CNG 495 Cloud Computing

Fall 2025-2026

Capstone Project Final Report: StudyMate Mobile Application

Arisha Ahmad (2751923)

Faiz Rashid (2460384)

Table of Contents

1. Introduction

1.1 Project Overview

StudyMate has been designed as an advanced academic management system tailored to maximize the day-to-day operations of academic students at METU-NCC through a simplified, easy to operate interface system. Created with Flutter technology for its enhanced Android platform experience and developed on Supabase for its secure and real-time data synchronization features, it functions as an all-inclusive platform for academic activities, embedded not only with academic organization but also equipped with basic productivity services such as class organization, a timer, and a to-do list along with a highly functional help forum, private messaging, and note taking, thus allowing for peer collaboration on the cloud for enhancing consistency and productivity in academic pursuits.

1.2 Benefits and Novelties

StudyMate distinguishes itself from traditional planners by offering a collaborative ecosystem rather than just a solitary tool. The major innovations brought by this tool are:

1.2.1 Integrated Productivity Features

It integrates administrative management (schedules and notes) and active execution (Pomodoro timers and task management) in a mobile interface.

1.2.2 Anonymous Help Forum & Private Messaging

One of the major novelties within the system is the communication part where students can put up questions within the help forum and participate in private messaging to help each other reach the possible solutions while keeping their identity private. In the help forum as well, the identity of the respondent is anonymous.

1.2.3 Multimedia Note Management

The multimedia aspect of the notes management module enables users to incorporate PDFs or pictures within the text notes, providing a broader study manual.

1.2.4 Cloud Native Synchronization

Using Supabase's PostgreSQL and Storage buckets, user information, including their schedules, tasks, and any media attachment, is replicated in real-time and is accessible from any device.

1.3 Similar Existing Projects

While our application StudyMate is specifically designed to benefit METU-NCC students, still there are several other open-source GitHub repositories which somehow align with the features used in our application, the two most beneficial ones are mentioned below.

1.3.1 AppFlowy

A massive open-source project built in Flutter that demonstrates advanced note-taking and organizational layouts.

GitHub Repo Link: https://github.com/AppFlowy-IO/AppFlowy.git

1.3.2 Holom Said E-Learning

A comprehensive educational platform that integrates user management and course tracking using the same tech stack as StudyMate

GitHub Repo Link: https://github.com/jaliil-9/holom-said-elearning-flutter-mobile-app.git

2. Project Features and Functional Components

StudyMate is designed as a multi-modular system where every piece of software helps to fulfill a particular aspect of being a METU NCC student. By incorporating all these services in one place, StudyMate prevents students from using various un-interconnected tools to get their work done.

Feature	Primary Function	Technical Implementation
User Profiles	Identity and Academic Context	Supabase Auth + PostgreSQL
Class Schedule	Weekly Time Management	Flutter GridView + CRUD Operations
To-Do Tracker	Assignment and Task	Real time Database Synchronization

	Deadlines	
Pomodoro Timer	Focus & Interval Training	Dart Streams & Timers
Help Forum	Community Q&A	Real time PostgreSQL Listeners
Private Messaging	Anonymous Peer Support	Secure Message Request Logic
Anonymous Replies	Identity Masking	Frontend abstraction of User IDs
Multimedia Notes	Centralized Study Materials	Supabase Storage Buckets

2.1 Authentication & Profile Management

The gateway module incorporates the use of Supabase Auth to ensure safe, encrypted login sessions.

2.2 Class Schedule Feature

A dynamic grid timetable organizer that enables students to input their lecture schedule, which helps in efficiently organizing the campus schedule. The student can save their time table similar to the CET system provided by METU. There is space for adding the classroom as well.

2.3 To-Do List

This is a task management interface where users can manage their assignments, deadlines, and mark them as completed in order to promote responsibility and organise their time better.

2.4 Pomodoro Timer

To counter burnout and increase deep work, the Pomodoro Timer in this module has a customizable feature where the user can alternate periods of study and breaks.

2.5 Collaborative Help Forum

An interactive forum in which students can raise their study related queries. The most interesting feature of StudyMate is that responses to queries can be given anonymously, thus lowering any possible social hurdles in assisting one's peers.

2.6 Private Message Requests

The inbox allow the students to discuss their issue and solution with each other without revealing their identity. Users have the option to make private message requests in the event the question requires a more in-depth conversation.

2.7 Multimedia and Note-Taking

Unlike text-based applications, StudyMate enables the addition of academic files (PDFs) and/or images to be attached to the notes and stored on Supabase Storage.

3. Utilized cloud services

Cloud Service	Category	Technical Implementation	Elaborated Functions
Supabase Auth	SaaS	Uses JSON Web Tokens (JWT)	Handles student registration. It generates Uuids used as foreign keys in the “profiles” table to refer to user-specific schedules and tasks.
PostgreSQL Database	PaaS	Row Level Security (RLS) Policies	Stores information related to class schedules, to-do items, and forum threads. The RLS protects a user from altering class sessions or personal items of other students.
Supabase Storage	IaaS	Utilizes S3-compatible buckets with specialized folders organized by user UUID	Hosts binary assets such as PDFs of the lecture and the study images attached to the notes. It also provides authenticated URLs to enable safe download of the file in the Flutter application.
Edge Functions	PaaS	Built on a Deno logic	Handles complex server side tasks such as Anonymous Forum Reply routing and real time broadcasts of notifications when new

			help requests are made.

4. User Manual

4.1 Getting Started

4.1.1 Account Registration (Sign Up)

Purpose: Creates a profile linked to the student, which is personalized with the academic background.

- Input Fields: Students should enter their Full Name, Email, and create a Password.
- Academic Details:
  - Department: Can be chosen from a drop-down menu which is later used for the help forum (e.g., Computer Engineering).
  - Year: Can be chosen from a drop-down menu which is the current year of study (e.g., Year 4).
- Actions:
  - "Sign Up" Button: Establishes the account and verifies the input. Upon successful registration, the Dashboard is accessed right away.
  - "Log In" Link: Redirects existing users to the Login screen.

