Intro to Software

Engineering

Test Cases Report

Team:

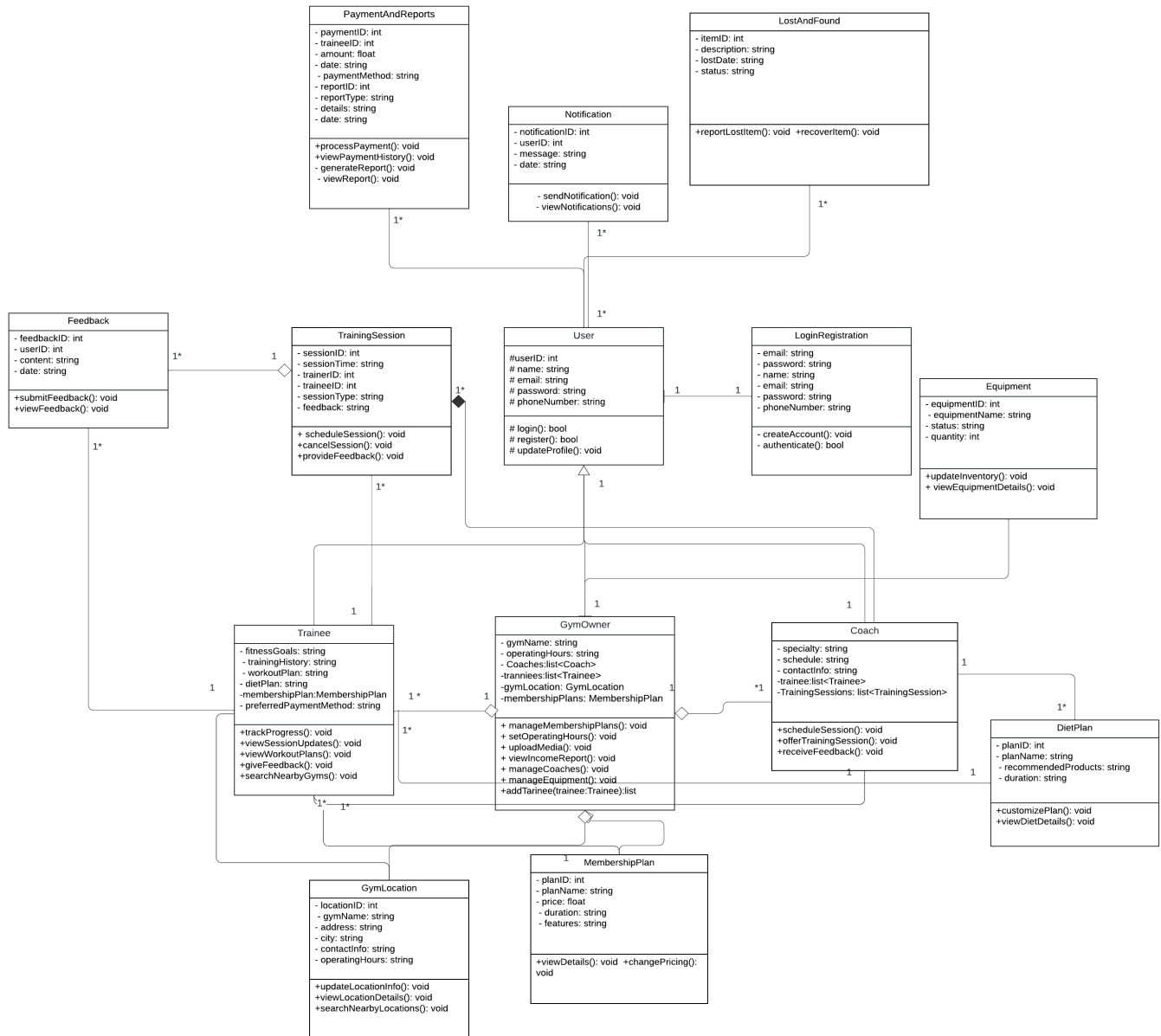| ID	Name	Email
20237004	Rabab Mohamed Abd El-Gafor Okasha	20237004@stud.fci-cu.edu.eg
20237003	Jumanah Muhammad Ali	20237003@stud.fci-cu.edu.eg
20237015	Malak Mostafa Abd El-maboud	20237015@stud.fci-cu.edu.eg
20237017	Nour Hassan Mahmoud	20237017@stud.fci-cu.edu.eg

Supervised by
Dr Manar El Kady

Sarah Alsisi

Content

- Phase 2 Update
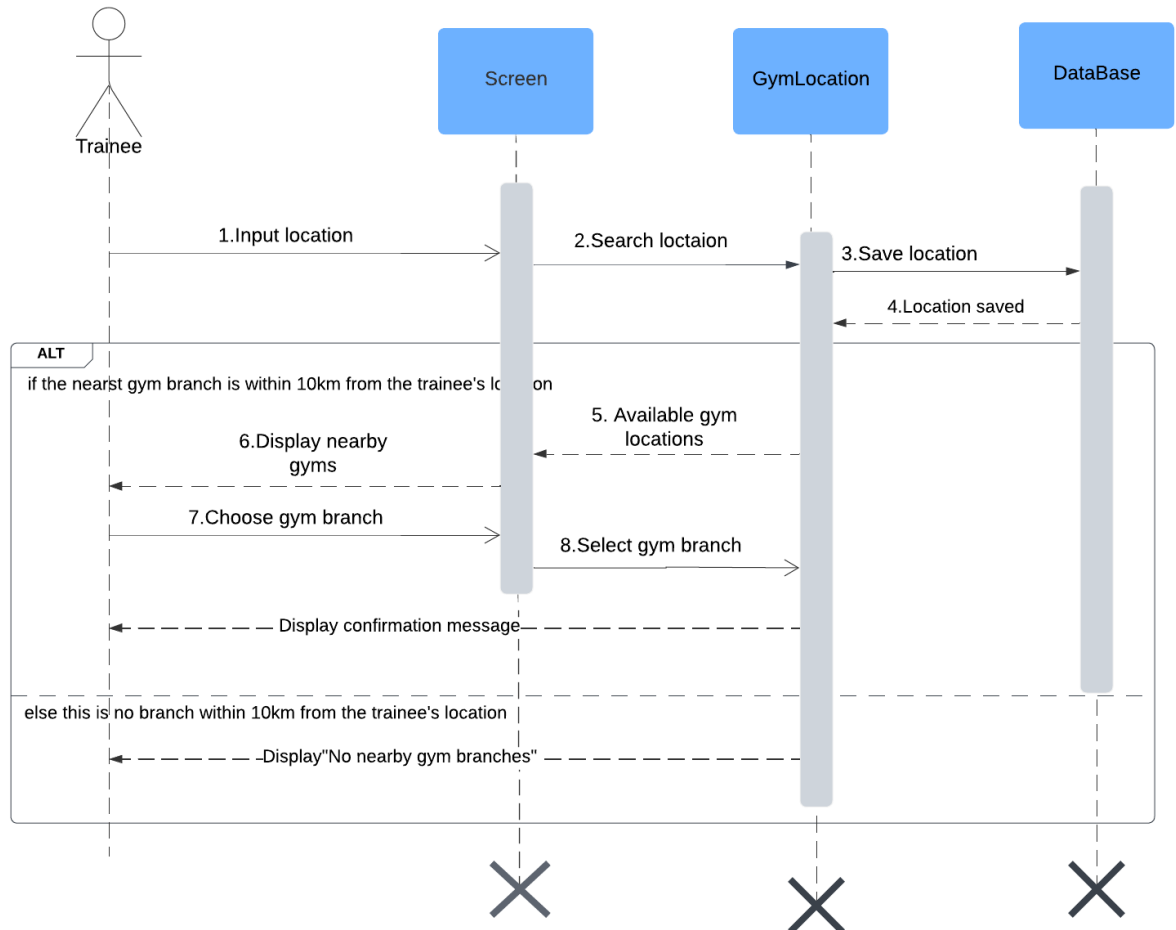- Scope
- Test Case Table template

