Object-Oriented Programming
Semester 2025-III

Workshop No. 1 — Object-Oriented Design

Alejandro Escobar
Jhon Gonzales
Sebastián Zambrano

Computer Engineering Program
Universidad Distrital Francisco José de Caldas

1   Object-Oriented Concepts

In the design of the application **GetClasses**, object-oriented concepts are used to structure the system into classes/objects that represent roles, operations, and data.

Main Entities (Classes)

- Student

- Tutor

- Administrator

- Academic Profile

- Scheduling

- Virtual Class

- Payment System

- Chat/Message

- Review

Main Operations

- Student: register, log in, search tutors, book class, pay, leave review.

- Tutor: register, manage profile, accept/reject requests, teach classes, receive payments.

- Administrator: validate identities, solve disputes, supervise reviews and transactions.

- Payment System: process payments, transfers, refunds.

- Scheduling: show availability, reserve slots, notify users.

- Chat/Message: send, receive, and store messages.

Data Flows (Simplified)

1. Registration/Login: Student/Tutor enters data → System validates → Access granted.

2. Search and Booking: Student searches → selects slot → Scheduling confirms.

3. Payment: Student pays → Payment System processes → Tutor receives balance.

4. Virtual Class: Scheduling starts session → Virtual Class connects both.

5. Review: Student leaves feedback → Stored in Tutor Profile.

6. Support: Administrator intervenes in case of disputes.

2 Requirements Documentation

Functional Requirements

1. Registration and login for students and tutors.

2. Profile management including experience, degrees, rate, and availability.

3. Tutor search with filters by subject, rate, language, and country.

4. Scheduling with notifications.

5. Secure payments (credit card, e-wallet).

6. Virtual classes with video call, screen sharing, and whiteboard.

7. Chat and messaging between users.

8. Reviews and ratings after sessions.

9. Dispute management by the administrator.

Non-Functional Requirements

- **Performance:** Search responses in less than 3s, support 1000 concurrent users.

- **Usability:** Intuitive interface, available on web and mobile, multilingual (EN/ES).

- **Security:** Encrypted data and credentials, PCI DSS compliance, tutor identity validation.

- **Availability:** 99% uptime, daily backups.

- **Scalability:** Architecture prepared for new services (e.g., group classes).

3   User Stories

Here are the main user stories of the platform.

Tutor

- As a tutor, I want to register and create a profile with my expertise areas, academic level, hourly rate, and personal introduction, so that students can know my skills and contact me.

- As a tutor, I want to upload a profile picture so that students can recognize me easily.

- As a tutor, I want to rate students after a class, so that I can provide feedback on their participation and punctuality.

- As a tutor, I want to set an auto-responder when unavailable, so students get notified automatically.

Student

- As a student, I want to search tutors by subject, level and price, so I can find the best fit.

- As a student, I want to view tutor profiles in a list with photo, name, subjects and hourly rate, so I can compare them.

- As a student, I want to chat with tutors to coordinate lessons.

- As a student, I want to rate tutors after a class, so I can evaluate their clarity and performance.

General Users

- As a user, I want to view my ratings and comments in my profile, so I can know my reputation.

- As a user, I want to contact support/admin, so I can solve technical or account issues.

