YEDITEPE UNIVERSITY

FACULTY OF ENGINEERING

ISE 402 - CSE 344

Interdisciplinary Project

Analysis Report

“TrackMyStudent”

ISE TEAM

Beren Özder – 20210703019

Aylin Akpınar – 20200703070

Yaren Sancar – 20200703039

CSE TEAM

Berkay Ahkemoğlu - 20210702019

Buğrahan Aydın – 20210702018

Furkan Can Aydın – 20210702017

Oğuzhan Çakan - 20210702128

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to analyze the challenges faced by driving school instructors in managing lesson schedules, student information, and communication with course administration. Based on this analysis, the document proposes a centralized software solution that aims to streamline the entire educational process by providing a unified and user-friendly platform. This report outlines the functional and non-functional requirements of the system and sets the foundation for its design and implementation phases.

1.2. Background

In many driving schools, instructors rely on manual or fragmented tools to manage their daily operations. Lesson scheduling is often done using paper-based calendars or Excel spreadsheets. Student attendance and lesson progress are tracked separately, and communication with course management or fellow instructors is handled via tools like WhatsApp or phone calls. This scattered structure leads to time loss, inconsistent information, communication breakdowns, and a lack of traceability in student progress.

Moreover, there is no common platform for instructors to share notes, past lesson insights, or updates on students. As a result, candidate tracking becomes inefficient, repetitive lessons are delivered unnecessarily, and candidates' true performance levels are not easily assessed. Errors made during student registration or scheduling require direct communication with course management to be resolved, further slowing down the process.

1.3. Motivation

1.3.1. Problems of Existing Systems

The current process of managing students and schedules in driving schools is inefficient and fragmented. Key issues include:

- **Manual Scheduling:** Instructors spend excessive time creating and updating daily schedules manually, which is error-prone and inflexible.
- **Scattered Student Information:** Candidate attendance and schedule changes are managed across different platforms, making it hard to access up-to-date information quickly.
- **Communication Gaps:** Communication with management and among instructors is done via separate channels, often resulting in unsynchronized data and misunderstandings.
- **Lack of Shared Notes:** Instructors cannot easily share observations or lesson notes, which affects continuity and lesson planning.
- **Error Handling Inefficiencies:** Any error in student records (e.g., incorrect names or attendance) requires manual correction and administrative involvement, which adds extra workload and delays.

1.3.2. The New System

The proposed system aims to eliminate the inefficiencies mentioned above by providing a centralized application tailored for driving school instructors. Key features and benefits include:

- **Centralized Scheduling and Tracking:** Instructors will be able to easily create, view, and modify schedules in one place with built-in time slot automation.
- **Unified Student Records:** All student data, including names, attendance, and notes, will be accessible in one system, ensuring continuity and consistency.
- **Improved Instructor Collaboration:** Shared lesson notes and progress tracking will help instructors coordinate and avoid content repetition.
- **Automated Notifications:** Reminders and updates regarding student sessions will be sent automatically to minimize forgetfulness or confusion.
- **Streamlined Management Communication:** The system will facilitate more structured and documented communication between instructors and course administrators.

With these capabilities, the system will significantly reduce workload, enhance time management, and improve the quality and consistency of driver education.

1.4. Structure of the Document

This document is organized as follows:

- **Section 2** describes the functional requirements of the system, including detailed use cases and expected user interactions.
- **Section 3** outlines the non-functional requirements such as performance, usability, and system constraints.
- **Section 4** provides the system models, including UML class diagrams and interface mockups.
- **Section 5** defines the abbreviations and terminology used throughout the document.
- **Section 6** includes a glossary and references that support the design and development processes.

This report serves as a foundational analysis document for the system's subsequent design, implementation, and evaluation stages.

2. Functional Requirements

- The system shall allow instructors to edit existing lesson schedules.
- The system shall allow instructors to delete lesson schedules.
- The system shall prevent scheduling conflicts between instructors and students.
- The system shall display scheduled lessons in a list format sorted by date.
- The system shall allow instructors to mark student attendance for each lesson.
- The system shall save attendance history for every student.
- The system shall allow instructors to write lesson-specific notes for each student.
- The system shall allow instructors to view previous notes related to a student.
- The system shall store each student's profile, including name, status, and assigned instructor.
- The system shall allow instructors to view the full profile of a student.
- The system shall allow instructors to share student-related notes with other instructors.
- The system shall allow instructors to submit correction requests for student data.
- The system shall notify administrators when a correction request is submitted.
- The system shall allow administrators to approve or reject correction requests.
- The system shall allow instructors to view a summary of lessons completed for each student.
- The system shall send automatic reminders to instructors before scheduled lessons.
- The system shall send automatic notifications to instructors when a schedule is updated.
- The system shall allow instructors to take notes about overall student performance.
- The system shall generate student progress summaries with attendance and notes.
- The system shall allow administrators to update instructor and student information.
- The system shall allow instructors to define their weekly availability.
- The system shall allow administrators to assign students to instructors.
- The system shall provide a dashboard showing key metrics to instructors.

- The system shall provide a dashboard showing key metrics to administrators.
- The system shall track user actions such as schedule edits and note entries.
- The system shall allow users to log in securely with email and password.
- The system shall allow users to reset passwords via email verification.
- The system shall enable communication between instructors and students through in-app messaging.
- The system shall support structured communication between instructors for lesson coordination.

2.1. Description of the System Functionalities

- **Sign Up**

The system allows new users to create an account by entering their first name, last name, email, and password.

- **Log in**

Users can log in to the system using their email and password.

- **Access the Main Screen**

After logging in, instructors and administrators are redirected to the main screen, where they can view their lesson schedules, student lists, and notifications.

- **Get Lesson Settings**

Only administrators can define lesson names, dates, durations, and assign instructors.

- **Schedule Lessons**

Instructors can set lesson times, update assigned lessons for students, and cancel lessons if needed.

- **Track Student Progress**

Instructors can track which lessons students have taken, their attendance records, and exam performances.

- **Editing Invitation Method**

Instructors can invite students to the system using a QR code or an invitation link.

- **Notifications and Reminders**

The system must send reminder notifications to students and instructors before lesson start times.

- **Error Handling**

In cases such as incorrect student registration or scheduling errors, the system must automatically detect errors and notify the instructor or administrator.

2.2. Description of the System Users

2.2.1.User

Individuals who have logged into the system but have not yet been assigned to a lesson or an instructor group are classified as users.

2.2.2.Instructor

Instructors are system users responsible for planning lessons, tracking student progress, and guiding students.

2.2.3.Student

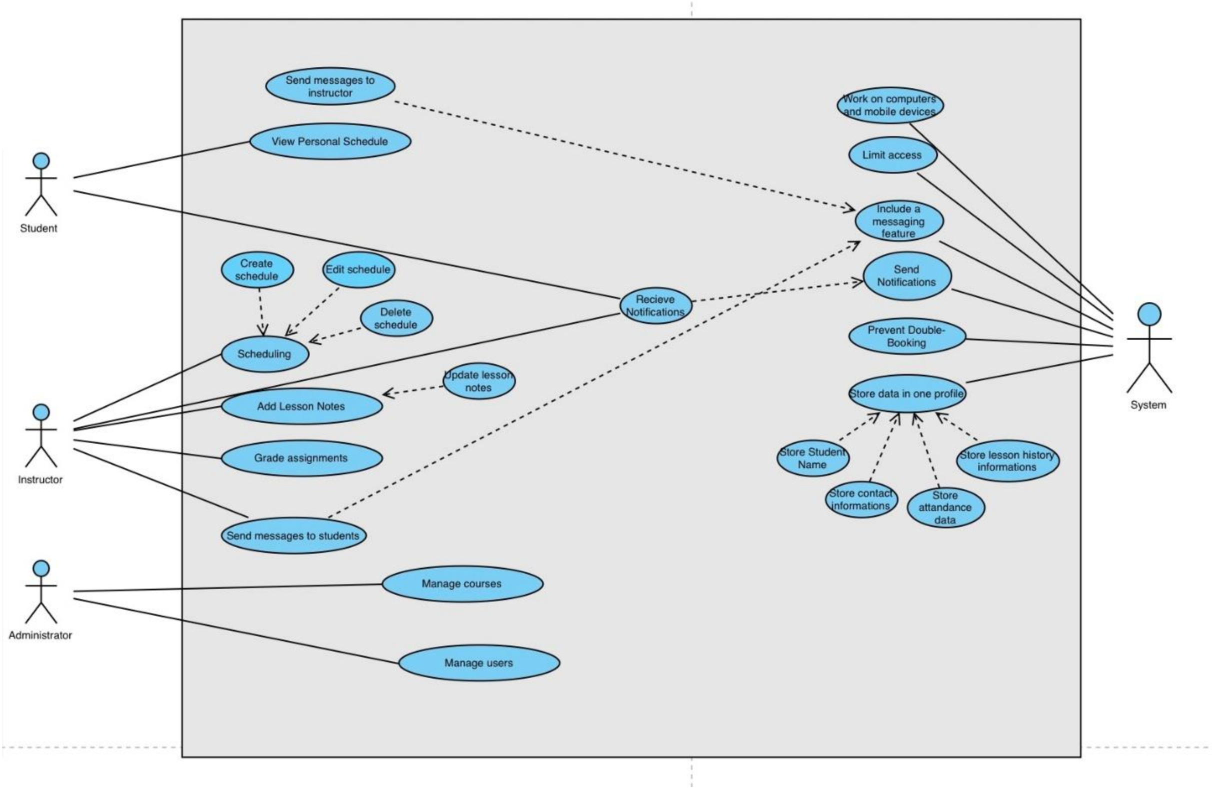
Students are individuals enrolled in the driving school, attending lessons, taking exams, and completing the training process.

