SEG 2105 Report

OTAMS PROJECT

Submitted by

GROUP 13

Faculty of Engineering

Experiment Date: Tuesday, September 17, 2024
Submission Date: Saturday, September 21, 2024

Table of Contents

Introduction

GitHub Repository https://github.com/uOttawaSEG/project-project_group_13

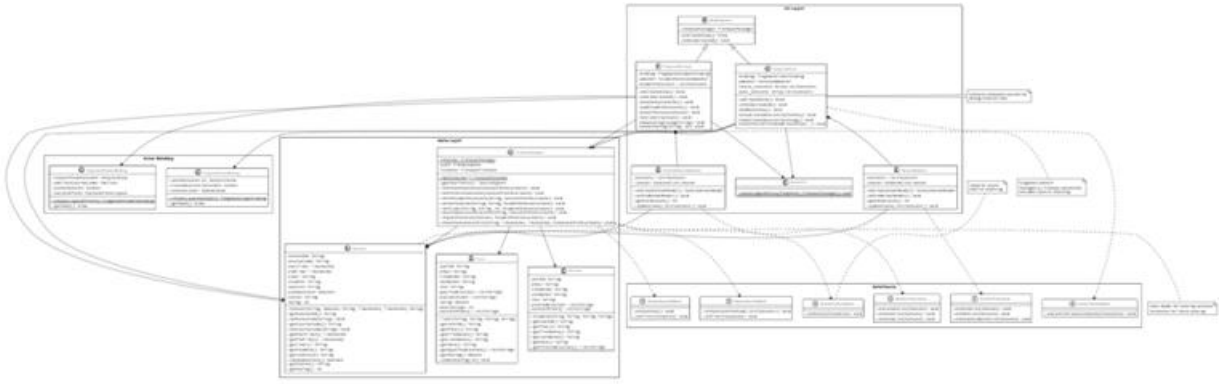
OTAMS (Online Tutoring Appointment Management System) is a comprehensive Android application designed to streamline the tutoring appointment process at the University of Ottawa Help Centre. The system supports three distinct user roles: Students, Tutors, and Administrators, each with specific functionalities tailored to their needs in the tutoring ecosystem.

Over four deliverables, we have incrementally developed a full-featured application that enables:

- Students to browse available tutoring slots, book/cancel sessions, and rate tutors
- Tutors to manage availability slots, approve/reject session requests, and track sessions
- Administrators to manage user registration and system oversight

This report documents the complete implementation of Deliverable 4, which focuses on Student features and system integration, building upon the foundation established in previous deliverables.