Phase 2
class diagram:

PaymentAndReports

- paymentID: int
- traineeID: int
- amount: float
- date: string
- paymentMethod: string
- reportID: int
- reportType: string
- details: string
- date: string

+processPayment(): void
+viewPaymentHistory(): void
- generateReport(): void
- viewReport(): void

LostAndFound

- itemID: int
- description: string
- lostDate: string
- status: string

+reportLostItem(): void +recoverItem(): void

Notification

- notificationID: int
- userID: int
- message: string
- date: string

- sendNotification(): void
- viewNotifications(): void

Feedback

- feedbackID: int
- userID: int
- content: string
- date: string

+submitFeedback(): void
+viewFeedback(): void

TrainingSession

- sessionID: int
- sessionTime: string
- trainerID: int
- traineeID: int
- sessionType: string
- feedback: string

+ scheduleSession(): void
+cancelSession(): void
+provideFeedback(): void

User

#userID: int
name: string
email: string
password: string
phoneNumber: string

login(): bool
register(): bool
updateProfile(): void

LoginRegistration

- email: string
- password: string
- name: string
- email: string
- password: string
- phoneNumber: string

- createAccount(): void
- authenticate(): bool

Equipment

- equipmentID: int
- equipmentName: string
- status: string
- quantity: int

+updateInventory(): void
+ viewEquipmentDetails(): void

Trainee

- fitnessGoals: string
- trainingHistory: string
- workoutPlan: string
- dietPlan: string
-membershipPlan:MembershipPlan
- preferredPaymentMethod: string

+trackProgress(): void
+viewSessionUpdates(): void
+viewWorkoutPlans(): void
+giveFeedback(): void
+searchNearbyGyms(): void

GymOwner

- gymName: string
- operatingHours: string
- Coaches:list<Coach>
-tranniees:list<Trainee>
-gymLocation: GymLocation
-membershipPlans: MembershipPlan

+ manageMembershipPlans(): void
+ setOperatingHours(): void
+ uploadMedia(): void
+ viewIncomeReport(): void
+ manageCoaches(): void
+ manageEquipment(): void
+addTarinee(trainee:Trainee):list

Coach

- specialty: string
- schedule: string
- contactInfo: string
-trainee:list<Trainee>
-TrainingSessions: list<TrainingSession>

+scheduleSession(): void
+offerTrainingSession(): void
+receiveFeedback(): void

DietPlan

- planID: int
- planName: string
- recommendedProducts: string
- duration: string

+customizePlan(): void
+viewDietDetails(): void

GymLocation

- locationID: int
- gymName: string
- address: string
- city: string
- contactInfo: string
- operatingHours: string

+updateLocationInfo(): void
+viewLocationDetails(): void
+searchNearbyLocations(): void

MembershipPlan

- planID: int
- planName: string
- price: float
- duration: string
- features: string

+viewDetails(): void +changePricing():
void
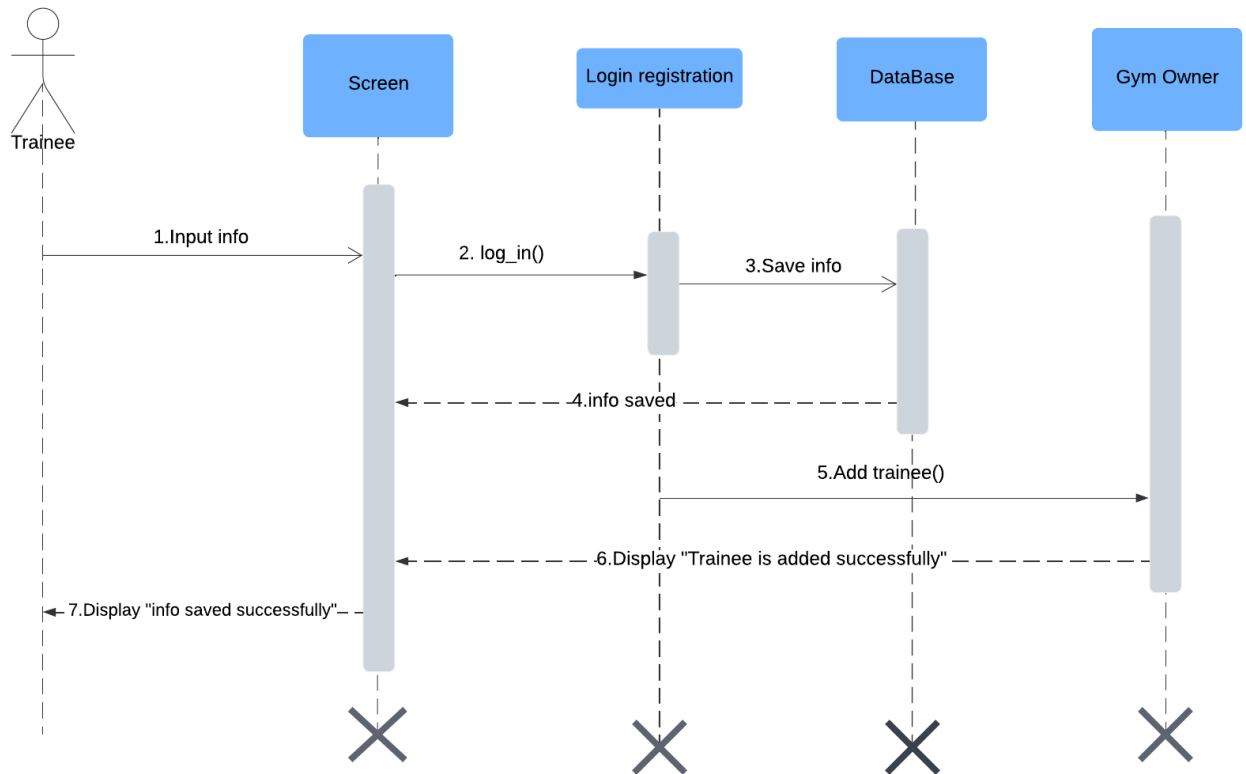
Sequence diagrams:

1) Requirement 18: The Trainee will be able to search for nearby gym branches depending on his location. (From User Story 19)
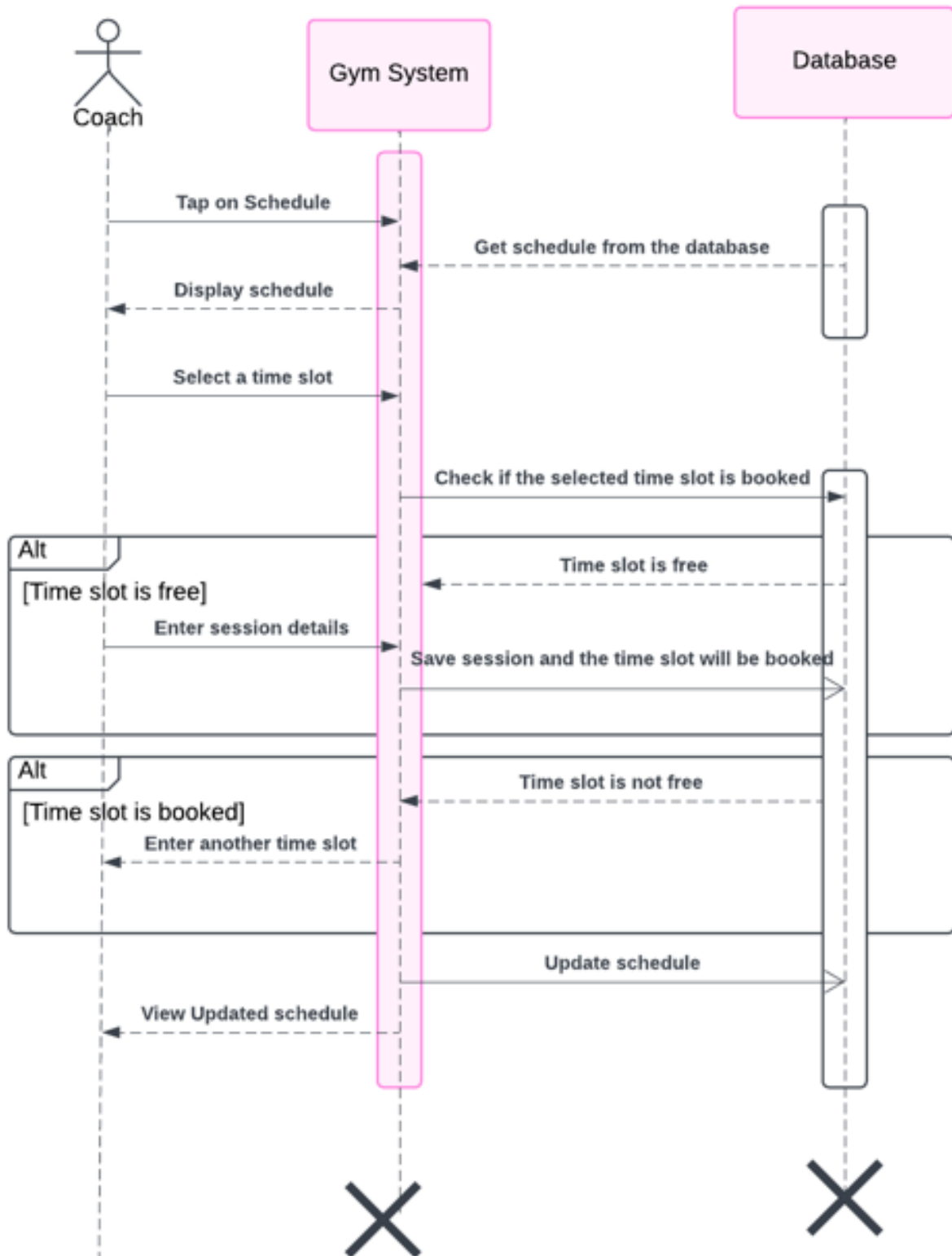
2) The system should save the user information to facilitate future actions and sign in process (From user story 23)