2.2.4.Admin

Administrators have the authority to create lesson schedules, assign instructors, edit student records, and manage the overall system.

2.3. Specific Requirements

2.3.1. Use Case Diagram



2.3.2. Use Case Priority List

Priority Rank	**Use Case**	**Rational**
High	*Sign Up*	Essential for user onboarding, allowing new users to create accounts and access the system.
High	*Log In*	Ensures secure access to user accounts and system functionalities.
High	*Access the Main Screen*	Represents the core entry point for users, enabling navigation to system features.
Medium	*Get Lesson Settings*	Provides customization options for lesson-related preferences, enhancing user control.
High	*Schedule Lessons*	Enables instructors to organize and plan lessons efficiently, ensuring structured learning.
High	*Track Student Progress*	Allows monitoring of student performance and attendance, facilitating better academic tracking.
Low	*Editing Invitation Method*	Gives flexibility to instructors by allowing modifications in how students join lessons.
Medium	*Notifications and Reminders*	Ensures students and instructors stay updated with lesson schedules and important updates.
Medium	*Error Handling*	Enhances system reliability by identifying and resolving inconsistencies in student or lesson data.

2.3.3. Use Case Specifications

USE CASE ID: UC1
USE CASE: Send Messages to Instructor
ACTORS: Student
PRECONDITIONS: • The student must be logged into the system. • The messaging feature must be enabled.
MAIN FLOW: 1. The student navigates to the messaging section. 2. The student selects the instructor to message. 3. The student composes and sends a message.
POST CONDITIONS: 4. The message is sent successfully, and the instructor can respond.
ALTERNATIVE FLOW: 4A. If the instructor is offline, the system stores the message for later delivery. 4B. If the message fails to send, an error message is displayed.

USE CASE ID: UC2
USE CASE: View Personal Schedule
ACTORS: Student
PRECONDITIONS: • The student must be logged in.
MAIN FLOW: 1. The student accesses the schedule section. 2. The system retrieves and displays their schedule.
POST CONDITIONS: 3. The student can view and manage their lesson schedule.
ALTERNATIVE FLOW: 3A. If no lessons are scheduled, the system displays an empty schedule. 3B. If access is restricted due to technical issues, an error message is shown.

USE CASE ID: UC3
USE CASE: Scheduling
ACTORS: Instructor, Admin
PRECONDITIONS: • The instructor or admin must be logged into the system.
MAIN FLOW: 1. The instructor/admin accesses the scheduling section. 2. They select a date, time, and lesson details. 3. The system validates the input and saves the schedule.
POST CONDITIONS: 4. The schedule is saved and notifications are sent if necessary.
ALTERNATIVE FLOW: 4A. If the selected time conflicts with another event, the system suggests an alternative. 4B. If mandatory fields are missing, the system prompts for completion.

USE CASE ID: UC4
USE CASE: Add Lesson Notes
ACTORS: Instructor
PRECONDITIONS: • The instructor must be logged in.
MAIN FLOW: 1. The instructor accesses the lesson notes section. 2. The instructor writes or edits lesson notes. 3. The system saves the notes.
POST CONDITIONS: 4. The notes are stored and accessible for future reference.
ALTERNATIVE FLOW: 4A. If the connection is lost, the notes are saved as a draft. 4B. If the instructor cancels before saving, no changes are applied.

USE CASE ID: UC5
USE CASE: Grade Assignments
ACTORS: Instructor
PRECONDITIONS: • The instructor must have grading permissions. • The student must have submitted an assignment.
MAIN FLOW: 1. The instructor navigates to the grading section. 2. The instructor reviews the student's submission. 3. The instructor assigns a grade and provides feedback. 4. The system saves the grade.
POST CONDITIONS: 5. The student's grade is recorded and visible to them.
ALTERNATIVE FLOW: 5A. If the assignment is missing, the system displays an error. 5B. If grading permissions are restricted, access is denied.

USE CASE ID: UC6
USE CASE: Send Messages to Students
ACTORS: Instructor
PRECONDITIONS: • The instructor must be logged in. • The student must be enrolled in a lesson.
MAIN FLOW: 1. The instructor selects a student from the messaging section. 2. The instructor composes and sends a message.
POST CONDITIONS: 3. The message is sent successfully, and the student can respond.
ALTERNATIVE FLOW: 3A. If the student is offline, the system stores the message for later delivery. 3B. If the message fails to send, an error message is displayed.

USE CASE ID: UC7
USE CASE: Manage Courses
ACTORS: Actor, Instructor
PRECONDITIONS: • The user must have course management permissions.
MAIN FLOW: 1. The user accesses the course management section. 2. They create, edit, or delete a course. 3. The system updates course information accordingly.
POST CONDITIONS: 4. The course catalog is updated.
ALTERNATIVE FLOW: 4A. If the user lacks permissions, access is denied. 4B. If course details are incomplete, an error message appears.

USE CASE ID: UC8
USE CASE: Manage Users
ACTORS: Admin
PRECONDITIONS: • The user must have administrative privileges.
MAIN FLOW: 1. The admin accesses the user management section. 2. The admin adds, edits, or removes user accounts. 3. The system updates user information accordingly.
POST CONDITIONS: 4. The user database is updated successfully.
ALTERNATIVE FLOW: 4A. If the admin lacks sufficient privileges, access is denied. 4B. If required user details are missing, an error message is displayed.

USE CASE ID: UC9
USE CASE: Limit Access
ACTORS: Admin
PRECONDITIONS: • The user must be an admin.
MAIN FLOW: 1. The admin selects a user account. 2. The admin modifies the access level.
POST CONDITIONS: 3. The access level is updated in the system.
ALTERNATIVE FLOW: 3A. If the user does not have admin privileges, access modification is denied. 3B. If an error occurs, changes are not saved.

USE CASE ID: UC10
USE CASE: Include a Messaging Feature
ACTORS: Admin
PRECONDITIONS: • The system must have messaging capabilities.
MAIN FLOW: 1. The admin enables the messaging feature. 2. Users can send and receive messages.
POST CONDITIONS: 3. Messaging is enabled and functional.
ALTERNATIVE FLOW: 3A. If the feature is disabled, users cannot access messaging. 3B. If a technical issue occurs, messages are not sent.

USE CASE ID: UC11
USE CASE: Send Notifications
ACTORS: System
PRECONDITIONS: • There must be an upcoming event or update.
MAIN FLOW: 1. The system identifies relevant notifications. 2. The system sends notifications to users.
POST CONDITIONS: 3. Users receive necessary notifications.
ALTERNATIVE FLOW: 3A. If a user has disabled notifications, they do not receive alerts. 3B. If sending fails, the system retries later.

USE CASE ID: UC12
USE CASE: Prevent Double-Booking
ACTORS: System
PRECONDITIONS: • A lesson must be scheduled.
MAIN FLOW: 1. The system checks for scheduling conflicts. 2. If no conflicts exist, the lesson is scheduled.
POST CONDITIONS: 3. Double-booking is prevented.
ALTERNATIVE FLOW: 3A. If a conflict exists, the system prompts for a new time.

USE CASE ID: UC13
USE CASE: Store Data in One Profile
ACTORS: System
PRECONDITIONS: • The user must have an account.
MAIN FLOW: 1. The system consolidates all user data into one profile. 2. Updates are stored in the profile.
POST CONDITIONS: 3. User data remains centralized.
ALTERNATIVE FLOW: 3A. If data is incomplete, the system requests missing information.

USE CASE ID: UC14
USE CASE: Receive Notifications
ACTORS: Instructor, Student, Admin
PRECONDITIONS: • The user must be registered for notifications.
MAIN FLOW: 1. The system checks for new notifications. 2. The system sends notifications for upcoming lessons, schedule changes, or important updates.
POST CONDITIONS: 3. The user receives the notification and can take necessary actions.
ALTERNATIVE FLOW: 3A. If the user has disabled notifications, no alerts are sent. 3B. If the notification fails to send, the system retries after a delay.

USE CASE ID: UC15
USE CASE: Work on Computers and Mobile Devices
ACTORS: System
PRECONDITIONS: • The system must be accessible on multiple devices.
MAIN FLOW: Students and instructors log in from mobile devices, while the admin logs in from a computer. The system adapts to the screen size.
POST CONDITIONS: The system provides a responsive experience.
ALTERNATIVE FLOW: 3A. If the device is unsupported, an error message is displayed.

3. Non-Functional Requirements

- The system shall respond to user actions (such as viewing a schedule or saving a note) within 2 seconds for 95% of all interactions under normal usage conditions.
- The system shall be accessible and operational at least 99.5% of the time per calendar month, excluding planned maintenance periods.
- At least 90% of new users shall be able to complete core tasks (e.g., scheduling a lesson, entering notes) without training, within 5 minutes of first use.
- The system shall implement role-based access control, ensuring that only authorized users can access, modify, or delete sensitive data such as student records and lesson schedules.
- The system shall be designed in a modular structure and documented well enough that developers can implement updates or bug fixes within 2 working days.
- The system shall be fully compatible with all modern web browsers (e.g., Chrome, Firefox, Safari, Edge) and mobile operating systems (iOS and Android) without functional limitations.
- The system shall encrypt all sensitive user data both in transit and at rest using industry-standard encryption protocols to ensure compliance with GDPR and other relevant data protection regulations.
- The system shall log all critical errors and security-related events in a centralized logging system.
- The system shall give instant notification for every transaction.

