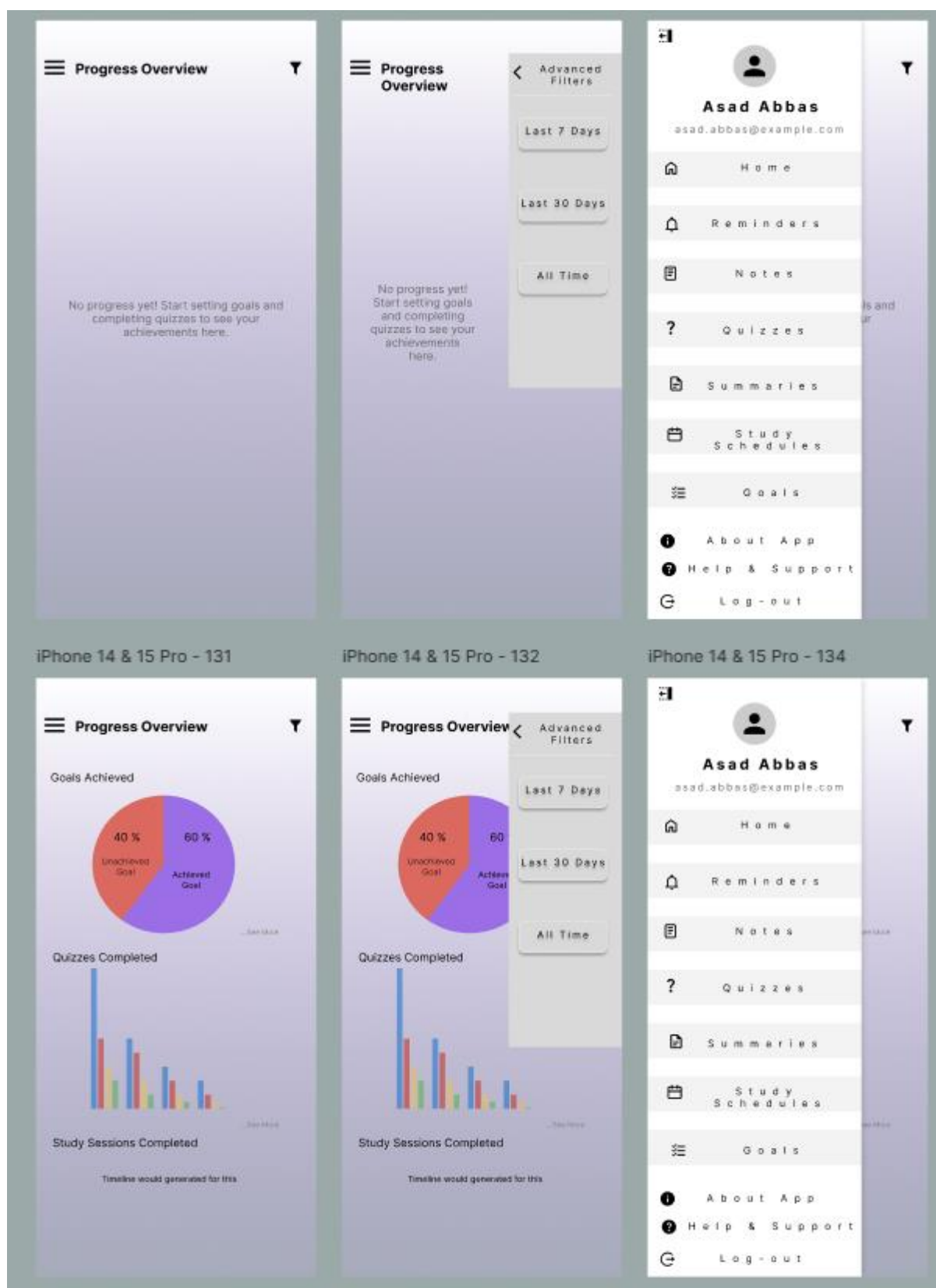


Figure 51 Progress Overview Screens

Initial Study Schedule Screen:

- This screen displays a calendar view, allowing users to visualize their study schedule for the month.
- Users can tap on specific dates to view scheduled tasks or create new ones.

Side Panel:

- The side panel provides access to other app features like reminders, notes, quizzes, goals, progress overview, summaries, app information, and help & support.

Adding a New Study Schedule:

- This screen allows users to create a new study schedule by entering the following information:
 - **Subject/Topic:** The relevant subject or topic area.
 - **Date:** The target date for the study session.
 - **Start Time:** The start time of the study session.
 - **End Time:** The end time of the study session.

