



SCREEN SKETCHES

PREPARED FOR

COMS 3090

Iowa State University

Fall 2024

PREPARED BY 3_roy_2

Autrin Hakimi

Nhat Le

Tam Minh Nguyen

Bach Nguyen

1. Actors

- **Administrators**
 - Manage student and teacher accounts.
 - Edit and oversee class schedules.
 - Control access to various app functionalities.
 - Send system-wide announcements and urgent messages.
- **Teachers/Managers**
 - Access and modify course schedules.
 - Communicate with students through announcements and messages.
 - Manage course-related materials and resources.
- **Students/Users**
 - View and manage personal class schedules and calendars.
 - Communicate with peers through real-time chat and study groups.
 - Access campus maps, building information, bus schedules, and weather updates.
 - Explore and review local amenities such as restaurants and entertainment centers.

2. Non-functional Requirements

1. **Multi-users:** The application should support different types of users.
2. **Performance:** Ensure that the app responds to user interactions within a certain timeframe, and this should be less than 2 seconds
3. **Resource Utilization:** Optimize the use of server and client resources, including memory usage and data transfer rates, to maintain a smooth experience
4. **Data Fetching:** Any external API data (maps, weather) should be fetched and displayed within 1 second of the request.
5. **Calendar Sync:** Users should be able to sync their schedules with external calendar apps (Google Calendar, Outlook, etc.) without data loss or errors.
6. **Authentication:** The app must use ISU's **NetID** for authentication to ensure only authorized users (students, teachers, administrators) can access the platform.
7. **Data Privacy:** The system must comply with relevant data privacy standards (e.g., FERPA) to ensure user data is protected and cannot be accessed by unauthorized personnel.
8. **Session Management:** Sessions should automatically expire after 30 minutes of inactivity to prevent unauthorized access on shared devices.
9. **Ease of Use:** The app should follow UI/UX best practices to be intuitive, requiring no more than 5 clicks to access key features (e.g., schedule, maps).

- 10. Learnability:** New users should be able to navigate the app's main features (schedule, maps, announcements) without needing external instructions or a manual.
- 11. Modularity:** Design the system with clear modularity to simplify updates and maintenance.
- 12. Documentation:** The system must include well-documented code and a developer guide to ensure smooth handover and future maintenance by other teams.
- 13. Real-time capabilities:** Websocket products need to ensure real-time capabilities and handle network connections seamlessly. For example, what if a student gets disconnected and then reconnected?
- 14. Testing:** The system should have at least 90% unit test coverage, with automated tests running for each new build to catch errors early.

3. Tables and Fields

1. Users Table

- `user_id` (Primary Key)
- `username`
- `email`
- `phone_number`
- `hashed_password`
- `active` (boolean, whether the user account is active)
- `last_login_time`
- `role` (admin, teacher, student)
- `net_id` (for ISU authentication)
- `profile_picture_url` (optional)
- **Notes:** This includes both identification and account status information. It's essential for distinguishing roles.

2. Courses Table

- `course_id` (Primary Key)
- `course_name`
- `description`
- `credits`
- `department_id` (Foreign Key, references Department table)
- **Notes:** Courses will be assigned to departments, making it easier to manage them by academic departments.

3. Class Sections Table

- `section_id` (Primary Key)
- `course_id` (Foreign Key, references `course_id`)
- `term` (Fall, Spring, Summer)
- `year`
- `schedule_id` (Foreign Key, references `schedule_id`)
- `professor_name`
- `start_time`
- `end_time`
- **Notes:** This table links specific sections of a course to a professor, schedule, and time.

4. Schedule Table

- `schedule_id` (Primary Key)
- `user_id` (Foreign Key, references `user_id`)
- `class_section_id` (Foreign Key, references `section_id`)
- `event_name`
- `start_time`
- `end_time`
- `recurring` (boolean, whether it's a recurring event)
- **Notes:** Users' individual schedules, linking class sections to time slots. Recurring events make it flexible for repetitive events like classes.

5. Tasks (Assignments) Table

- `user_id` (Foreign Key, references `user_id`)
- `course_id` (Foreign Key, references `course_id`)
- `due_date`
- **Primary Key:** (`user_id`, `course_id`)
- **Notes:** This table stores assignments or tasks associated with specific courses, personalized per user.

6. Enrollments Table

- `enrollment_id` (Primary Key)
- `student_id` (Foreign Key, references `user_id`)
- `course_id` (Foreign Key, references `course_id`)
- `enrollment_date`
- **Notes:** Tracks which students are enrolled in which courses.

7. Messages Table

- `message_id` (Primary Key)
- `sender_id` (Foreign Key, references `user_id`)
- `recipient_id` (Foreign Key, references `user_id`)
- `group_id` (Foreign Key, optional, for group messages)
- `content`
- `timestamp`
- **Notes:** This stores all messages, allowing for both individual and group messaging.

8. Announcements Table

- `announcement_id` (Primary Key)
- `sender_id` (Foreign Key, references `user_id`)
- `recipient_role` (student, teacher, or admin)
- `content`
- `timestamp`
- **Notes:** Manages announcements that are sent out by admins or teachers.

9. Study Groups Table

- `group_id` (Primary Key)
- `group_name`
- `creator_id` (Foreign Key, references `user_id`)
- `created_at`
- **Notes:** Study groups formed by students, with the ability to add members and chat.

10. Group Members Table

- `group_member_id` (Primary Key)
- `group_id` (Foreign Key, references `study_groups` table)
- `user_id` (Foreign Key, references `user_id`)
- `joined_at`
- **Notes:** Tracks group membership for study groups.

11. Bus Schedules Table

- `bus_id` (Primary Key)
- `route_name`
- `stop_name`

- **arrival_time**
- **departure_time**
- **Notes:** Manages bus routes and schedules.

12. Map Locations Table

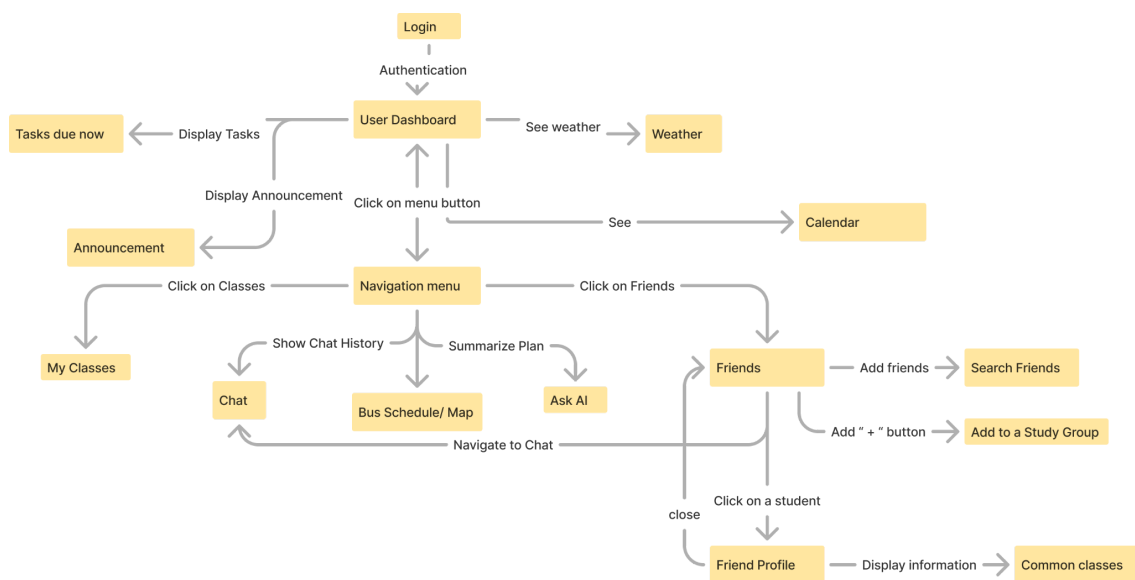
- **location_id** (Primary Key)
- **building_name**
- **room_number**
- **latitude**
- **longitude**
- **operating_hours**
- **Notes:** Provides detailed building and room location data for navigation.