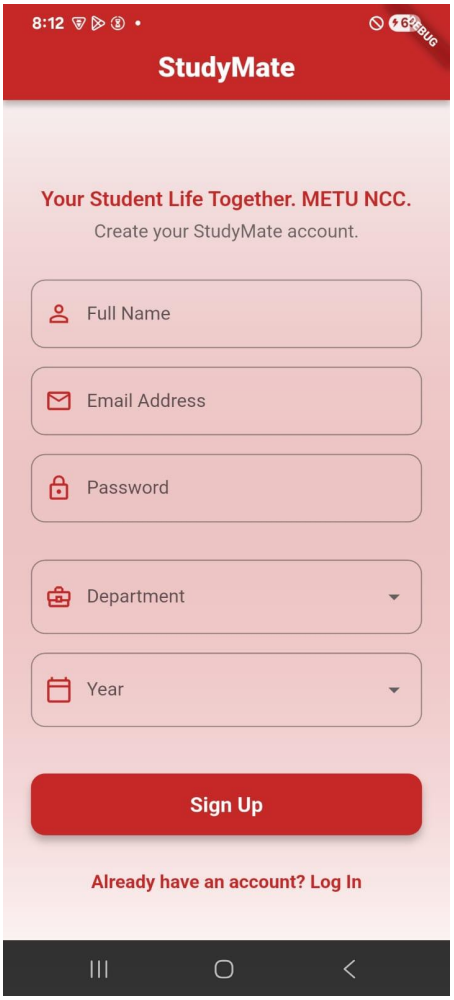


Figure 1 Sign up

4.1.2 Student Login

Purpose: Allows returning users to enter securely.

- Input Fields: Registered Email Address and Password are required.
- Actions:
  - "Log In" Button: Verifies credentials. Access to the Dashboard is granted upon success; an error message is sent upon failure.
  - "Register Now" Link: Directs new users to the Sign Up screen.

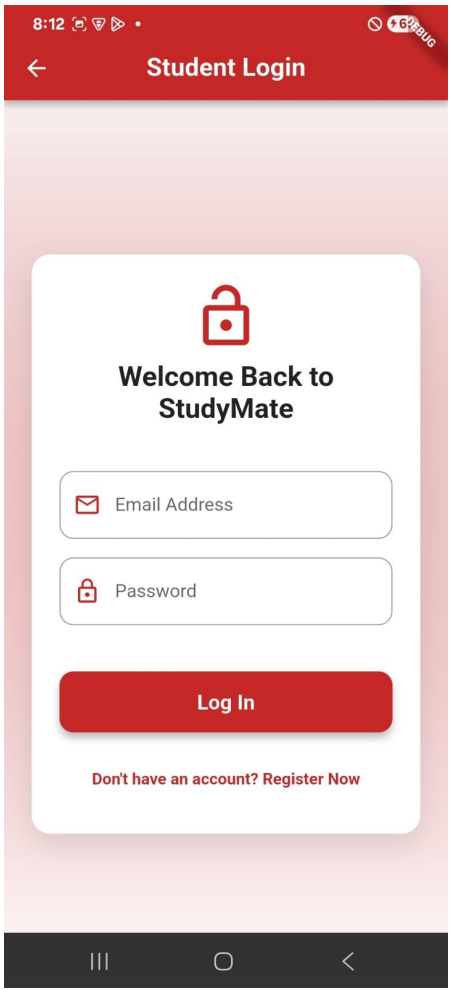


Figure 2 Login

4.2 The Workspace

4.2.1 Dashboard (Main Hub)

Purpose: The central landing page providing quick access to all application features.

- Profile Header: Displays the user's name and department. Tapping this area opens the User Profile screen.
- Feature Grid: A scrollable layout of cards representing the main tools:
  - Pomodoro: Focus timer tool.
  - Schedule: Weekly timetable view.
  - Tasks: To-Do list manager.

- Help Forum: Community request board.
- My Inbox: Private chat conversations.
- Note Taking: Digital notebook.

· Sign Out: A button at the bottom securely logs the user out and returns them to the Login screen.

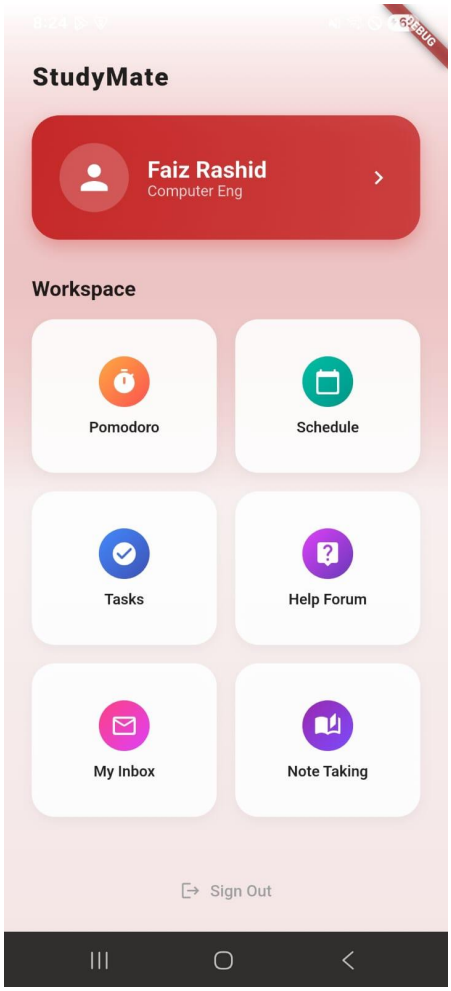


Figure 3 Dashboard

4.2.2 User Profile

Purpose: Shows the student's academic and personal details.

· Academic Info: Displays the student's Year and Department.

- Enrolled Courses: Small “tiles” show the courses enrolled for the semester for e.g. (CNG 495, CNG 463, CNG 465).
- Navigation: A back arrow redirects to the Dashboard.