Populated Study Schedule Screen:

- This screen displays a list of created study schedules. Each schedule shows its subject/topic, date, and start/end times.
- Users can tap on a specific schedule to view more details or edit it.

Detailed Study Schedule:

- This screen indicates that a study schedule has been completed.
- It may display a confirmation message or provide a summary of the study session.

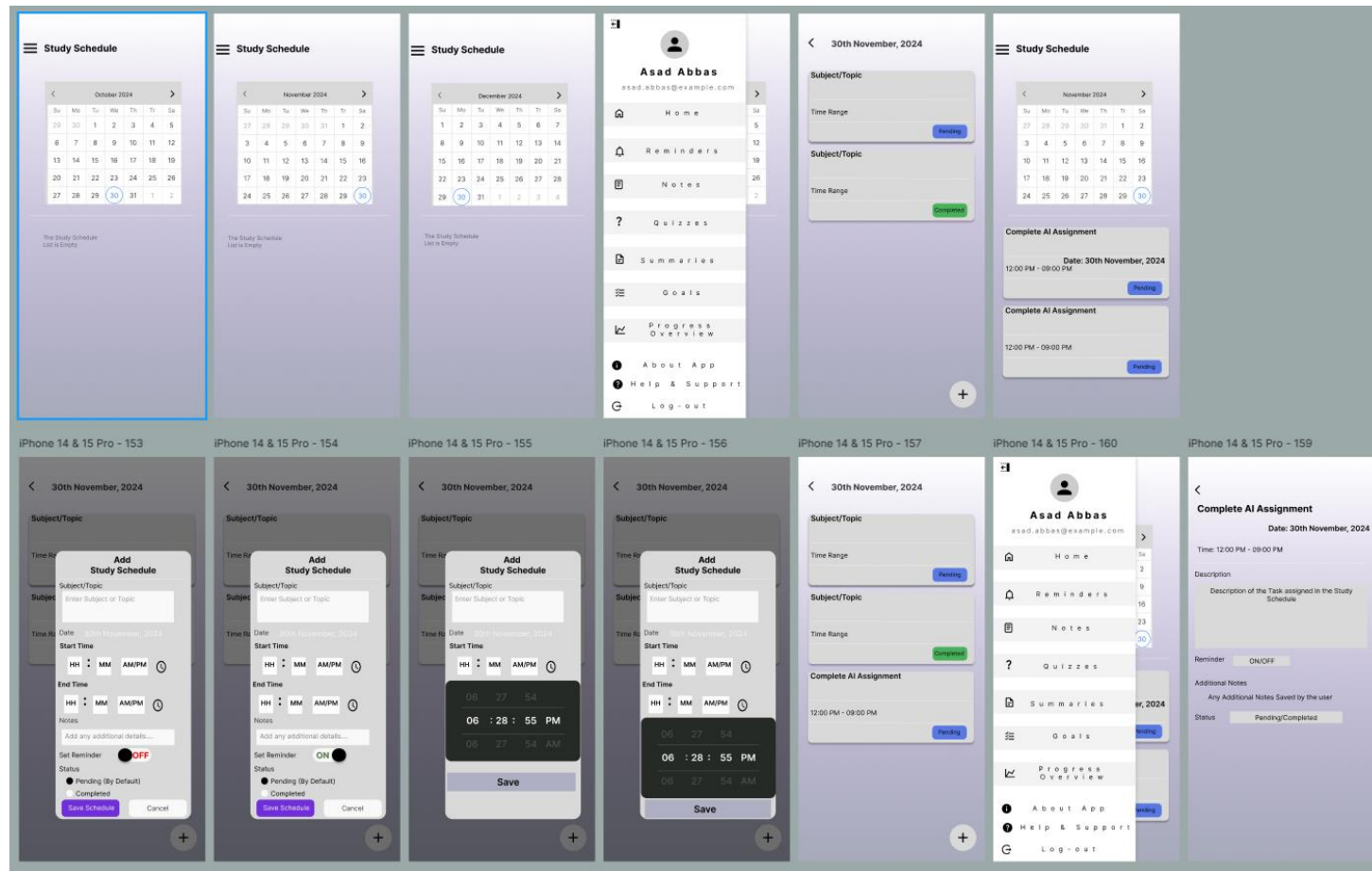


Figure 52 Study Schedule Screens

Link to the Figma Prototypes: <https://www.figma.com/design/KWo76dKg0V2Zf1GIpUECV7/final-year-project?node-id=469-2&t=t1A5YbrKqFtlVWW6-l>