13. Weather Data Table

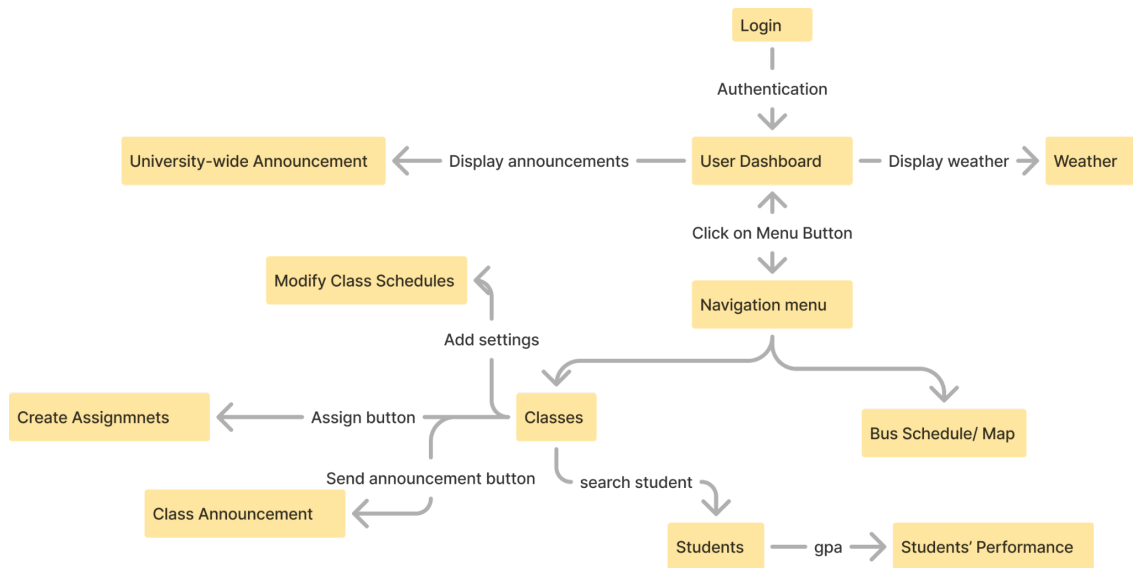
- **weather_id** (Primary Key)
- **location**
- **date**
- **temperature**
- **conditions**
- **Notes:** Stores weather data for displaying real-time conditions.