3.1 VOLERE Template

Requirement ID: NFR1	**Requirement Type:** Non-functional – Performance
Description: The system shall respond to user actions within 2 seconds for 95% of all interactions under normal usage conditions.	
Rational: Ensures efficient user experience and prevents delays during daily operations.	
Fit Criteria: Measured using performance testing tools; 95% of user interactions must complete in ≤2 seconds.	
Priority: Necessary	

Requirement ID: NFR2	**Requirement Type:** Non-functional – Avaliability
Description: The system shall be available at least 99.5% of the time per calendar month, excluding planned maintenance.	
Rational: Provides reliable system access for instructors and administrators.	
Fit Criteria: Uptime logs will show minimum 99.5% monthly availability.	
Priority: Necessary	

Requirement ID: NFR3	**Requirement Type:** Non-functional – Usability
Description: At least 90% of first-time users shall complete basic operations (e.g., creating a schedule) within 5 minutes without external help.	
Rational: Ensures ease of use for non-technical users like instructors.	
Fit Criteria: Results from usability testing sessions must meet the 90% threshold.	
Priority: Valuable	

Requirement ID: NFR4	**Requirement Type:** Non-functional – Security
Description: The system shall restrict access to sensitive data through role-based access control.	
Rational: Prevents unauthorized users from accessing or altering important records.	
Fit Criteria: Security audit logs confirm proper enforcement of access rules.	
Priority: Necessary	

Requirement ID: NFR5	**Requirement Type:** Non-functional – Maintainability
Description: The system shall allow developers to fix bugs or make updates within 2 working days, using provided documentation.	
Rational: Quick maintenance minimizes system downtime.	
Fit Criteria: Maintenance tasks must be completed within 2 days in test environments.	
Priority: Desirable	

Requirement ID: NFR6	**Requirement Type:** Non-functional – Portability
The system shall function on all major browsers and mobile platforms (Chrome, Firefox, Safari, Edge, iOS, Android).	
Rational: Users access the system from various platforms.	
Fit Criteria: Compatibility testing must confirm full functionality across platforms.	
Priority: Valuable	

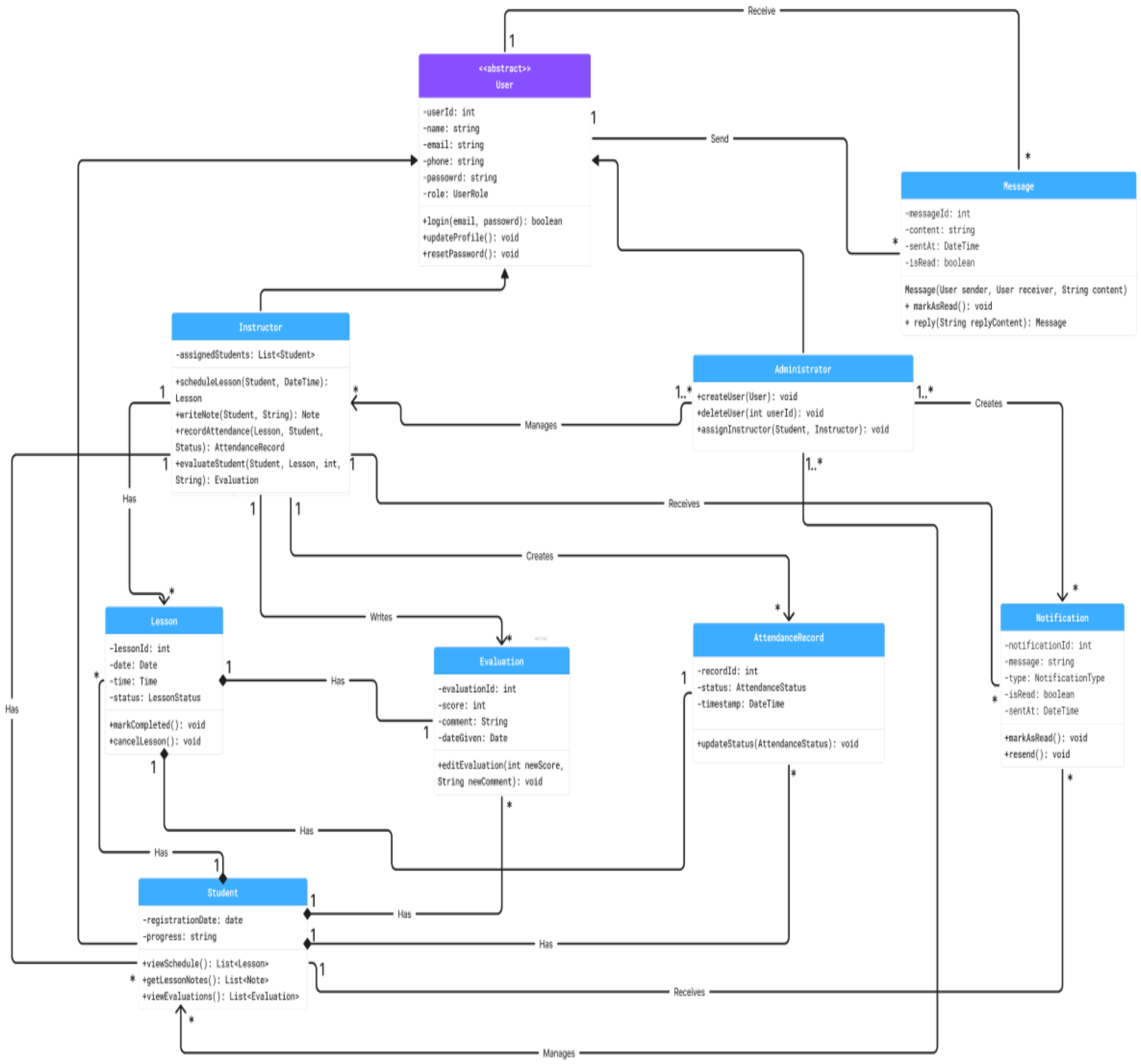
Requirement ID: NFR7	**Requirement Type:** Non-functional – Security & Complience
The system shall encrypt all sensitive user data both in transit and at rest using industry-standard encryption protocols to ensure compliance with GDPR and other relevant data protection regulations.	
Rational: Encrypting sensitive data prevents unauthorized access, ensuring user privacy and legal compliance (e.g., GDPR).	
Fit Criteria: The system shall use AES-256 encryption for stored data and TLS 1.2+ for data in transit, ensuring all sensitive information remains protected.	
Priority: High	

Requirement ID: NFR8	**Requirement Type:** Non-functional – Security & Maintenance
The system shall log all critical errors and security-related events in a centralized logging system.	
Rational: Centralized logging enables quick issue detection, security auditing, and regulatory compliance.	
Fit Criteria: The system shall capture and store all critical errors and security-related events in a centralized logging system, retaining logs for at least 12 months	
Priority: High	

Requirement ID: NFR9	**Requirement Type:** Non-functional – Performance & User Experience
The system shall give instant notification for every transaction.	
Rational: Instant notifications help users confirm their actions, reducing errors and improving efficiency.	
Fit Criteria: The system shall display a relevant notification within 1 second after an action is performed	
Priority: High	

