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

User Story 1 – Tutor Registration

User Story: As a tutor (professional or advanced student), 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.

Acceptance Criteria:

- Given that I am on the tutor registration page,
- When I fill in my details and click “Create Account”,
- Then my profile is created and I can access the platform as a tutor.

—

User Story 2 – Tutor Profile Picture

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

Acceptance Criteria:

- Given that I am on my profile page,
- When I upload a valid image,
- Then the picture appears on my profile and is visible to students.

—

User Story 3 – Tutor Search

User Story: As a student, I want to filter tutors by subject, academic level, and price range, so that I can quickly find the most suitable tutor for me.

Acceptance Criteria:

- Given that I am on the tutor search page,
- When I select filters and click “Search”,
- Then the system shows tutors that match those criteria.

—

User Story 4 – Tutor Listing

User Story: As a student, I want to see tutor profiles in a list with photo, name, title, subjects, and hourly rate, so that I can compare them easily before selecting one.

Acceptance Criteria:

- Given that I have applied filters,
- When the results are displayed,
- Then the system shows a list of tutors with their basic information.

—

User Story 5 – Chat with Tutor

User Story: As a student, I want to chat with the tutor through the platform, so that I can coordinate lessons, ask questions, and stay in direct contact.

Acceptance Criteria:

- Given that I have selected a tutor,
- When I access the "Chat" option,
- Then I can send and receive real-time messages with that tutor.

—

User Story 6 – Tutor Rates Student

User Story: As a tutor, I want to rate the student after a tutoring session, so that I can leave feedback about their participation, punctuality, and commitment.

Acceptance Criteria:

- Given that a tutoring session has ended,
- When the tutor selects "Rate Student",
- Then they can assign a score and a comment that will appear on the student's profile.

—

User Story 7 – Student Rates Tutor

User Story: As a student, I want to rate the tutor after a tutoring session, so that I can evaluate their performance, clarity, and punctuality.

Acceptance Criteria:

- Given that I have completed a tutoring session,
- When I select "Rate Tutor",
- Then I can assign a score and a comment that will appear on the tutor's profile.

—

User Story 8 – View Ratings and Reviews

User Story: As a user (student or tutor), I want to view ratings and comments on my personal profile, so that I can know my reputation and receive feedback.

Acceptance Criteria:

- Given that I am on my profile page,
- When I access the "Ratings" section,
- Then I can see my average score and all comments left by other users.

—

User Story 9 – Tutor Auto-Responder

User Story: As a tutor, I want to have an auto-responder when I am unavailable, so that students receive an automatic reply informing them of my status and don't feel ignored.

Acceptance Criteria:

- Given that a student sends me a message while I am offline or marked as "unavailable",
- When the system triggers the auto-response,
- Then the student sees a predefined message indicating I will reply later.

—

User Story 10 – Contact Support/Admin

User Story: As a user (student or tutor), I want to have a section to contact an administrator or support, so that I can solve technical issues, ask questions, or report problems on the platform.

Acceptance Criteria:

- Given that I am logged into my account,
- When I go to the "Support" section and send a message,
- Then the system forwards my request to the administrator and I receive confirmation that it was registered.

4 Mockups

Below are some initial sketches of the application screens.