4. Screen Flow Diagram

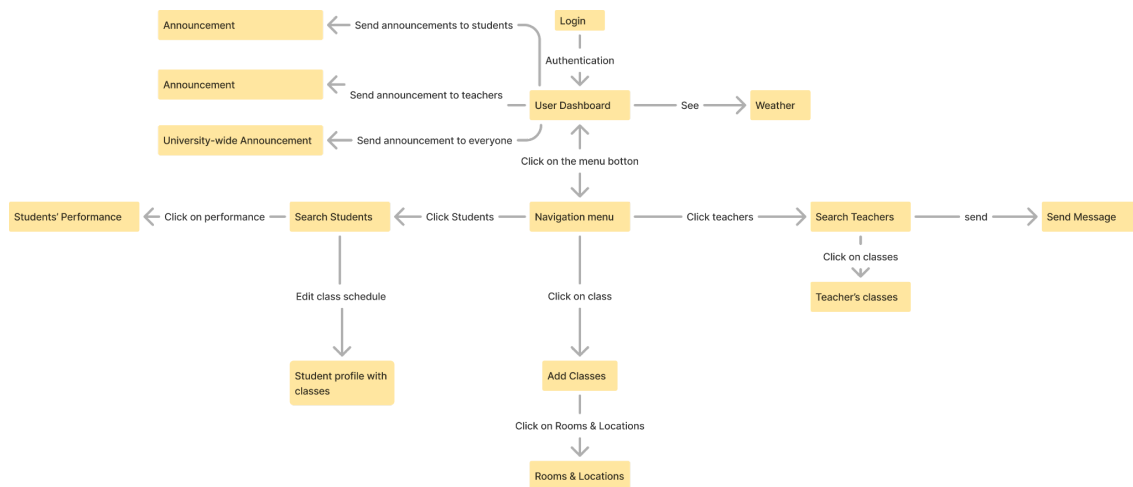
a) Student's View



b) Teacher's View

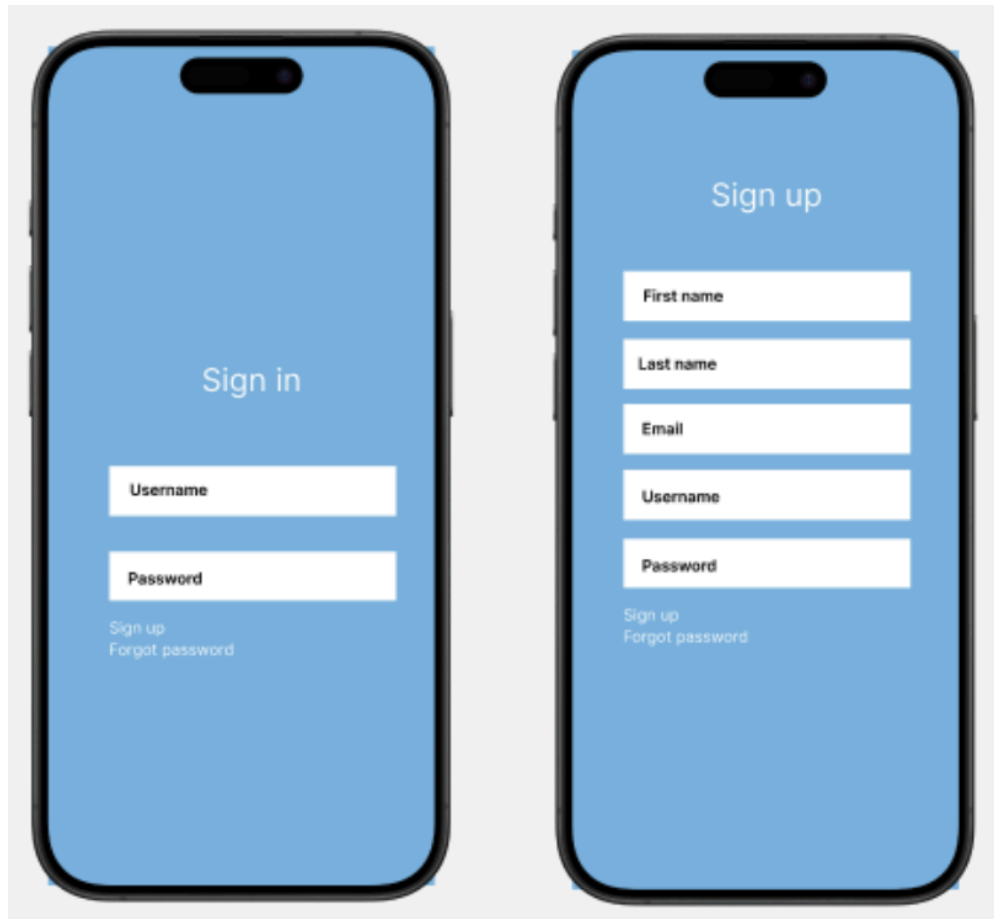


c) Administrator's View



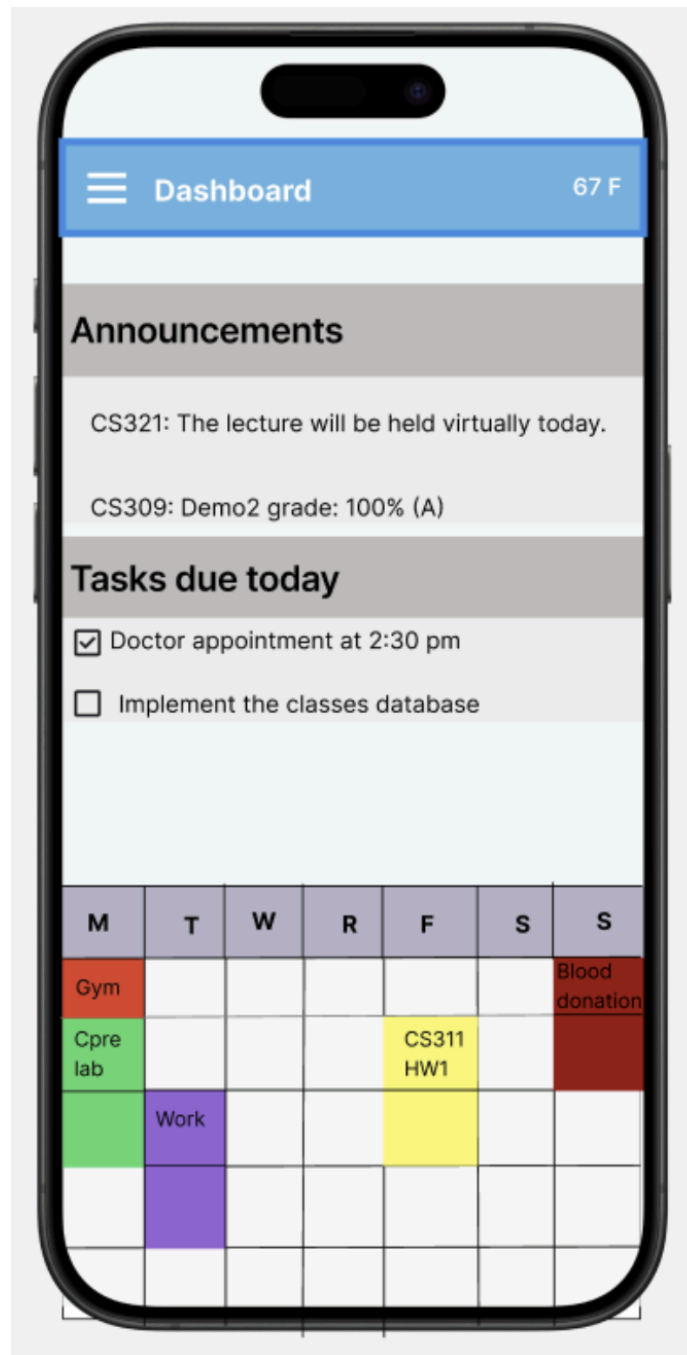
5. Screen Sketches

a) Sign in - Sign up - Forgot password



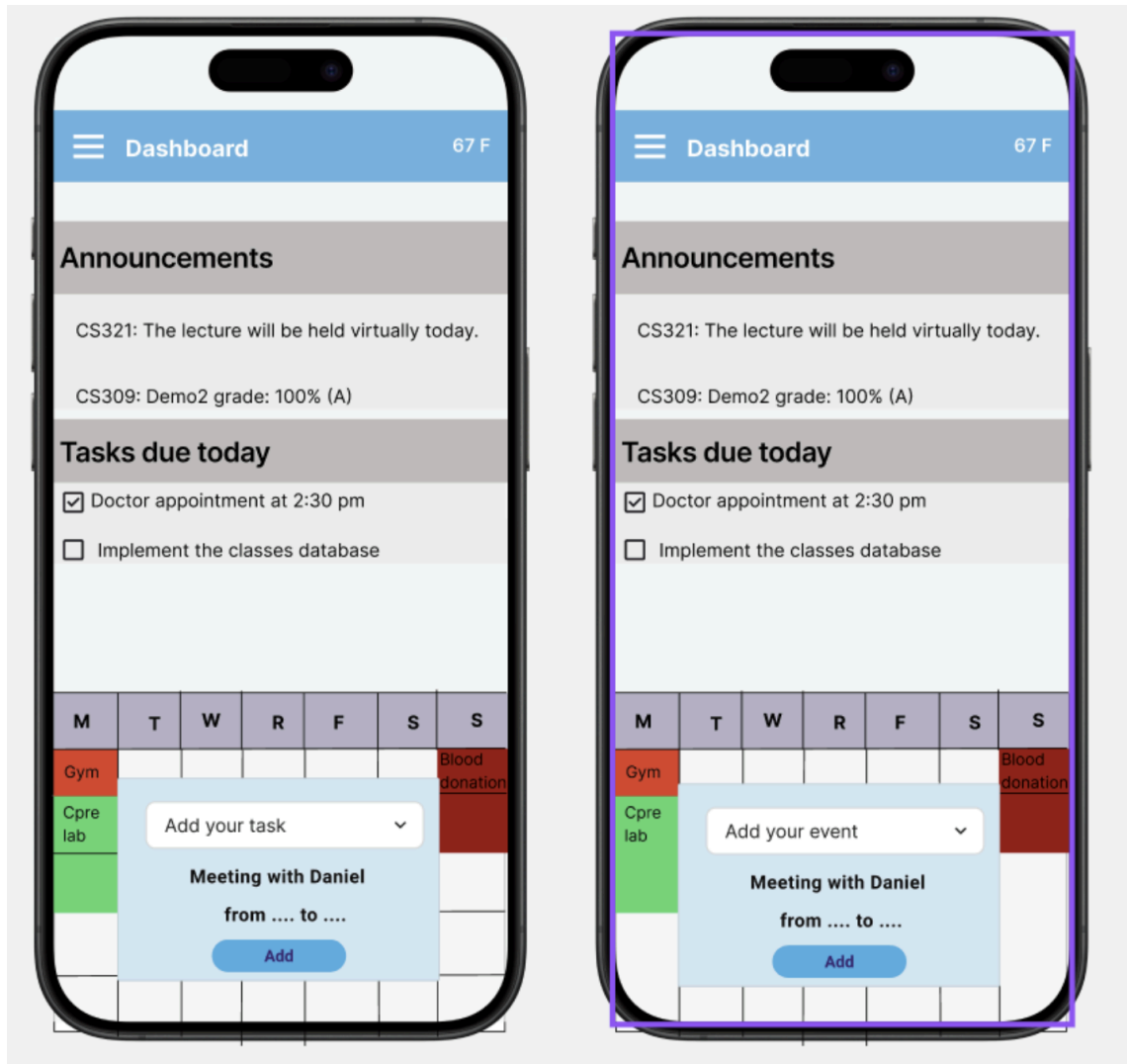
The initial screen of our application is the **Sign-In page**, providing users with a seamless entry point into ISU Pulse. Upon launching the app, users are directed to this page, where they can log in with their credentials. A "Sign-Up" button is also available, which navigates users to the registration page for account creation. Please note that while the sketches provide a basic visual guide, the final design of ISU Pulse will feature a significantly more polished and user-friendly interface, enhancing the overall user experience.