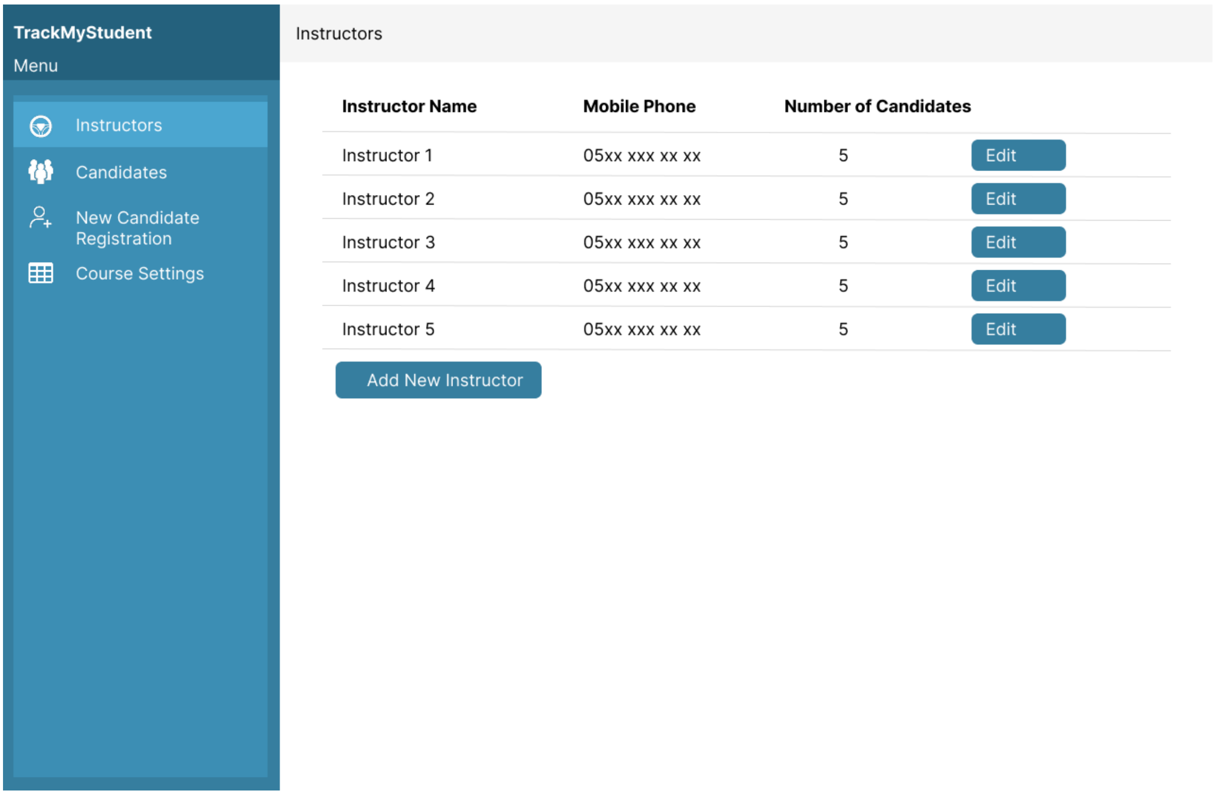
4. System Models

4.1 Class Model



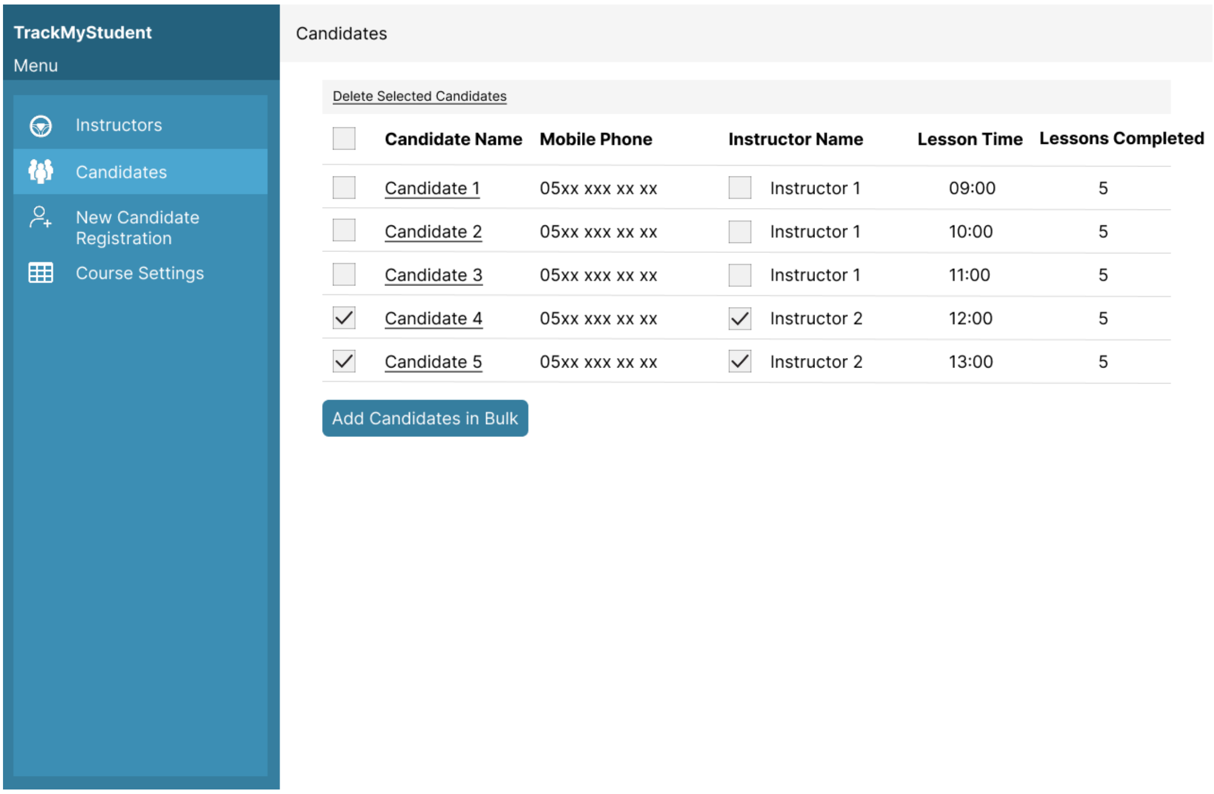
4.2 User Interface - Navigational Paths and Screen Mock-Ups

4.2.1 Administrator



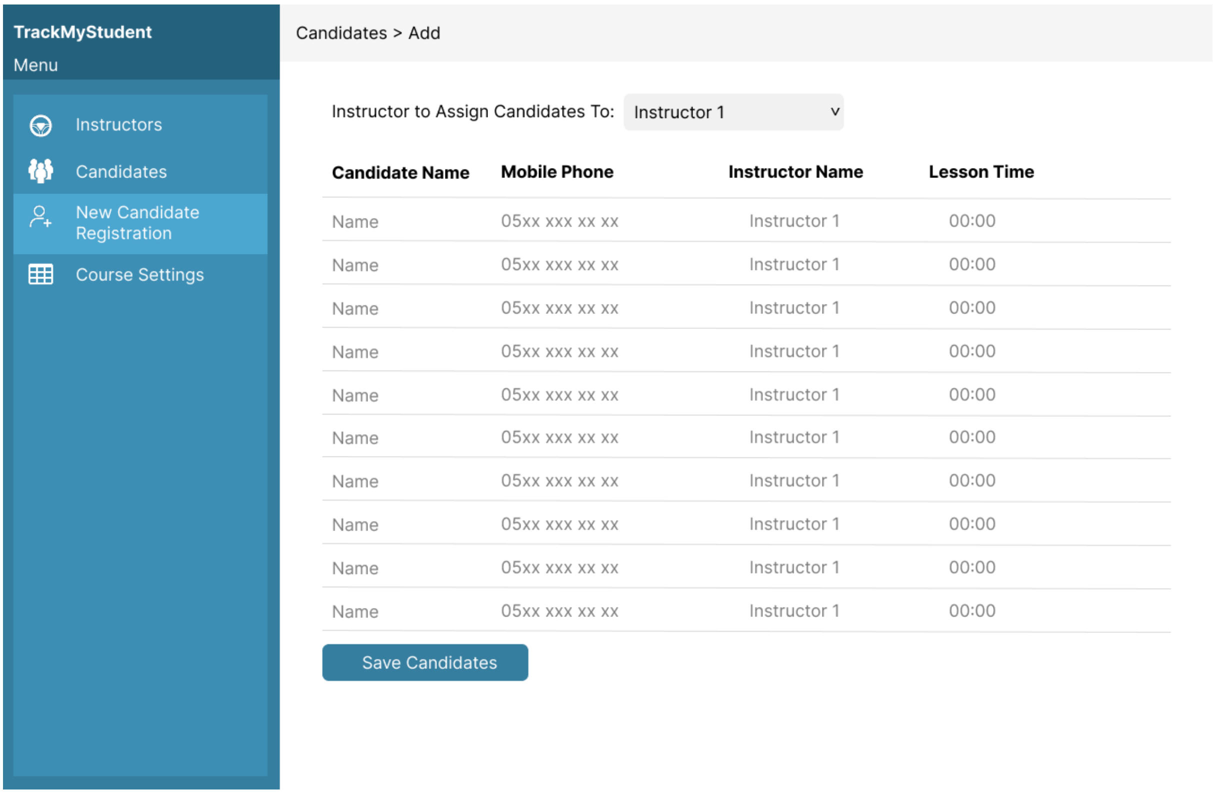
Instructors

Instructor Name	Mobile Phone	Number of Candidates	
Instructor 1	05xx xxx xx xx	5	Edit
Instructor 2	05xx xxx xx xx	5	Edit
Instructor 3	05xx xxx xx xx	5	Edit
Instructor 4	05xx xxx xx xx	5	Edit
Instructor 5	05xx xxx xx xx	5	Edit



Candidates

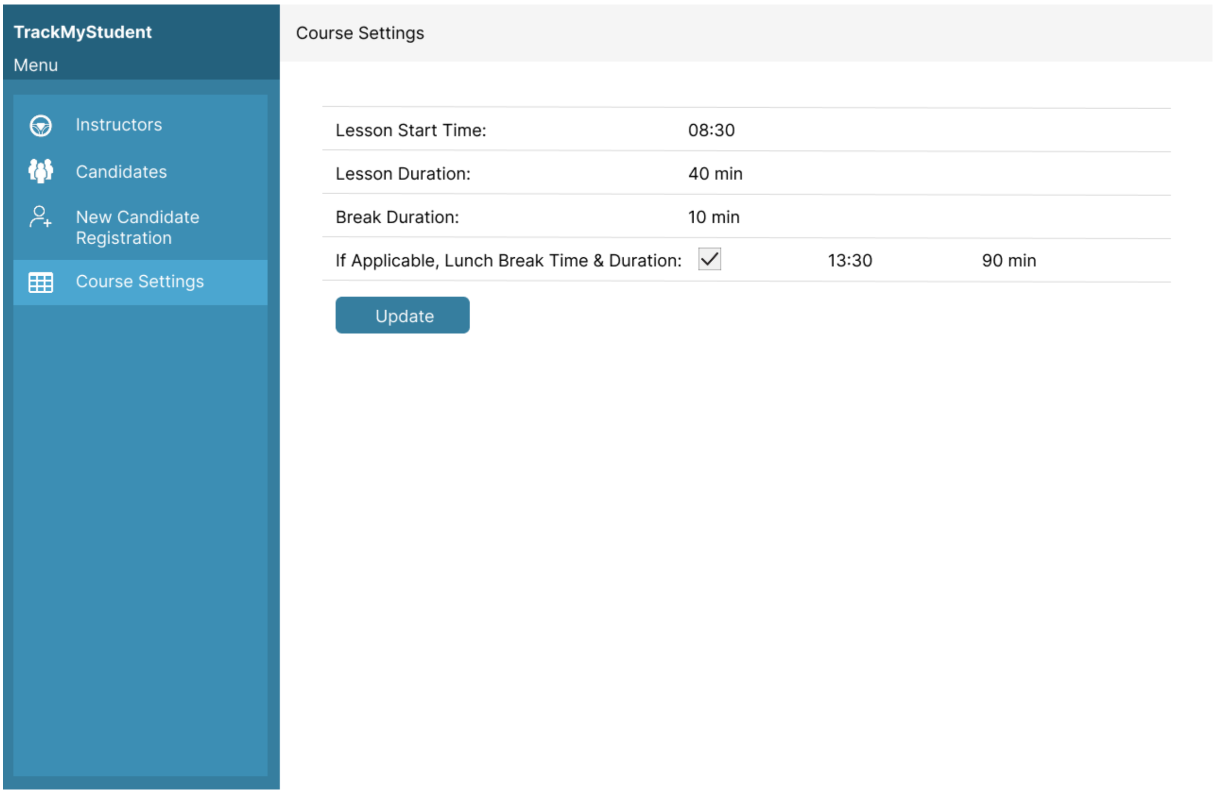
□	Candidate Name	Mobile Phone		Instructor Name	Lesson Time	Lessons Completed
□	Candidate 1	05xx xxx xx xx	□	Instructor 1	09:00	5
□	Candidate 2	05xx xxx xx xx	□	Instructor 1	10:00	5
□	Candidate 3	05xx xxx xx xx	□	Instructor 1	11:00	5
☑	Candidate 4	05xx xxx xx xx	☑	Instructor 2	12:00	5
☑	Candidate 5	05xx xxx xx xx	☑	Instructor 2	13:00	5