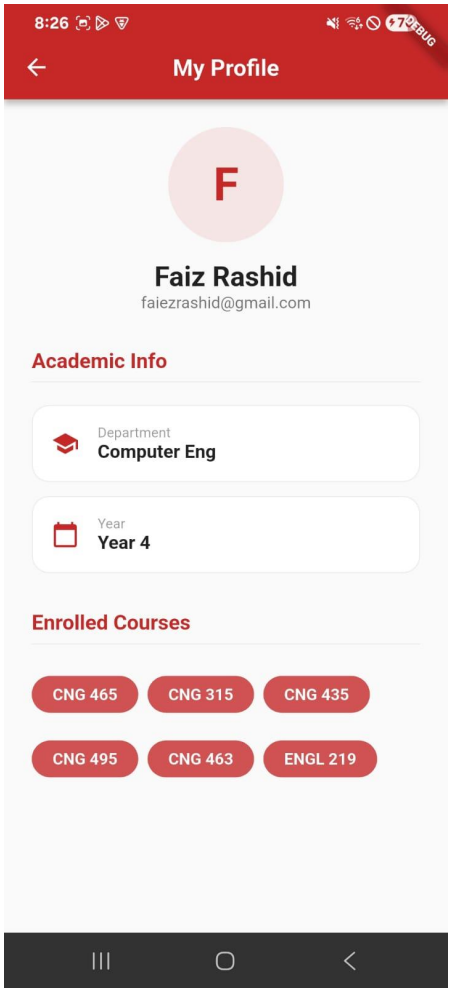


Figure 4 Profile

4.3 Productivity Features

4.3.1 My Schedule

Purpose: A schedule for tracking weekly classes.

- Weekly Grid: Shows a read-only view of the week (Mon–Sun) with time slots on the left. Classes appear as colored blocks showing the Course Code and Room Number.
- Add Class: A red Floating Action Button (+) opens a dialog to add new sessions. Users input the Course, Room, Day, and Time.

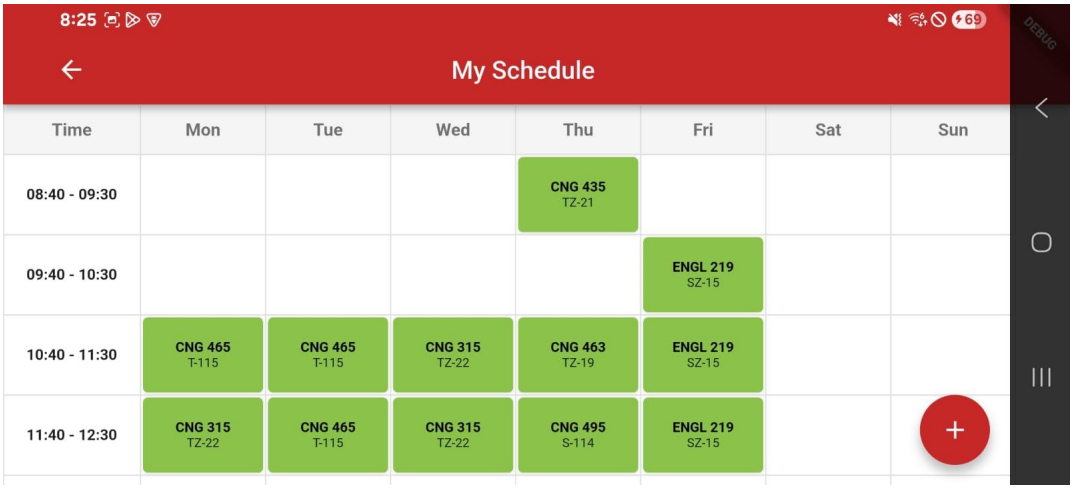


Figure 5 Schedule

4.3.2 To-Do List

Purpose: Tracks assignments and deadlines.

- Task Input: A text field allows quick entry of task names. A date picker allows users to assign specific Due Dates.
- Task Management:
  - Completion: Allow users to mark tasks as done using checkboxes.
  - Deletion: Users can swipe a task card left or right to delete the task.

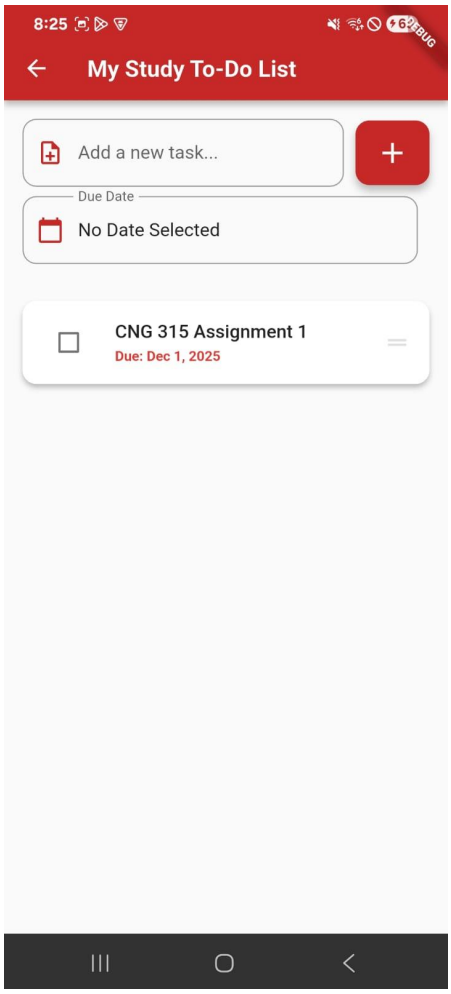


Figure 6 ToDo List

4.3.3 Pomodoro Timer

Purpose: A Pomodoro technique (work/break time) tool which lets users focus.

- Timer Display: Displays the phase you are in ("Focus Time" or "Break Time") and a countdown clock.
- Controls:
  - Start/Pause: Toggle button to turn the timer on or off.
  - Reset: Restarts the session to the full duration.
- Customization: Users can adjust the duration using simple plus/minus buttons for either “Study mode” or “Break mode”.