b) Dashboard

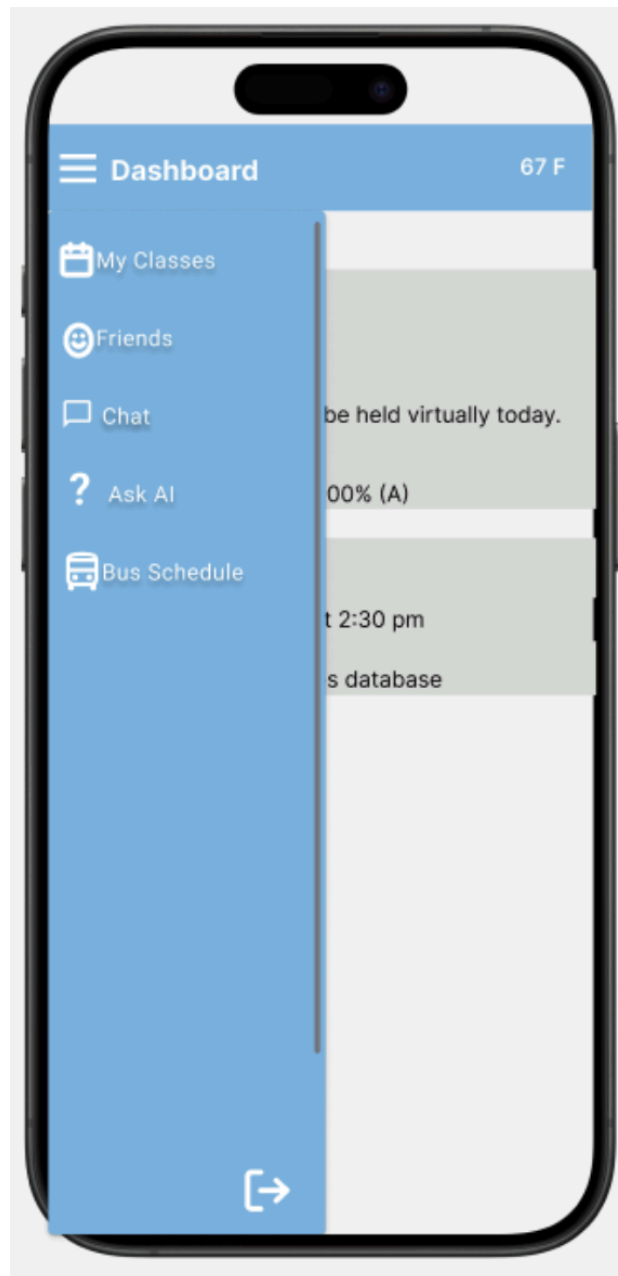


Once signed in, users are directed to the **Dashboard**, which serves as the central hub for key information. Here, users can view their current tasks, important announcements, and local weather updates at a glance. Tasks can easily be marked as complete by clicking the checkbox next to each task.

At the bottom of the screen, a weekly calendar is displayed, showcasing upcoming events and tasks, providing a clear overview of the user's schedule for the week.

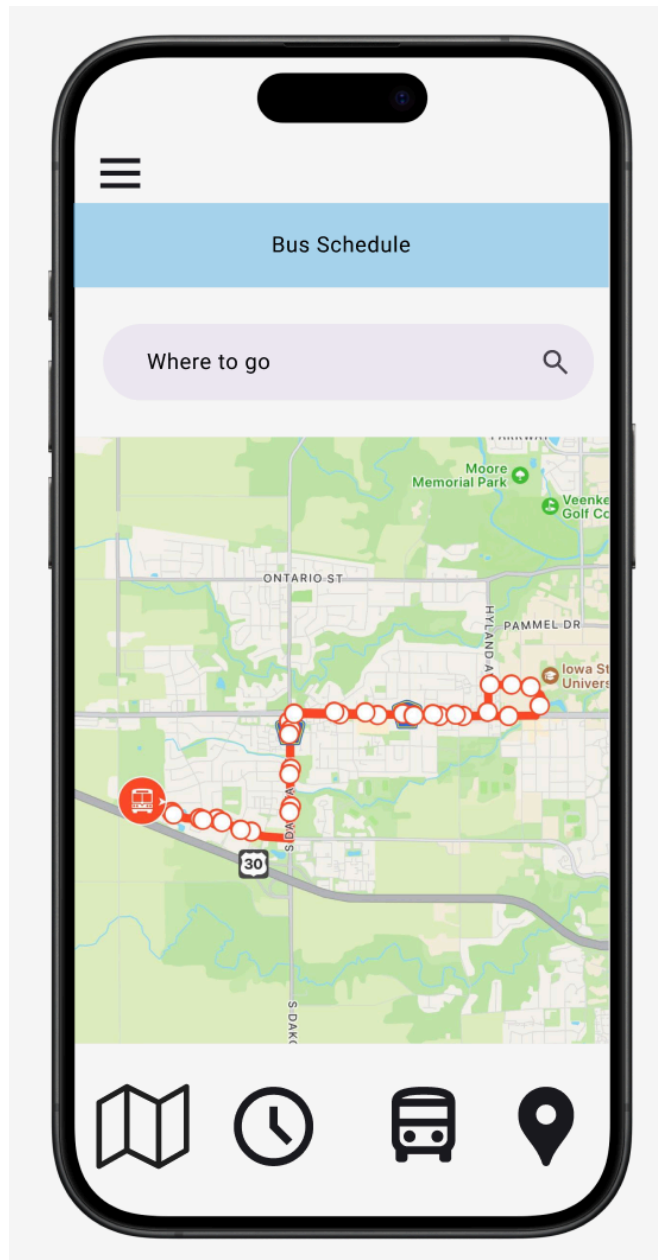


Users can easily create new events or tasks by clicking anywhere on the calendar. This action triggers a flyout or modal, where they can select between an event or task using a dropdown menu. After specifying the necessary details, such as the time and description, the user can add the item to their calendar by clicking the 'Add' button.



Our application includes a convenient **navigation menu** for easy access to different features. By clicking the menu icon (the three stacked lines), the navigation drawer slides out, allowing users to select the page they want to navigate to. The available options include: "**My Classes**," "**Friends**," "**Chat**," "**Ask AI**," and "**Bus Schedule**." This ensures users can quickly and intuitively move between key sections of the app.

