SafePass Intelligent Student Transit Platform – Phase I Report

Crew Sirius

Bekir Emre Sarıpınar, Mehmet Efe Hırkalı, Ahmet Burak Durukanoğlu, Ömer Nacar, Bekir Göktepe, Edip Zahit Güney

CSE343 – Software Engineering, Gebze Technical University

Abstract

SafePass is a multi-tenant SaaS platform that digitalizes student transportation for company administrators, schools, drivers, and parents. The solution spans a NestJS backend (REST + WebSocket gateway), a Next.js administrative portal, and Flutter-based driver/parent apps. This report consolidates project-wide artifacts—covering the foundational sprints (infrastructure and authentication), sprint 2 (route management and observability), and ongoing sprint 3 planning (attendance and notifications). It summarizes the main user story, software specification, backlog health, and UI prototypes to provide a holistic status snapshot for stakeholders.

Keywords: student transit, real-time tracking, software engineering, backlog, SafePass

I. INTRODUCTION

School transportation still depends on phone calls, spreadsheets, and WhatsApp groups. When a bus is late, parents worry, school staff scramble for answers, and drivers try to respond while juggling their routes. SafePass was born from the simple idea that everyone should share the same reliable picture of what is happening on the road. Instead of chasing information, parents receive timely notifications, drivers record boarding with a single tap, and administrators monitor the entire fleet from a common dashboard.

From the very first sprint, the team set out to replace manual coordination with an ecosystem that feels natural to each stakeholder. We started by ensuring that every role—company admin, school admin, driver, parent—can sign in securely and land on an experience tailored to them. Next, we focused on route authoring, student management, and reporting so that the daily operation (who goes where and when) could be handled in a single workflow. As of this report, we are extending that foundation with digital attendance, absence notifications, and panic-alert procedures so that SafePass can respond gracefully when plans change or emergencies arise.

This document captures that journey. It highlights the user story that drove our first release, the requirements we distilled from interviews and field observations, the backlog items that keep us moving, and the prototype interfaces we use to validate the experience with real users. Above all, it reminds us why we are building SafePass: to give parents peace of mind, to help schools and transport companies stay ahead of the day, and to let drivers focus on the road rather than the phone.

II. USER STORY AND SCENARIOS

User Story US-001 – “A Calm Morning for Every Stakeholder”

> The transportation company manager wants to publish daily routes and assign drivers,
>
> the **school administration** wants to see which students boarded in real time,
>
> the **driver** wants to record attendance and communicate status with a single tap,
>
> and the **parent** wants precise updates about the bus so they can plan ahead.
>
> SafePass gives all of these parties one shared, transparent view of the same morning ride.

Main Scenario – “Smooth Morning Journey”

1. The transportation manager defines the route, stops, vehicle, and driver the night before on SafePass.
2. The driver opens the mobile app in the morning, taps “Start Trip,” and parents immediately receive “the bus is on its way” notifications.
3. Parents watch the bus on the live map and rely on smart ETA reminders to send their child downstairs at the right moment.

4. When the student boards, the driver updates the digital attendance list; the school dashboard instantly marks the student as “onboard.”
5. Upon arrival, the driver completes the trip; parents get an “arrived safely” message, and the company sees the trip in its daily summary.

Alternative Scenario 1 – “Absence Notice”

1. A parent reports that their child will skip the bus that morning.
2. SafePass alerts the driver, school, and company simultaneously, marking the stop as “skipped.”
3. The driver avoids unnecessary waiting, the company saves fuel, and the school keeps an accurate absence record.

Alternative Scenario 2 – “Delay & Communication”

1. The driver anticipates heavy traffic and triggers the delay option.
2. Parents receive “expect an 8–10 minute delay,” and the school panel displays the same warning.
3. Families spend less time waiting outside, the school adjusts its timetable, and the driver can focus on the road instead of phone calls.

Alternative Scenario 3 – “Panic Situation”

1. The driver encounters an emergency (accident, mechanical issue) and holds the panic button for three seconds.
2. Transportation management and the school are auto-notified; parents receive a reassurance message that support is on the way.
3. The operations center tracks the bus location, dispatches backup if necessary, and keeps the driver informed.

Alternative Scenario 4 – “Evening Return & Parent Communication”

1. When the evening route starts, the driver taps “Return Trip”; parents get an “evening bus has departed” notification.
2. The driver marks each stop as students get off; parents see when their child leaves the bus.
3. The company and school maintain a digital log of who arrived home.

Alternative Scenario 5 – “Quick School Dashboard Check”

1. School staff opens the morning dashboard and sees every student labeled as “onboard,” “absent,” or “waiting.”
2. If a student is still “waiting,” the school sends a reminder notification through SafePass instead of calling.
3. The parent receives the reminder, sends the child to the stop, and the status updates automatically once the driver records the boarding.

III. SOFTWARE REQUIREMENT SPECIFICATION

A. Purpose and Scope

SafePass digitizes end-to-end school transportation by connecting transportation companies, school administrators, drivers, parents, and students in a single ecosystem. The scope includes:

- Route planning, driver assignment, stop management, and reporting for transportation companies.
- Attendance monitoring, absence handling, and audit trails for school administrators.
- Live tracking, notifications, and historical journeys for parents.
- Guided route execution, digital attendance, and panic alert workflows for drivers.
  Out of scope: in-class safety systems, tuition/billing modules, and vehicle maintenance tracking.