UML Class Diagram



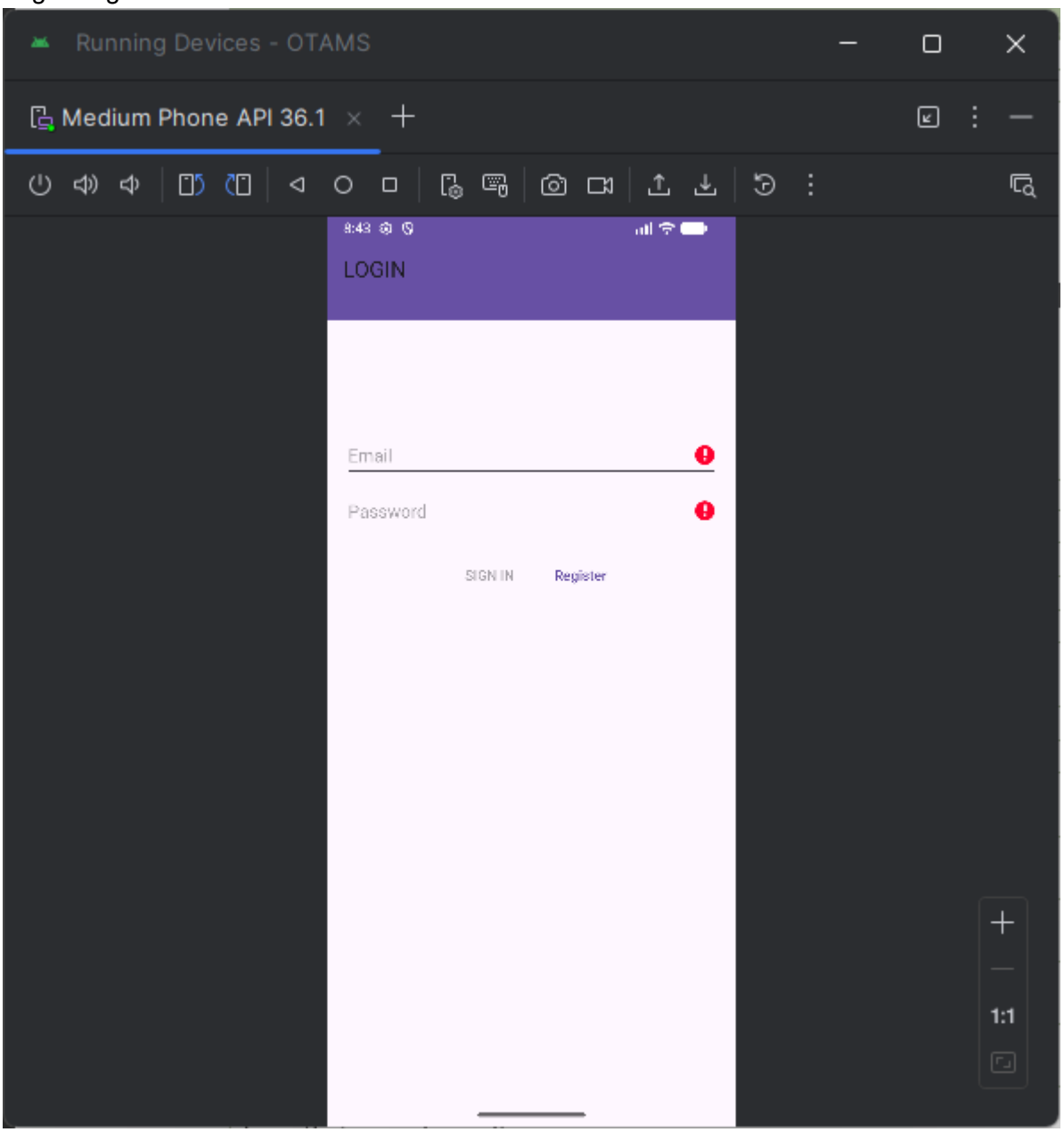
Team-Member Contribution

For team-member contribution; Forever-Uselesss (Joe) is the only person who worked on the app's code. The rest team-members did not create much pull requests or did not bother to do anything at all. Some made Joe's life miserable by suggesting bad naming scheme (and uml) and he decided not to push them to pull their weight in the project. The group chat was mostly dead, and many tasks given were not completed. That way Joe have a minimum viable product all made. Joe did all the work from setting up the GitHub landing page to organizing the app.