Candidates > Add

Instructor to Assign Candidates To: Instructor 1

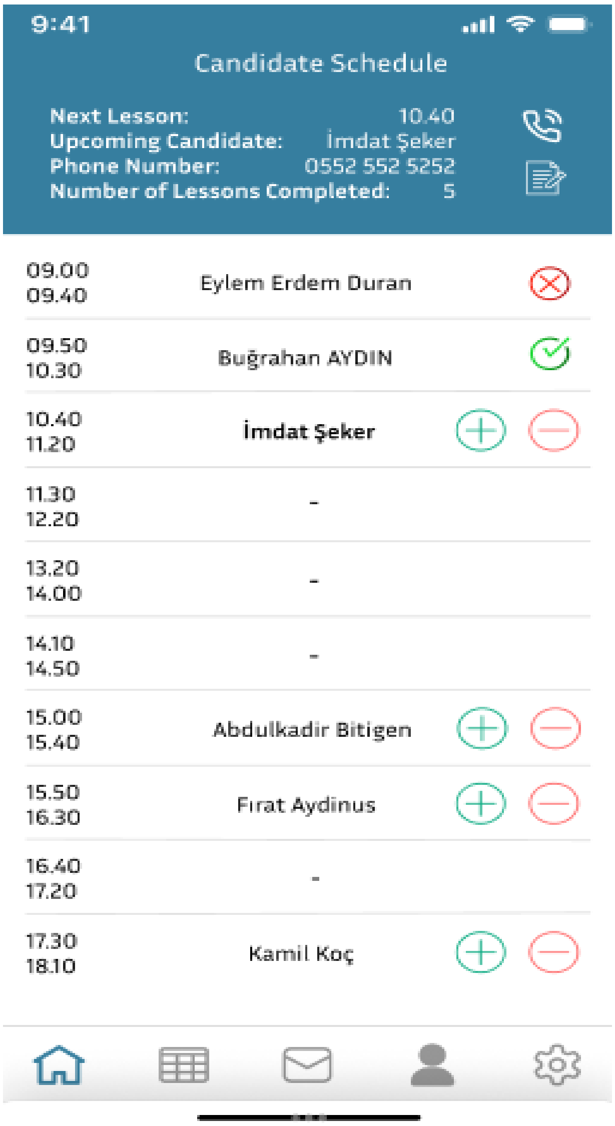
Candidate Name	Mobile Phone	Instructor Name	Lesson Time
Name	05xx xxx xx xx	Instructor 1	00:00
Name	05xx xxx xx xx	Instructor 1	00:00
Name	05xx xxx xx xx	Instructor 1	00:00
Name	05xx xxx xx xx	Instructor 1	00:00
Name	05xx xxx xx xx	Instructor 1	00:00
Name	05xx xxx xx xx	Instructor 1	00:00
Name	05xx xxx xx xx	Instructor 1	00:00
Name	05xx xxx xx xx	Instructor 1	00:00
Name	05xx xxx xx xx	Instructor 1	00:00
Name	05xx xxx xx xx	Instructor 1	00:00



Course Settings

Lesson Start Time:	08:30		
Lesson Duration:	40 min		
Break Duration:	10 min		
If Applicable, Lunch Break Time & Duration:	☑	13:30	90 min

4.2.2 Instructor

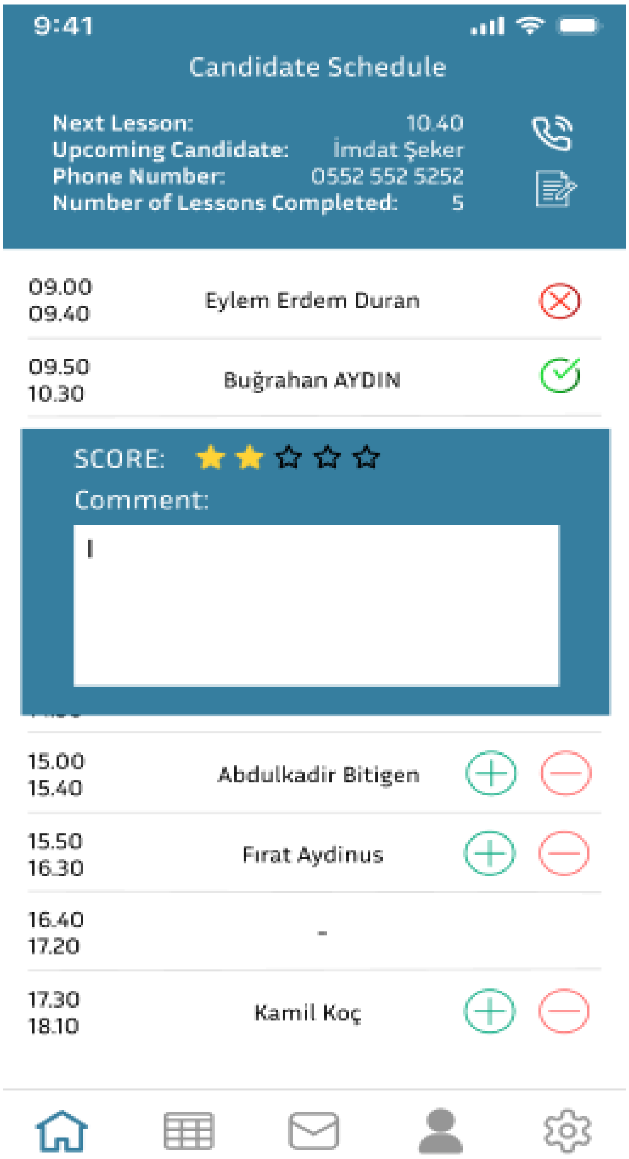


9:41

Candidate Schedule

Next Lesson: 10.40
Upcoming Candidate: İmdat Şeker
Phone Number: 0552 552 5252
Number of Lessons Completed: 5

09.00 09.40	Eylem Erdem Duran	⊗
09.50 10.30	Buğrahan AYDIN	✓
10.40 11.20	**İmdat Şeker**	⊕ ⊖
11.30 12.20	-	
13.20 14.00	-	
14.10 14.50	-	
15.00 15.40	Abdulkadir Bitigen	⊕ ⊖
15.50 16.30	Fırat Aydinus	⊕ ⊖
16.40 17.20	-	
17.30 18.10	Kamil Koç	⊕ ⊖



9:41

Candidate Schedule

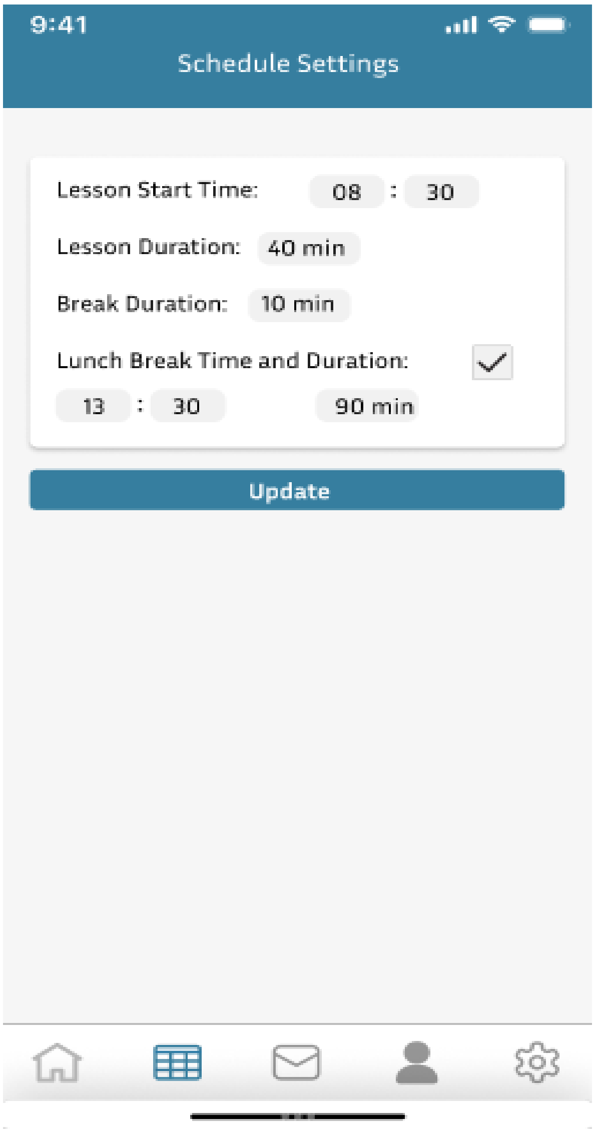
Next Lesson: 10.40
Upcoming Candidate: İmdat Şeker
Phone Number: 0552 552 5252
Number of Lessons Completed: 5