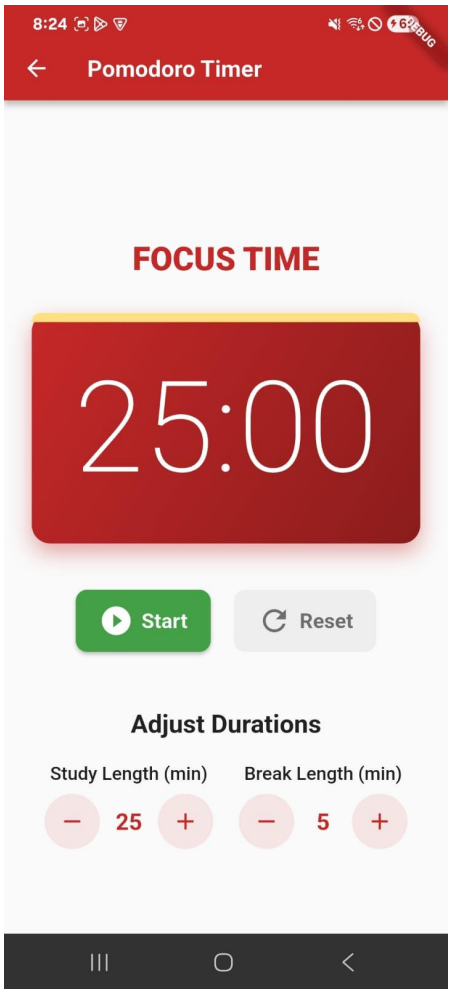


Figure 7 Timer

4.3.4 My Notebook

Purpose: A rich-text editor for saving study notes.

- Editor Interface: Provides fields for a Title and Note Body.
- Attachments:
  - Image Icon: Attaches photos (e.g., whiteboard snapshots) from the gallery.
  - PDF Icon: Uploads PDF documents to the note.
- Save: Stores the note and its attachments to the cloud database.

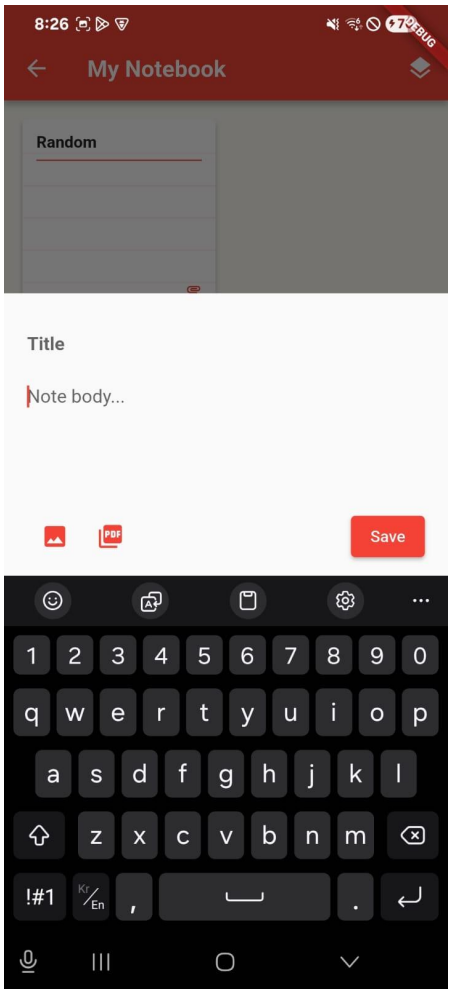


Figure 8 Notes

4.4 Communication Features

4.4.1 Student Help Forum

Purpose: A public board for academic assistance and collaboration.

- Forum Feed: Lists help requests posted by other students. Each card displays the Course Tag, Title, Description, and Date.
- Interactions:
  - "Message" Button: Immediately initiates a private conversation with the student who made the request.
  - The "Ask for Help" Button: Enables the user to submit their own assistance request.

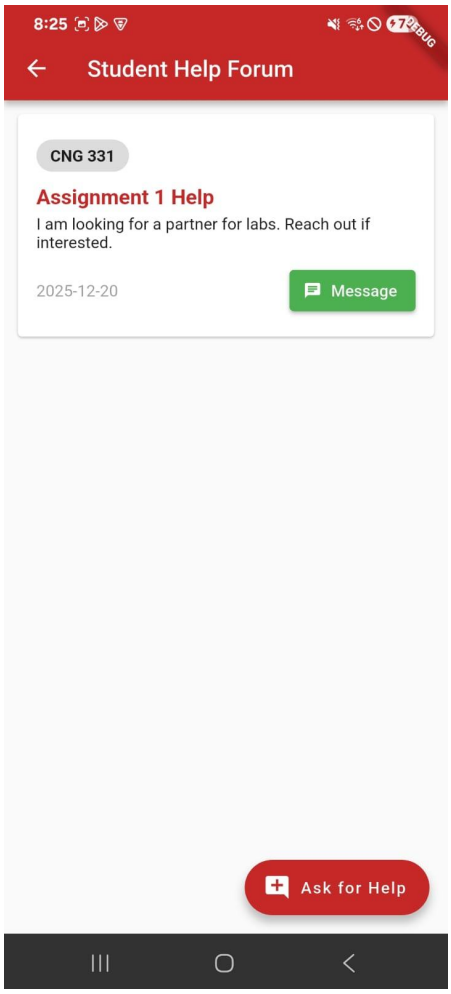


Figure 9 Help Forum

4.4.2 Inbox & Chat

Purpose: Peer to peer connection using private connection.

- Inbox List: Displays every conversation that is currently taking place. For improved organization, users can rename contacts by tapping the Edit (Pencil) icon (e.g., "Salar - Lab Partner").
- Chat View: Sending messages are shown in red bubbles on a typical messaging interface, while receiving messages are shown in grey. It has auto-scrolling and real-time delivery.