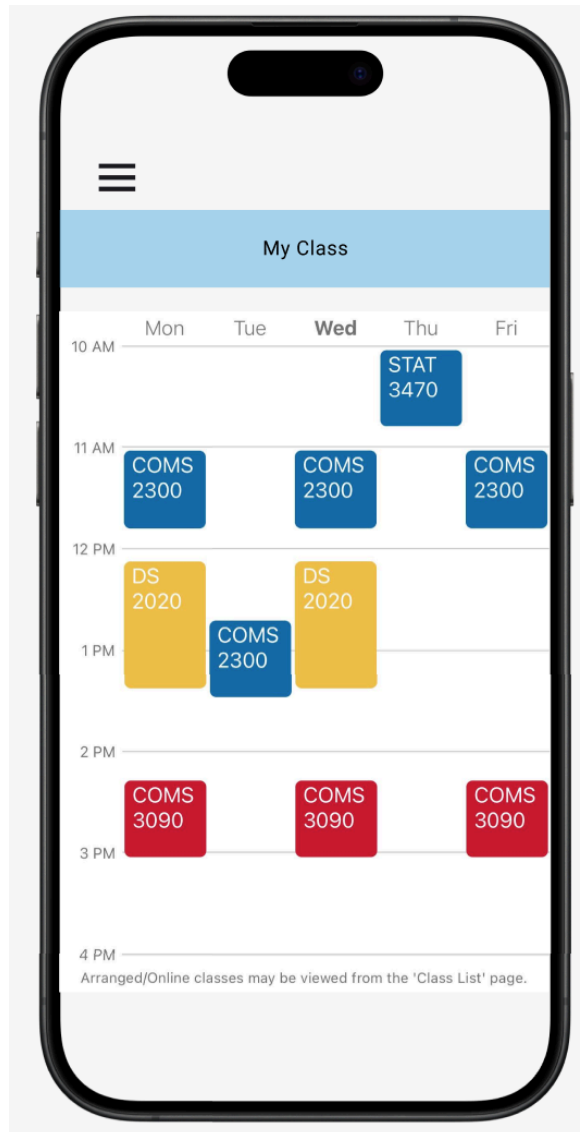
c) Bus Schedule



The **"Bus Schedule"** view helps users plan their routes efficiently by displaying bus routes on an interactive map. Users can search for their destination, and the corresponding bus route will be highlighted for easy navigation. To access this feature, simply select the **"Bus Schedule"** option from the navigation menu.

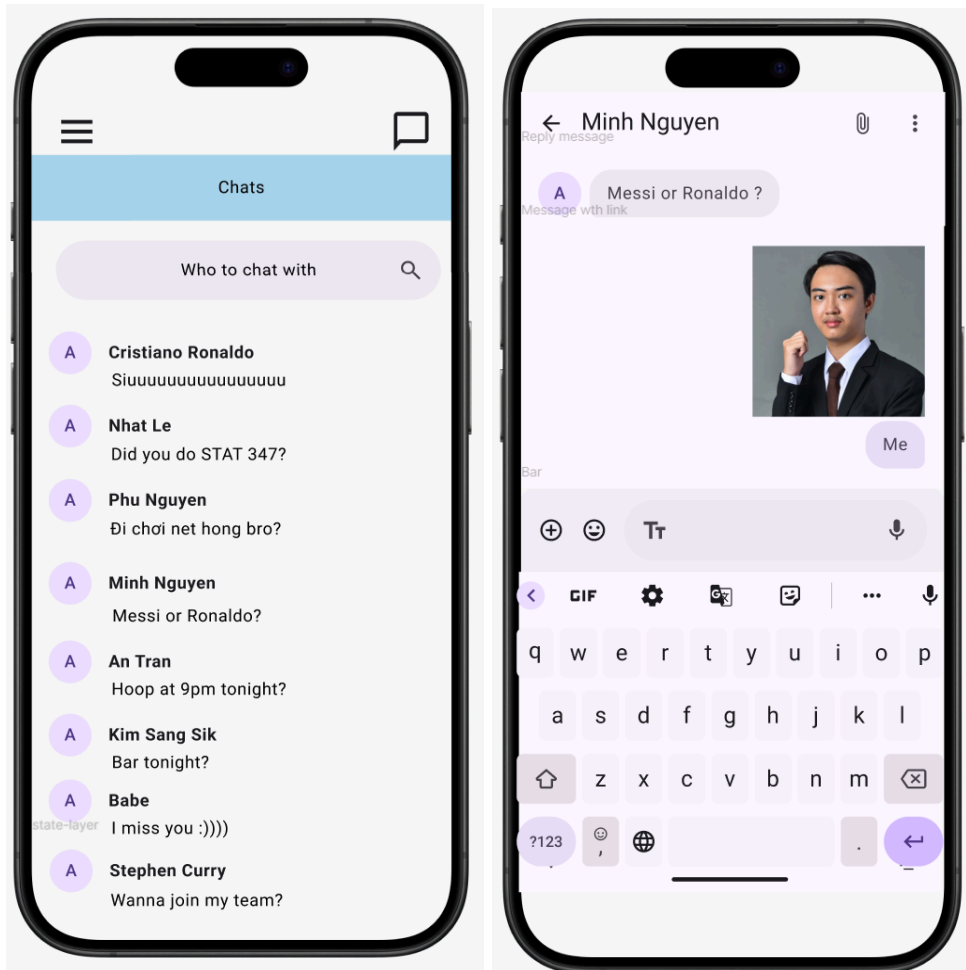
Students can pinpoint their current location and enter their destination in the search bar to find the most relevant bus route to reach their destination smoothly.

d) My Classes



The **"My Classes"** view displays your weekly class schedule in a clean and organized format. Classes are shown as blocks arranged by day and time, making it easy to visualize your schedule. To access your schedule, navigate to the **"My Classes"** section from the navigation menu. Each day of the week is clearly divided, with classes color-coded for quick identification. This allows you to easily see which classes are scheduled for each day, helping you manage your time efficiently.

e) Chat



The **"Chats"** view presents a list of people you've messaged, making it easy to continue conversations or start new ones. You can also search for specific contacts to initiate a chat. To access this view, navigate to the **"Chat"** section from the main menu. You'll see a list of recent contacts, and to start a conversation, simply click on the avatar of the person you'd like to message.