4 Mockups

Below are some initial sketches of the application screens.
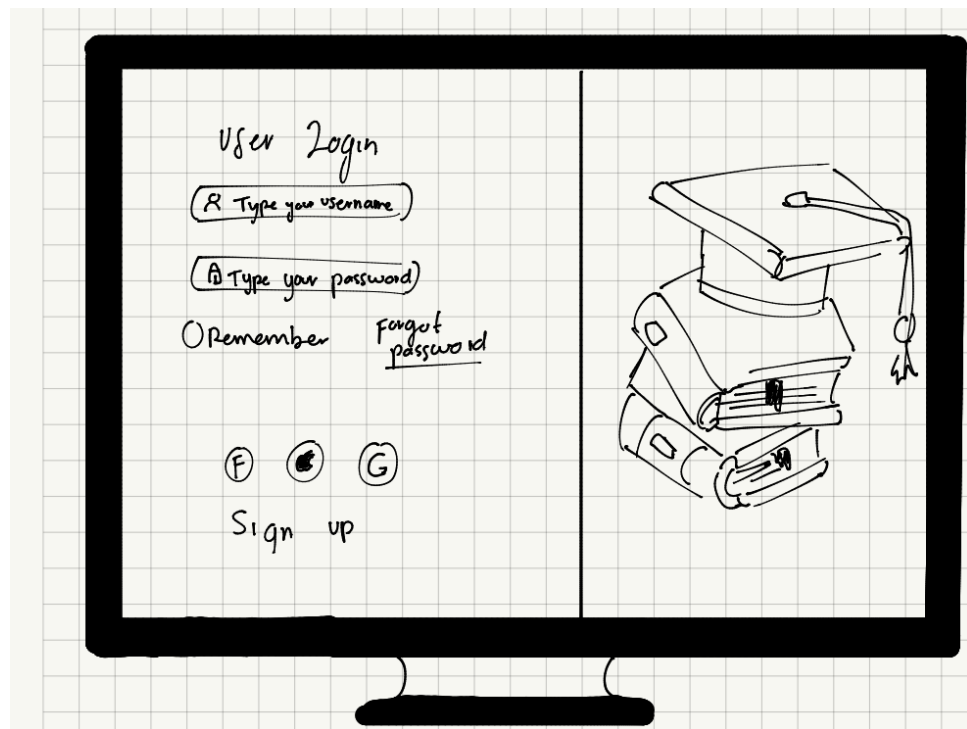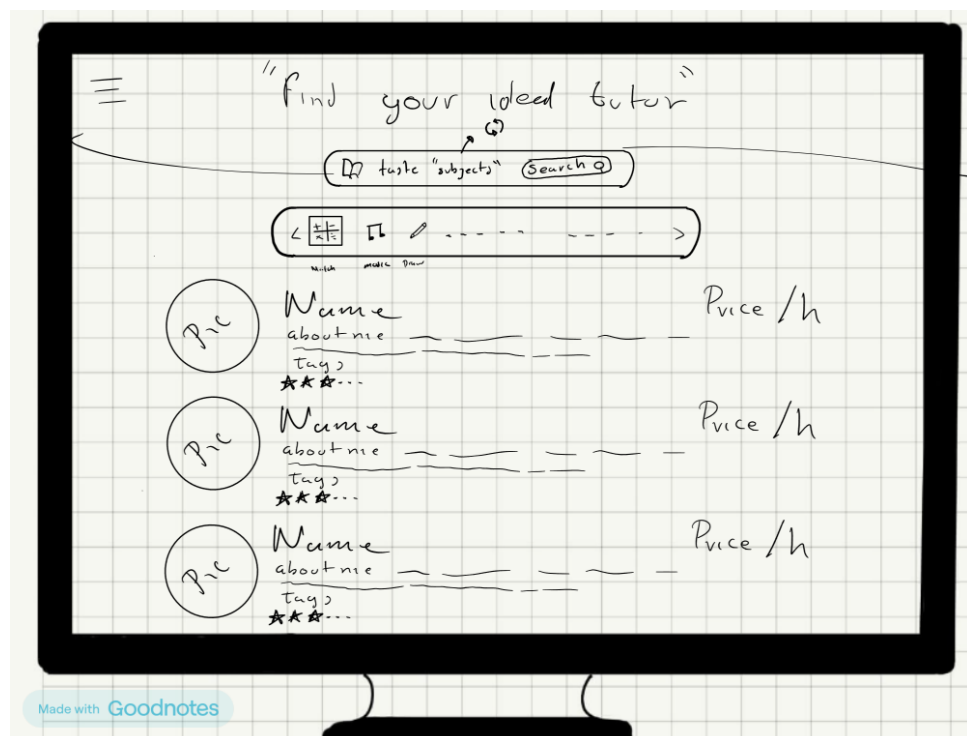

