



Birzeit University

Faculty of Engineering & Technology

Computer Science Department

Software Engineering- Comp433

G #2

FINAL (COMPLETE PROJECT) REPORT

Prepared by:

Yara Darabumukho (Project manager) 1211269

Dana Assad (Programmer) 1211452

Mohammad Manasrah (Technical architect) 1211407

Roa Daraghme (Requirement Engineer) 1221349

Section: 2

Instructor: Prof. Adel Taweel

Date: 17/1/2024

Contents

1	Chapter 1: Project Planning and Management	2
1.1	Names of Editors/writers of the Report	2
1.2	Business Title	2
1.3	Group Name: Group Number	2
1.4	Name of students/members	2
1.5	Role of each member	2
1.6	Project management strategy	2
1.7	Project Manager Report	2
1.8	Group members report	4
2	Chapter 2: Requirement Elicitation, Analysis and Modeling	6
2.1	Requirement Statement/Business Description	6
2.2	USER and SYSTEM requirements	9
2.3	SCENARIOS	14
2.4	Effort/Time estimation calculation	19
2.5	ACTORS analysis	21
2.6	USE-CASE diagram	22
2.7	Description of key USE-CASES	24
2.8	ACTIVITY diagram	34
2.9	Instance Activity diagrams	35
3	Chapter 3: System Analysis and Modelling	39
3.1	System CLASS Diagrams	39
3.1.1	Analysis class model	39
3.1.2	detailed class model	40
3.2	SEQUENCE Diagrams	41
4	Chapter 4: System Design and Modelling	45
4.1	Description of chosen Design Goals	45
4.2	Component Diagram	46
4.3	Overall architecture diagram	47
4.4	Deployment diagram	48
5	G2 team meetings	49
6	course feedback	49

1 Chapter 1: Project Planning and Management

1.1 Names of Editors/writers of the Report

writers of the Report: Mohammad manasrah 1211407

1.2 Business Title

Online flight ticket booking System

1.3 Group Name: Group Number

G#2

1.4 Name of students/members

Yara Darabumukho 1211269

Mohammad Manasrah 1211407

Dana Assad 1211452

Roa DaraghmeH 1221349

1.5 Role of each member

Yara Darabumukho (Project manager)

Dana Assad (Programmer)

Mohammad Manasrah (Technical architect)

Roa DaraghmeH (Requirement Engineer)

1.6 Project management strategy

Meetings: We planned to convene at least ones a week as a face-to-face meeting to do specific phases and ensure timely completion before the deadline. But due to the constraints of face-to-face interaction, our meetings primarily done online using Google Meet.

Decisions: we took everyone's opinion and discuss each one of them, then vote for the best idea for the project and in case of tie occurred, the manager take the final decision. Process

models: we use waterfall in our project, to ensure that everything is perfect. Because we see it's the best model for our project.

1.7 Project Manager Report

It was a big project for all of us, and the work was divided fairly while considering the individual workload and pressure on each member. Each member worked on their individual tasks, while for group tasks, the entire team collaborated. We used **Lucidchart** for graphical tasks, which allowed us to work together in real-time. The following summarizes the contributions of each team member:

- **Yara Darabumukho:** As a **Group Manager**, her responsibilities encompassed establishing project milestones, coordinating and scheduling meetings, and fostering clear

communication among team members. She led the creation of the **Business Description, User Requirements Specifications, Use-Case Activity Modelling, and System and Component Modelling**. Additionally, she discussed the **System Requirements Specifications** and reviewed and edited the **Actor Analysis, Use-Case Modelling, System Class Modelling and Analysis, System Design Goals, System and Architectural Design, and System and Deployment Modelling**. Individually, she developed a scenario detailing the **ticket booking process**, including a **Use-Case Description, Activity Model, and Sequence Diagram**.

