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:

- To 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:

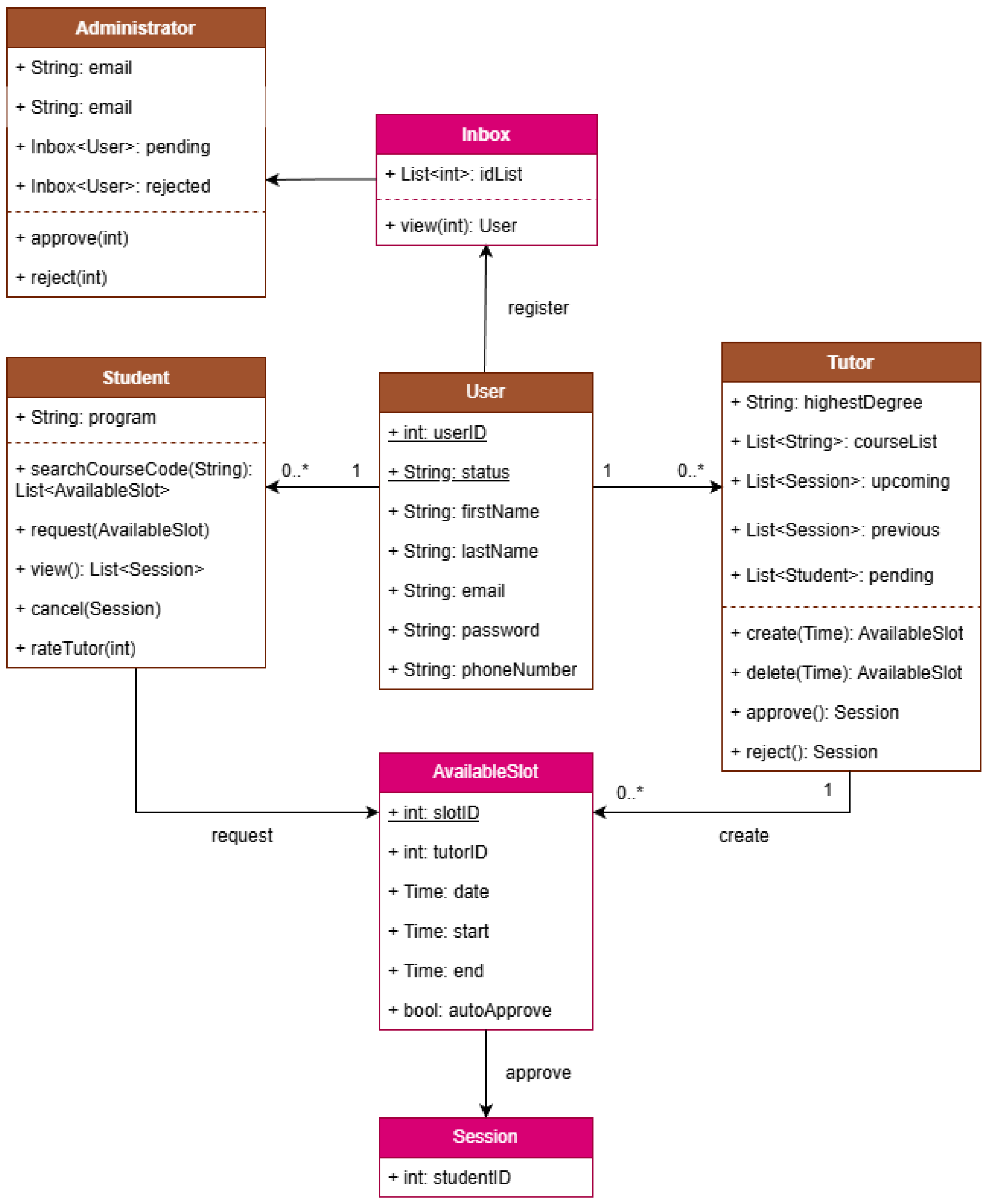
- Android Studio
- GitHub
- Diagrams.net

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 be 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)

In this project, the goal was to design and create a working mobile app from scratch. When starting such a task, it is easy to get overwhelmed and frustrated. However, by dividing the workload into smaller parts, it made the design process simpler to understand. Nonetheless, the project was still a challenge. Our group had not worked with Android Studio before, so we had to familiarize ourselves with the software; understanding its Gradle build system, resolving build errors and running the app on an emulated phone. Moreover, the deliverable marking scheme forced us to write partial bits of code that compile, despite only being a fraction of the final design. We could not simply write the code at once and work to debug at the end, which was different than assignments and labs. Furthermore, by collaborating on Git and using the push pull commit functions, we learned how to split the workload and stay up to date with the other parts of the project the group members were working on. Finally, through the UML Class Diagrams and the design pattern showcased in class (notably the player-role pattern), it was easier to come up with ideas for functionality. Overall, the project was successful, and although problems were encountered during its course, it served as a good learning experience.