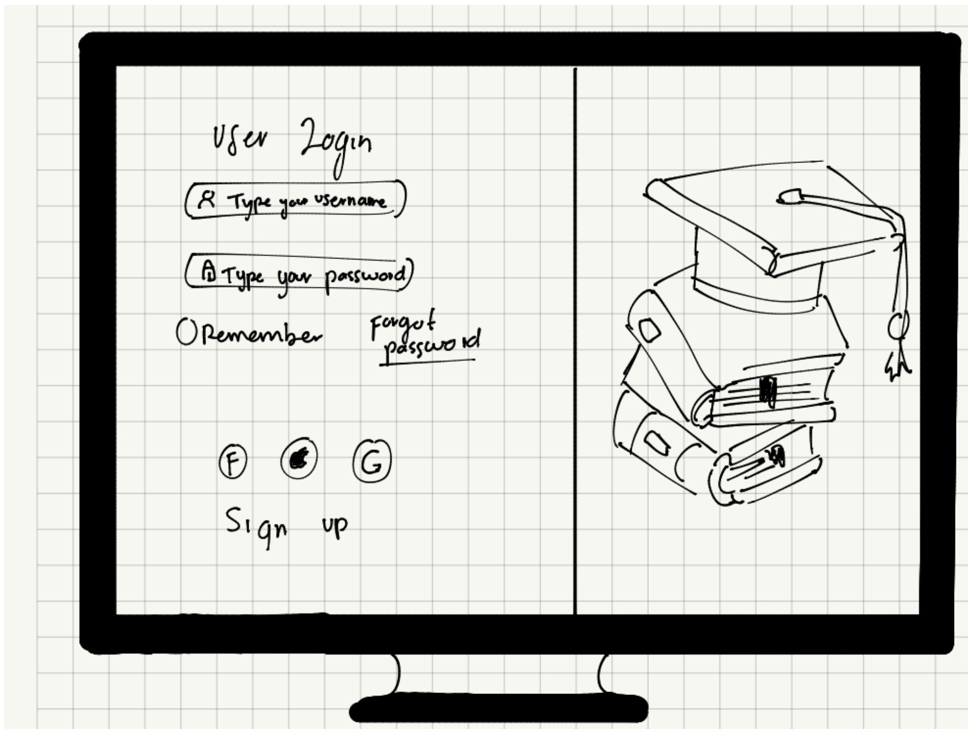


Figure 1: Login Page

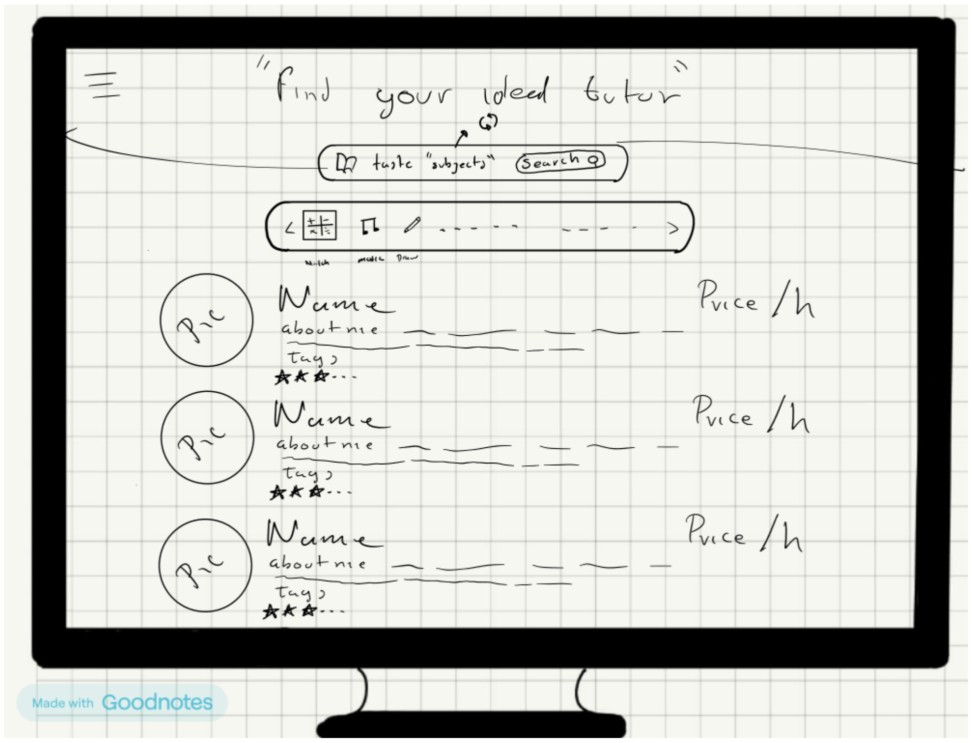


Figure 2: Main page

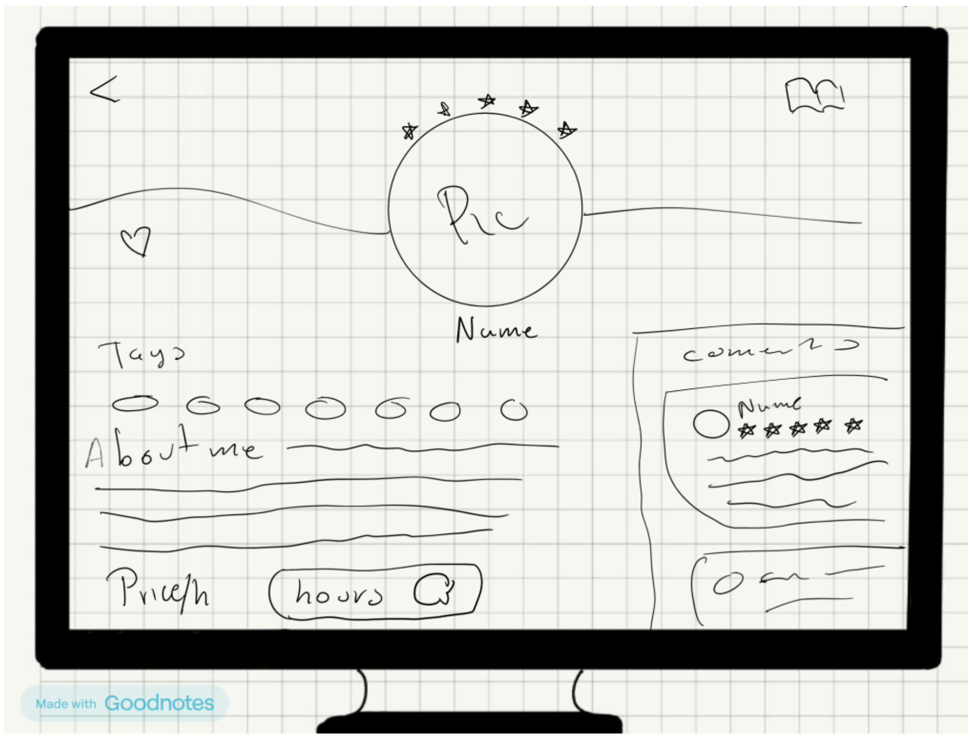


Figure 3: Tutor Perfil

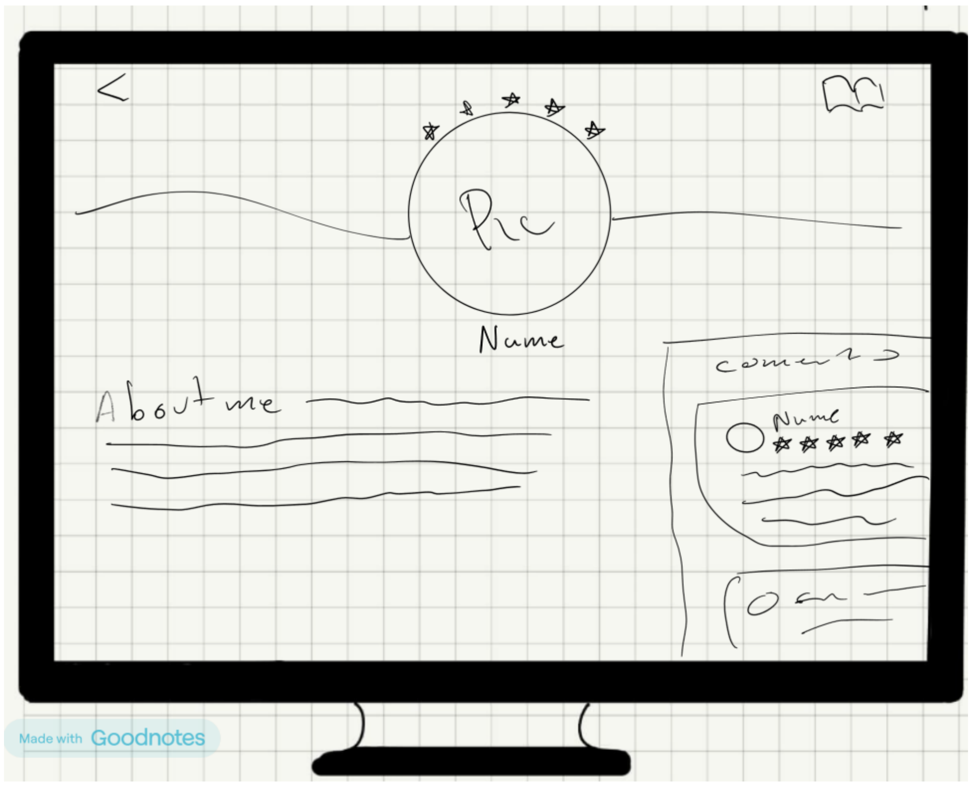