- **Roa Daraghma:** As the **Requirements Engineer**, she was responsible for defining **User and System Requirements** and ensuring their correct application. Roa led the **Effort and Time Estimation**, guided the **System Design Goals**, participated in discussions on **System Requirements**, and reviewed **Use-Case Modelling, Actor Analysis, System Class Modelling, System Component Modelling, and System Deployment Modelling**. She individually created a scenario for **"Create an Account"**, including a **Use-Case Description, Activity Model, and Sequence Diagram**.
- **Mohammad Manasrah:** As a **Technical Architect**, he spearheaded the formulation of **Design Goals** and provided justification for utilizing a **Layered Architecture**. Mohammad contributed to **Actor Analysis** through reviewing and editing, led **Use-Case Modelling**, and contributed to **Use-Case Activity Modelling**. Additionally, he was the **Team Lead** for **System Class Modelling and Analysis** and contributed to reviewing and editing **System Design Goals, System and Component Modelling, and System Deployment Modelling**. He also led the **System and Architectural Design** and wrote the **Description of Design Goals**, which was later reviewed by all team members. Individually, he created the scenario for the **"Make Payment"** process, including a **Use-Case Description, Activity Model, and Sequence Diagram**.
- **Dana Assad:** As a **Programmer**, she was responsible for ensuring the correct application of requirements in the coding phase. Dana led **Actor Analysis, Deployment Modelling, and System Requirements Specification**. She participated in discussions on **User Requirements, Effort and Time Estimation, Use-Case Modelling, and Activity Modelling**. Additionally, she reviewed the **Design Goals, System Class Modelling Analysis, and Architectural and Component Modelling**. Individually, she created a scenario for **"Client Log In"**, including a **Use-Case Description, Activity Model, and Sequence Diagram**.

As fourth-year **Computer Engineering** students and third-year **Computer Science** students, this semester was particularly challenging. This course was no less demanding than any other technical course and required significant effort. One of the biggest challenges was finding common time for group meetings amid our busy schedules. However, this project was a great opportunity to experience the reality of professional software development and different types of projects, which was both exciting and insightful. It significantly enhanced our **teamwork, communication skills, and problem-solving abilities**, allowing us to gain new experiences and a deeper understanding of how real-world projects operate.

1.8 Group members report

Technical Architect: Mohammad Manasrah

The project was successfully built with efficiency, following a well-structured strategy and maintaining strong communication despite the challenges of an emergency state due to the ongoing war situation. Regular virtual meetings via Google Meet ensured smooth collaboration, with each phase involving discussions, task division, and feedback to keep the project on track. My contributions included engaging in user requirements development by contacting customers, reviewing system requirements, and participating in effort and time estimation. I was involved in Actor Analysis through reviewing and editing, led the Use-Case Modelling efforts, and contributed to the Use-Case Activity Modelling. Additionally, I was the team lead for System Class Modelling and Analysis, contributed to reviewing and editing System Design Goals, System and Component Modelling, and System and Deployment Modelling, while also leading the System and Architectural Design. Individually, I wrote the Use-Case Specification and designed the Use-Case Activity Modelling Diagram for the Make Payment Use Case, developed the Sequence Diagram, and wrote the description of design goals, which was reviewed by all team members. Furthermore, I wrote an individual scenario, providing a detailed representation of system interactions in the Make Payment process. This project was a valuable learning experience, allowing us to apply software engineering principles effectively, and through strategic planning and teamwork, we successfully delivered a well-developed system.

Programmer: Dana Assad

Reflecting on this course, I have gained valuable insights into project management and software development, particularly in user requirements gathering and leading system requirements to ensure practical and well-defined specifications. I actively participated in effort and time estimation discussions, contributing to accurate planning and resource allocation. Additionally, I conducted a scenario analysis and detailed use case scenario for “Client Log In,” ensuring a comprehensive understanding of system interactions. As the leader in Actor Analysis, I played a key role in refining system roles and responsibilities. I also contributed to Use-Case Modelling and Activity Modelling, ensuring logical workflow representation. My role extended to reviewing system class modelling analysis and design goals, ensuring clarity and consistency. Furthermore, I took the lead in deployment modelling and reviewed architectural and component modelling, ensuring a well-structured system design. I was also responsible for Use-Case Activity Modelling for “Client Log In” and contributed to System Sequence Modelling and Analysis for “Log In”, ensuring a well-defined and seamless authentication process. The guidance provided by Dr. Adel Tawil significantly enhanced my decision-making skills, and this course has strengthened my technical expertise while boosting my confidence in handling real-world project execution and future challenges.

