GETCLASSES: AN OBJECT-ORIENTED TRANSACTIONAL PLATFORM FOR ONLINE TUTORING

ALEJANDRO ESCOBAR BARRIOS, JHON JAIRO GONZALEZ, SEBASTIAN ZAMBRANO

INTRODUCTION

Online education requires platforms that connect students and tutors efficiently and securely. Existing systems such as Coursera or Udemy offer broad content but lack personalized interaction. GetClasses addresses this gap by integrating tutoring, payments, and feedback in one system, applying object-oriented programming (OOP) principles and a layered architecture.

OBJECTIVES

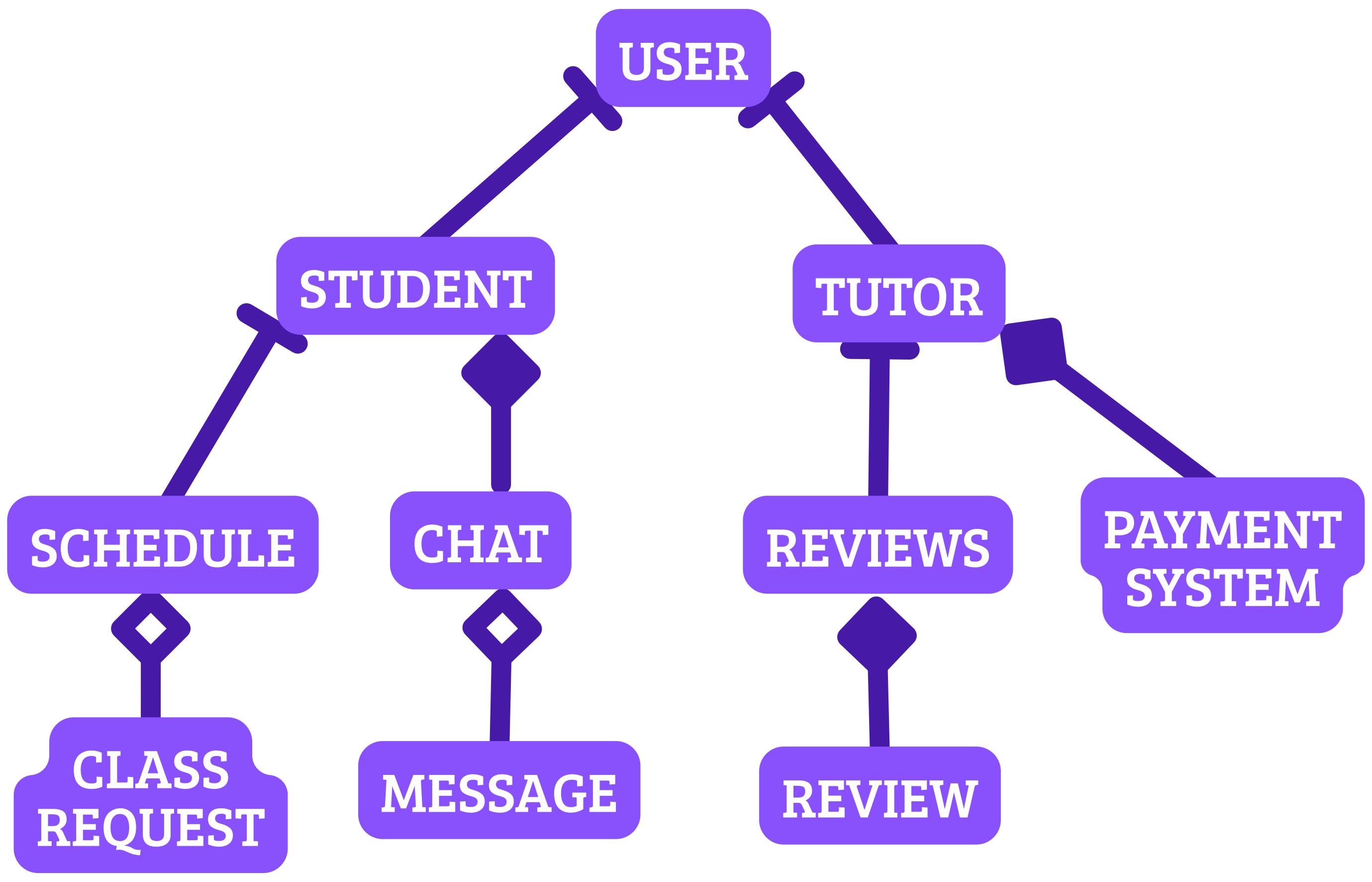
Design and implement an object-oriented tutoring platform that efficiently connects students with qualified tutors, using a modular and maintainable JavaFX architecture.