B. Stakeholders and Definitions

- **Transportation Company Manager (Company Admin):** Owns the fleet, defines routes, and monitors performance.

- **School Administrator:** Oversees daily attendance, parent communication, and incident escalation.
- **Driver:** Uses the mobile app to start/end trips, mark boarding events, and trigger panic or delay notices.
- **Parent:** Follows real-time bus status, receives notifications, and reports absences through the app.
- **Student:** Passive beneficiary whose status (onboard, absent, arrived) is tracked digitally.

Key Terms

- **Route:** A scheduled service with ordered stops, assigned vehicle, driver, and time window.
- **Assignment:** A link between a driver, vehicle, route, and date (single trip).
- **Digital Attendance:** In-app confirmation of each student boarding or leaving the bus.
- **Panic Button:** Emergency trigger that notifies company and school after being held for ≥3 seconds.
- **ETA:** Estimated Time of Arrival combining live traffic and historical data (via Google Maps API).

C. Functional Requirements

1. The system shall allow company admins to create, update, and publish routes with ordered stops.
2. The system shall enforce that each route is tied to exactly one school and one vehicle per trip.
3. The system shall prevent double booking of drivers or vehicles within overlapping time windows.
4. The system shall allow drivers to start/end a trip, update location in real time, and record boarding via the mobile app.
5. The system shall notify parents when the bus departs, approaches their stop, and arrives at school or home.
6. The system shall allow parents to submit absence notices that automatically remove the student from upcoming trips.
7. The system shall display attendance status to school administrators in real time.
8. The system shall provide digital logs of all trips (start/end times, stops visited, attendance) for auditing.
9. The system shall allow drivers to declare delays and broadcast the updated ETA to parents and schools.
10. The system shall provide a panic mechanism that alerts company admins, school admins, and parents with the latest location.

D. Data and Business Rules

1. A student must be linked to at least one parent account; a parent may manage multiple students.
2. Each route belongs to exactly one school but can include stops outside the school’s district.
3. Panic button events require a press duration of ≥3 seconds to avoid accidental triggers.
4. ETA calculations must use an external mapping service (Google Maps or equivalent) combining traffic and historical data.
5. Absence requests must specify at least one date; the system shall remove the student from affected assignments.
6. Attendance records shall include timestamp, location, driver ID, and route ID.
7. Every assignment must have exactly one vehicle and driver; substitutions must be logged.

E. Non-Functional Requirements (System Infrastructure Excluded)

- **Performance:** Critical user actions (attendance, panic, delay) must be acknowledged within 2 seconds under normal load.
- **Reliability:** The system shall queue offline driver events and sync them once connectivity returns; no attendance data loss is acceptable.
- **Availability:** The target service uptime is ≥99.5 %; maintenance windows must be announced 24 hours in advance.
- **Security:** All requests must be authenticated; role-based access controls must prevent data leakage between schools.
- **Usability:** Mobile actions (attendance, panic, delay) shall require at most two taps; parent notifications must be customizable.

- **Observability:** Metrics for active connections, message throughput, and errors shall be exposed for monitoring dashboards.

IV. PROJECT BACKLOG (Epic-Level Summary)

Epic 1 – Project Infrastructure & DevOps Setup

Goal: Establish the monorepo, CI tooling, backend skeleton, and platform-specific shells.

Stories:

1. Story 1.1 – Repository and development-environment bootstrap.
2. Story 1.2 – NestJS + PostgreSQL foundation with migrations and env wiring.
3. Story 1.3 – Next.js admin portal skeleton with Tailwind layout/routing.
4. Story 1.4 – Flutter mobile workspace with Riverpod, GoRouter, Dio setup.

Priority & Rationale: High – enables every other module and ensures reliable build/test automation.
Status: Completed in Sprint 1.

Epic 2 – Authentication System

Goal: Provide secure role-based login flows across web and mobile.

Stories:

1. Story 2.1 – Firebase Authentication integration (client + Admin SDK).
2. Story 2.2 – Parent mobile login experience.
3. Story 2.3 – Driver mobile login experience.
4. Story 2.4 – Web admin login with role-aware redirects.
5. Story 2.5 – Profile synchronization API between Firebase and Postgres.

Priority & Rationale: High – without auth, downstream features cannot be exercised safely.
Status: Completed in Sprint 1 (hand-offs finalized early Sprint 2).

Epic 3 – Real-Time Tracking (MVP)

Goal: Deliver live bus tracking for drivers, parents, and administrators.

Stories:

1. Story 3.1 – WebSocket gateway (`/location`) with room management and guards.
2. Story 3.2 – Driver-side streaming pipeline (GPS permissions, reconnection).
3. Story 3.3 – Parent mobile live map powered by Socket.IO updates.
4. Story 3.4 – Admin live-tracking console aggregating all active routes.

Priority & Rationale: High – differentiating capability for SafePass.
Status: Completed by the end of Sprint 1 / start of Sprint 2.

Epic 4 – Route Management System

Goal: Let transportation companies plan routes, assign drivers, and monitor trips.