Requirement Engineer: Roa Daraghmeh

I am proud to evaluate our project as a remarkable success, achieving all objectives with high quality and within the set timeline through excellent teamwork. As the Requirements Engineer, I played a vital role in defining and documenting the system’s requirements, leading the Effort and Time Estimation task, guiding the System Design Goals, and actively participating in discussions on System Requirements. I also contributed to reviewing key tasks such as Use Case Modelling, Actor Analysis, System Class Modelling, System Component Modelling, and

System Deployment Modelling, ensuring clarity and consistency. Additionally, I worked individually on Scenario Analysis for "Create an Account," Use-Case Specifications, Use-Case Activity Modelling, and System Sequence Modelling and Analysis, ensuring a structured and well-defined approach to system functionality. This project has been an invaluable experience, allowing me to enhance my expertise in software engineering and system analysis while reinforcing my ability to manage project requirements effectively.

2 Chapter 2: Requirement Elicitation, Analysis and Modeling

2.1 Requirement Statement/Business Description

La Petite Boulangerie

Business Description

La Petite Boulangerie specializes in offering a wide variety of freshly baked breads, pastries, and desserts through an **online ordering system**. Our platform enables customers to conveniently browse our menu, select their preferred baked goods, and place orders for **home delivery** or **in-store pickup**. The system assists us in **managing product procurement, inventory listing, order processing, and scheduling deliveries efficiently**. Our goal is to ensure a seamless and satisfying experience for customers by simplifying the **ordering and delivery process**, making high-quality baked goods accessible anytime.

Business Services

Our bakery provides the following business services to ensure a **smooth customer experience**:

- **Online Ordering System:** Customers can place orders via our website or mobile application, ensuring a hassle-free shopping experience.
- **In-Store Purchases:** Customers can visit our shop and select from available baked goods.
- **Customized Orders:** Special requests such as personalized cakes and large event catering can be placed in advance.
- **Delivery Service:** Orders are delivered to customers' doorsteps, ensuring fresh and timely deliveries.
- **Secure Payment Options:** Customers can choose to pay via **cash on delivery, credit/debit card, or online payment gateways**.
- **Loyalty Program & Discounts:** Regular customers can earn reward points and avail of discounts on special occasions.

Working Process

The process of handling and fulfilling a customer order at La Petite Boulangerie is structured as follows:

1. **Order Placement:** Customers place orders via:
 - The online ordering system (website or app).
 - Phone calls for customized or bulk orders.
 - In-store purchases.

2. **Order Details Collection:** The system records essential details such as:
 - List of ordered items.
 - Quantity and any special customizations.
 - Delivery or pickup preferences.
3. **Customer Information Gathering:**
 - Name, phone number, and email (for order tracking).
 - Delivery address (for home deliveries).
 - Payment method (cash, card, or online).
4. **Order Confirmation:** The system verifies the order and provides a confirmation notification with an estimated delivery time.
5. **Preparation and Packaging:** The bakery staff prepares the order, ensuring that it meets quality standards and is packaged appropriately for delivery or pickup.
6. **Delivery or Pickup:**
 - If the order is for delivery, our logistics team ensures that the package reaches the customer on time.
 - If the order is for pickup, the customer is notified when it is ready.
7. **Payment Processing:** Transactions are securely processed based on the customer's chosen payment method.

Company Capabilities

La Petite Boulangerie serves **500-1000 unique customers per month**, with variations based on **seasonal demand and promotions**. We handle an estimated **2500-3500 units of baked products per month**, ensuring that quality and freshness remain consistent. Our online ordering system operates for **five hours daily**, allowing **5-15 simultaneous customers** to place orders during peak hours.

We offer **four main product categories**:

- **Breads (1000-1500 units/month):** Sourdough, Focaccia, Bagel, Whole Wheat Bread, and Naan.
- **Desserts (700-900 units/month):** Cakes ("Vanilla and Chocolate"), Cookies ("Vanilla and Chocolate"), and Donuts ("Classic and Chocolate").
- **Croissants (500-700 units/month):** Chocolate, Butter, Almond, and Sandwiches.
- **Pastries (300-400 units/month):** Danish Pastries, Éclairs, and Puff Pastry Twists.

