Attendin

Attendance tracking made simple and easy

Objectives:

- **Attendance Integrity through Geolocation Verification**
- **Optimize User Experience to Minimize Administrative Overhead**
- **Establish a Secure and Scalable Data Architecture**

Mutazz Zeidan
- Computer Engineer -
Front-End Developer

Dominic Iquina
- Computer Science -
Back-End Developer

Elizabeth Pawley
- Assistant Professor -
Advisor

Built With:

Background & Goals

Background:

Traditional attendance systems like QR codes or paper sign-ins are easily exploited, compromising academic integrity. Attendin solves this by requiring students to be physically present, using geofencing technology.

Project Goals:

- Attendance Integrity through Geolocation Verification
- Optimize User Experience to Minimize Administrative Overhead
- Establish a Secure and Scalable Data Architecture

Intellectual Merits

- **Frictionless User Experience (UX) Design**
- We created a one tap check in workflow that minimizes time spent checking into class to under 5 seconds. This directly supports our goal of minimal disruption to class

- **Security-Preserving Architecture:**
- We handle the raw GPS data in real-time which is then immediately discarded, ensuring strict FERPA compliance by never storing student location history.

- **Caching Strategy:**
- We integrated a caching layer for static data (like student profiles and class schedules). This reduces database hits for frequently accessed data, lowering latency from milliseconds to microseconds

Broader Impacts

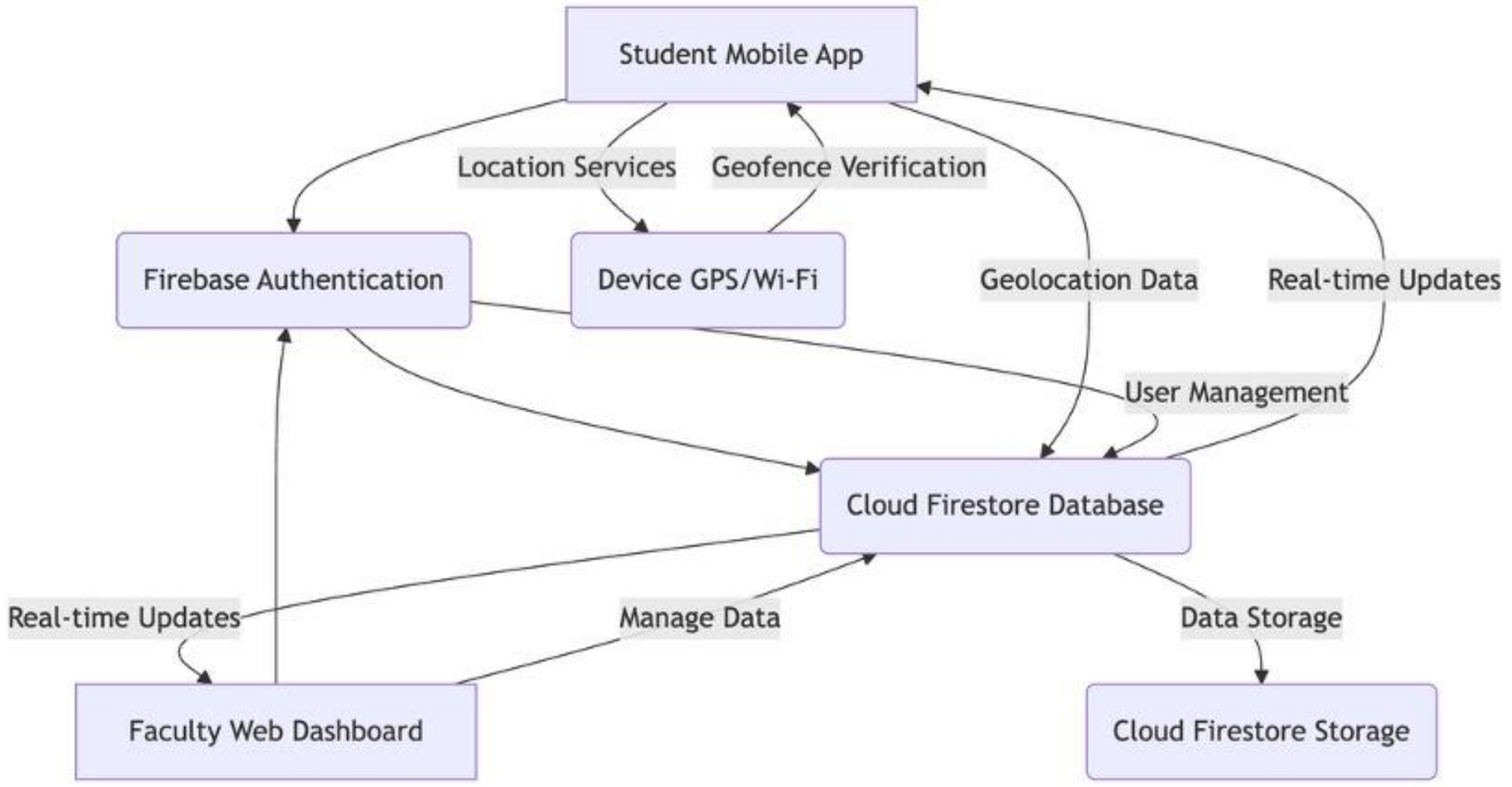
- **Reducing Wasted Class Time**
- By automating the data collection process, the system reclaims valuable instructional time that is currently lost to manual roll calls or passing around sign-in sheets.