Stories:

1. Story 4.1 – Route creation/editing (forms, map stops, CRUD endpoints).
2. Story 4.2 – Driver route visualization with ordered stops and mobile UX.
3. Story 4.3 – Operational daily reports (endpoint, UI card, SQL view).

Priority & Rationale: High – foundational for every other module; enables live tracking, reporting, and safety features.
Status: Completed in Sprint 2; current UI tweaks are refinements built on top of the delivered core.

Epic 5 – Student & Parent Management

Goal: Keep student rosters accurate and give parents visibility into journeys.

Stories:

1. Story 5.1 – Student management (entity, CRUD APIs, admin UI).
2. Story 5.2 – Parent-facing children list and selection flows.

Priority & Rationale: High – required for attendance, notifications, and compliance.
Status: Completed in Sprint 2 with roster APIs, parent linkage flows, and parent dashboard widgets.

Epic 6 – WebSocket Optimization & Monitoring

Goal: Ensure scalable, observable real-time communication.

Stories:

1. Story 6.1 – Performance hardening (connection pooling, reconnection, leak detection, load tests).
2. Story 6.2 – Metrics/logging collection and dashboard integration.

Priority & Rationale: Medium – critical for long-term stability.
Status: Completed during Sprint 2 (load testing + Prometheus integration).

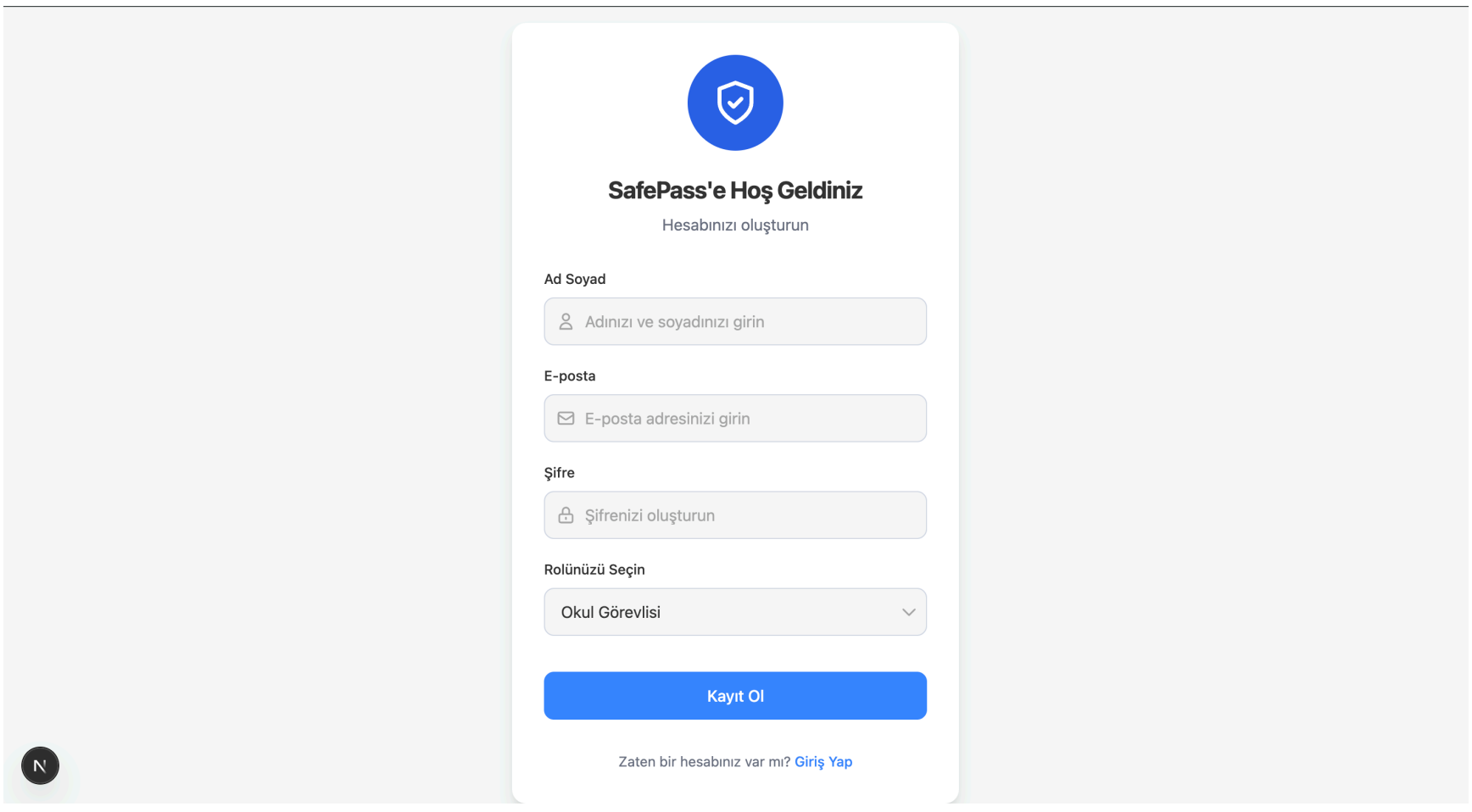
Epic 7+ (Attendance, Notifications, Panic) – To Be Completed