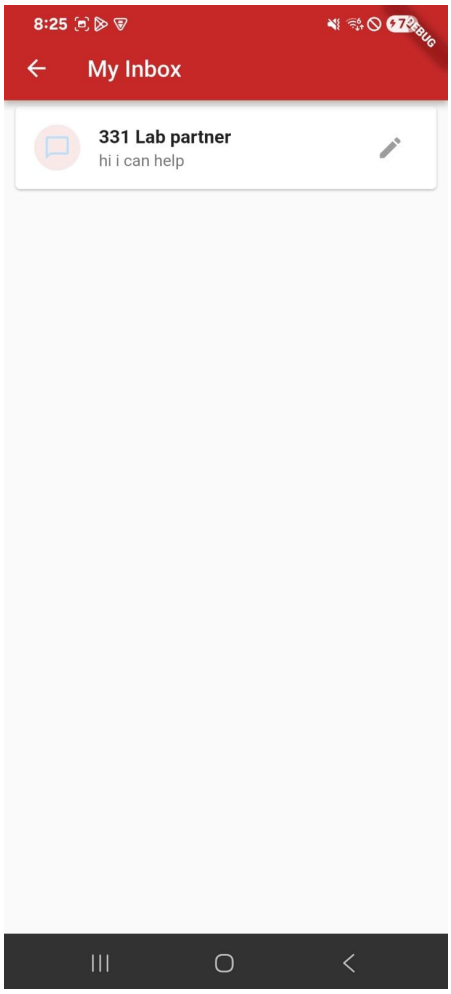


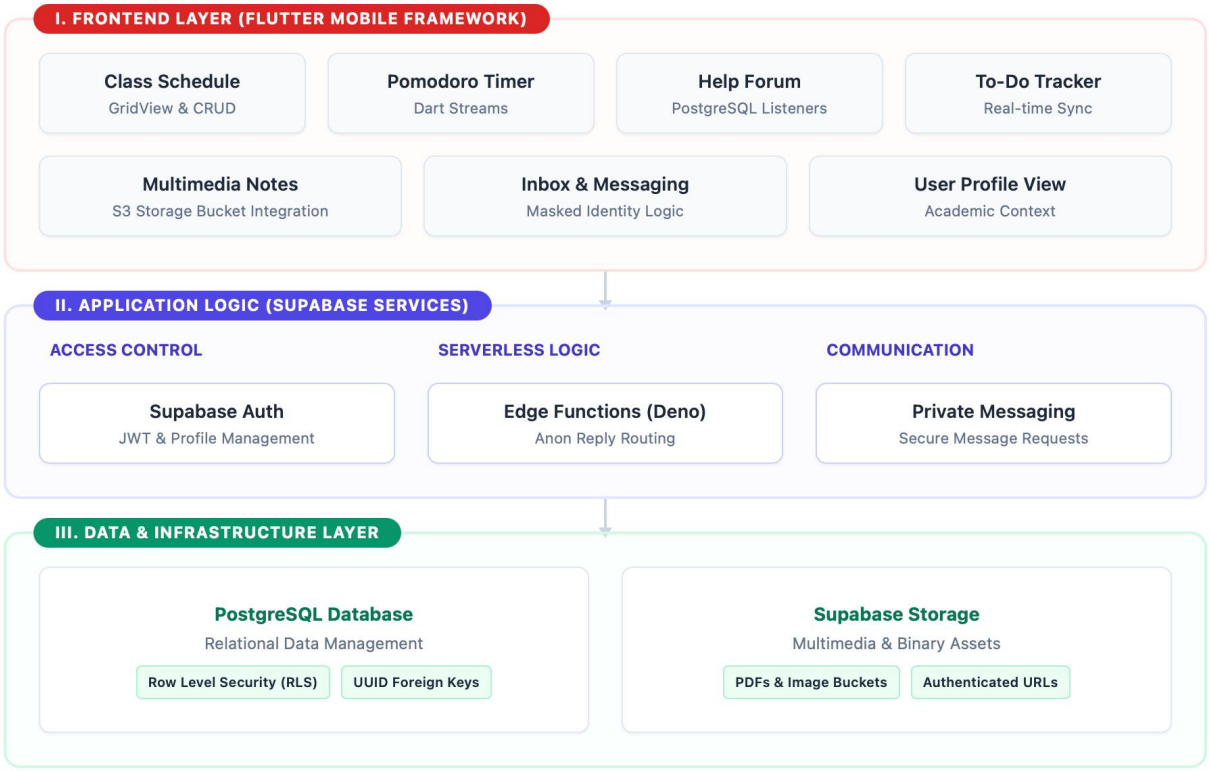
Figure 10 Inbox

5. Important Diagrams

This section illustrates the architectural design and logical flow of StudyMate, providing a visual blueprint of how the mobile client interacts with the cloud-backend.