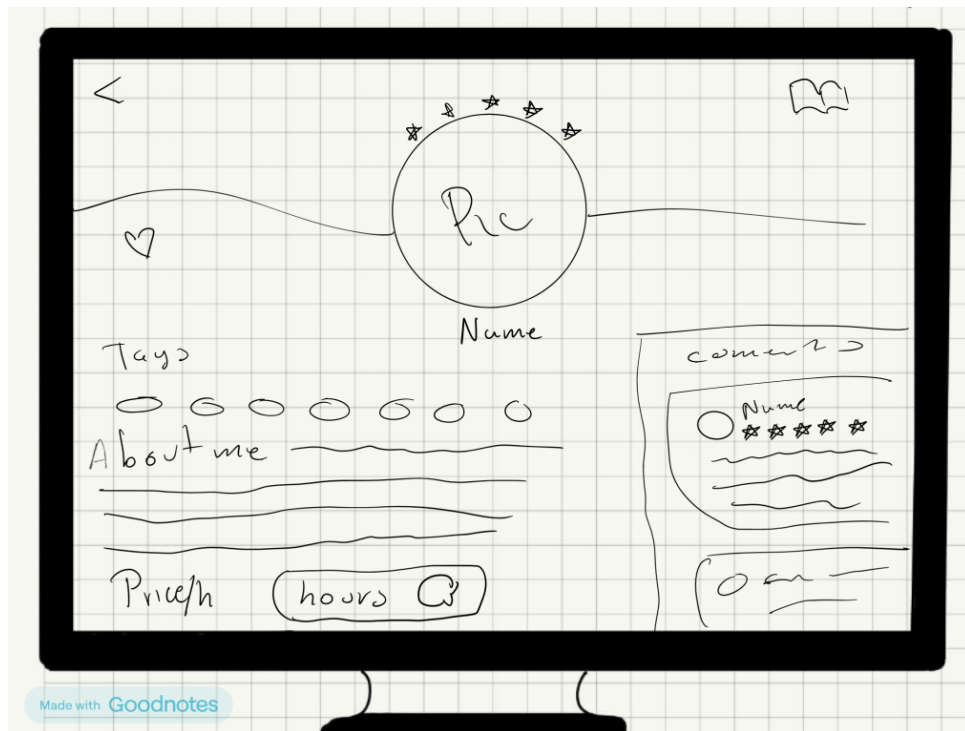Figure 1: Login Page

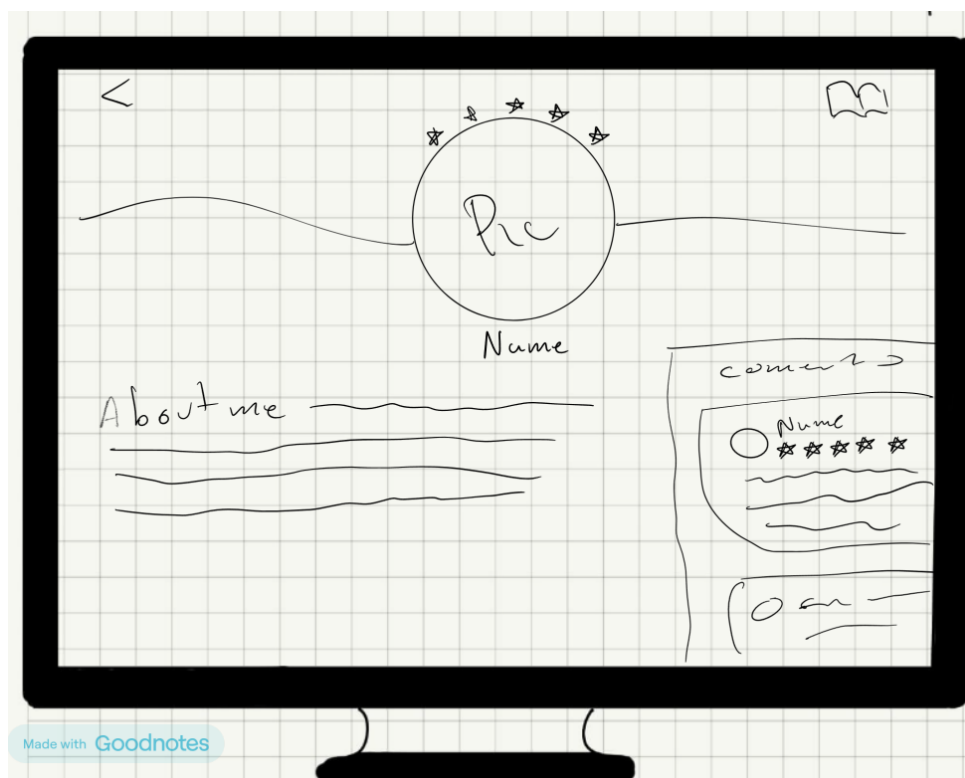

Figure 2: Main page

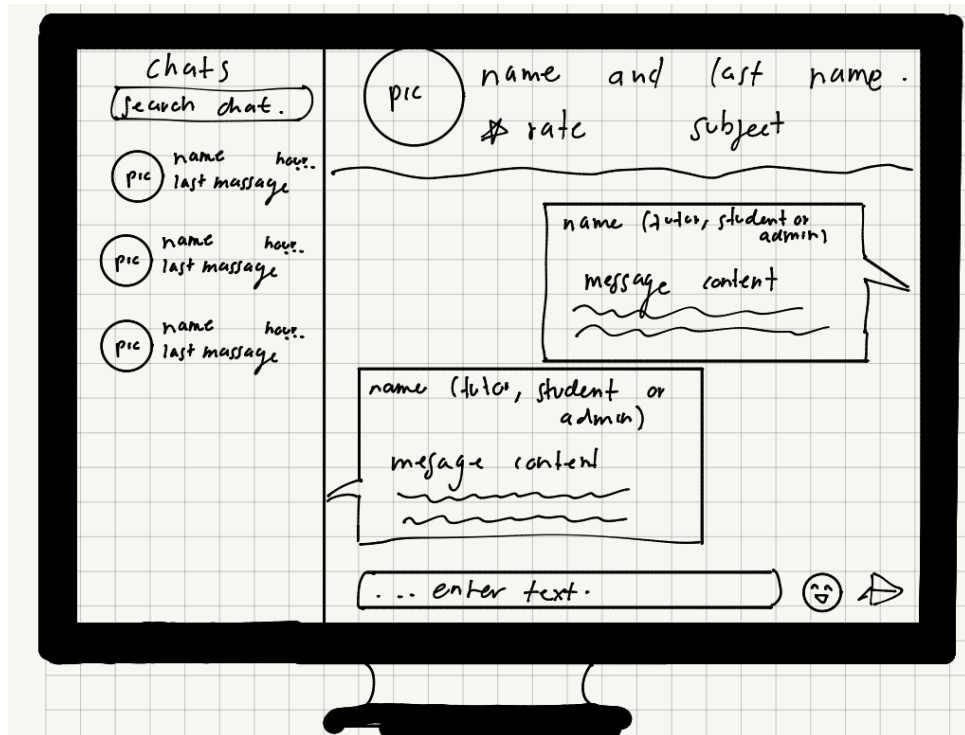Figure 3: Tutor Perfil

Figure 4: User Perfil

Figure 5: char Page

5   CRC Cards (Textual)

Tutor

Responsibilities:

- Register and manage profile.

- Manage availability and pricing.

- Teach classes, rate students.

- Enable auto-responder when unavailable.

Collaborators: Profile, Student, Chat, Rating System.

Student

Responsibilities:

- Search and filter tutors.

- View tutor profiles.

- Schedule and attend classes.

- Rate tutors.

Collaborators: Tutor, Chat, Rating System, Support.

Profile

Responsibilities:

- Show tutor and student information.

- Display ratings and reviews.

Collaborators: Tutor, Student, Rating System.

Chat

Responsibilities:

- Enable real-time communication.

- Provide auto-responses if tutor unavailable.

Collaborators: Tutor, Student.

Rating System

Responsibilities:

- Store and manage ratings for both tutors and students.

- Display reviews on profiles.

Collaborators: Tutor, Student, Profile.

Admin/Support

Responsibilities:

- Receive and resolve support requests.

- Supervise disputes and transactions.

Collaborators: Tutor, Student.