Workforce and Roles

Our bakery operates efficiently with a team of **10-15 employees**, each assigned specific roles to ensure smooth operations:

- **(5-7) Bakers and Kitchen Staff** – Responsible for baking, quality control, and order preparation.
- **(2-3) Delivery Personnel** – Ensures timely delivery and customer satisfaction.
- **(1) Accountant** – Manages financial transactions, payroll, and business expenses.
- **(2) Digital Marketing and Social Media Specialists** – Handles online promotions, customer engagement, and brand awareness.
- **(1-2) Customer Support Representatives** – Assists customers with inquiries, complaints, and order tracking.
- **(1) Manager** – Oversees bakery operations, staff coordination, and business strategy.

flight ticket booking management Description

Our **online flight booking system** is designed to provide clients with a **fast, secure, and user-friendly** platform to book, modify, and manage their flights with ease. By integrating advanced search features, real-time flight recommendations, and a seamless booking process, we aim to eliminate the hassle of traditional flight reservations. Customers will be able to **search for flights, select their seats, specify special requirements, and complete payments** through secure gateways like Visa and PayPal. Additionally, an **AI-powered chatbot** will be available 24/7 to assist customers with inquiries, ensuring a smooth user experience.

For airlines, the system offers a **comprehensive staff dashboard** to manage flights, monitor seat availability, and adjust pricing dynamically. Real-time updates on **flight status, cancellations, and customer service queues** will help airlines provide better service while optimizing operations. Furthermore, **automated sales reports** will be generated weekly, offering key insights into sales performance and business growth. By combining **convenience, security, and automation**, our platform ensures that both customers and airline staff benefit from a more efficient and reliable booking experience.

2.2 USER and SYSTEM requirements

USER REQUIREMENTS

[Group Task] **Lead:** [Yara]; **contributors:** [Roa: reviewing], [Dana: discussion], [Mohammad: Contact with customers]

- **UR1:** The system shall enable users to create personal accounts by providing their email, password, full name, contact information.
- **UR2:** The system shall include a secure login mechanism where users can access their accounts using their email and password, ensuring privacy and security of personal data.
- **UR3:** The system shall provide users with the top 10 available flights before they initiate a search. Additionally, users shall be able to search for flights by specifying their departure city, destination city, travel dates, and the number of passengers
- **UR4:** The system shall support a chatbot, to answer the questions from the customers.
- **UR5:** The system shall provide a secure booking process. where Users shall input passenger details, specify special requirements such as meals or wheelchair assistance, select baggage options, and choose their preferred seats.
- **UR6:** The system should allow users to cancel or modify their booking within 48 hours before the flight.
- **UR7:** The system shall prompt users to confirm their bookings and complete the payment using secure methods, such as Visa and PayPal.
- **UR8:** The system shall offer staff a comprehensive dashboard to manage various flight-related operations. Additionally, staff shall be able to oversee customer service queues, handle system alerts, and review performance metrics.
- **UR9:** The system shall automatically generate sales reports every week for the administrative staff and provide a button for manual report creation

SYSTEM REQUIREMENTS

[Group Task] **Lead:** [Dana]; **contributors:** [Mohammad: reviewing], [Roa: discussion], [Yara: discussion]

UR1: The system shall enable users to create personal accounts by providing their email, password, full name, contact information.

- **SR1.1:** When creating an account, the system shall allow the user to enter the following fields: full name, password, email address, and phone number.
- **SR1.2:** The system shall automatically checks the input data during the account creation process to make sure all required fields are completed correctly
- **SR1.3:** The password should contain eight or more characters and one capital letter, while the email address shall be in a valid format and unique, not already associated with an existing account.
- **SR1.4:** the system shall block the account when the user enter invalid informatoins many times like invalid Email
- **SR1.5:** The system shall display a success message to the user upon successful registration and activation of the account.

