uOttawa

Faculté de génie
Faculty of Engineering

SEG 2105 – OTAMS Report

Dec 1, 2025

Authors:

Estifanos Wondem, 300403119

Fey Adediran, 300448160

Simon Bose, 300363330

Youssef Boussabah, 300445491

Table of Contents

Objective:

- Design and to construct a student tutor scheduling application
- Collaborate on GitHub to code as a group and submit four Deliverables
- Learn the design and coding processes behind a large project

Software:

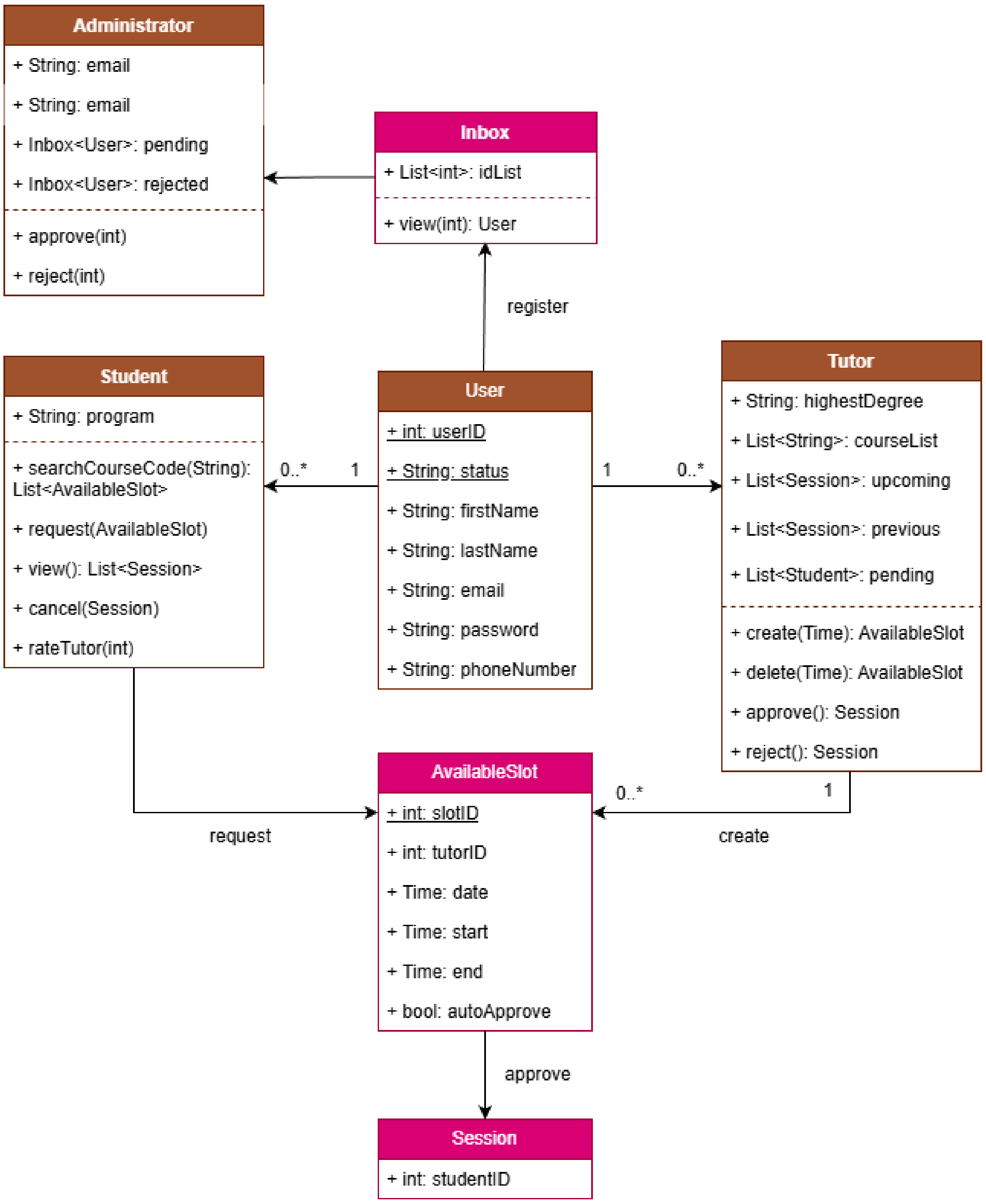
- Android Studio
- GitHub
- Diagrams.net