METHODS AND MATERIALS

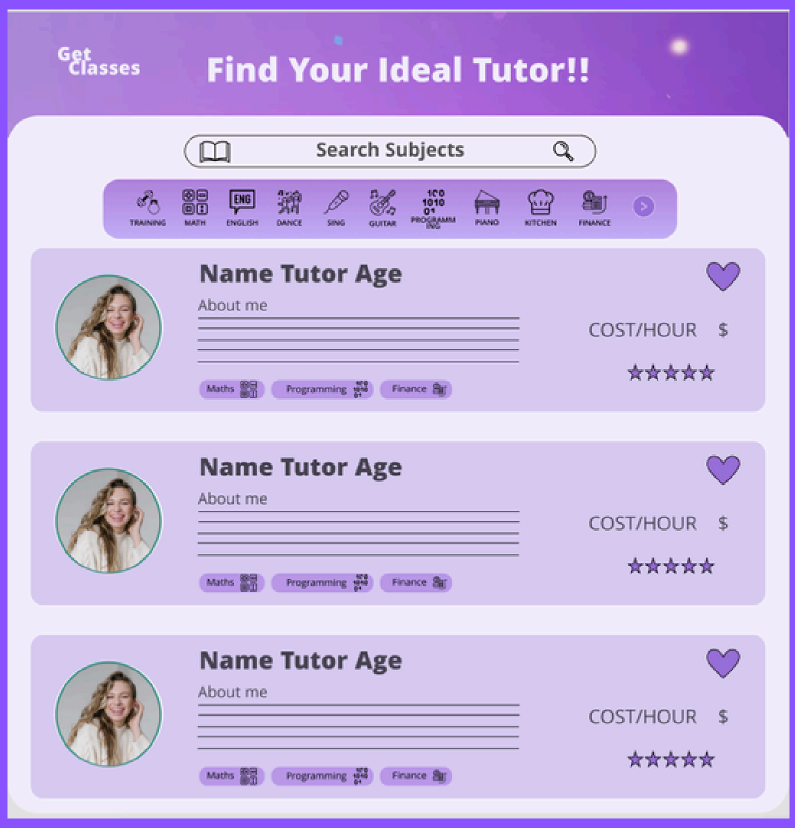
The system follows a layered architecture:

- Presentation: JavaFX interface for registration, search, and session management.
- Business Logic: handles transaction and scheduling rules.
- Persistence: manages data storage, initially file-based.

The design applies OOP principles—inheritance, polymorphism, and encapsulation—using a base User class and subclasses StudentUser, TutorUser, and AdminUser to ensure modularity and code reuse.



MOCKUPS



EXPERIMENTS

The prototype was tested through unit and integration tests on modules such as registration, scheduling, and review.
Each test verified functionality, data consistency, and response time.

TEST MODULE	STATUS	ESTIMATE EXECUTION TIME
REGISTRATION	IN PROGRESS	1.8 S
REVIEW SYSTEM	IN PROGRESS	1.9 S
SCHEDULING	IN PROGRESS	2.5 S
CHAT MODULE	PENDING	2.0 S
PAYMENT SIMUL.	PENDING	2.7 S

RESULTS

The prototype achieved an intuitive and smooth interface. Tests confirmed stable communication between modules. Compared with traditional platforms, GetClasses offers a more cohesive user flow and modular structure for future enhancements.

CONCLUSIONS

GetClasses demonstrates how OOP and SOLID principles enable the creation of modular, scalable educational software. Future work includes database integration, AI-based tutor matching, and web deployment.