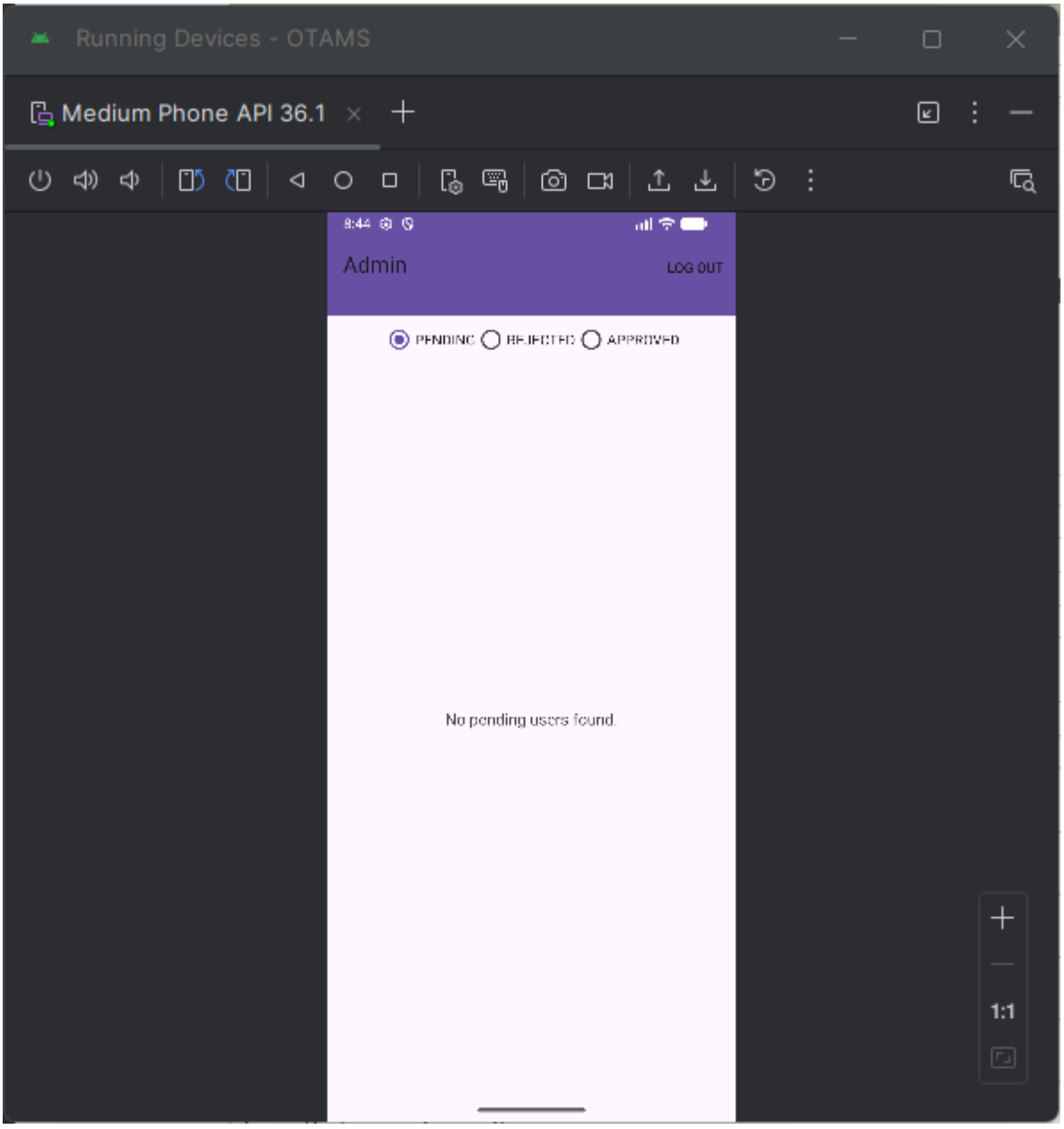
Please do grant the other team members' fair share as they helped a lot by staying out of Joe's way and not causing any more headaches. To note: Mani took up ui – did not pull in (Joe choose to use fragments and over wrote his); Douaa and Safa implemented the data structure which was heavily modified by Joe for use; Ziad also took up some part but Joe had to use a lot of effort to commit his name to the readme file; Yassin took up payment system and nobody knows why /: This group is just here because a group was needed.