6 CRC Cards (Visual)

Below are CRC cards represented as images.
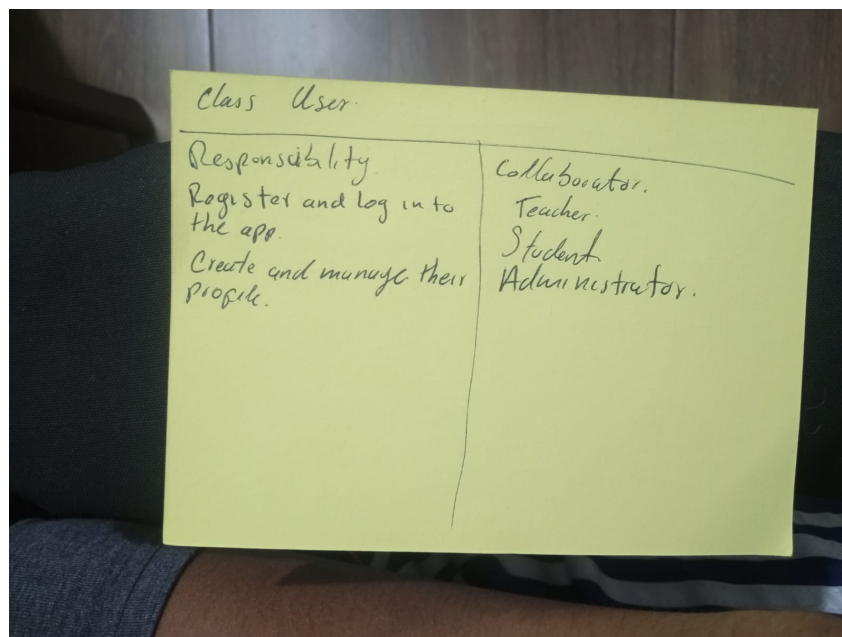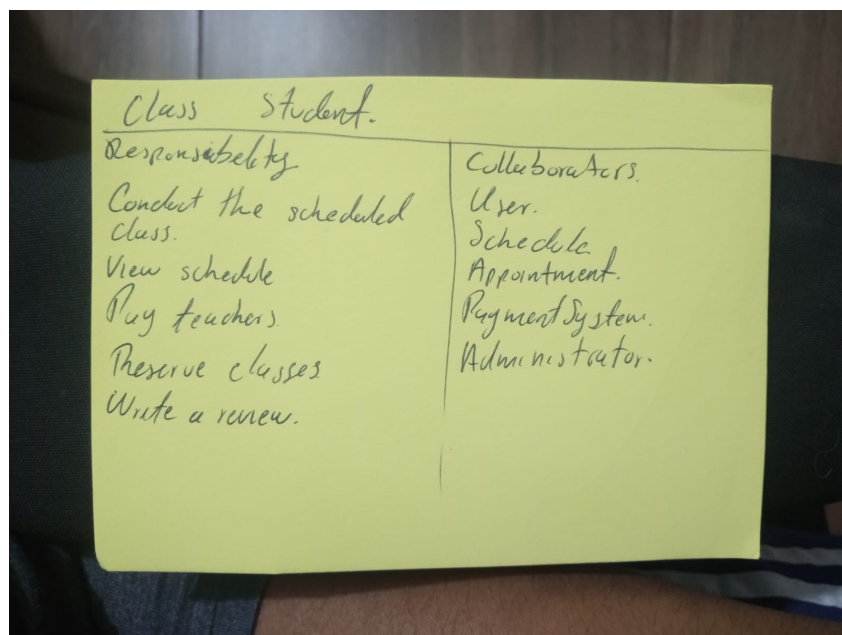

Figure 6: CRC Card - User

Figure 7: CRC Card - Student

Figure 8: CRC Card - Tutor

Figure 9: CRC Card - Administrator

Figure 10: CRC Card - Schedule

Figure 11: CRC Card - PaymentSystem

Figure 12: CRC Card - Appointment

Figure 13: CRC Card - Message

Figure 14: CRC Card - Chat

Figure 15: CRC Card - Reviews

Notes and Reflection

- All documentation is in English.

- References should be cited if external resources were used.

- This is a draft, subject to improvement in future workshops.

Reflection: During this first stage, the main challenge was defining clear user interactions and ensuring the platform covered both tutor and student needs. Decisions were made to prioritize usability, security, and scalability for future growth.