

Frontend Documentation

Overview

MeetMe is a Flutter-based appointment scheduling application that connects professors and students. The frontend is organized into shared service files, professor screens, and student screens. It includes navigation via PageView and bottom navigation bar using GNav, real-time chat via WebSocket, and persistent user sessions.

This document outlines the structure and functionality of each file used in the Flutter frontend.

Shared Logic (`api_service.dart`, `main.dart`, `auth_controller.dart`, `app_navigation.dart`)

- '`api_service.dart`': Handles all API and WebSocket communication. It abstracts HTTP requests and provides methods for authentication, class creation, enrollment, appointment booking, chat, and notifications.
- '`auth_controller.dart`': Manages sign-up and login logic, including validation, navigation based on role, and error handling.
- '`main.dart`': Initializes the app, sets the theme, and presents a draggable login/signup sheet.
- '`app_navigation.dart`': Provides a static method for page jumps using PageController. Used by both professor and student sections.

Professor Pages

- '`professor_pages.dart`': Main PageView and GNav for navigating between professor screens.
- '`professorWelcomePage.dart`': Dynamic greeting, appointment summary, course categories with student count, quick actions.
- '`prof_page1.dart`': List of created classes, create/delete/search functionality, and student list redirection.
- '`prof_page2.dart`': Slot management page with AM/PM groupings, time picker, deletion logic.
- '`prof_page3.dart`': Calendar view (Syncfusion), color-coded by course, highlights appointments by day.
- '`prof_page4.dart`': Loads all students across professor's classes and opens chat via WebSocket.
- '`prof_page5.dart`': Real-time chat screen with message sending/receiving logic.

- ‘professorappointments.dart’: Appointment management, reschedule/cancel, filters for today/upcoming, chip selector for rescheduling.
- ‘professor_notifications.dart’: Lists and marks notifications as read, refreshable.

Student Pages

- ‘student_pages.dart’: Student PageView + navigation using StudentNavigation class.
- ‘student_navigation.dart’: Static method holder for jump-to-page from any widget.
- ‘studentwelcomepage.dart’: Personalized dashboard with greetings, appointments, reminders, quick links, enrolled class list.
- ‘stu_page1.dart’: Displays joined classes, allows joining via class code, searches by course or instructor.
- ‘stu_page2.dart’: Displays enrolled classes and available appointment slots for booking by date, grouped AM/PM, filtered.
- ‘studentappointmentlist.dart’: Lists student appointments by Today, Upcoming, and Past. Allows cancellation and viewing details.
- ‘student_notifications.dart’: Notification screen for students, marks notifications as read.
- ‘classes.dart’: Class student list, fetched via class code, loads usernames.
- ‘CourseGroupPage.dart’: Displays grouped classes by category (used in welcome page), shows count and class access.

API Design and Backend Integration (api_service.dart)

The ‘ApiService’ class centralizes all HTTP requests and WebSocket logic. It uses a RESTful API architecture:

- ‘GET’ for fetching data: ‘getAppointments’, ‘getAvailableSlots’, ‘getClasses’
- ‘POST’ for creating resources: ‘createClass’, ‘createAppointment’, ‘addAvailableSlot’, ‘rescheduleAppointment’
- ‘DELETE’ for removing resources: ‘deleteClass’, ‘cancelAppointment’, ‘deleteAvailableSlot’
- ‘PUT’ and ‘PATCH’ are not used, but could be used for partial updates.

Each method encodes requests as JSON and sets appropriate headers. SharedPreferences is used to persist user session data locally. The chat system is powered by WebSockets, allowing real-time bidirectional communication.

UI Rendering Techniques (ListView, Grid, Modal Sheets)

Appointment data is rendered using 'ListView.builder' for both professor and student dashboards. The dynamic rendering allows efficient updates for:

- Appointments (today/upcoming/past): shown with 'ListTile' inside styled 'Card' widgets
- Classes: rendered as scrollable lists with search filters
- Slots: presented using 'OutlinedButton' and 'ActionChip' widgets grouped into AM/PM slots
- Modal bottom sheets are used for interactive slot booking (students) and rescheduling (professors), allowing contextual UI without navigating to new screens. 'showModalBottomSheet' is used alongside stateful updates.
- Slot management is handled with 'AlertDialog' and 'Wrap' for button arrangement. Calendars are implemented using 'Syncfusion SfCalendar' for professors and 'DatePickerTimeline' for both roles.

State and Navigation Management

Each 'HomePage' (Professor and Student) uses a 'PageView' with a 'PageController'. Bottom navigation ('GNav') allows quick switching between screens. Navigation across unrelated pages is handled via:

- Static function callbacks ('AppNavigation.jumpToPage', 'StudentNavigation.jumpToPage')
- 'Navigator.push' for screen transitions like chat and detail views
- 'Navigator.pushAndRemoveUntil' for logout flows

All screens are 'StatefulWidget' to allow dynamic updates after API calls. Notifications, chat messages, and appointments update via 'setState' calls or refresh handlers.

Authentication Logic (auth_controller.dart and main.dart)

The 'AuthController' manages input validation (email/password), role selection (student or professor), and navigation post-login.

It supports:

- Field validation using regex and length checks
- Role-based navigation using SharedPreferences

- Error handling with snackbar feedback

The login/signup UI uses a 'DraggableScrollableSheet' that transforms into form fields or role selection views based on user interaction. The 'main.dart' file includes theming and branding with logo assets and color schemes.

Professor Feature Logic Summary

- Class Management ('prof_page1.dart'): Professors create classes with auto-generated codes, search through them, and delete using dialog prompts.
- Slot Management ('prof_page2.dart'): Professors pick available 30-minute time slots for classes. Duplicate slots are blocked. Slots are grouped AM/PM.
- Appointment Calendar ('prof_page3.dart'): Uses 'SfCalendar' to visualize appointments per date. Color tints are dynamically generated.
- Chat with Students ('prof_page4.dart'): Loads students across all classes, shows usernames, and launches 'ChatScreen'.
- Notifications ('professor_notifications.dart'): Unread messages are tracked and marked via API.
- Appointment Management ('professorappointments.dart'): Includes cancel and reschedule with dynamic chip slot selectors inside modals.
- Dashboard ('professorWelcomePage.dart'): Categorizes classes by type (math, science), displays appointment previews and student counts.

Student Feature Logic Summary

- Class Enrollment ('stu_page1.dart'): Students view joined classes and join via a 4-digit class code.
- -Duplicate joins are blocked. Includes class search.
- Booking Appointments ('stu_page2.dart'): Loads available slots for a class and selected date. AM/PM grouping with availability filtering. Booking uses 'createAppointment'.
- Calendar & Notifications ('studentappointmentlist.dart', 'student_notifications.dart'): Categorizes appointments by Today, Upcoming, and Past with cancel support.
- Dashboard ('studentwelcomepage.dart'): Greets user, previews upcoming appointments, shows reminders and quick navigation buttons.

- Student Home UI ('student_pages.dart'): Navigation via 'PageView', notification badge via 'FutureBuilder', and logout handled with 'pushAndRemoveUntil'.