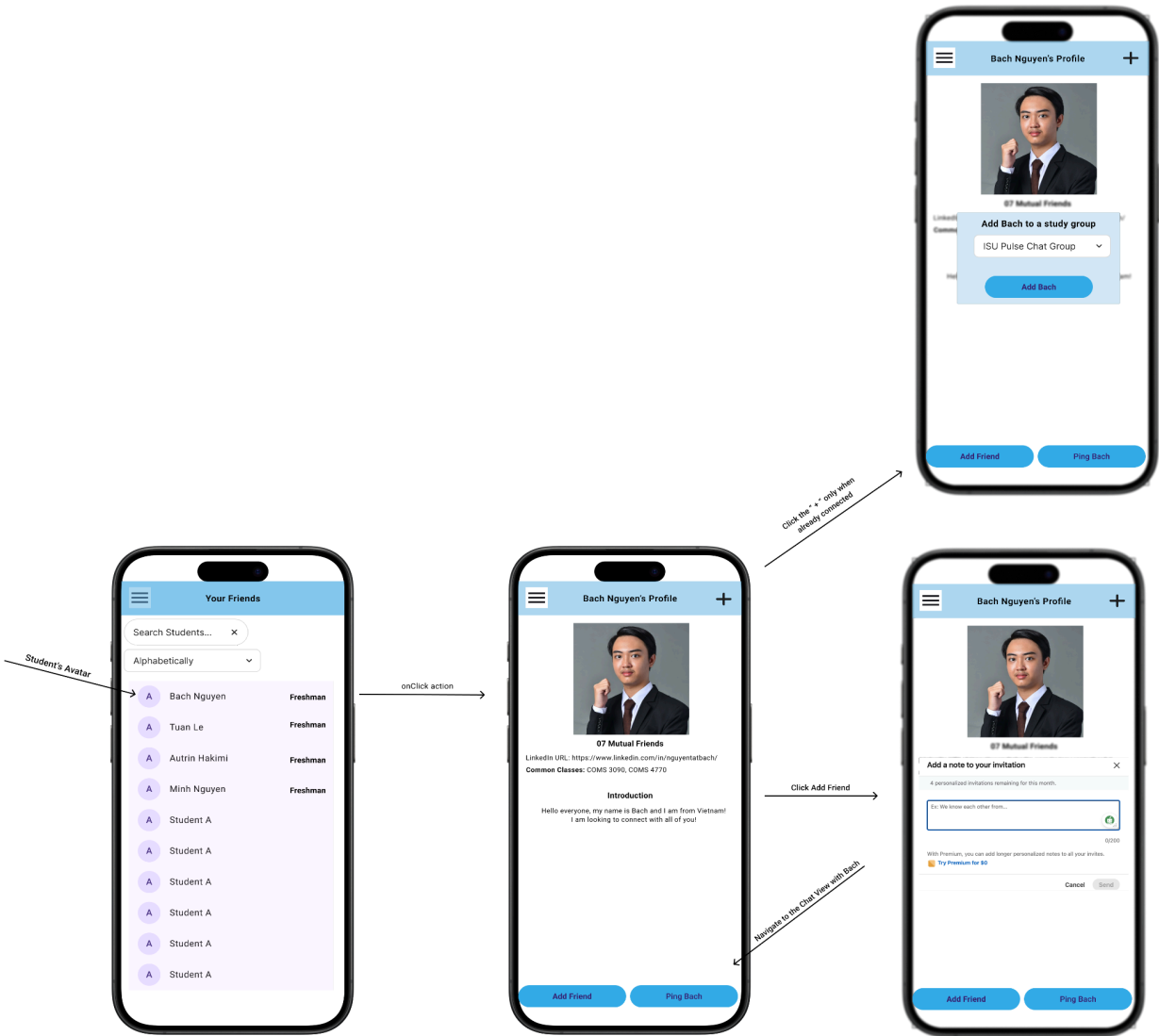
f) Ask AI



The "**AI Assistant**" view provides a text input field for users to interact with the AI. Users can type their questions and submit them using the on-screen keyboard. To access this view, navigate to the "**AI**" section from the navigation menu. Simply type your question into

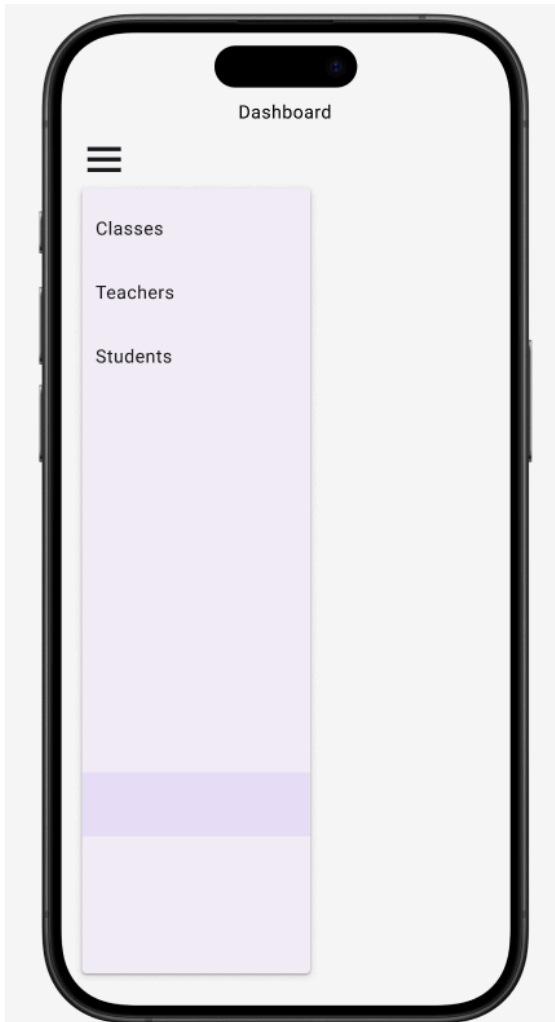
the input box, and when ready, send it by pressing the "send" button on the keyboard. The AI will respond to assist with your queries.

g) Friends



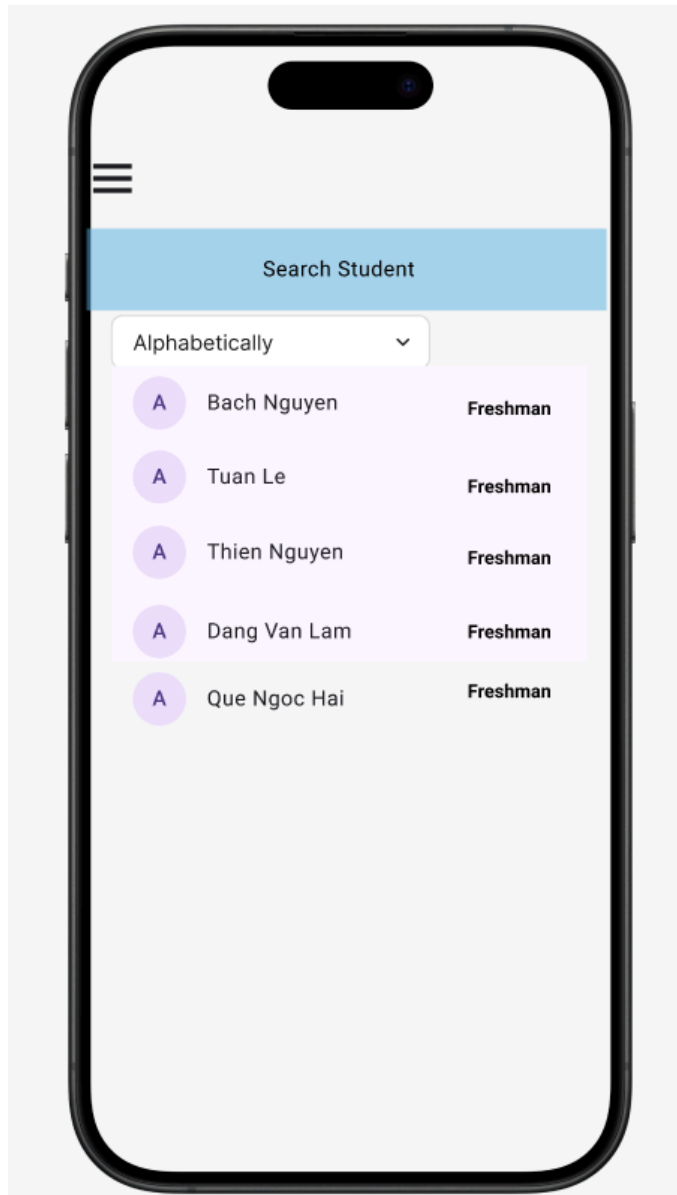
The **"Friends"** view displays a list of your friends and allows you to search for other students using the search box. From a student's profile page, you can either connect with them or send a message to those who have already become friends, which will redirect you to the **Chat** view for easy communication.