3) Requirement 21: The User should be able to choose the payment method whether using credit card or cash. (From User Story 21)
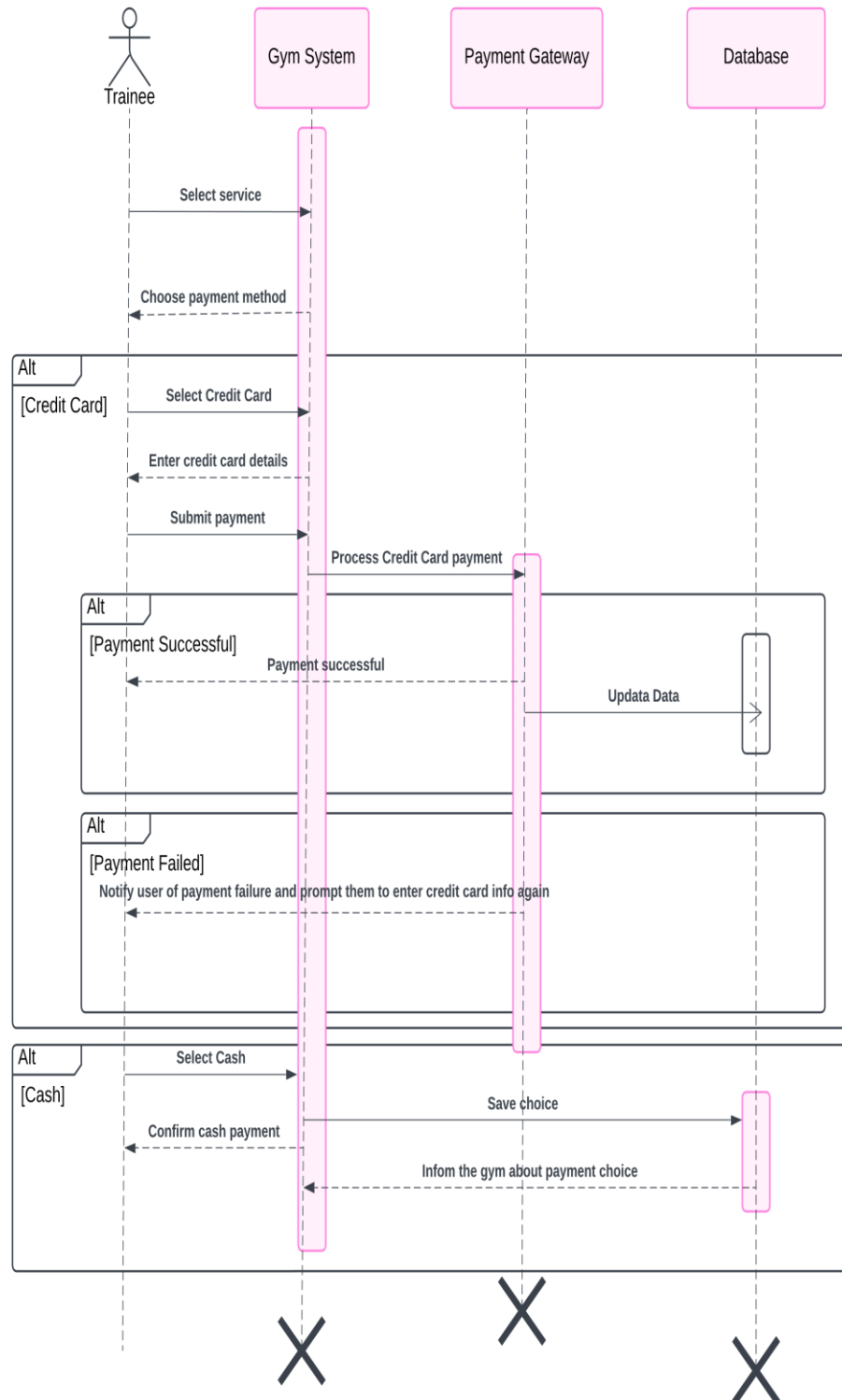
4) Requirement 6: The System should allow the coach to set and schedule their sessions with the trainees. (From User Story 6)