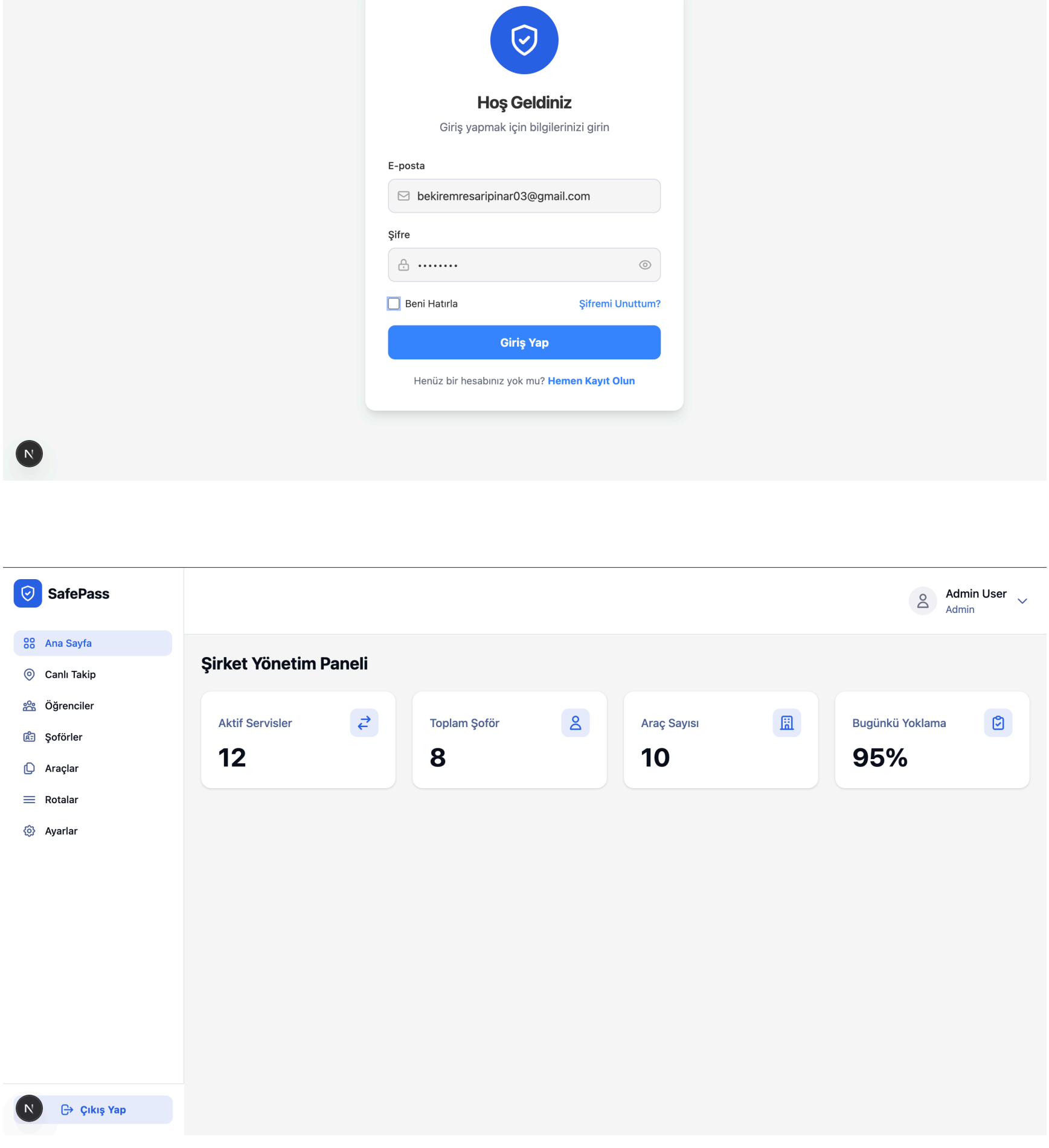
Goal: Epic 7 (Attendance Automation), Epic 8 (Absence Management & Notifications), Epic 9 (Panic & Safety Workflows), Epic 10 (Analytics), Epic 11 (Billing), Epic 12 (Fleet Management), Epic 13 (Integrations), Epic 14 (Mobile UX polish), and Epic 15 (Compliance) remain planned for subsequent sprints.