Figure 4: User Perfil

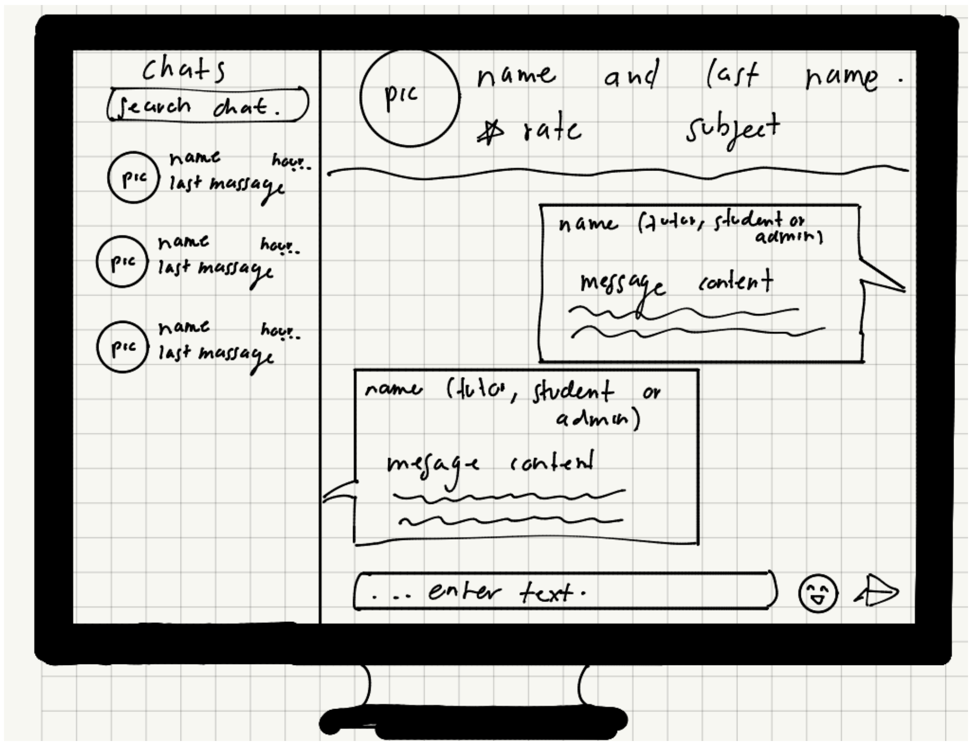


Figure 5: char Page

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.

Reviews

Responsibilities:

- Store and manage ratings for both tutors and students.
- Display reviews on profiles.