5) User Story 5: As a gym owner, I want to set my gym's operating hours, so that clients know when the gym is open.
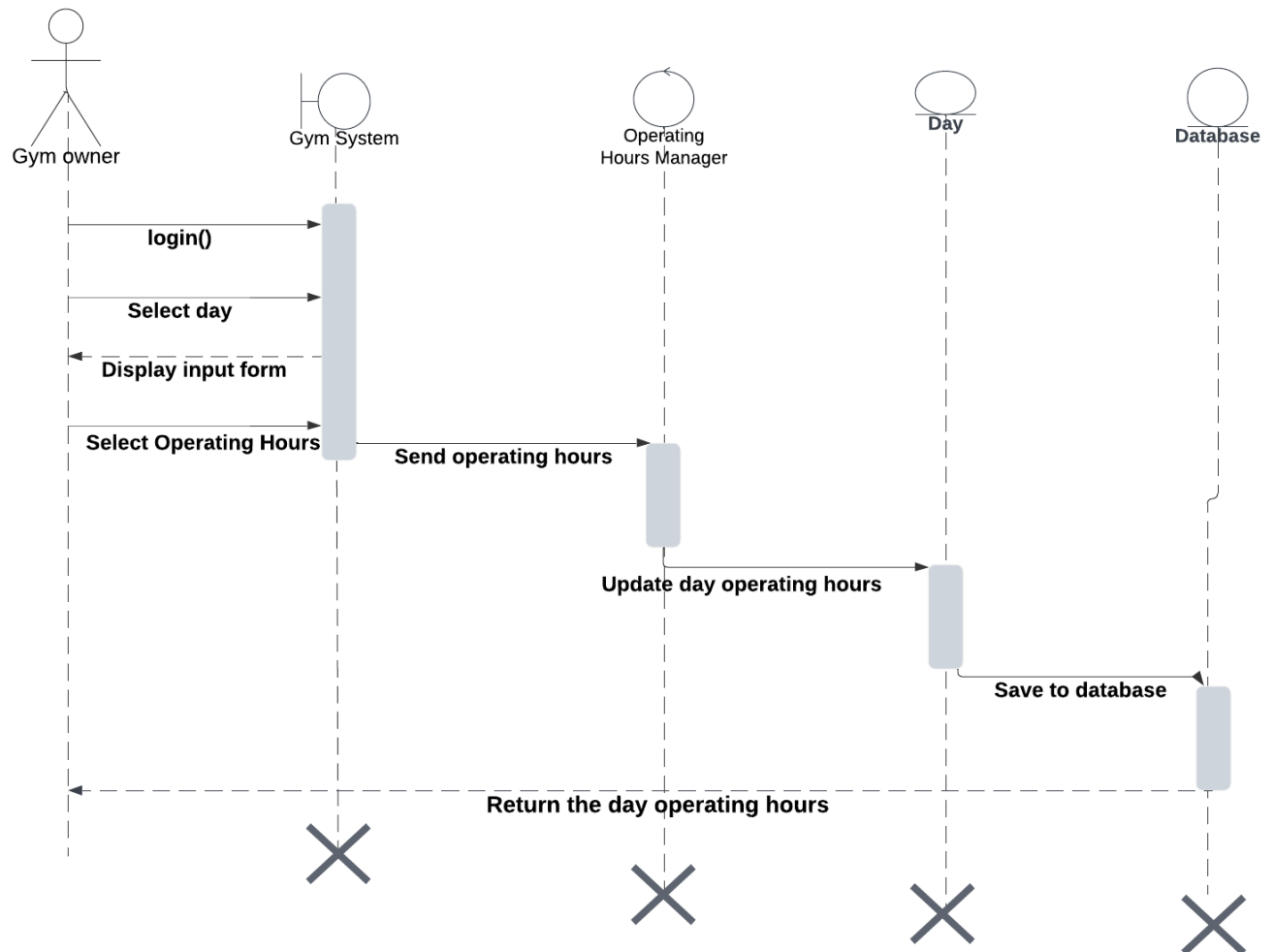
6) User Story 4: As a gym owner, I want to manage membership plans, like trial or premium with the option of changing their pricing or even the entire plan, so that I can focus on different clients' needs.
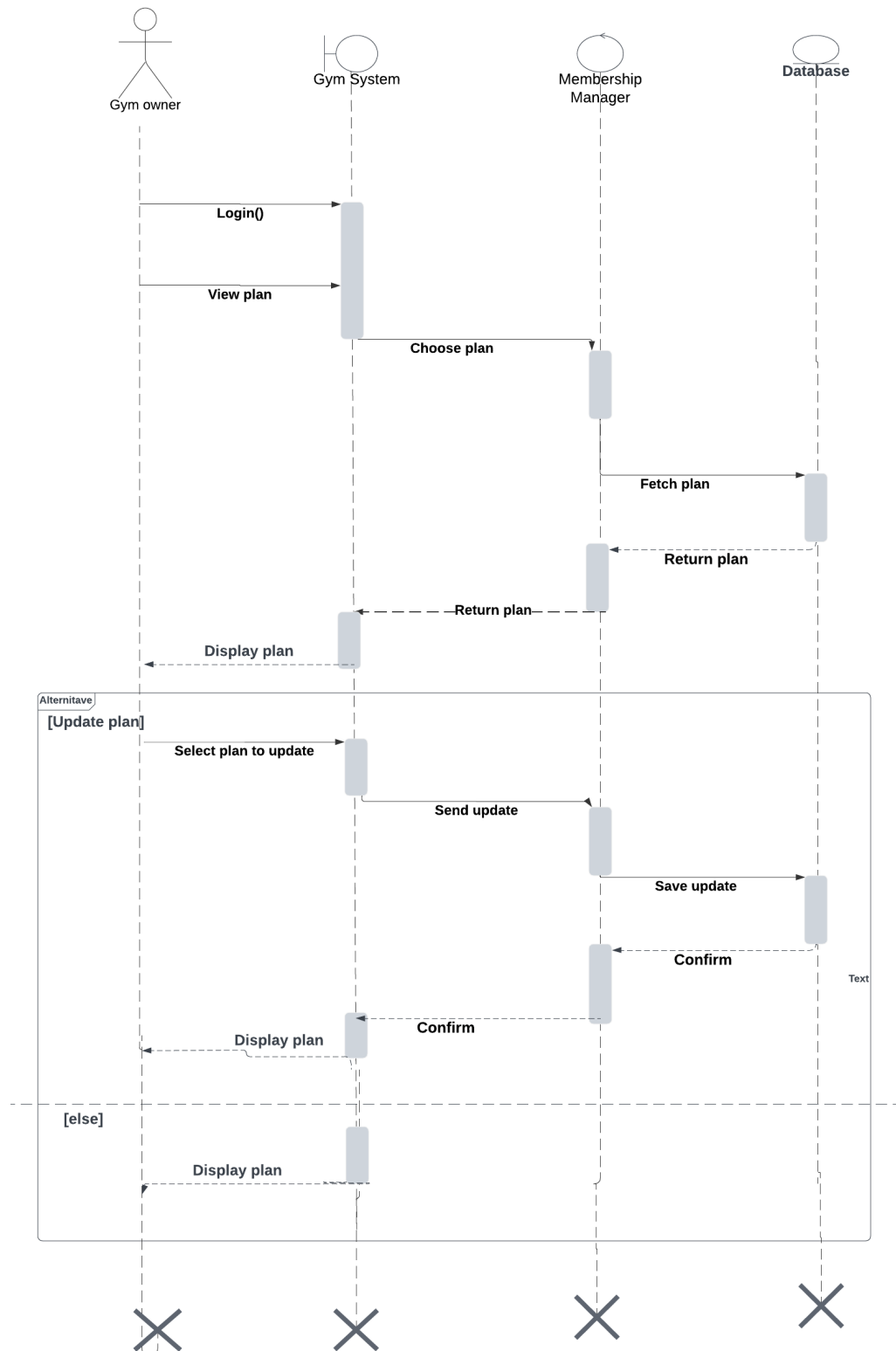