UR2: The system shall include a secure login mechanism where users can access their accounts using their email and password, ensuring privacy and security of personal

- **SR2.1:** The system shall display an appropriate error message when the user enters an incorrect email or password combination.
- **SR2.2:** The system shall ensure that passwords are securely Hashed using SHA256 during transmission and storage to protect users' sensitive information
- **SR2.3:** The system shall lock the user account after 5 failed login attempts to prevent brute-force attacks and notify the user by email.
- **SR2.4:** The system shall provide an optional **Remember Me** feature that allows users to stay logged in across sessions on trusted devices.

UR3: The system shall provide users with the top 10 available flights before they initiate a search. Additionally, users shall be able to search for flights by specifying their departure city, destination city, travel dates, and the number of passengers

- **SR3.1:** Before the user performs any search. The system shall automatically fetch and display the top 10 most relevant flights on the home page. With data refreshed every 30 minutes.
- **SR3.2:** The system shall determine the top 10 flights based on Current popularity (booking frequency), price offerings, search frequency of routes, and user ratings.
- **SR3.3:** The system shall display each flight entry with the following essential information: airline name, flight number, departure city and time, arrival city and time, and base fare.
- **SR3.4:** The system shall provide a search interface that accepts required inputs (departure city, destination city, travel date(s), and passenger count) and enables users to sort/filter displayed results by price (ascending/descending), departure time, or travel duration
- **SR3.5:** The system shall validate all search inputs (departure city, destination city, travel dates, passenger count) and display specific error messages when validation fails (e.g.: Cities must exist in airport database, Travel dates must be current or future dates or required fields cannot be empty)
- **SR3.6:** The system shall process the search request and display matching flight results within 3 seconds.

UR4: The system shall support a chatbot, to answer the questions from the customers.

- **SR4.1:** The system shall provide users with a chat interface that includes a message input field and a conversation display area.
- **SR4.2:** The system shall clearly show the difference between user questions and chatbot answers while displaying chat messages sequentially.
- **SR4.3:** The system should process user input and respond within 3 seconds.
- **SR4.4:** The system shall encrypt all chat conversations and user data using AES encryption algorithm with a 128-bit key length

UR5: The system shall provide a secure booking process. where Users shall input passenger details, specify special requirements such as meals or wheelchair assistance, select baggage options, and choose their preferred seats.

- **SR5.1:** The system shall allow the user to modify their booking details before confirming the reservation.
- **SR5.2:** If any changes are made to the booking, users shall be notified of the updated details
- **SR5.3:** The system shall provide users with a summary of their booking, including costs, baggage options, and selected seats before final confirmation.

UR6: The system should allow users to cancel or modify their booking within 48 hours before the flight.

- **SR6.1:** The system must allow the user to modify or cancel the booking up to 48 hours before the flight.
- **SR6.2:** The system shall provide a full refund to the client if the flight is canceled.

UR7: The system shall prompt users to confirm their bookings and complete the payment using secure methods, such as Visa and PayPal.

- **SR7.1:** If any error occurs during payment, the process shall not exceed 10 seconds, and no funds shall be deducted from the user.
- **SR7.2:** user's card information shall be securely stored for future payments.
- **SR7.3:** Users shall receive detailed payment information for every online transaction
- **SR7.4:** All payment transactions shall be encrypted using RSA algorithm to protect sensitive user information
- **SR7.5:** Users shall have the option to avoid saving their card information for future payments.
- **SR7.6:** A notification confirming the successful completion of the payment shall be sent to users
- **SR7.7:** Users shall be able to review their payment history through their account

UR8: The system shall offer staff a comprehensive dashboard to manage various flight-related operations. Additionally, staff shall be able to oversee customer service queues, handle system alerts, and review performance metrics.