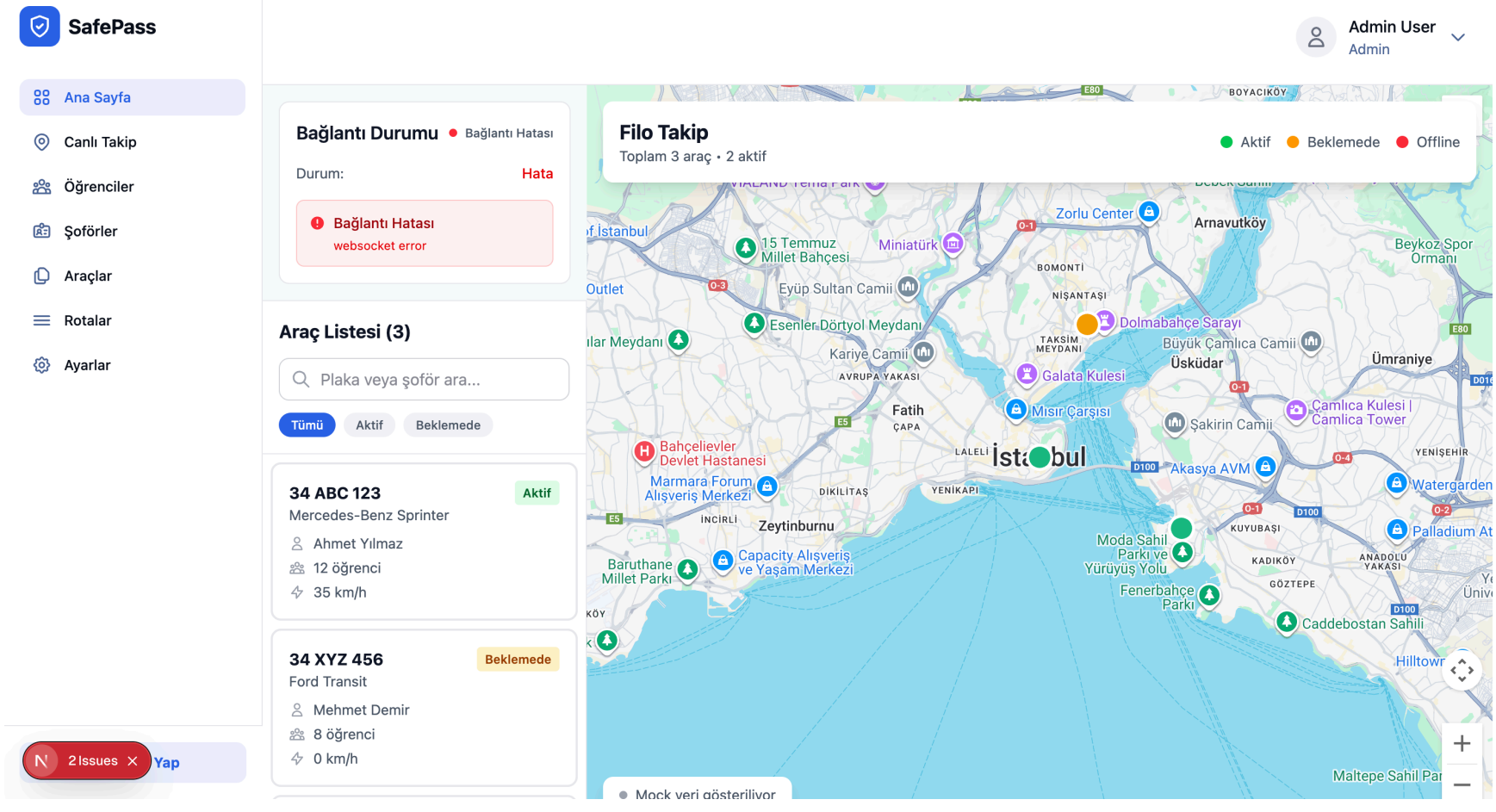
Status: Not yet started; they are slated for Sprint 3 and beyond, and will be detailed