7) User Story 7: As a coach, I want to offer both online and offline training sessions, so can choose their preferred training mode.
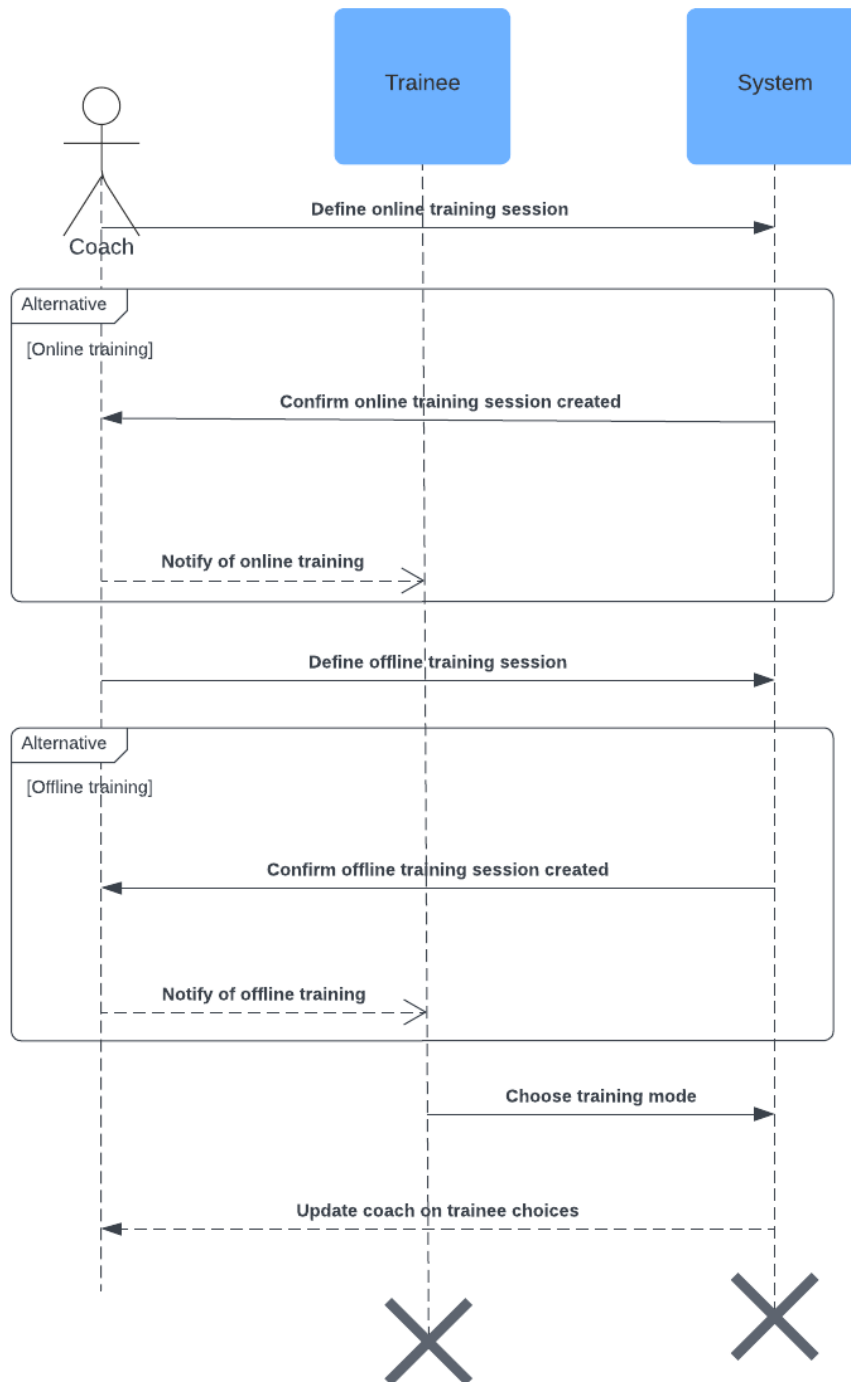
8)User Story 1: As a user (a user could be a trainee, coach, or gym owner), I want to register and log in the gym system, so that I can create a portfolio and explore the platform.
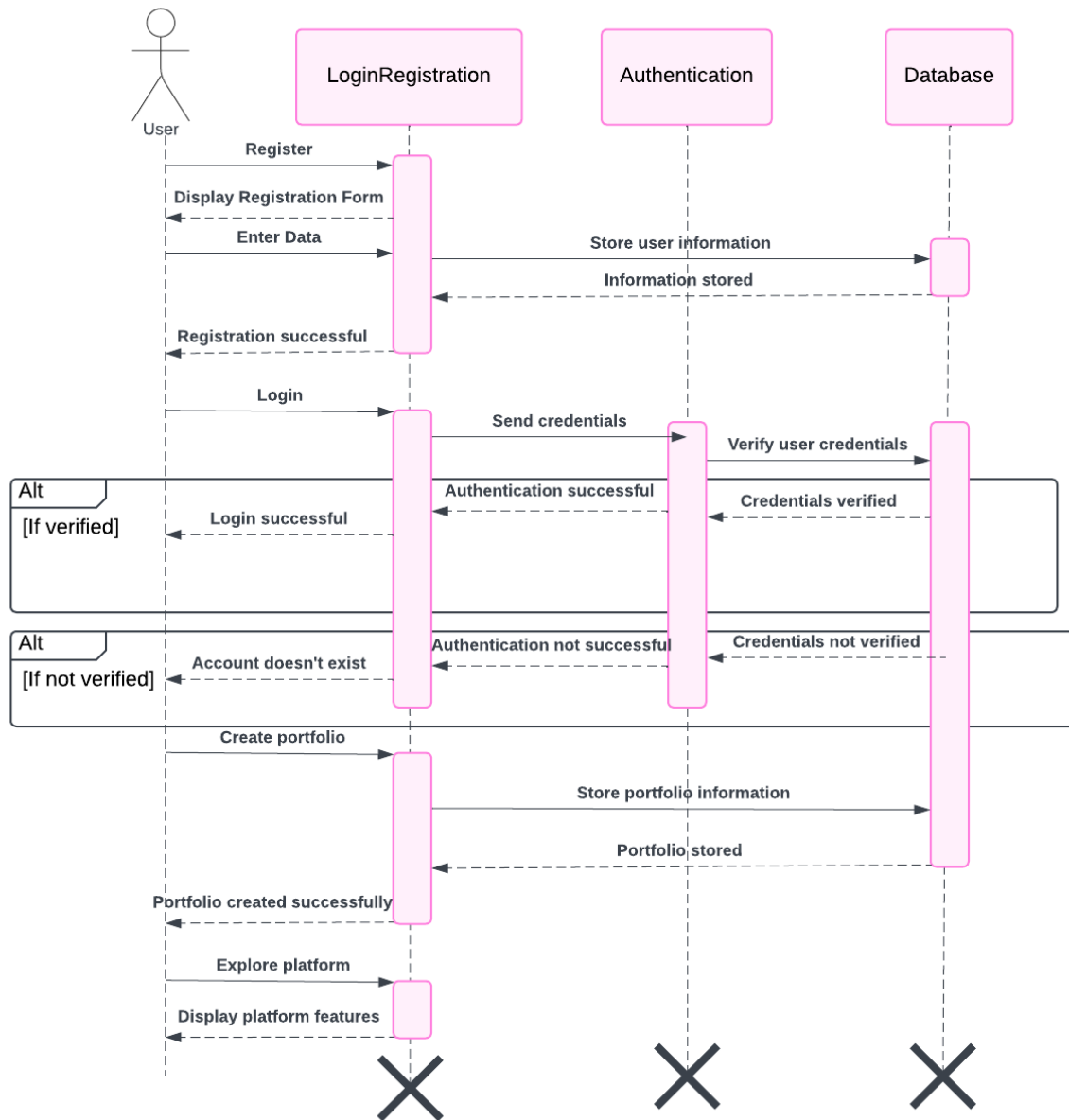
<u>Scope</u>