Introduction

The aim of this project was to develop the Online Tutoring Appointment Management System (OTAMS), a mobile application designed for the University of Ottawa Help Centre. For this design, OTAMS has three main users: Students, Tutors, and the Administrator. Students are able to see available time slots, request and cancel sessions, and rate Tutors after completing their session. Tutors can update and manage their schedule, accept or reject incoming session requests, and oversee their upcoming and past sessions. Finally, the Administrator oversees the registration process by approving or rejecting account requests from new Students and Tutors.

This report summarizes each members' contribution across the four deliverables, as well as the lessons learned from its development.

UML Class Diagram



Contribution Table

Deliverable 1: Getting Started

Task	Member(s)
Create repository on GitHub	Fey
Design UML Class Diagram	Simon
User can create Student and Tutor entities with required fields	Youssef
Admin can log in using credentials	Youssef, Simon
Login and Welcome screen	Fey, Estifanos
Demonstration video	Estifanos
Bonus: Implement DB	Fey, Youssef

Deliverable 2: Administrative Functions

Task	Member(s)
Registration request, along with user info, are sent to inbox	Simon, Youssef
Admin can interact with inbox for registration requests	Estifanos, Fey
Admin can interact with inbox for previously rejected	Estifanos, Fey
User can login once Admin permits	Youssef, Simon
User receives notification if request pending or rejected	Youssef, Fey
UML Class Diagram	Simon
Demonstration video	Estifanos

Deliverable 3: Tutor Functions

Task	Member(s)
Tutor can create an availability slot	Fey, Youssef
Tutor can delete availability slot	Fey, Youssef
Tutor can accept requests manually or automatically	Simon
Tutor can see requested, upcoming and past sessions	Estifanos, Youssef
Tutor can approve, reject, and cancel sessions	Estifanos
UML Class Diagram	Simon
Demonstration video	Estifanos

Deliverable 4: Student Functions and Final Report

Task	Member(s)
Student can view requested, scheduled and rejected sessions	Estifanos, Simon
Student can search for slot via course code	Youssef, Estifanos
Student can rate Tutors out of 5 stars	Youssef

Updated sessions so Students cannot delete within 24h	Simon, Fey
Updated slots so Tutors cannot delete once booked	Simon, Youssef
Test Cases	Youssef, Estifanos
UML Class Diagram	Simon
Final Report	Simon, Fey
Demonstration video	Estifanos

Conclusion (Lessons Learned)

Challenges

At the beginning, the idea of building a full mobile app from scratch honestly felt overwhelming. None of us had done anything like this before, so just figuring out where to start was intimidating. Learning Android Studio was a hurdle. We had to figure out the interface, understand Gradle builds, fix error after error, and get the app running on the emulator. Sometimes solving one problem would lead to three more, which was frustrating, but we learned a lot along the way.

The marking scheme made things trickier, too. Unlike typical assignments where you can write most of the code first and debug later, this project expected us to submit small, working sections throughout. It forced us to plan carefully, write organized code, and develop step by step rather than all at once.

Working as a team had its own challenges. Git was new to many of us, and at first, committing, pushing, pulling, and dealing with merge conflicts felt confusing. But as we got the hang of it, it became a useful way to stay on the same page and coordinate our work.

Takeaways

Even with all the challenges, the project taught us so much. Breaking the work into smaller chunks made the process feel manageable, and every little success gave us confidence. Learning to troubleshoot Android Studio and using Git improved not just our technical skills, but also our problem-solving and teamwork abilities.

Designing with UML diagrams and class diagrams was a big help, too. Being able to visualize the app's structure and see how everything connected made it easier to plan features and avoid confusion later. Patterns like the player-role pattern gave us a framework for thinking about how the app should work, which made coding smoother and more organized.

Final Reflection

Looking back, despite all the bumps along the way, the project was a big success. We built a working app, learned new skills, and gained confidence in our ability to handle a big, complicated project. Every challenge from technical errors to teamwork issues turned into a learning opportunity. By the end, we didn't just have a finished app; we also had a better understanding of coding, design, and collaboration. The project showed us that even when something feels impossible at first, breaking it down and working together can make it achievable.