09.00 09.40	Eylem Erdem Duran	⊗
09.50 10.30	Buğrahan AYDIN	✓

SCORE: ★★☆☆☆

Comment:

15.00 15.40	Abdulkadir Bitigen	⊕ ⊖
15.50 16.30	Fırat Aydinus	⊕ ⊖
16.40 17.20	-	
17.30 18.10	Kamil Koç	⊕ ⊖



Schedule Settings

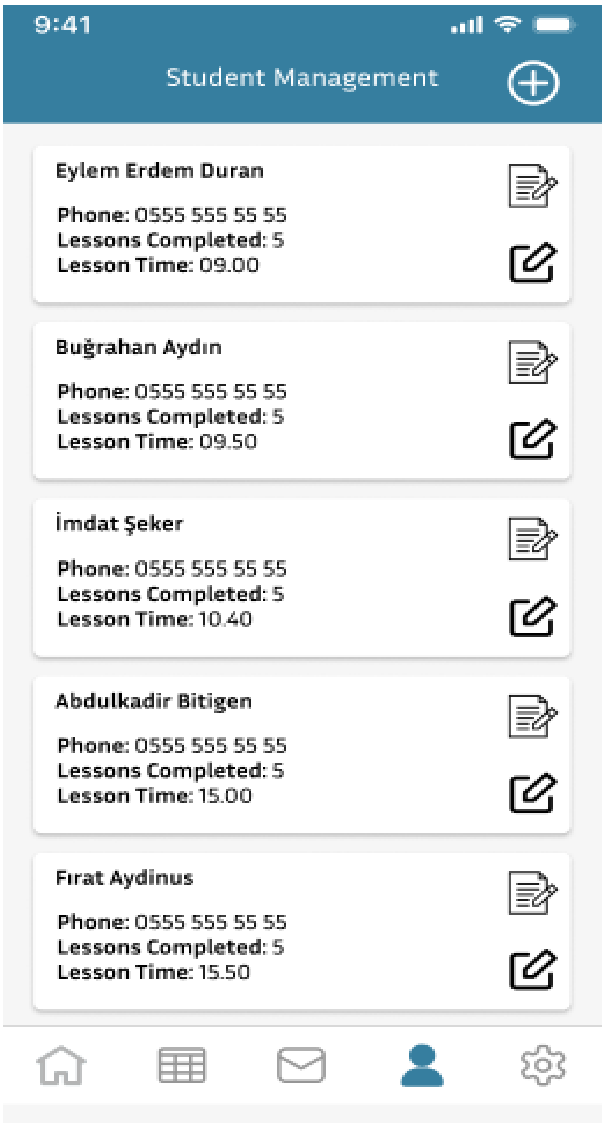
Lesson Start Time: 08 : 30

Lesson Duration: 40 min

Break Duration: 10 min

Lunch Break Time and Duration: ☑

13 : 30 90 min



Student Management

Eylem Erdem Duran

Phone: 0555 555 55 55
Lessons Completed: 5
Lesson Time: 09.00

Buğrahan Aydın

Phone: 0555 555 55 55
Lessons Completed: 5
Lesson Time: 09.50

İmdat Şeker

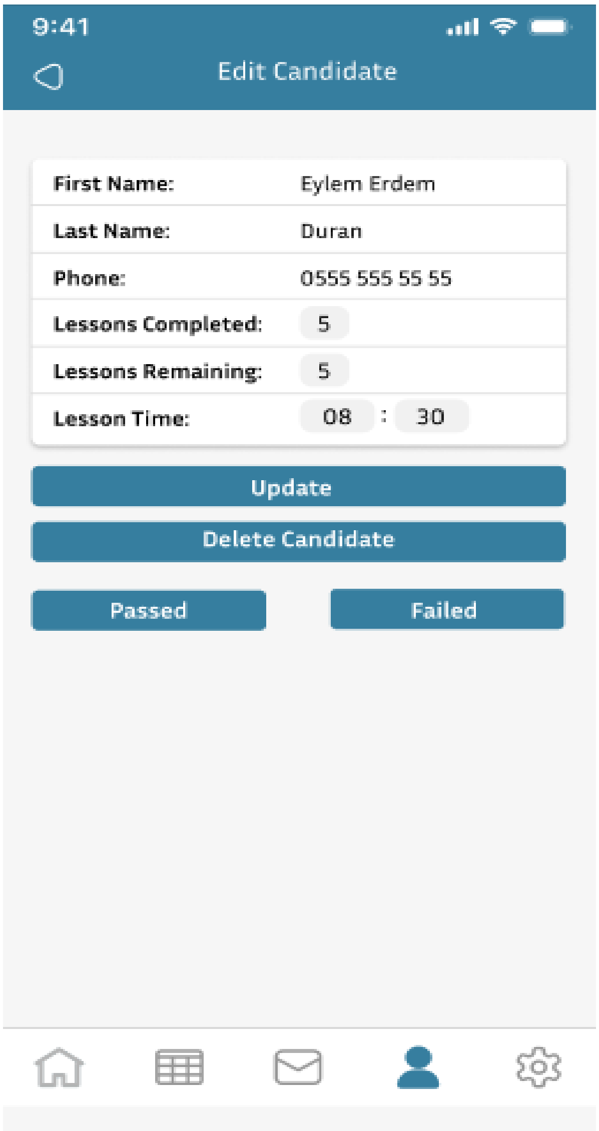
Phone: 0555 555 55 55
Lessons Completed: 5
Lesson Time: 10.40