App Screenshots

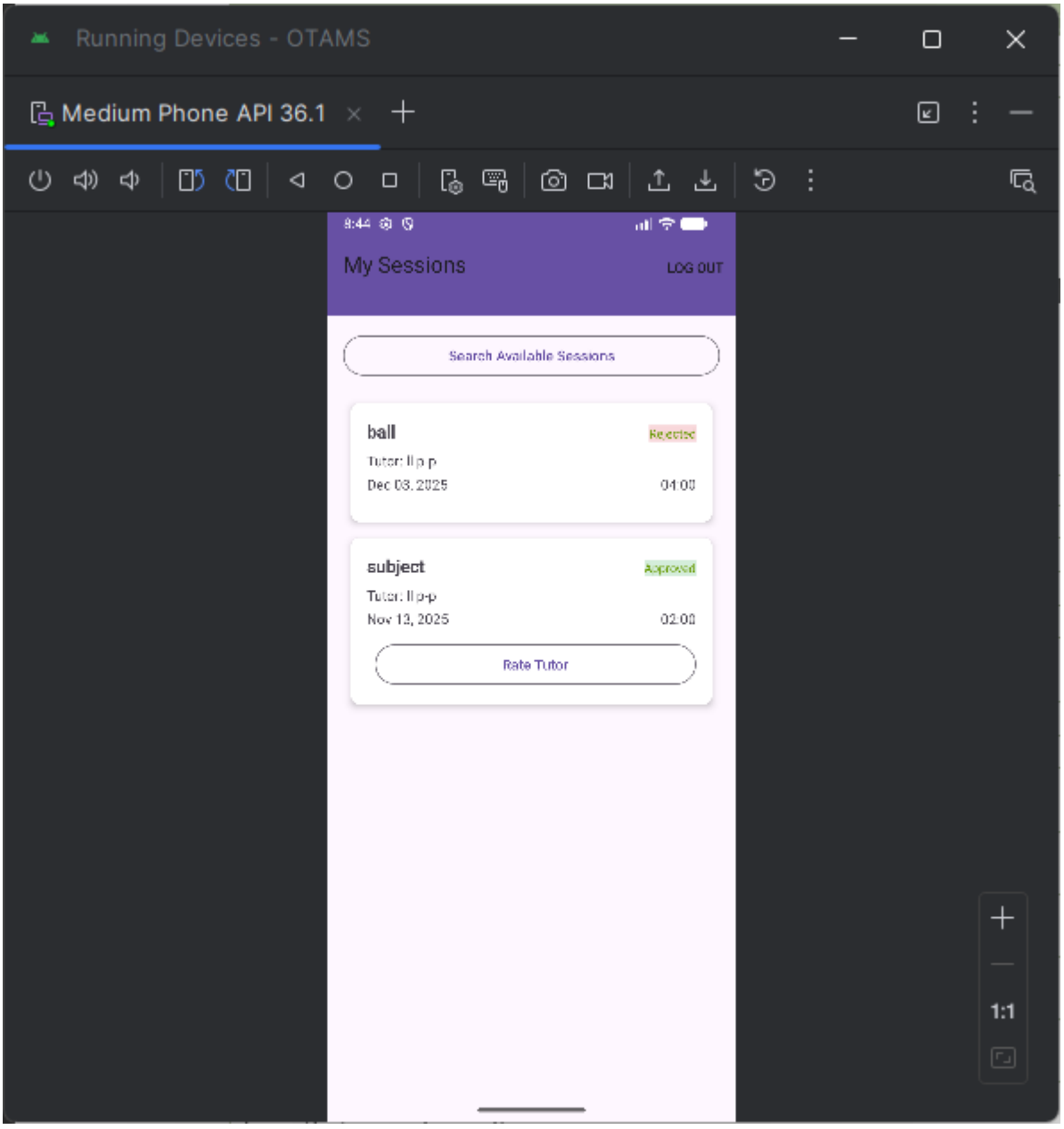
- Login Page

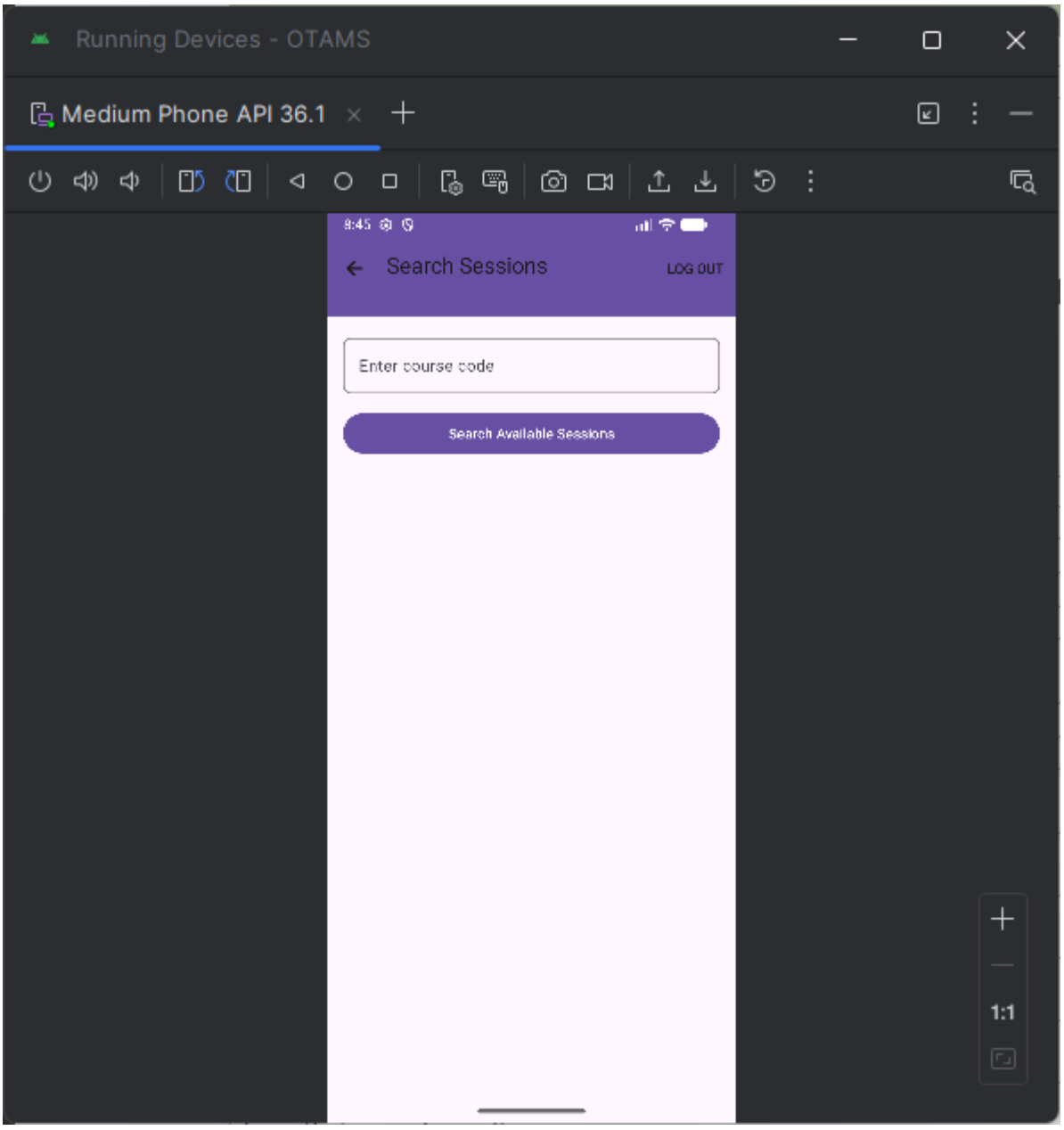