- **SR8.1:** The system shall allow staff to add new flights, including specifying flight details such as destination, departure time, and aircraft.
- **SR8.2:** The system shall enable staff to update and modify flight schedules, including changes to departure and arrival times, ensuring the system reflects the most accurate information.
- **SR8.3:** The system shall permit staff to adjust seat availability for flights, ensuring that the number of seats available aligns with flight capacity and bookings.
- **SR8.4:** The system shall allow staff to modify flight pricing, including adjusting fare categories, discounts, or special offers.
- **SR8.5:** The system shall provide staff with real-time updates on flight status, allowing them to track delays, cancellations, and other operational changes.
- **SR8.6:** The system shall enable staff to view, modify, and manage current bookings, including customer reservations, seat assignments, and special requests.
- **SR8.7:** The system shall allow staff to send emergency notifications, such as flight cancellations, accidents, or other critical updates, to customers and relevant parties promptly.

UR9: The system shall automatically generate sales reports every week for the administrative staff and provide a button for manual report creation

- **SR9.1:** The system shall allow configuration of the report schedule “Day and time”.
- **SR9.2:** The report generation shall allow the staff to choose the reporting period “Daily, monthly, yearly”.
- **SR9.3:** The report shall contain a key matrix for example: sales by region, average sales, net profits, sales by companies and so on.
- **SR9.4:** the system shall provide a button to generate the report manually
- **SR9.5:** The reports shall be in a readable format.

2.3 SCENARIOS

Mohammad's Scenario: Make Payment

Initial Assumption

The user has successfully logged into their account and selected a service they wish to purchase. The system has calculated and displayed the total amount. The system is in a valid state and ready to process payments through multiple payment methods (e.g., Visa, PayPal). The user has navigated to the payment page.

Normal Flow

The user accesses the payment interface, where they are presented with the available payment methods for their booking. The system displays valid payment options, including Visa and PayPal. The user selects their preferred payment method. If they choose a credit or debit card, the system prompts them to enter the required payment details, including the card number, expiry date, and CVV code. If they select PayPal, they are redirected to PayPal's authentication page.

After entering their payment details, the system asks whether they want to save their payment information for future transactions. If the user chooses to save their payment details, the system encrypts all payment data using the RSA algorithm. The system then securely processes the transaction and validates the payment details. Upon successful payment, the system generates a detailed transaction record, sends a confirmation notification to the user, and updates their payment history accordingly.

Alternative Flow 1: Unsupported or Invalid Payment Method

If the selected payment method is unsupported or invalid (e.g., unsupported card type, invalid PayPal account), the system displays an error message and prompts the user to choose a different payment method. The user is returned to the payment selection screen, where they can select another available payment method.

Alternative Flow 2: User Chooses Not to Save Card Information

If the user decides not to save their card details, the system proceeds with processing the payment without storing any payment information. Once the transaction is successfully completed, the user receives a confirmation notification, and their payment history is updated accordingly.

Error Flow

If the payment processing fails due to reasons such as invalid card details, network failure, system timeout, or bank authorization failure, the system halts the transaction within 10 seconds to prevent any unintended deductions. The system ensures that no funds are withdrawn, displays an error message, and prompts the user to retry the payment process or select a different payment method.

Other Activities

Transaction logs are maintained throughout the process. All payment details are securely encrypted using the RSA algorithm. The system also monitors transaction processing time to ensure it complies with the required 10-second limit.

System State on Completion

The user remains in an authenticated session. A payment record is created in the system database, containing essential details such as the transaction ID, payment amount, selected payment method, transaction timestamp, encryption status, and user preferences regarding card storage. A confirmation notification is then sent to the user's registered contact information.

Yara's Scenario: Manage Flight Ticket Booking Process

Initial Assumption

The client shall be logged into their account with stable internet device, and search for a flight by enter leaving and destination locations, travel dates, and the number of the travelers from these categories "Adults, Children, infants on lap, infants on seats". Then the system shows the clients the available flights labeled by the airline and the tickets cost, after the client choose the flight, the system has verified flight availability, and the client has proceeded to the booking section.

Normal Flow

The client accesses the booking interface and enter his personal information such as full name, date of birth, contact information of the client as displayed on their travel documents. The system then moves to the special requirements part where the client can select preferred meals or request for wheelchair. The client then moves to the next step where he choose the seat by interacting with the drop-down displayed customary aircraft layout. After that, for baggage options, presented options in terms of carry-on and checked baggage allowances together with the costs. During the booking process the client can edit prior entries as deemed necessary. In case no payment is made online, the system produces a summary of the booking that states all the options selected, the total cost and all the details of a booking session. The client goes through this information. The client confirms the booking and once confirmed a confirmation email sent to the client with all booking information.