once design and architecture spikes conclude.

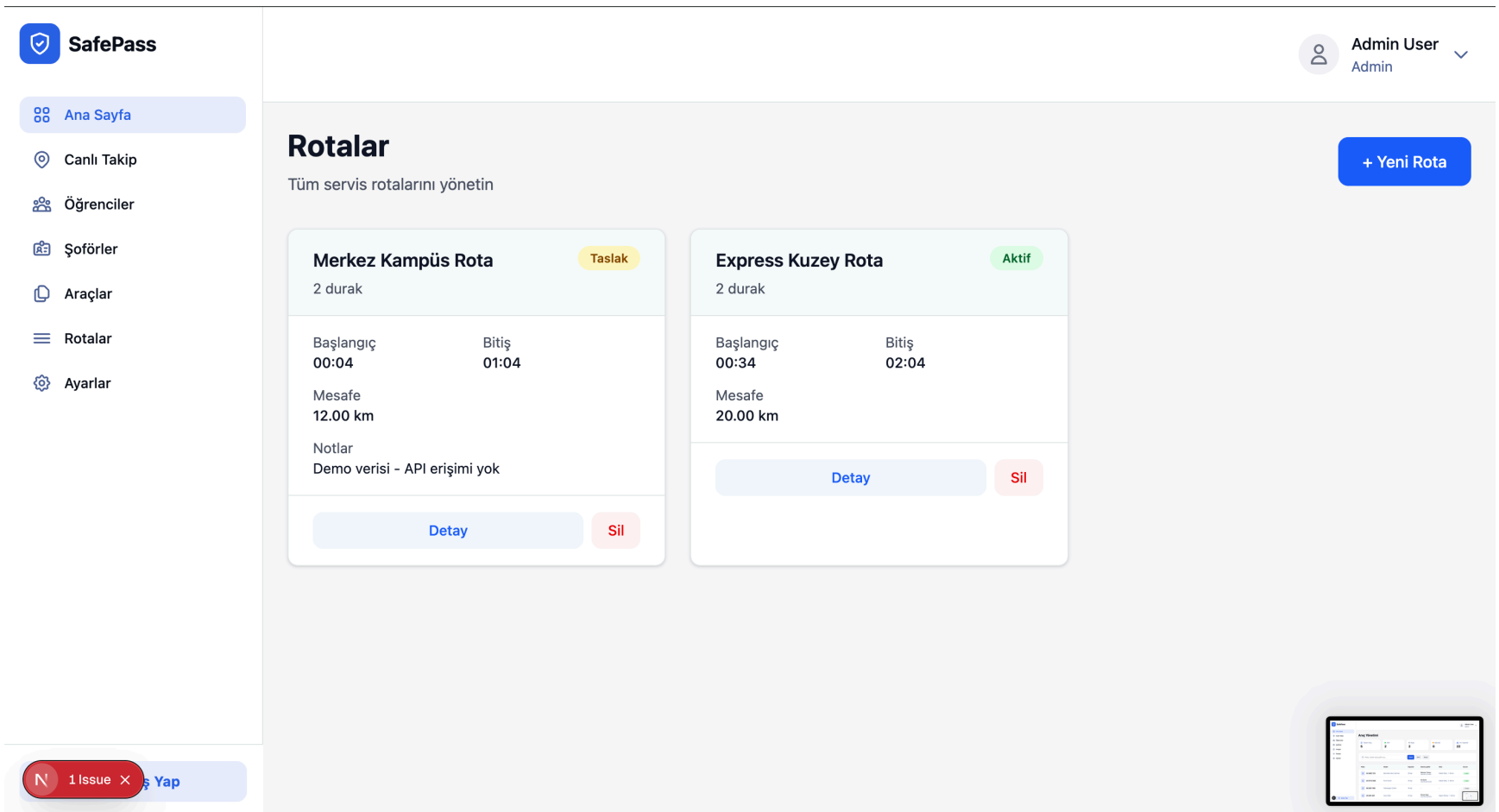
V. DUMMY / SIMULATED GRAPHICAL INTERFACE