1)Payment methods:

Requirement 21: The Trainee should be able to choose the payment method whether using credit card or cash. (From User Story 21)

- **Payment Methode selection**: Validate that the trainee can successfully choose their preferred payment method (credit card or cash).
- **Payment Validation:** Ensure that the system correctly processes and validates credit card payments, including card details and transaction confirmation.
- **User Interface:** Confirm that the payment method selection interface is responsive and provides clear feedback to the user on the selected option.
- **Cash Option:** Ensure that when cash is selected, the system appropriately records this choice and provides necessary instructions for cash payments and a receipt with payment details.

2)Session Creation and Scheduling:

Requirement 6: The System should allow the coach to set and schedule sessions with the trainees (From User Story 6).

- **Session Creation:** Validate that a coach can successfully create a session with the required details (trainee, date, time, and session type).
- **Double-booking prevention:** Ensure that the system prevents double-booking by disallowing overlapping sessions for the same coach.
- **Data integrity:** Ensure that the system stores session data securely and correctly updates user profiles.
- **Performance Validation:** Validate that the session creation process completes within the system's performance standards which is 2 seconds.

3) Account Creation:

The system should allow all users to create their accounts. The focus of test cases will be on the validation of account creation for gym owners, trainees, and coaches, we need to cover all the requirements associated with the process of creating an account on our system. This is derived from requirement 1, security and performance requirements.

Functional Requirements:

1- Account Creation Process:

- o Ensure that gym owners, trainees, and coaches can create accounts with required details (name, email, password).
- o Ensure important fields are filled, and error messages are displayed if something is missing or incorrect.
- o Ensure verification for email and password meets strength criteria.

Non-Functional Requirements:

2. Performance:

- o Ensure that account creation process is done within 2 seconds.

3. Security:

- o Ensure password data is encrypted to protect user data.

4) Session type and feedback collection:

Requirement 8: The System gives the trainee an option to choose the type of training session, whether online or offline. (From User Story 11) .

- **Session Type Selection Functionality:** Verify that the system allows selection of training session type (online or offline).
- **Feedback Collection Functionality:** ensure that the system properly collects and handles feedback from trainees after a training session.
- **System Performance Under Load:** It ensures that the system can handle many users without slowing down or crashing.

Test Case Specification

1)Payment methods:

1.1 **Unit Test case:** Here we are only checking if the credit card field will detect expired cards and gives error message to the user.

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	status
TC21-01	Check if the credit card field will detect an error if the credit card is expired.	1. The trainee login into the system. 2. The trainee chooses a membership plan (trial or premium). 3. The Trainee is assigned to a coach	1. Open the payment page. 2. Select "Credit Card" as the payment method. 3. Enter a valid card details (I.16 digit card number, II. card expiry date, III. 3 digits CVV number) 4. Confirm the payment.	Card details: I. Card Number: 1111 1111 1111 1122 II. Expiry date: 01/**22** III. CVV: 123	Display message to the user depends on the specific reason for the failure: Your card has **expired**. Please use a valid card to complete the payment.	Your card has **expired**. Please use a valid card to complete the payment.	Pass

1.2 **Unit Test case:** Here we are only checking if the credit card field gives an error message if the user (Trainee) enters a card number less than 16-digit number or a CVV number less than 3-digit number.

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	status
TC21-02	Check if the credit card field will detect an error if one or more of the credit card numbers are missing.	1. The trainee login into the system. 2. The trainee chooses a membership plan (trial or premium). 3. The Trainee is assigned to a coach	1. Open the payment page. 2. Select "Credit Card" as the payment method 3. Enter a valid card details (I.16-digit card number, II. card expiry date, III. 3 digits CVV number) 4. Confirm the payment.	Card details: I. Card Number: 1111 1111 1111 12**--** II. Expiry date: 01/25 III. CVV: 12**-**	Display message to the user depends on the specific reason for the failure: **Invalid card details**. Please check your card number, expiry date, and CVV and try again.	**Invalid card details**. Please check your card number, expiry date, and CVV and try again.	Pass

1.3 Unit Test case: Here we are only checking if the credit card field makes a successful payment if all input numbers are correct and valid.

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	status
TC21-03	Check if the credit card field will make a successful Payment if all card numbers are correct and valid.	1. The trainee login into the system. 2. The trainee chooses a membership plan (trial or premium). 3. The trainee is assigned to a coach.	1. Open the payment page. 2. Select "Credit Card" as the payment method. 3. Enter a valid card details (I.16-digit card number, II. card expiry date, III. 3 digits CVV number) 4. Confirm the payment.	Card details: I. Card Number: 1111 1111 1111 1122 II. Expiry date: 01/**29** III. CVV: 123	A success message is displayed. Payment **successful!** Your subscription has been activated.	Payment **successful**! Your subscription has been activated.	Pass

1.4 **Unit Test case:** Here we are only checking if the cash payment field makes a receipt with the payment details and the payment instructions are displayed to the user (Trainee).

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	status
TC21-04	Check if the cash payment field makes a successful Payment.	1. The trainee login into the system. 2. The trainee chooses a membership plan (trial or premium). 3. The trainee is assigned to a coach.	1. Open the payment page. 2. Select "Cash" as the payment method 3. Confirm the payment.	1. The trainee selects "Pay by Cash" 2. The trainee clicks on "Confirm Payment Option"	• The system generates a receipt includes: 1. Subscription details (plan type, duration, start date) 2. Amount due 3. A unique reference number for tracking purposes. • The system provides instructions for completing the cash payment: 1. The location where cash payment can be made (e.g., gym front desk) 2. Payment hours or availability 3. Payment deadline	The system fails to generate receipts or display payment instructions. A blank page with an error message "**Unable to generate receipt**" is displayed.	Failed

2) Session Creation and Scheduling:

1) Unit Test case: Here we are only checking if creation process successfully creates a session with the required details (trainee, date, time, and session type).

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	Status
TC06-01	Validate session creation for a coach.	1.The coach is logged into the system and has a list of trainees assigned to them	1. The coach navigates to the "Create Session" page. 2. Coach selects a trainee from the list. 3. Coach selects the session type (individual/group). 4. Coach enters the session date and time. 5. Coach clicks the "Create Session" button to confirm.	Trainee: John Session Type: Individual Date: 27.12.2024 Time: 10:00 AM	The system successfully schedules the session and adds it to the coach's and trainee's calendar. A confirmation message is displayed.	A confirmation message is displayed "Your session has been successfully scheduled."	Pass

2) Unit Test case: Here we are only checking if the system prevents double-booking by disallowing overlapping sessions for the same coach.

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	Status
TC06-02	Ensure that the system prevents double-booking for the same coach at the same time	Coach is logged in and has an existing session for the selected time	1. The coach navigates to the "Create Session" page. 2. Coach selects a trainee from the list. 3. Coach selects the session type (individual/group). 4. Coach enters the session date and time. 5. Coach clicks the "Create Session" button to confirm. 6. Coach attempts to create another session at the same date and time.	Trainee: John Session Type: Individual Date: 27.12.2024 Time: 10:00 AM	The system should prevent double-booking and display an error message indicating the time slot is already taken.	An error message is displayed: **"Error: The selected time slot is already booked. Please choose a different time slot!"**	Pass

3) Unit Test case: Here we are only checking if the system stores session data securely and correctly updates user profiles.