Collaborators: Tutor, Student, Profile.

Administrator

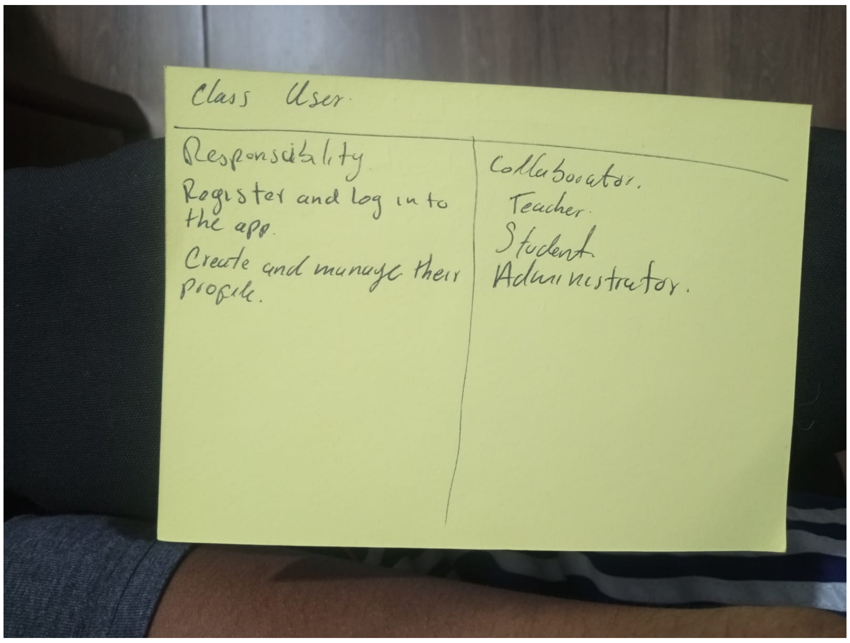
Responsibilities:

- Receive and resolve support requests.
- Supervise disputes and transactions.

Collaborators: Tutor, Student.