- **Restoring Academic Integrity**
- Current systems like static QR codes are easily exploited by students, sign in sheets can sometimes be marked off by friends, by enforcing physical presence through geolocation verification, this ensures that attendance records accurately reflect a student’s actual participation.

- **Modernizing Classroom Infrastructure**
- This project replaces outdated, insecure methods with a secure, scalable digital architecture. By utilizing the devices students already own, this project demonstrates that secure, verifiable attendance tracking can be implemented at zero additional cost to the university.

System Architecture

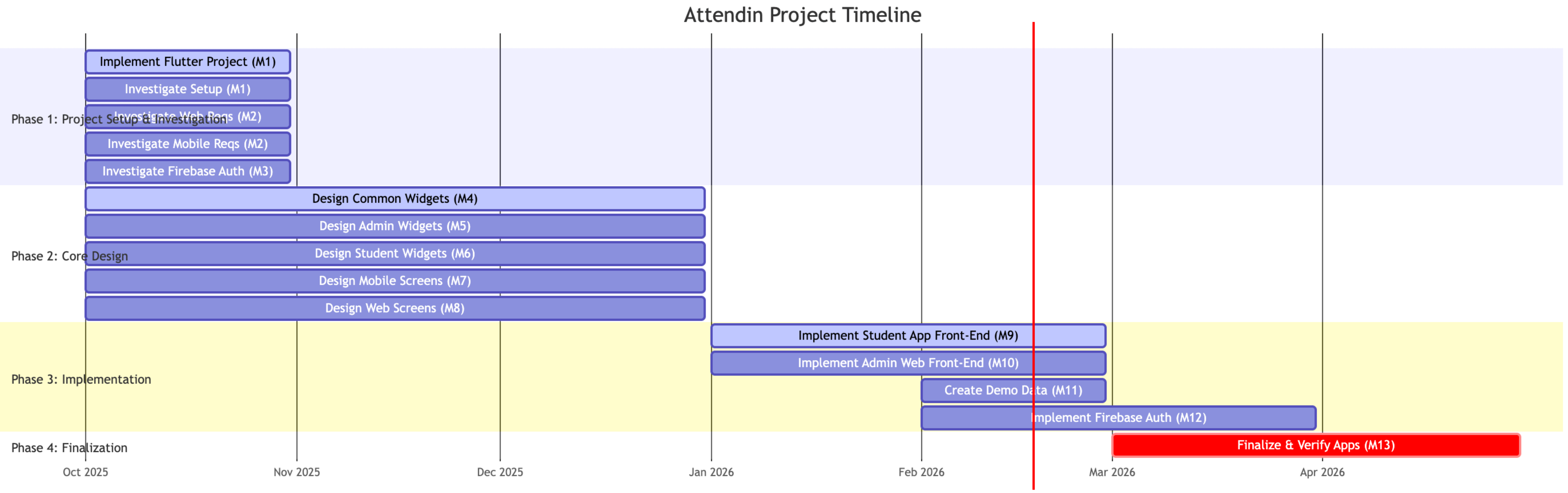
- **Architecture:**
- Client Layer (Student/Faculty Apps)
- Security Layer (Firebase Authentication)
- Data Layer (Cloud Firestore Database)