- Admin view

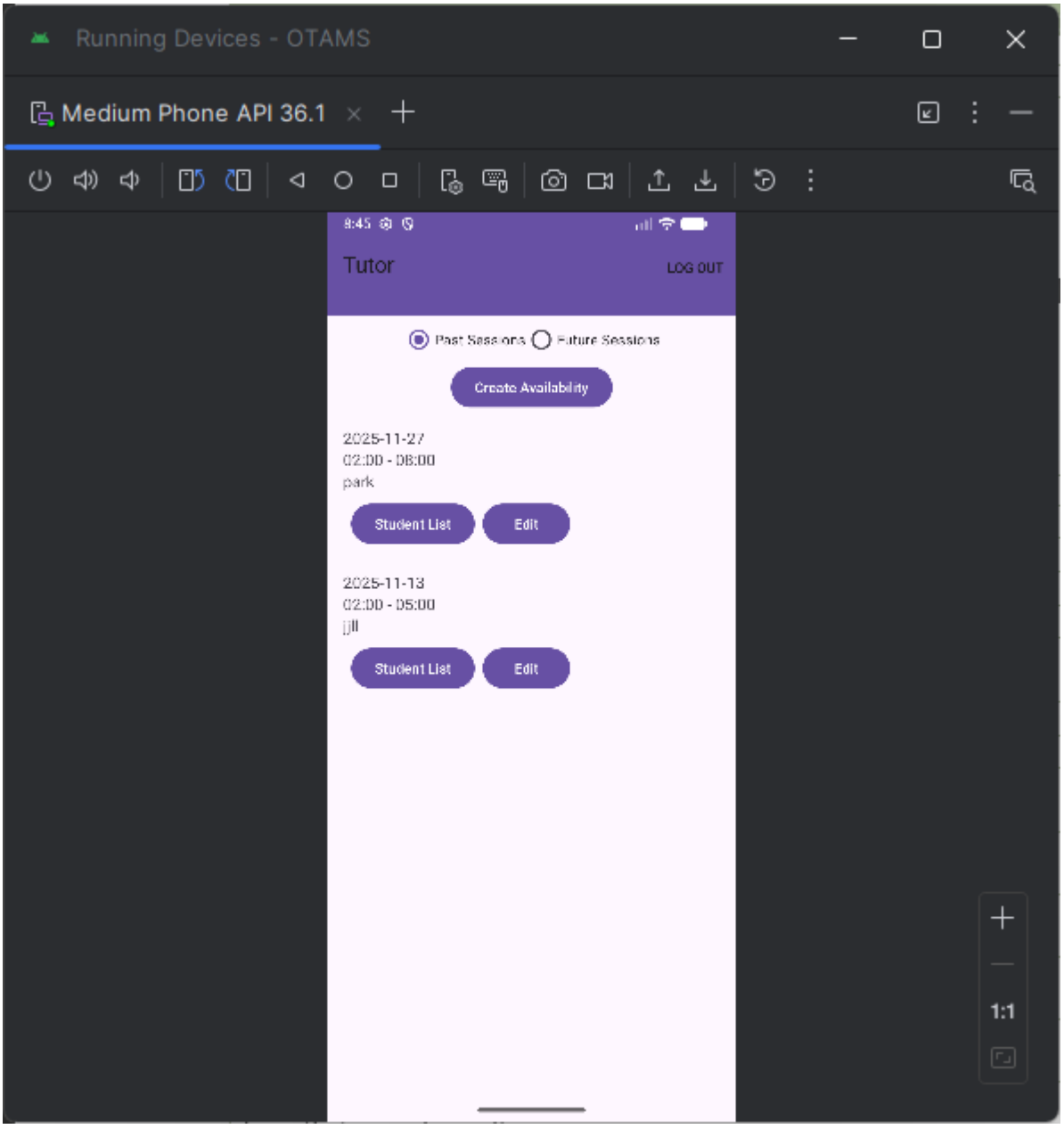

Running Devices - OTAMS
Medium Phone API 36.1
Admin
LOG OUT
PENDING
REJECTED
APPROVED
No pending users found.
1:1