h) Administrator's view



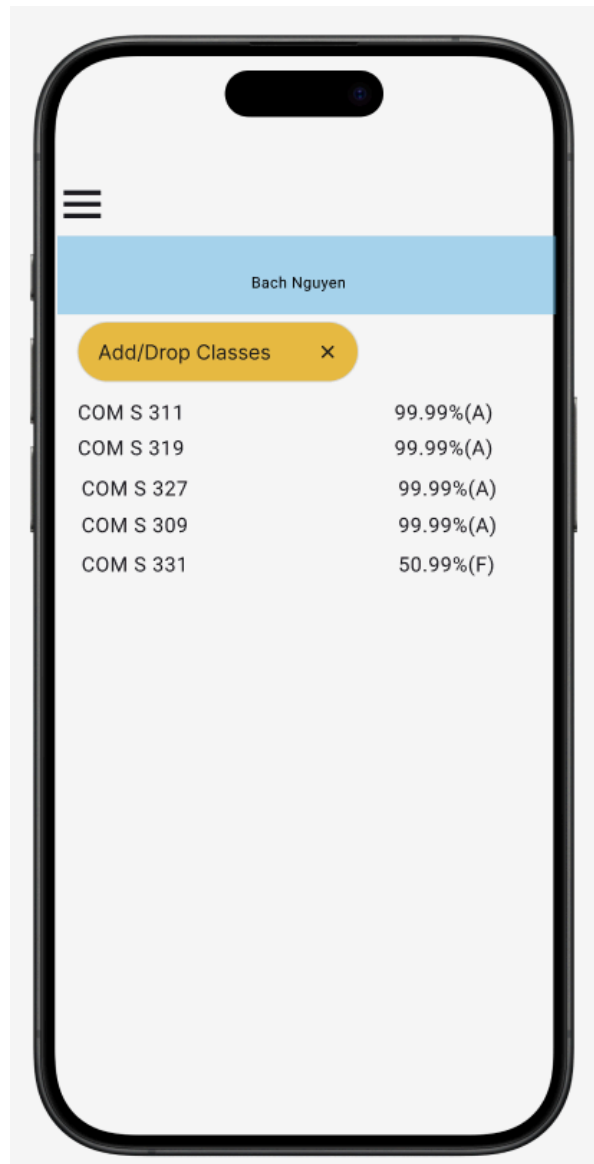
Administrators have a more streamlined, minimalistic interface. When they click the menu button, a navigation bar appears, providing access to manage **C**lasses, **T**eachers, and **S**tudents. This simplified layout ensures that administrators can efficiently navigate and manage essential functions.

i) Search Student



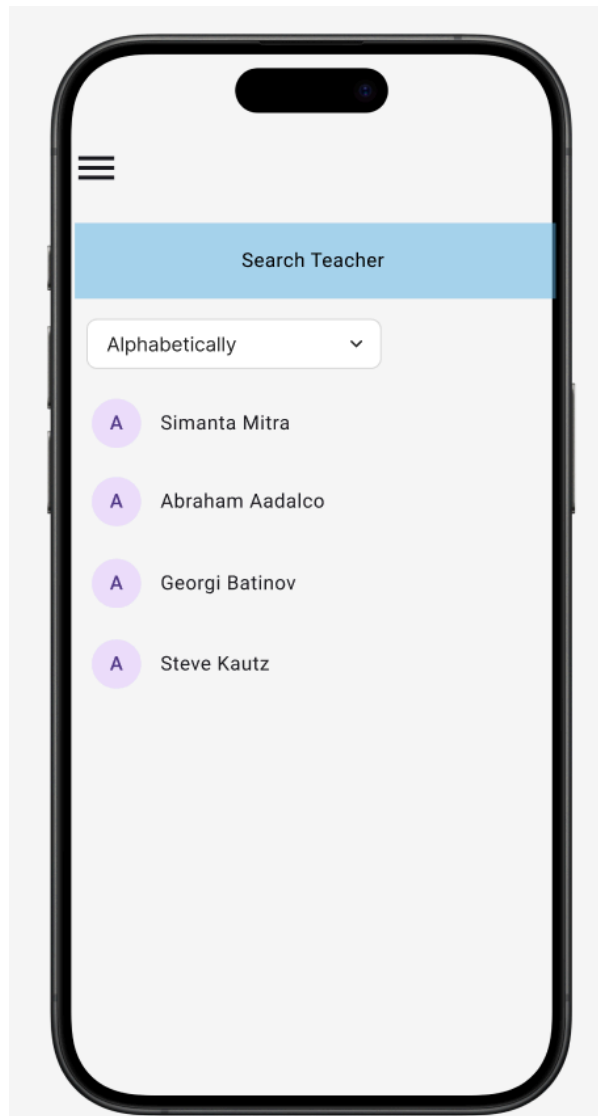
When administrators select **Students** from the menu, they are directed to a screen where they can search for students enrolled at the university. Administrators can browse the list and view each student's class level, providing a clear overview of student enrollment and progress.

j) Student's profile



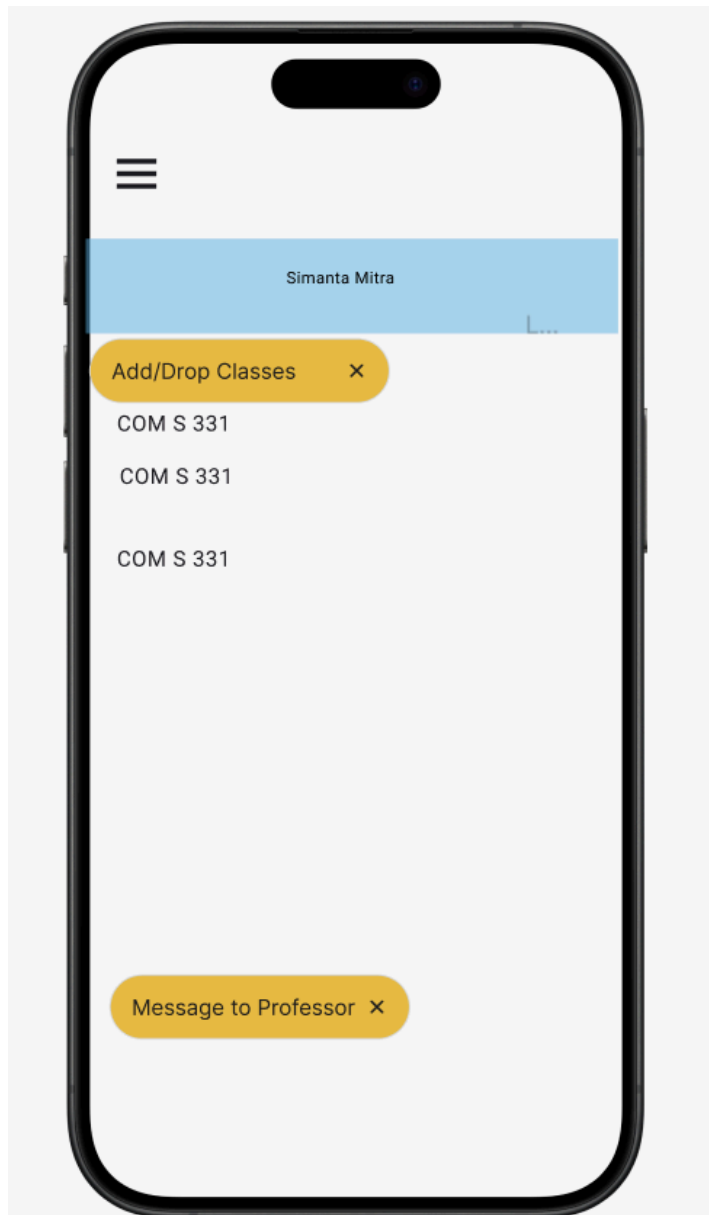
When administrators click on a student's name, they are taken to the **Student Profile** screen. Here, they can view the student's current class grades and manage their enrollment. Administrators have the option to add or drop classes for the student, providing a convenient way to update their academic record.