TC06-03	Ensure that the session is successfully added to the trainee's profile after scheduling	Coach and trainee are logged into the system	1. The coach navigates to the "Create Session" page. 2. Coach selects a trainee from the list. 3. Coach selects the session type (individual/group). 4. Coach enters the session date and time. 5. Coach clicks the "Create Session" button to confirm. (Schedule creation steps) 6. Trainee logs into their account to check their schedule.	Trainee: Emily Session Type: Group Date: 27.12.2024 Time: 11:00 AM	The scheduled session should appear in the trainee's calendar and profile.	The scheduled session appears in the trainee's calendar and profile.	Pass

4) Unit Test case: Here we are only checking if the session creation process completes within the system's performance standards which is 2 seconds.

TC06-04	Validate system performance for session creation	Coach is logged in and the session creation interface is available	1. Coach attempts to schedule a session. 2. Measure time taken for response	Trainee: Tom Session Type: Group Date: 23.12.2024 Time: 01:00 PM	The session should be created within 2 seconds after clicking " Create Session ".	The session is created within 2 seconds after clicking " Create Session ".	Pass

3) Account creation:

3.1 **Unit Test case:** Here we are only checking if the creating account process is done correctly.

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	status
TC01-01	Verify successful account creation.	The user should be online to access our gym system's website through a web browser.	1. Tap on "Sign Up". 2. Enter important data like their name, their email, and their type whether they are a trainee, coach, or gym owner. 3. Create password. 4. Submit their information.	Name: Malak Email: abcde@example.com Type: Trainee Password: P@ssw0rd	Account is successfully created and a confirmation mail is sent.	1. A confirmation message is displayed **"Your account has been created successfully"** 2. An email is sent to the user to inform them of their registration on our platform.	Pass

3.2 **Unit Test case:** Here we are only checking if the email field gives an error message if the user, of any type, enters an invalid email format.

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	status
TC01-02	Validation of email format during account creation process.	The email field must be filled.	1. Tap on "Sign Up". 2. Enter information that includes an invalid email format. 3. Submit form.	Name: Malak Email: whatever Type: Trainee Password: P@ssw0rd	A message that says "**Error: "Enter a valid email address."** should be displayed.	An error message is displayed: "**Enter a valid email address.**"	Pass

3.3 **Unit Test case:** Here we are only checking if the password gives a warning massage if the password is weak.

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	status
TC01-03	Ensure password meets the strength criteria.	The password field must be filled.	1. Tap on "Sign Up". 2. Enter information that includes a weak password. 3. Submit form.	Name: Malak Email: abcde@example.com Type: Coach Password: 12345	A message that says, **"Password must include uppercase, lowercase, number, and special character."** Should be displayed.	An error message is displayed: **"Password must include uppercase, lowercase, number, and special character."**	Pass

3.4 **Unit Test case:** Here we are only checking the performance if it's done within 2 seconds.

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	status
TC01-04	Validate account creation is done within 2 seconds.	System is online and active.	1. Tap on "Sign Up". 2. Enter valid data, that includes a right email format, their name and what type of user they are. 3. Create a strong password. 4. Submit their information.	Name: Malak Email: abcde@example.com Type: Gym owner Password: P@ssw0rd	The account should be created within 2 seconds after clicking on "Create Account".	The account is created in less than 2 seconds.	Pass

3.5 **Unit Test case:** Here we are only checking the security, the password should be encrypted before storing.

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	status
TC01-05	Ensure that the system encrypts passwords and securely saves data.	There must be data to securely store, which means we need a created account with valid information to test this feature.	1. Create an account. 2. Verify password encryption in the database.	Name: Malak Email: abcde@example.com Type: Gym owner Password: P@ssw0rd	Password should be encrypted and not stored in plaintext.	The system should store data encrypted, like for the input test data: Encrypted Password: b'$2b$12$Wj0TeF6A1.TFo8EmLe3ZpeCnVRfwom	Pass

4) Session type and feedback collection:

4.1 **Unit Test case:** Here we are only checking if the system allows the trainee to select the type of training session as Online and confirms the selection. (functional)

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	status
TC08-01	Verify the trainee can select the type of training session.	1. Trainee login. 2. Trainee books a training session.	1. Go to the session selection page. 2. Select "Online" option. 3. Click the "Confirm" button.	1. Select the "Online" option. 2. Click the "Confirm" button.	The system should accept and confirm the selected session type "Online" And display a confirmation message "**Online selection is successfully selected.** "	A confirmation message is displayed "**Online selection is successfully selected.** "	Pass

4.2 **Unit Test case:** Here we are only checking if the system correctly collects feedback after a training session and confirms the submission. (functional)

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	status
TC08-02	Verify the system collects feedback from trainees after a session.	1. Trainee login 2. Trainee books a training session. 3. Trainee has completed a training session.	1. Trinee navigates to the feedback page. 2. Enter feedback text. 3. Submit the feedback.	Trainee enters a feedback text ("Great session")	The system should successfully collect and submit the feedback provided by the trainee and display a message **'Thank you for your feedback!'**	The feedback was successfully submitted, and a confirmation message was displayed saying **'Thank you for your feedback!'**	Pass

4.3 **Unit Test case:** Here we are only checking if the system performs well under the load of multiple users selecting training session types and submitting feedback. (Non-functional)

Test Case ID	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	status
TC08-03	Verify system performance when multiple users select session types and submit feedback	Multiple users are logged in	1) Simulate 100 users going to the session selection page. 2) Each user picks either "Online" or "Offline" and submits their choice. 3) Each user provides feedback and submits it.	100+ concurrent users	The system should handle up to **100 concurrent users** without slowing down or crashing, and responses should be processed within 2 seconds.	The system handled 100+ users without crashes. The response time for selecting "Online" or "Offline" was under 2 seconds, and feedback submission was processed quickly.	Pass

Submission Guidelines

- Format: Teams are required to submit their project documentation as a PDF.

- Submission Method: The completed document should be uploaded through the classroom platform by the specified deadline.

- You should submit one pdf file with naming convention:

  ID1_ID2_ID3_ID4.pdf

- Each TA will post a submission link for his/her groups, so please submit the file to the link of your group

Deadline

Friday 27 December 2024 at 11.59 PM

No late submissions accepted; no apologies so be sure to submit on time

and don't wait until the last second

Grading Criteria

Phase 2 update	2
Scope defined correctly	3
TestCases Desing template	Each team member should design 3 test cases , total Grade 5