Technologies

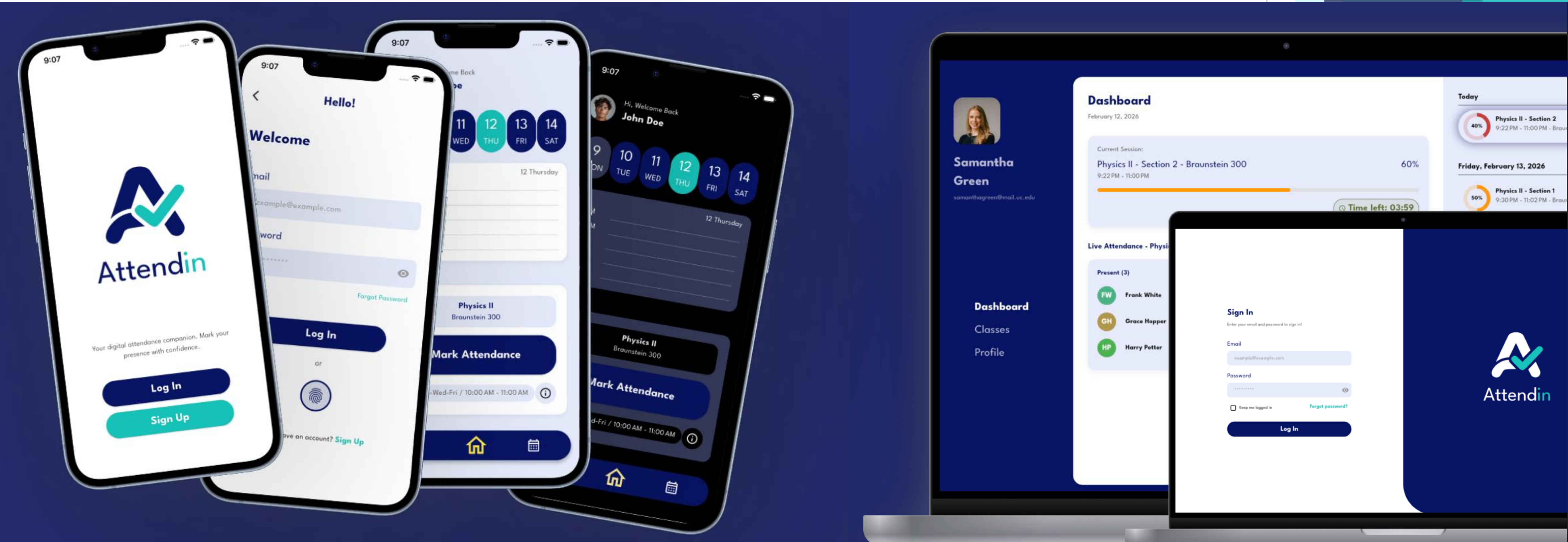
- **Frontend Framework: Flutter (Dart)**
  - Flutter gives us cross-platform development to deploy apps for both Android and IOS from a single codebase.
  - By using Flutter for both the Student Mobile App and the Faculty Web Dashboard, we can have high code reusability and keep widgets common between both the apps.
- **Backend Infrastructure: Google Firebase**
  - Authentication: We use Firebase Auth to delegate complex security tasks (like saving passwords and session management) to Google, ensuring industry standard security without having to build it from scratch.
  - Database: We use Cloud Firestore for its Real-Time Listeners. This allows the Faculty Dashboard to update instantly when a student checks in, without the professor needing to refresh the page.

Milestones


Attendin Project Timeline
Phase 1: Project Setup & Investigation
Implement Flutter Project (M1)
Investigate Setup (M1)
Investigate Web Reqs (M2)
Investigate Mobile Reqs (M2)
Investigate Firebase Auth (M3)
Phase 2: Core Design
Design Common Widgets (M4)
Design Admin Widgets (M5)
Design Student Widgets (M6)
Design Mobile Screens (M7)
Design Web Screens (M8)
Phase 3: Implementation
Implement Student App Front-End (M9)
Implement Admin Web Front-End (M10)
Create Demo Data (M11)
Implement Firebase Auth (M12)
Phase 4: Finalization
Finalize & Verify Apps (M13)
Oct 2025
Nov 2025
Dec 2025
Jan 2026
Feb 2026
Mar 2026
Apr 2026

Results

- **Authentication** - Secure Login/Sign-up via Firebase Auth
- **Navigation** – Core UI Completed for Student (Mobile) and Faculty (Web) dashboards
- **Database** – Cloud Firestore Connected with real-time listeners



Challenges

- Cross-Disciplinary Integration
  - We established a database schema which allowed us to first work on the UI skeletons separately before real database implementation. After this was finished, we were able to implement firebase smoothly without having to rework everything.
- Simulating High-Volume traffic
  - We needed to ensure the system could handle an entire class worth of students checking in at the exact same time, without physically gathering that many people for every test. We wrote automated seeding scripts that inject large volumes of mock user interactions into the database simultaneously. Which allowed us to stress-test the dashboards real-time listeners and verify that the app remains responsive.
- Designing for FERPA Compliance
  - We created a Privacy-First Database schema. We designed the system to store only the attendance timestamps and status, deliberately leaving out any continuous location history to ensure we met the legal constraints of the university environment.