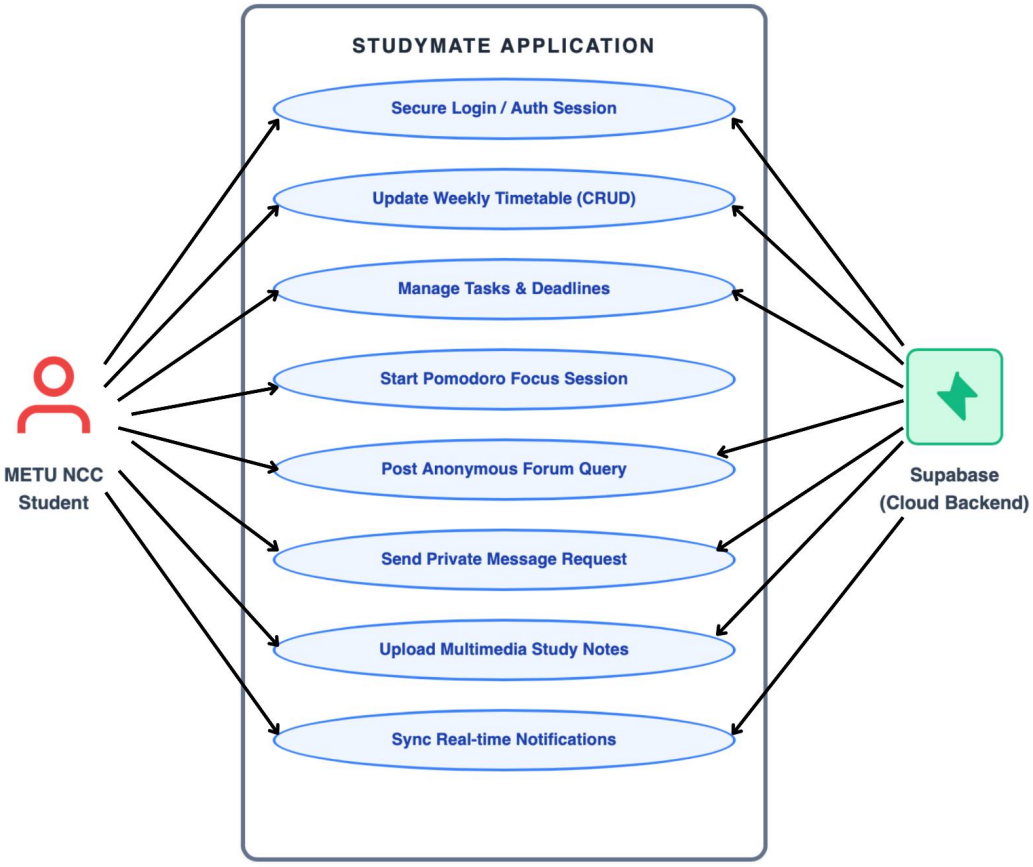
5.1 System Architecture Diagram

The System Architecture Diagram below showcases the structural components involved between the Flutter mobile interface and the Supabase cloud interface. It's centered on the decoupled aspect that exists within the services themselves. This includes all parts in terms of modules which are Authentication, Database, Storage.



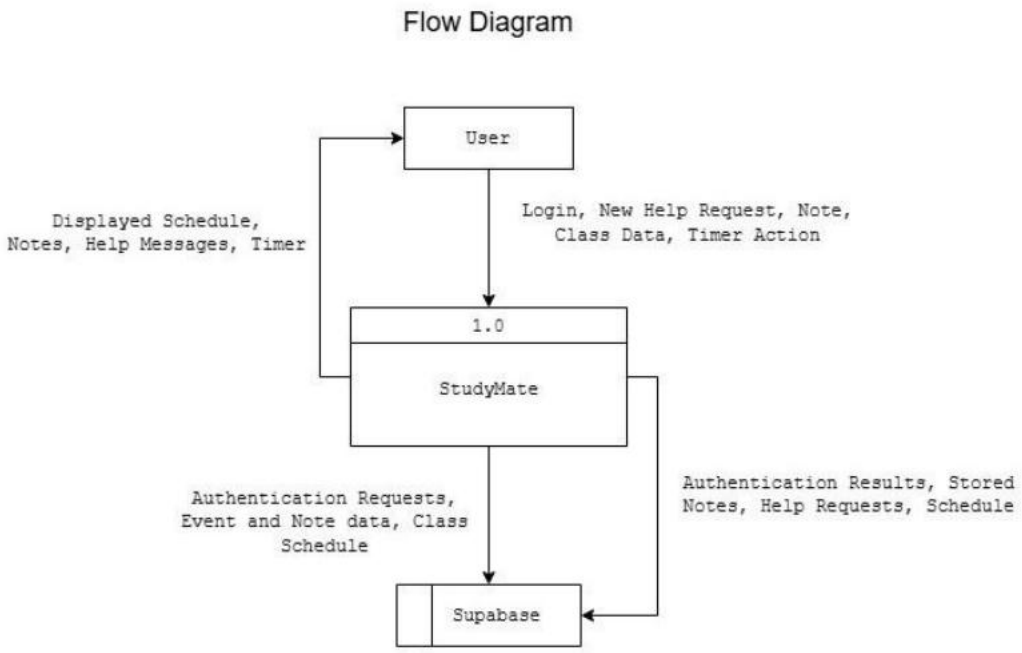
5.2 Use Case Diagram

The figure below illustrates the interactions between the Actors (User) and the System (StudyMate/Supabase).

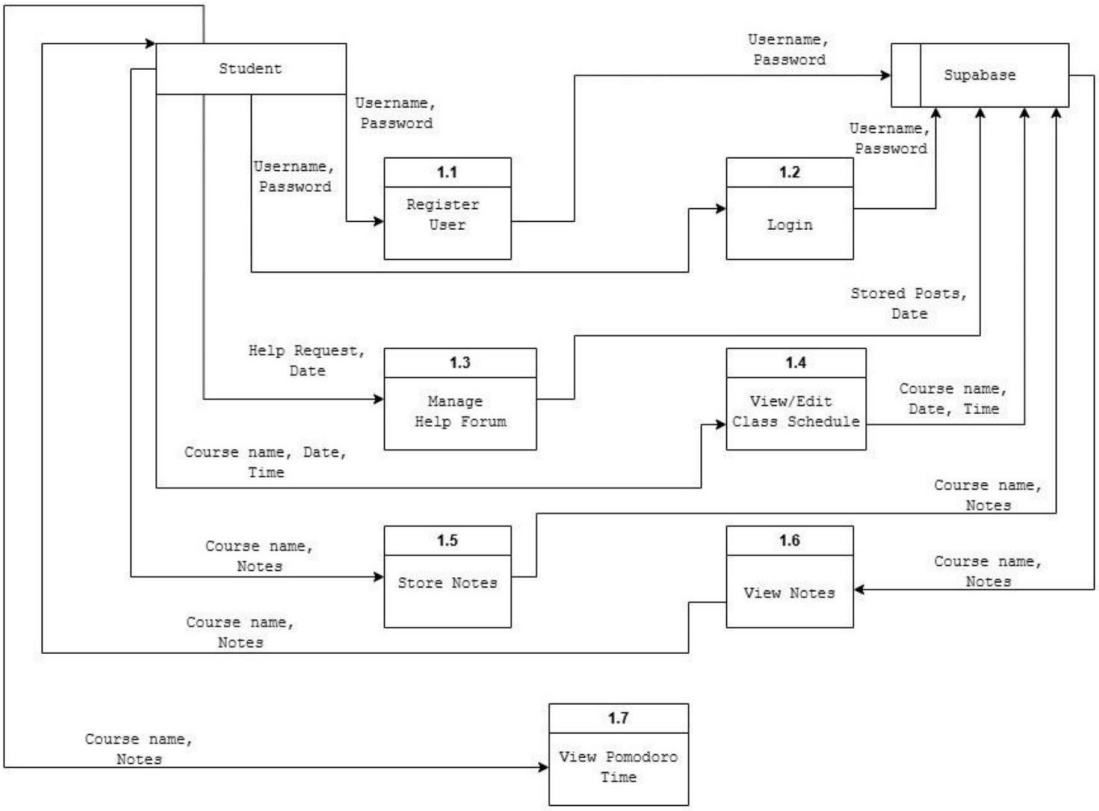


5.3 Data Flow Diagrams

The context-level diagram below shows how the user interacts with the Flutter app and its communication with the Cloud service.