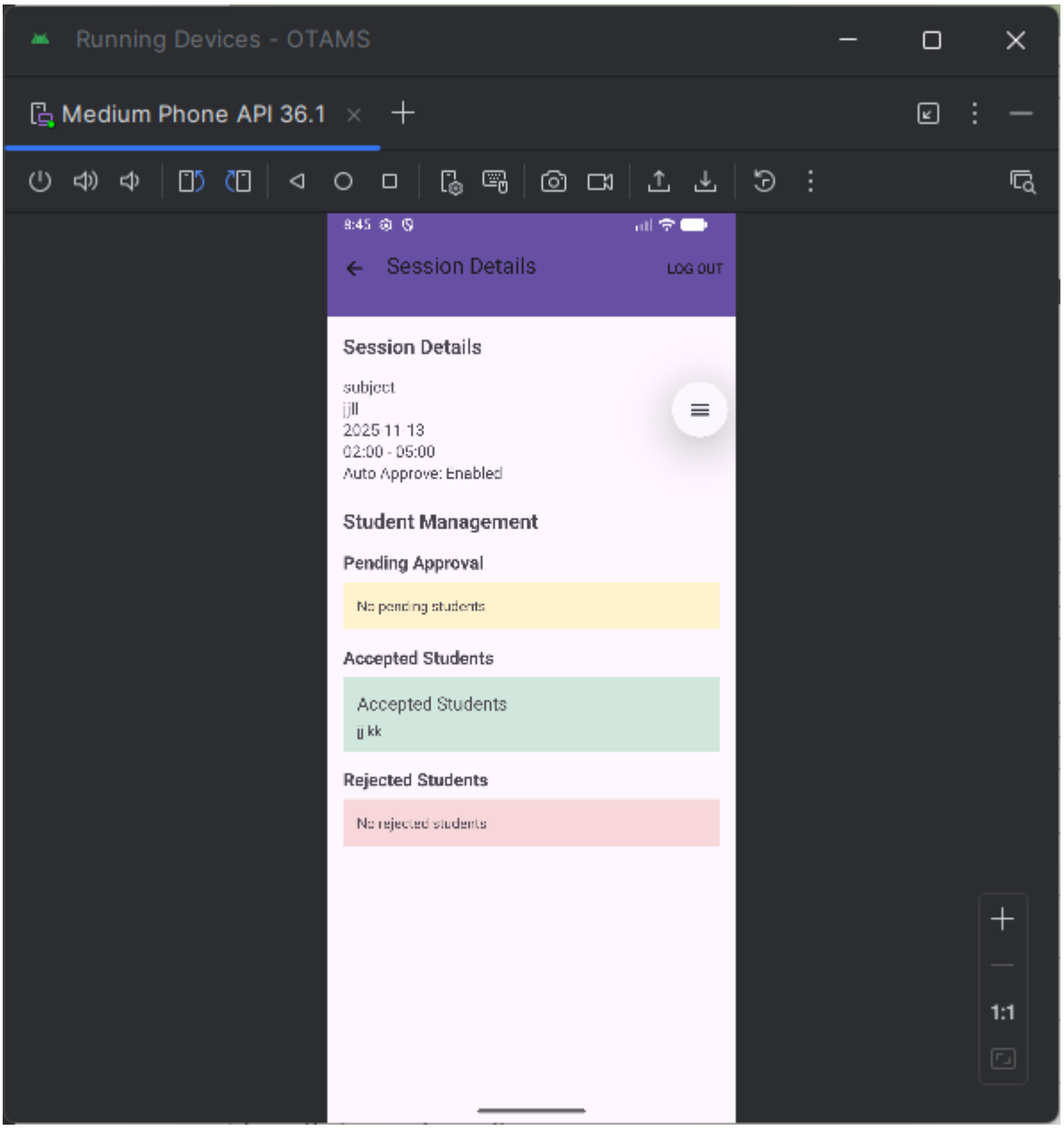
- Student view

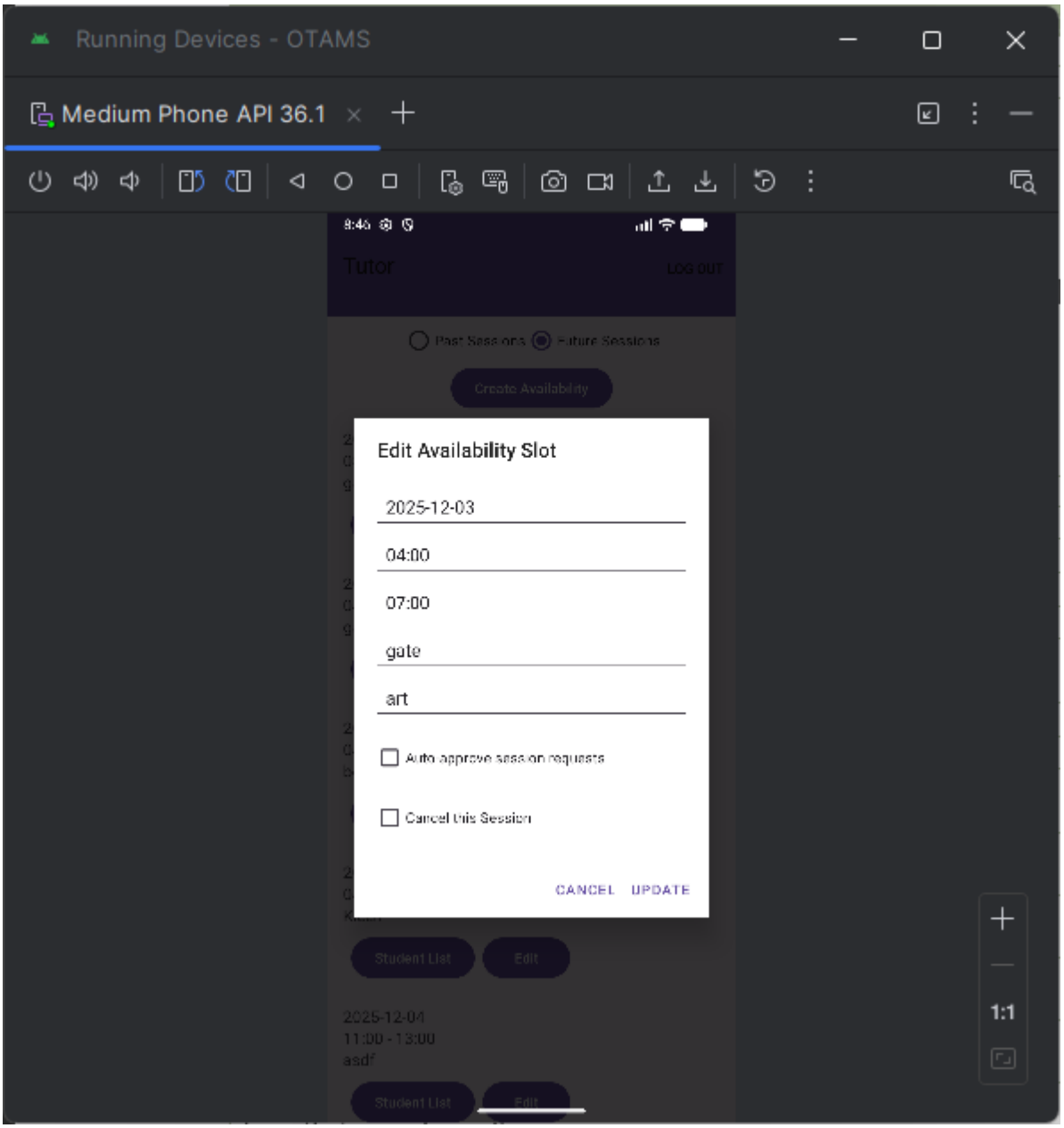



Running Devices - OTAMS
Medium Phone API 36.1
8:45
Search Sessions
LOG OUT
Enter course code
Search Available Sessions
1:1

- Tutor view


Running Devices - OTAMS
Medium Phone API 36.1
8:45
Tutor
LOG OUT
Past Sessions
Future Sessions
Create Availability
2025-11-27
02:00 - 08:00
park
Student List
Edit
2025-11-13
02:00 - 05:00
jjll
Student List
Edit
1:1


Running Devices - OTAMS
Medium Phone API 36.1
8:45
Session Details
LOG OUT
Session Details
subject
jjll
2025 11 13
02:00 - 05:00
Auto Approve: Enabled
Student Management
Pending Approval
No pending students
Accepted Students
Accepted Students
jj kk
Rejected Students
No rejected students
1:1



Technical Implementation Details

Database Schema

Collections:
- users: {userId, email, firstName, lastName, role, status, ...}
- sessions: {sessionId, courseCode, startTime, endTime, tutorId, studentId, status, ...}

Key Algorithms

Conflict Detection Algorithm: Checks for overlapping time slots using timestamp comparison
Rating Calculation: Weighted average of all Tutor ratings
Session Filtering: Real-time filtering based on course code and availability

Time Validation: 30-minute increment validation for slot creation

Security Measures

Password encryption using Firebase Authentication
Role-based access control
Input validation on all forms
Session validation before operations

Lessons Learned

During this project, I developed strong skills in Firebase integration by learning efficient data modelling techniques tailored for NoSQL databases. This experience allowed me to design data structures that are both scalable and performant. Additionally, I gained practical knowledge of Android architecture by implementing the Model-View-ViewModel (MVVM) pattern along with LiveData. This approach helped create reactive and maintainable user interfaces, improving the overall quality and responsiveness of the app.