6 CRC Cards (Visual)

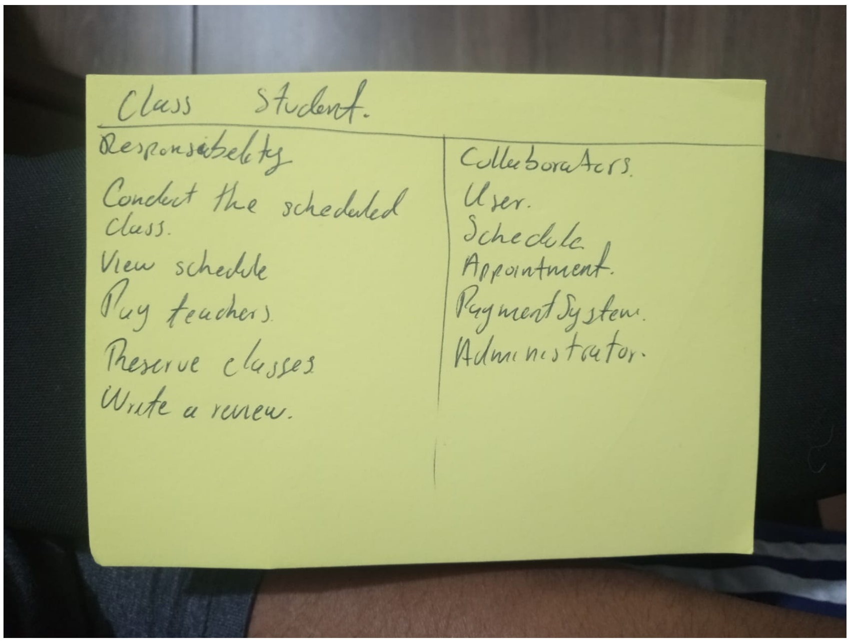
Below are CRC cards represented as images.



Class User.

Responsibility	Collaborator.
Register and log in to the app.	Teacher.
Create and manage their profile.	Student
	Administrator.

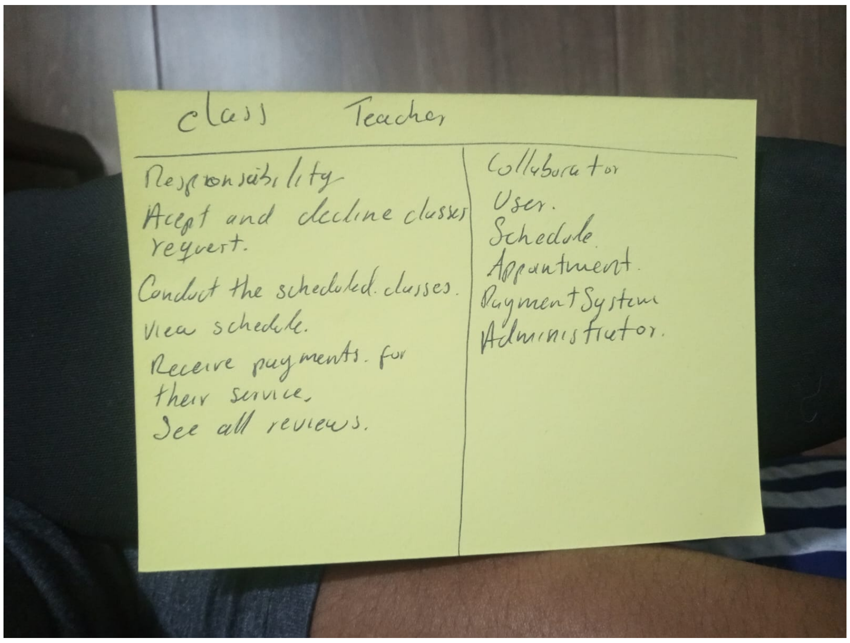
Figure 6: CRC Card - User



Class Student.

Responsibility	Collaborators.
Conduct the scheduled class.	User.
View schedule	Schedule
Pay teachers.	Appointment.
Reserve classes	Payment System.
Write a review.	Administrator.