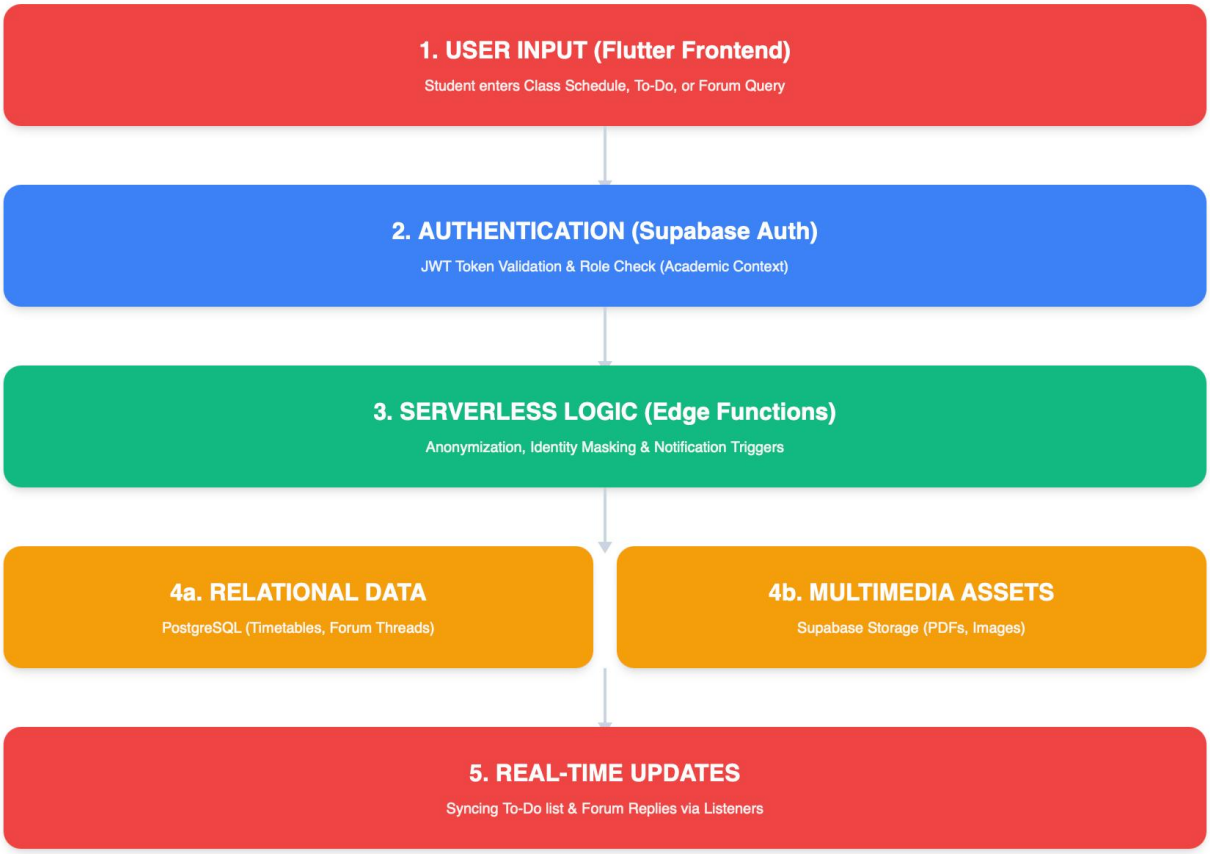


Now, the Level 1 data flow diagram is illustrated below which shows how different functions of our project handle data, like registering a new user account, using the notes forum or creating a Pomodoro timer.