A. WEB INTERFACE

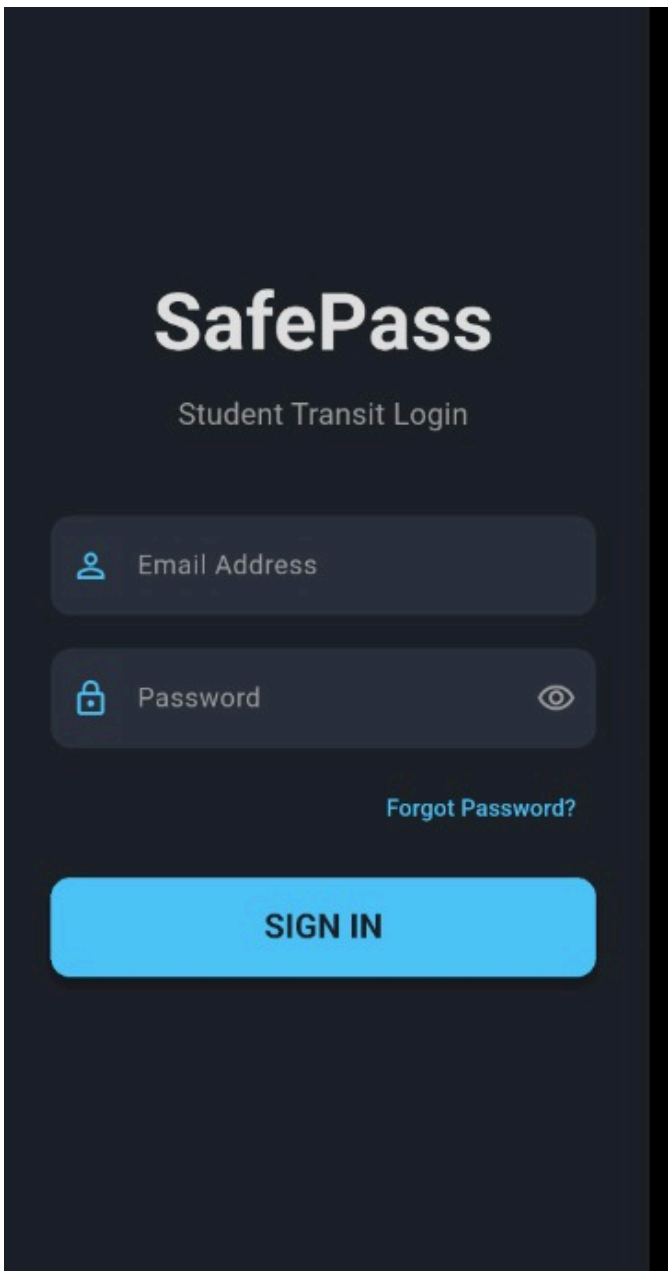


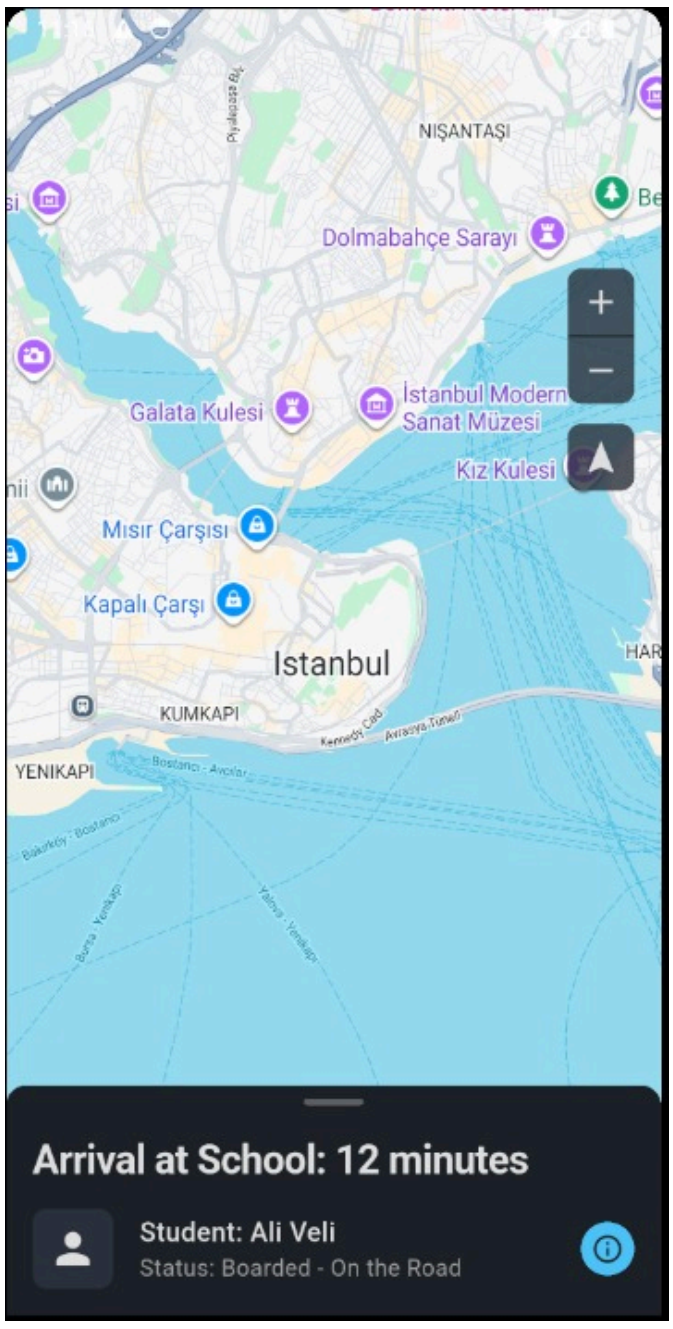

Hoş Geldiniz
Giriş yapmak için bilgilerinizi girin
E-posta
bekiremresaripinar03@gmail.com
Şifre
••••••••
Beni Hatırla
Şifremi Unuttum?
Giriş Yap
Henüz bir hesabınız yok mu? Hemen Kayıt Olun
SafePass
Ana Sayfa
Canlı Takip
Öğrenciler
Şoförler
Araçlar
Rotalar
Ayarlar
Çıkış Yap
Admin User
Admin
Şirket Yönetim Paneli
Aktif Servisler
12
Toplam Şoför
8
Araç Sayısı
10
Bugünkü Yoklama
95%