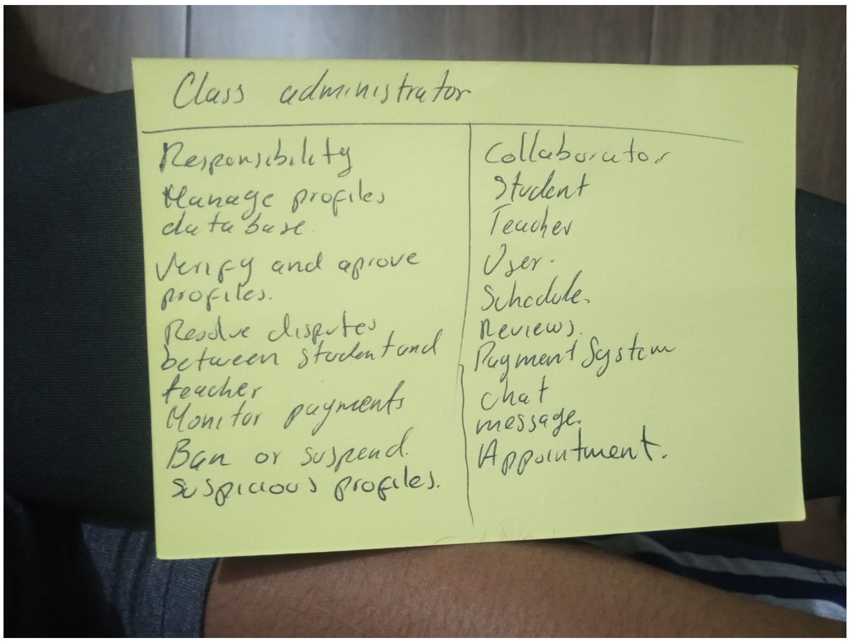
Figure 7: CRC Card - Student


class Teacher

Responsibility
Acept and decline classes request.
Conduct the scheduled classes.
View schedule.
Receive payments. for their service.
See all reviews.

Collaborator
User.
Schedule.
Appointment.
Payment System
Administrator.

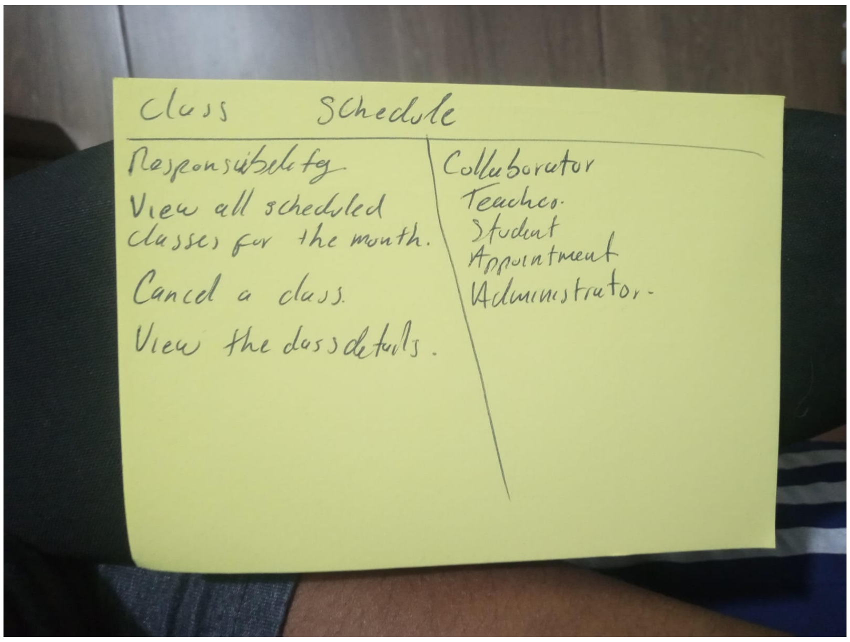
Figure 8: CRC Card - Tutor


Class administrator

Responsibility
Manage profiles database.
Verify and aprove profiles.
Resolve disputes between student and teacher
Monitor payments
Ban or suspend. suspicious profiles.

Collaborator
Student
Teacher
User.
Schedule.
Reviews.
Payment System
chat message.
Appointment.

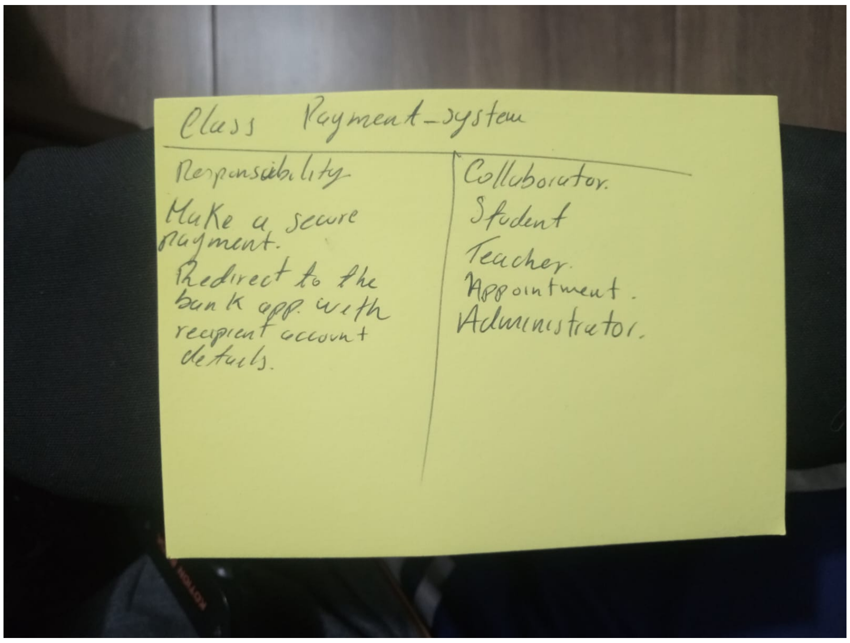
Figure 9: CRC Card - Administrator



Class Schedule

Responsibility	Collaborator
View all scheduled classes for the month.	Teacher.
Cancel a class.	Student
View the class details.	Appointment
	Administrator.