Working with distributed systems underscored the importance of conflict resolution through atomic transactions, ensuring data consistency across multiple users and devices. I also learned the value of a rigorous testing strategy; writing unit tests helped catch edge cases early in the development cycle, which greatly increased the reliability and robustness of the codebase. Breaking down complex features into smaller, incremental deliverables proved effective in managing project complexity and tracking progress efficiently.
Effective use of version control was crucial throughout the project. Regular commits and a clear branching strategy helped prevent merge conflicts and facilitated smooth collaboration among team members. Clear communication and well-defined roles within the team improved coordination and productivity. Finally, maintaining comprehensive and updated documentation streamlined integration efforts and onboarding processes, saving significant time and reducing potential confusion during collaboration.

Conclusion

OTAMS successfully delivers a comprehensive tutoring appointment management system that addresses the needs of all three user roles. Through four incremental deliverables, we have built a robust, user-friendly application that:

- Streamlines the tutoring process for Students seeking help
- Empowers Tutors to manage their availability efficiently
- Provides Administrators with control over system access
- Ensures data integrity through proper validation and conflict checking
- Delivers a user experience with intuitive interfaces

The project has provided valuable hands-on experience in Android development, Firebase integration, team collaboration, and software engineering principles. We have met most requirements outlined in the project description and delivered a fully functional application ready for deployment.