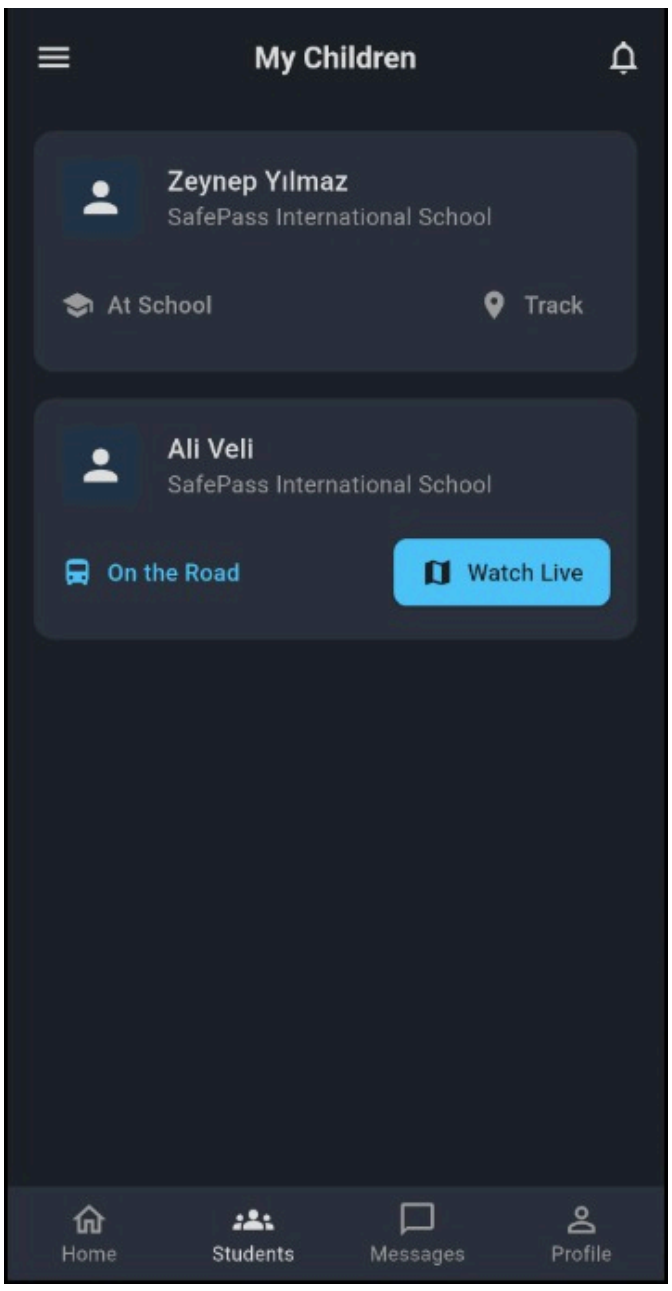


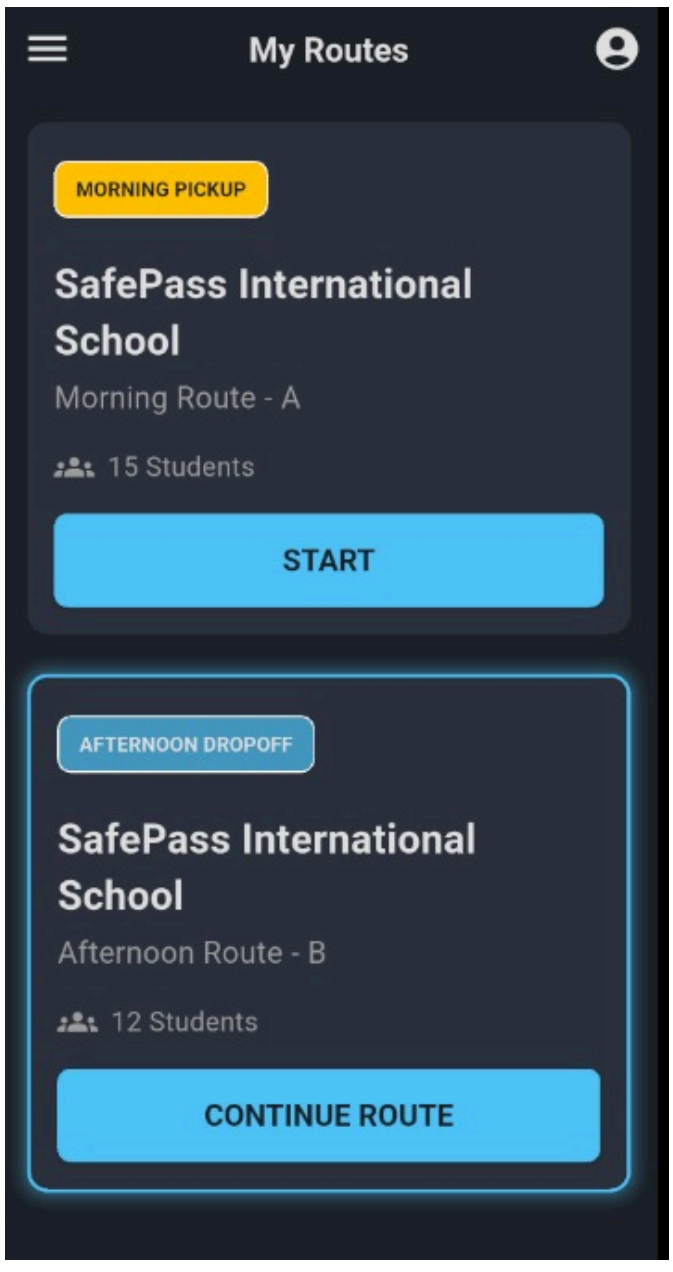


B. MOBILE INTERFACE


SafePass
Student Transit Login
Email Address
Password
Forgot Password?
SIGN IN


NIŞANTAŞI
Dolmabahçe Sarayı
Galata Kulesi
İstanbul Modern Sanat Müzesi
Kız Kulesi
Mısır Çarşısı
Kapalı Çarşı
Istanbul
KUMKAPI
YENIKAPI
Arrival at School: 12 minutes
Student: Ali Veli
Status: Boarded - On the Road


My Children
Zeynep Yılmaz
SafePass International School
At School
Track
Ali Veli
SafePass International School
On the Road
Watch Live
Home
Students
Messages
Profile


My Routes
MORNING PICKUP
SafePass International School
Morning Route - A
15 Students
START
AFTERNOON DROPOFF
SafePass International School
Afternoon Route - B
12 Students
CONTINUE ROUTE

VI. CONCLUSION

SafePass now offers an end-to-end digital transportation backbone: shared authentication across roles, real-time location streaming, rich route planning, and roster management are all deployed across backend, web, and mobile touchpoints. The engineering effort also established CI, observability, and load-tested WebSocket infrastructure, giving the team confidence to scale beyond pilot fleets. Phase I deliverables therefore demonstrate that the technical foundation is production-grade and already solving day-to-day coordination between parents, drivers, and school/company administrators.

Looking ahead, the roadmap focuses on deepening safety and communication: digital attendance flows, absence automation, FCM-based notifications, and panic response tooling (Epics 7–15) will build on the stable core described in this report. With the architectural groundwork complete and user feedback loops in place, the team is well positioned to deliver those advanced capabilities in the upcoming sprints and move SafePass toward a public launch.

REFERENCES

[1] IEEE Editorial Style Manual, “Guidelines for Conference Papers,” 2023.

[2] NestJS Docs, “WebSockets and Gateway Patterns,” v11.1.

[3] M. Fowler, *UML Distilled*, Boston: Addison-Wesley, 2003.