k) Search Teacher



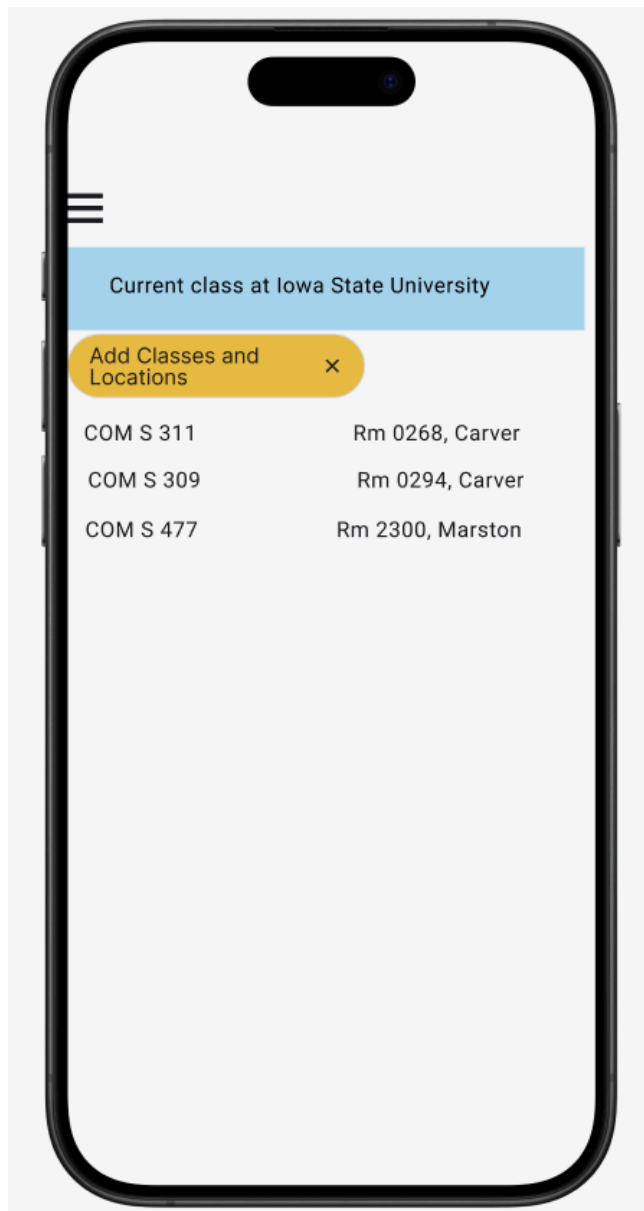
When administrators select **Teachers** from the navigation bar, they are directed to a screen where they can search for all teachers at the university. Administrators can view the list either alphabetically or by other sorting options, allowing for flexible and efficient navigation.

l) Teacher's classes

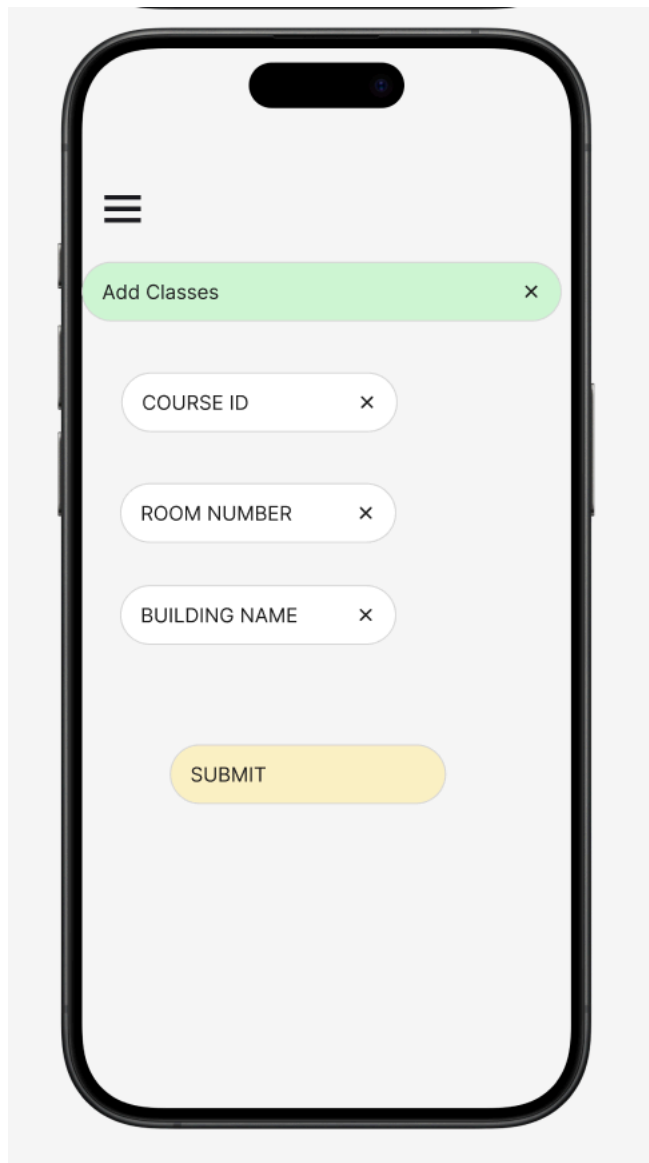


When administrators click on a professor's name, they are directed to the **Professor Profile** screen. Here, they can view the professor's current class assignments and have the ability to add or drop classes as needed.

m) Manage classes



When administrators select **Classes** from the navigation menu, they are taken to a screen displaying the class details, including room numbers and building locations. From this screen, administrators can add new classes and update or assign room locations, ensuring that class schedules and locations are properly managed.



The **Add Class** screen appears when administrators click on **Add Classes** and **Locations** from the previous screen. Here, administrators can input the **Course ID**, **Room Number**, and **Building Name**. Once the required information is entered, they can submit the form to add the new class to the system.

The **Teacher's View** will be very similar to the **Student's View** and **Administrator's view** but with fewer features and screens than the student's view, tailored to their role.