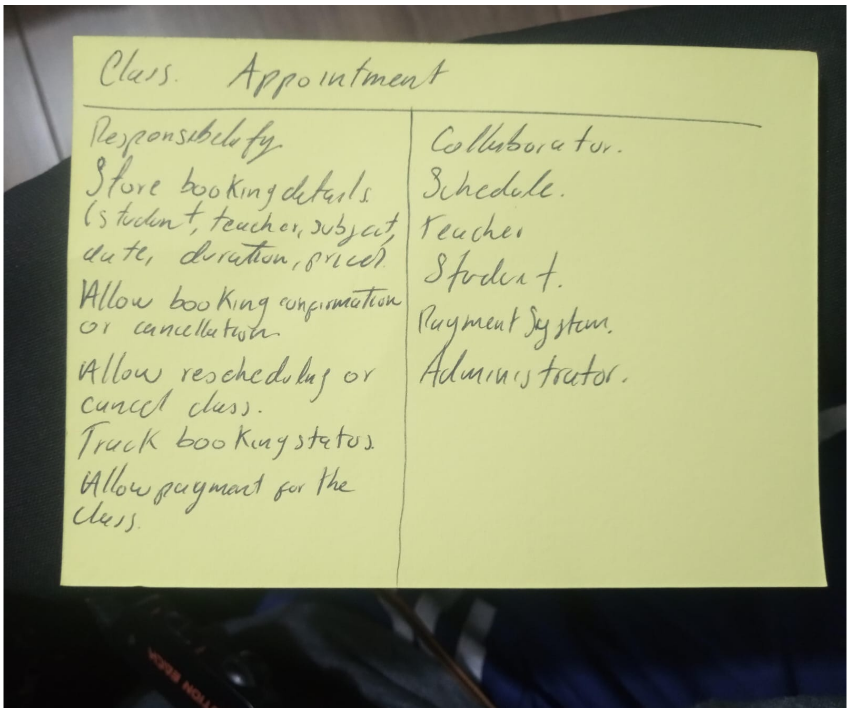
Figure 10: CRC Card - Schedule



Class Payment_system

Responsibility	Collaborator.
Make a secure payment.	Student
Redirect to the bank app with recipient account details.	Teacher.
	Appointment.
	Administrator.

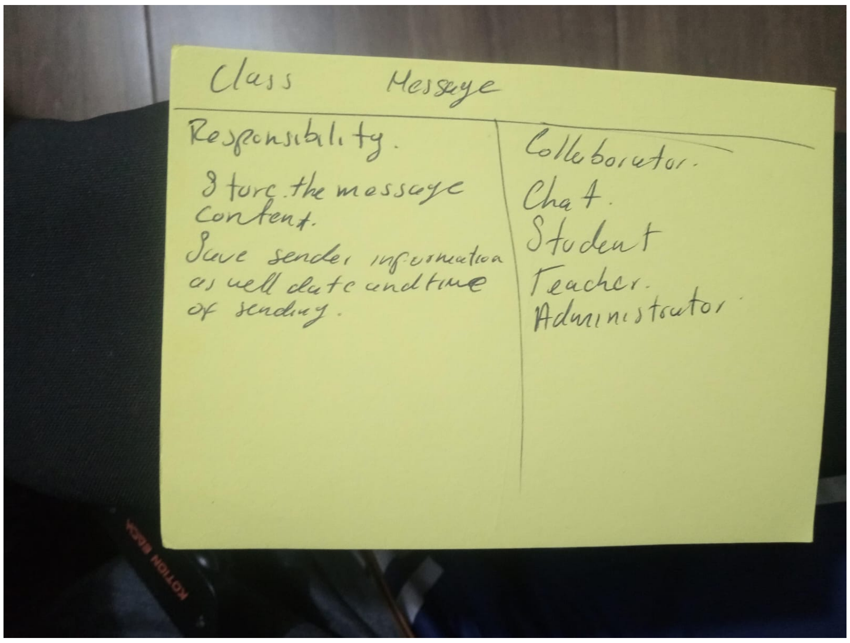
Figure 11: CRC Card - PaymentSystem



Class. Appointment

Responsibility	Collaborator.
Store booking details (student, teacher, subject, date, duration, price)	Schedule.
Allow booking confirmation or cancellation.	Teacher
Allow rescheduling or cancel class.	Student.
Track booking status.	Payment System.
Allow payment for the class.	Administrator.