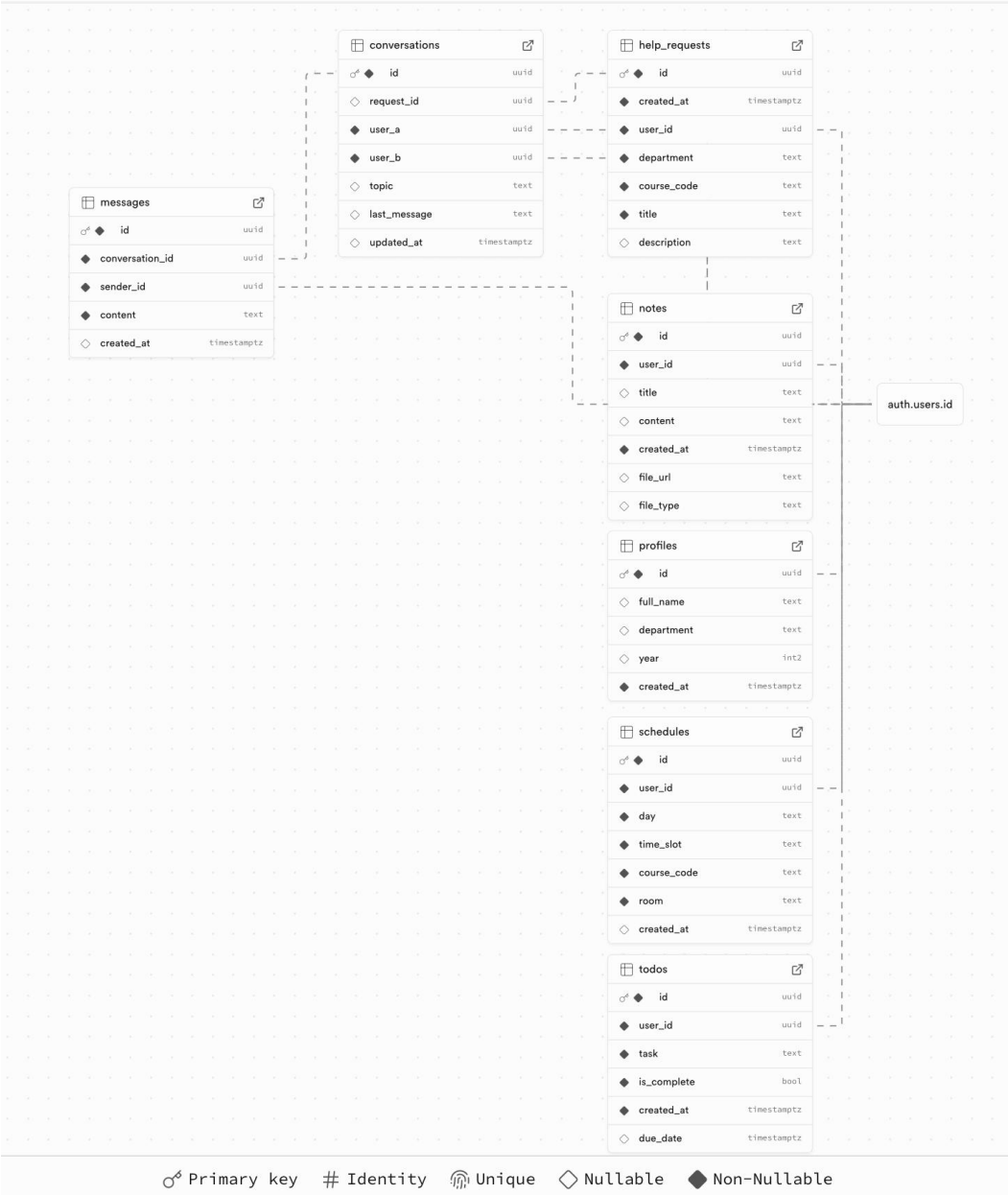
5.4 Information Flow Process Diagram

The Sequence Diagram illustrates the step-by-step chronological interaction between the Flutter app and Supabase during a live event.



5.5 Supabase Schema Visualizer

The Schema table is shown below.



6. Technologies Used

StudyMate features a BaaS architecture which integrates a high-performance frontend framework and a cloud native backend. The technologies used by us to develop our capstone project are explained in detail in this section of the report.

6.1 Programming Languages

6.1.1 Dart

This is the language in which the mobile application has been developed. It helps create a reactive interface for the application and contains all the logic of the application.

6.1.2 SQL (PostgreSQL)

We used SQL for the management of the database. Queries were entered in our supabase to develop the tables required. The code for this can be found in our Supabase.

6.1.3 JavaScript

This is used primarily for the Supabase edge functions which were needed for adding the anonymity of the users in the case of help forum and inbox messaging.

6.2 Frameworks and Libraries (APIs)

StudyMate is built on a modular framework that leverages specialized APIs to handle everything from user interface rendering to cloud synchronization.

6.2.1 Core Frameworks

Flutter SDK: The main UI component technology used in the construction of the compiled Android application. It applies the reactive widget-based state management paradigm.

Dart: This is an optimized programming language used on the client-side that performs all the logical operations, such as calculations for the Pomodoro Timer and asynchronous database operations.

6.2.2 Key dependencies

To extend the basic functions, the following external packages and APIs are used:

supabase_flutter: The main API client that's used for interactions on the backend. It facilitates interactions with the following components:

GoTrue API: This API is used for secure user authentication and management of sessions.

PostgREST API: For doing CRUD (Create, Read, Update, Delete) operations on PostgreSQL database tables such as schedules and todos.

google_fonts: Allows usage of the library of custom fonts, thus ensuring that the "StudyMate" brand is consistent across all screens.

path_provider & file_picker: These APIs are crucial for the Notes Module.

6.3 Development Tools

Git & GitHub: For Version Control, Coding, Collaborations, & to manage project documentation.

VS Code/Android Studio: These are the two major Integrated Development Environments (IDEs) that we employed in Dart/Flutter development of our application.

7. Task Distribution

The task was divided fairly among the two members. The details are listed below in this section.

7.1 Arisha Ahmad

- Backend Infrastructure: Responsible for the complete Supabase project setup and establishment of the connection between the Flutter application and cloud services.
- Authentication and Profiles: Developed the full stack logic for sign-up and login systems, including the design and backend integration for the Profile View screen.
- Core Productivity Modules: Implemented the UI and all backend logic for the To-Do List, Pomodoro Timer, and the Notes Section, including multimedia (PDF/Image) storage functionality.
- Dashboard Components: Designed and implemented specific modules of the centralized dashboard.

7.2 Faiez Rashid

- Collaboration Systems: Development of the Help Forum and the Inbox messaging system, managing both the UI and the real-time backend synchronization.
- Navigation & Dashboard: Developed the primary Dashboard interface and the grid-based navigation system to ensure a seamless user experience between modules.
- Class Organization: Implemented the Class Schedule module, including the scrollable grid view and the backend logic for managing course data.

8. Project Statistics

Category	Detail	Technical Description & Responsibilities
Programming Languages	Dart, SQL, TypeScript	Dart: Powers the Flutter frontend and app logic. SQL: Defines database schema and RLS policies. TypeScript: Used for custom Edge Functions
Lines of Code	4,000 – 6,000 lines	Includes all Dart widgets, state management logic, and SQL backend scripts across the shared repository.
Memory Requirements (Runtime)	120MB – 200MB RAM	Target memory footprint on Android to maintain stable 60fps performance during resource intensive tasks like PDF viewing.
Memory Requirements (Development)	8GB – 16GB RAM	Recommended RAM for running IDEs (VS Code/Android Studio) and mobile emulators simultaneously during the 12-week development cycle.
Database System	PostgreSQL (v15+)	Managed by Supabase.

9. References

Our GitHub Repo: https://github.com/arishaahmad/study-mate-application.git

[1] Middle East Technical University Northern Cyprus Campus. (2025). *CNG 495 Capstone project guidelines*.

[2] Supabase. (n.d.). *Supabase documentation*. https://supabase.com/docs

[3] Flutter. (n.d.). *Flutter documentation*. https://flutter.dev/docs

[4] PostgreSQL Global Development Group. (n.d.). *PostgreSQL documentation*.

https://www.postgresql.org/docs/