Abdulkadir Bitigen

Phone: 0555 555 55 55
Lessons Completed: 5
Lesson Time: 15.00

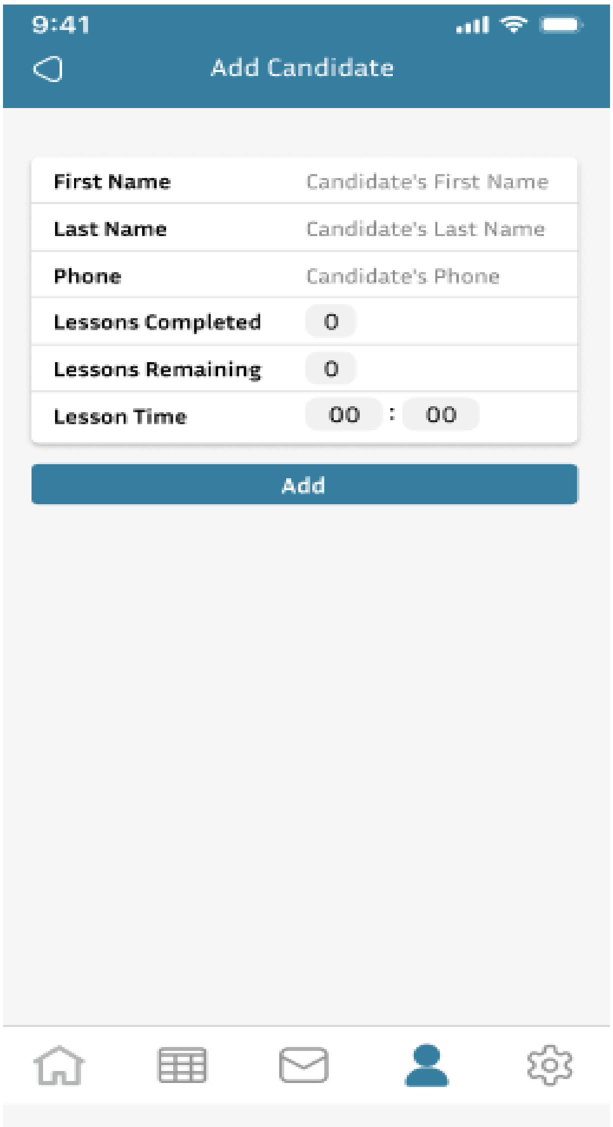
Fırat Aydinus

Phone: 0555 555 55 55
Lessons Completed: 5
Lesson Time: 15.50

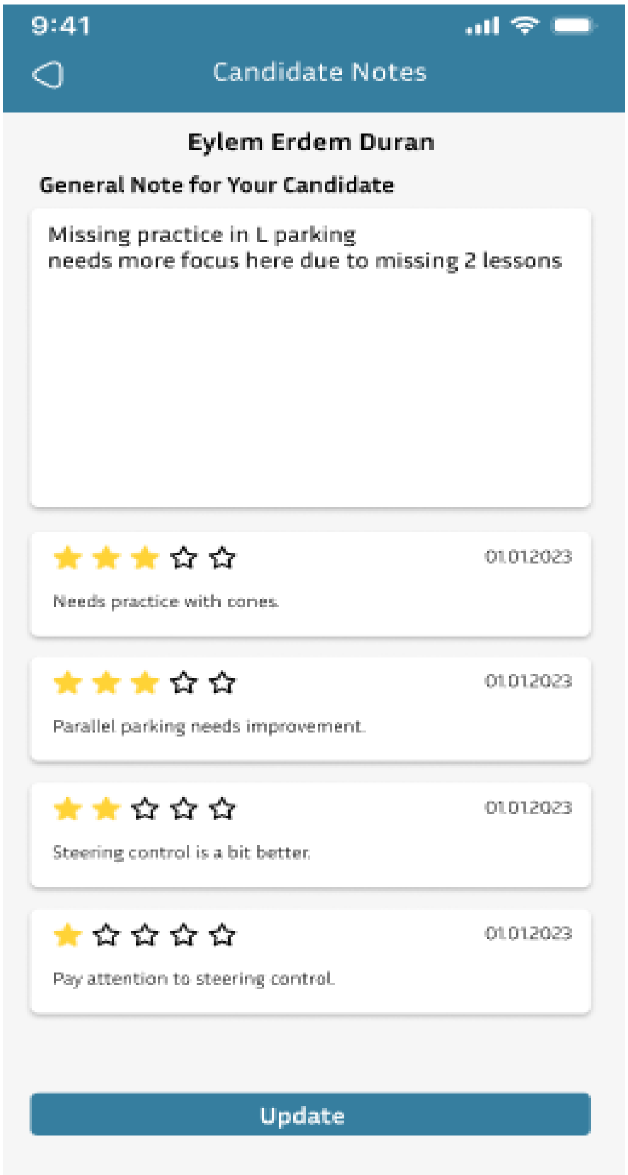


Edit Candidate

First Name:	Eylem Erdem
Last Name:	Duran
Phone:	0555 555 55 55
Lessons Completed:	5
Lessons Remaining:	5
Lesson Time:	08 : 30



First Name	Candidate's First Name
Last Name	Candidate's Last Name
Phone	Candidate's Phone
Lessons Completed	0
Lessons Remaining	0
Lesson Time	00 : 00



9:41

Candidate Notes

Eylem Erdem Duran

General Note for Your Candidate

Missing practice in L parking
needs more focus here due to missing 2 lessons

★★★☆☆ 01.012023

Needs practice with cones.

★★★☆☆ 01.012023

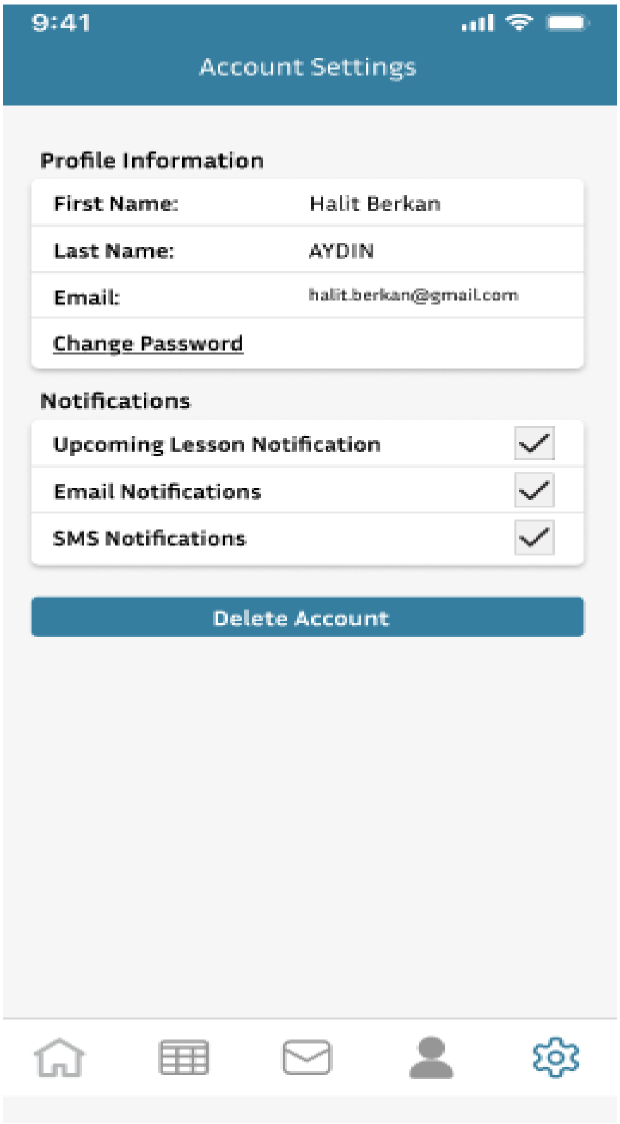
Parallel parking needs improvement.

★★☆☆☆ 01.012023

Steering control is a bit better.

★☆☆☆☆ 01.012023

Pay attention to steering control.



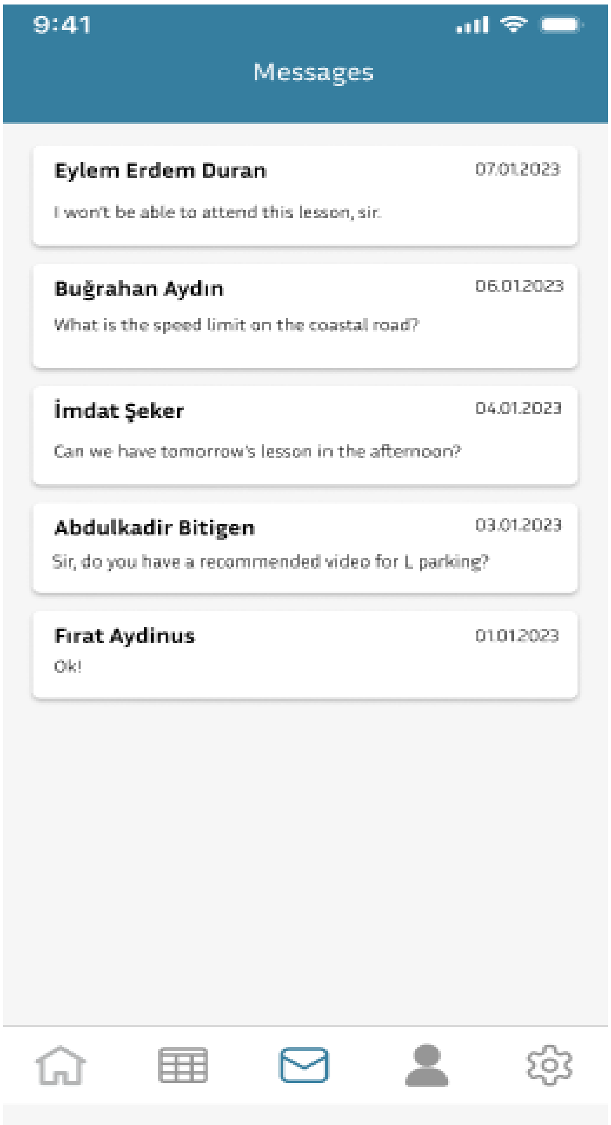
Account Settings

Profile Information

First Name:	Halit Berkan
Last Name:	AYDIN
Email:	halit.berkan@gmail.com
Change Password	

Notifications

Upcoming Lesson Notification	☑
Email Notifications	☑
SMS Notifications	☑



Messages

Eylem Erdem Duran 07.01.2023

I won't be able to attend this lesson, sir.

Buğrahan Aydın 06.01.2023

What is the speed limit on the coastal road?

İmdat Şeker 04.01.2023

Can we have tomorrow's lesson in the afternoon?

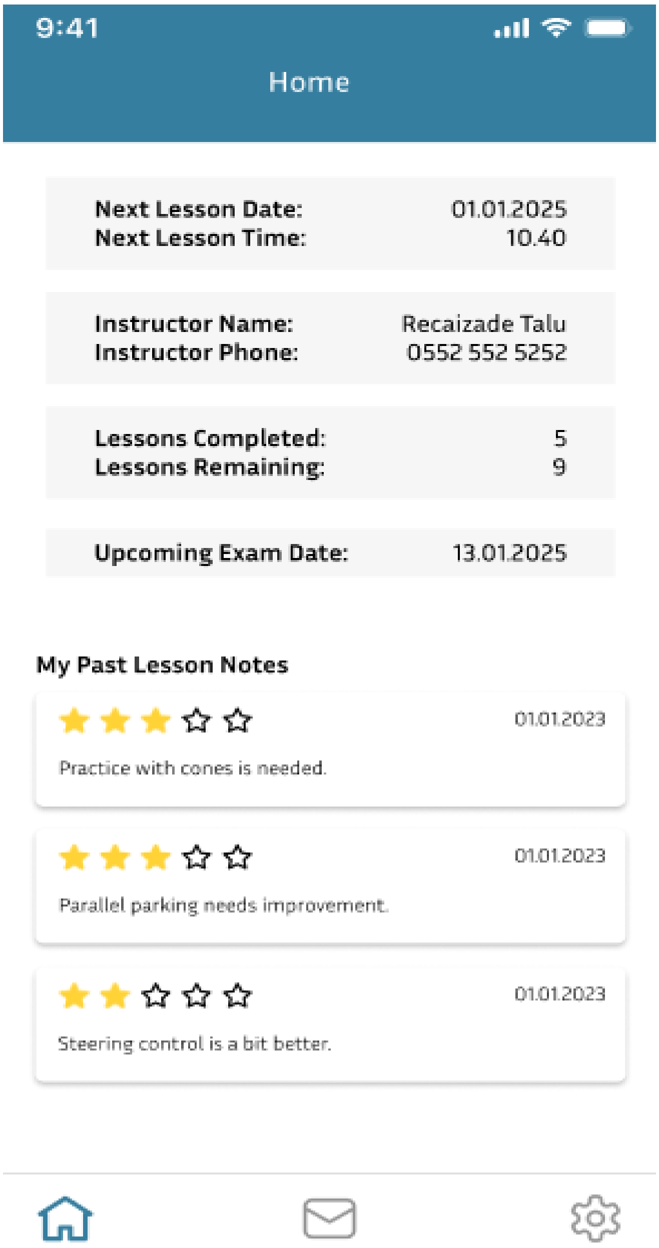
Abdulkadir Bitigen 03.01.2023

Sir, do you have a recommended video for L parking?