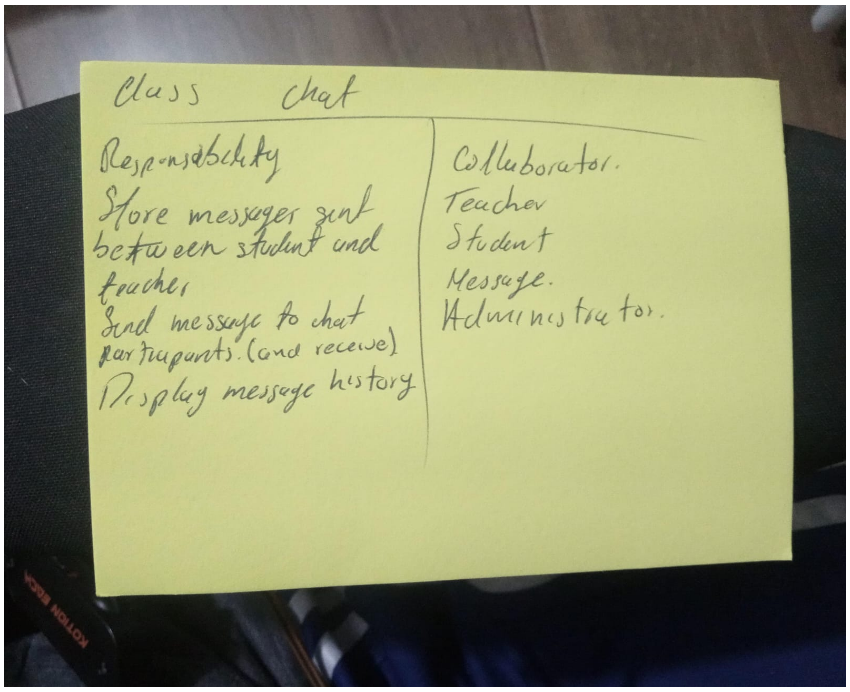
Figure 12: CRC Card - Appointment



Class Message

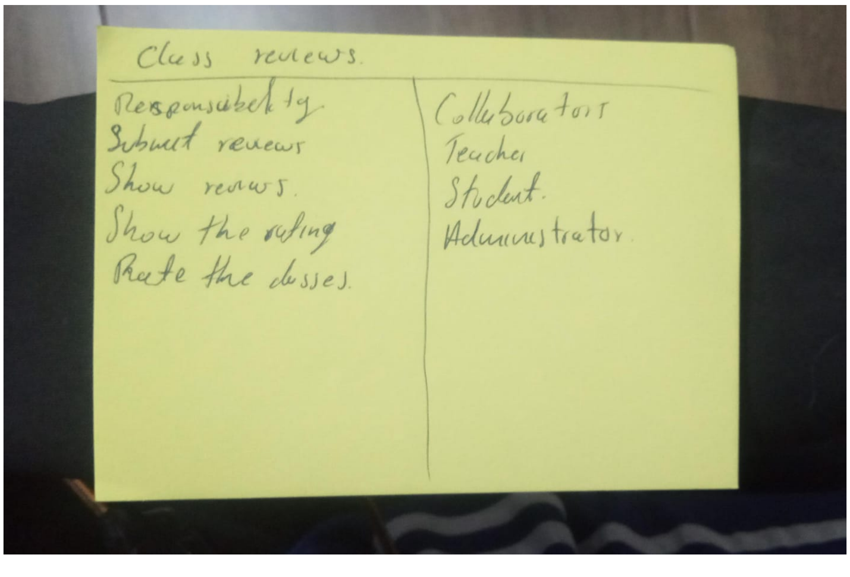
Responsibility.	Collaborator.
Store the message content.	Chat.
Save sender information as well date and time of sending.	Student
	Teacher.
	Administrator.

Figure 13: CRC Card - Message


Class Chat

Responsability	Collaborator.
Store messages sent between student and teacher	Teacher
Send message to chat participants. (and receive)	Student
Display message history	Message.
	Administrator.

Figure 14: CRC Card - Chat


Class reviews.

Responsability	Collaborators
Submit reviews	Teacher
Show reviews.	Student.
Show the rating	Administrator.
Rate the classes.	

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.