Fırat Aydinus 01.01.2023

Ok!

4.2.3 Student



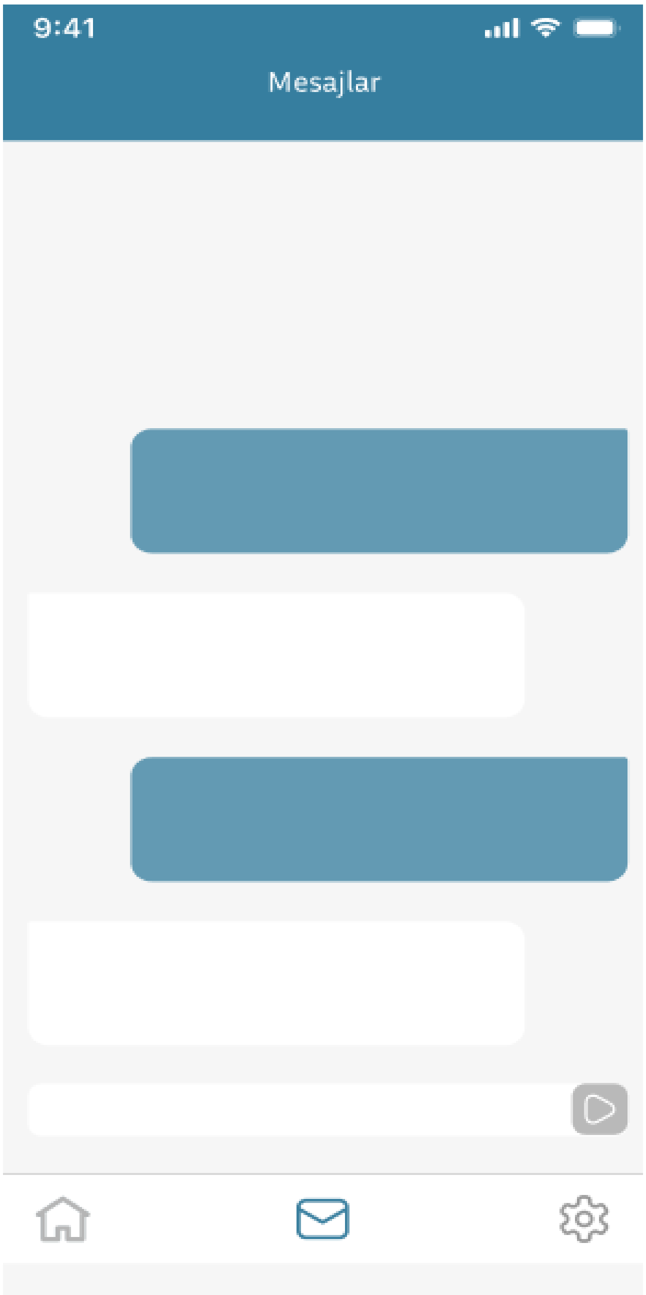
Next Lesson Date:	01.01.2025
Next Lesson Time:	10.40
Instructor Name:	Recaizade Talu
Instructor Phone:	0552 552 5252
Lessons Completed:	5
Lessons Remaining:	9
Upcoming Exam Date:	13.01.2025

My Past Lesson Notes

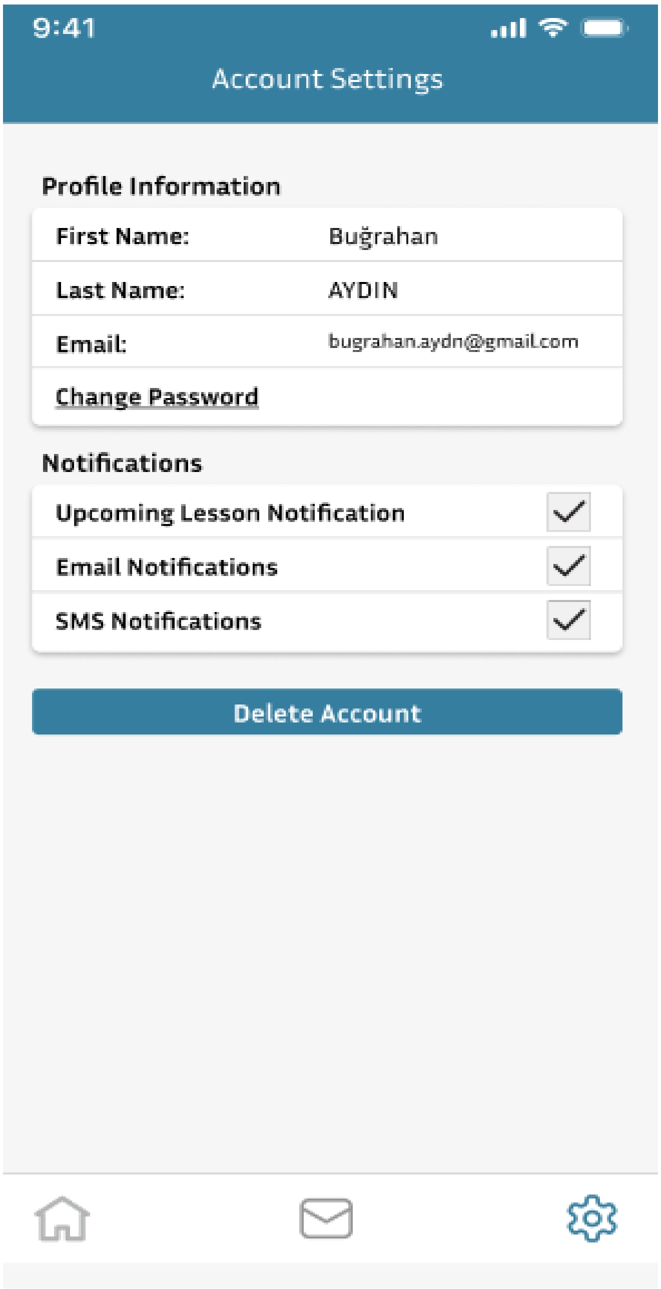
★★★☆☆ 01.01.2023
Practice with cones is needed.

★★★☆☆ 01.01.2023
Parallel parking needs improvement.

★★☆☆☆ 01.01.2023
Steering control is a bit better.



9:41
Mesajlar



9:41

Account Settings

Profile Information

First Name:	Buğrahan
Last Name:	AYDIN
Email:	bugrahan.aydn@gmail.com
Change Password	

Notifications

Upcoming Lesson Notification	☑
Email Notifications	☑
SMS Notifications	☑

Delete Account

5. Definitions, Acronyms and Abbreviations

This section lists and explains the abbreviations and technical terms used throughout the document:

- **UML**: Unified Modeling Language – a visual modeling language used to represent the structure and behavior of software systems.
- **GUI**: Graphical User Interface – an interface that allows users to interact with the system through visual elements.
- **CRUD**: Create, Read, Update, Delete – the four basic operations of persistent data management.
- **DTO**: Data Transfer Object – an object that carries data between processes to reduce method calls.
- **OOP**: Object-Oriented Programming – a programming paradigm based on the concept of objects and classes.
- **AES**: Advanced Encryption Standard – an encryption algorithm used to secure data.
- **TLS**: Transport Layer Security – a protocol for encrypting data transmitted over networks.
- **GDPR**: General Data Protection Regulation – a regulation in EU law on data protection and privacy.
- **RBAC**: Role-Based Access Control – a security mechanism that restricts system access based on user roles.
- **HTML/CSS/JS**: Technologies used to structure (HTML), style (CSS), and add interactivity (JavaScript) to web applications.
- **Notification**: A system-generated alert sent to users about lessons, schedule changes, or updates.
- **Message**: A two-way communication feature allowing users to exchange information in the system.

6. Glossary & References

6.1 Glossary

- **User**: A person registered in the system, either as a student, instructor, or administrator.
- **Instructor**: A user responsible for managing lessons, tracking student progress, and entering notes and evaluations.
- **Student**: A learner enrolled in the driving course who participates in lessons and assessments.
- **Admin**: The system administrator who oversees all users, schedules, records, and permissions.
- **Lesson**: A planned session where a student receives training from an instructor.
- **Attendance Record**: A log showing whether a student attended a specific lesson.
- **Evaluation**: A structured assessment of a student's performance, including grades and feedback.
- **Notification**: A system-generated message that informs users of upcoming lessons, updates, or warnings.
- **Message**: A user-to-user communication feature enabling direct text interaction within the system.
- **Dashboard**: The main page a user sees after logging in, which displays relevant information like schedules, messages, and tasks.
- **Volere Template**: A structured format for writing detailed and traceable software requirements.

6.2 References

1. Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.
2. Lucidchart UML Diagrams Documentation – https://www.lucidchart.com/pages/uml-diagram
3. ISO/IEC 25010:2011 – Systems and Software Quality Requirements and Evaluation.
4. Oracle Java Documentation – https://docs.oracle.com
5. OWASP Security Guidelines – https://owasp.org
6. GDPR Compliance Guidelines – https://gdpr.eu
7. MDN Web Docs (HTML, CSS, JavaScript) – https://developer.mozilla.org