Alternative Flow 1:

The selected seat is no longer available as another traveler has booked it during this client booking and the system did not update it frequently. The system displays updated seat availability and ask client to select a new seat.

Alternative Flow 2:

Client needs to modify their booking details or special requirements after initial entry before the last 48 hours. The system allows changes and updates the booking summary accordingly.

Error Flow

The client lost their internet connection during the booking process. As a result, they are unable to access the system, and all entered data will be lost if the client did not complete the payment.

Other Activities

Booking details are being temporarily saved while client completes the process, seat map is continuously updated to reflect the remains seats at the real-time, and system monitors session time to ensure booking completion within allowed time frame.

System State on Completion

Clients remain in active session while booking record is created with:

- Traveler details
- Selected seat assignments
- Special requirements
- Baggage selections
- Total cost breakdown
- Booking timestamp

Confirmation notification is prepared for sending after payment completion.

Roa's scenario: Create an account

Initial Assumption

The client wants to create an account using a device with a stable internet connection. This service can only be invoked by users who are not logged into the system and do not already have an account. The registration process requires the client to provide a valid email address, full name, password, and any other necessary personal details, such as date of birth and phone number, to successfully complete the account creation.

Normal Flow

The client begins the account creation process by clicking on the "Create Account" button. Before proceeding, the system checks if the client is already logged into an existing account. If the client is logged in, the system halts the process and displays the message: "You are already logged in." If the client is not logged in, the system displays the registration form, and the client starts the registration process by entering their full name and email address. The system validates the email format to ensure it adheres to standard email conventions and checks if the email is already associated with an existing account. If the email is invalid or already in use, the system halts the process and displays an appropriate error message: "Email already registered. Please log in or use another email." If the email is valid and unique, the system prompts the client to create a password and confirm it. The system ensures the password meets the required

security standards, such as minimum length, inclusion of special characters, and complexity, and verifies that the confirmation password matches the original password. If the password does not meet the security criteria, the system displays the message: “Password does not meet security criteria.” It also provides guidance to improve the password’s strength in real-time. Once the password is validated, the client is prompted to enter their date of birth. The system checks whether the client meets the minimum age requirement of 18 years. If the client does not meet the age requirement, the process stops, and an error message is displayed: “You must be at least 18 years old to create an account.” If the age requirement is met, the client provides their phone number. The system securely saves the client’s information. A verification email containing a unique link is sent to the provided email address. The system displays the message: “Verification email sent. Please check your inbox.” Once the client clicks the verification link, the system activates the account and displays the confirmation message: “Your account has been successfully created. You can now start using your services.”

Alternative Flow 1: Account Creation Using Phone Number

If the client prefers not to use an email address, they can create an account using their phone number. The client provides their full name, phone number, and password. Upon successful submission, the client provides their date of birth for age validation. If the client meets the age requirement, the account is successfully created, ensuring clients without email access can complete registration.

Alternative Flow 2: Guided Password Setup

If the password is weak or mismatched, the system displays a password guidance panel. This panel provides clear, real-time feedback on the password’s strength (e.g., “Weak,” “Moderate,” “Strong”) and outlines specific security requirements, such as minimum length, inclusion of special characters, and uppercase letters. The client updates their password based on the feedback, and the system validates it continuously. Once the password meets all requirements and matches the confirmation, the client proceeds with account creation.

Error Flow

If the client enters a date of birth indicating they do not meet the minimum age requirement of 18 years, the system’s age validation mechanism fails. As a result, the system halts the account creation process and displays the error message: “You must be at least 18 years old to create an account. “The system logs this error for auditing purposes and ensures no incomplete or invalid account is created in the database.

System State on Completion

After successfully creating an account, the system securely stores the client’s information and designates the account as “*pending verification*” until email confirmation is completed.

Upon clicking the verification link:

- The account status transitions to “*active*,” enabling full access.
- During this process, the system captures critical details, such as the sign-up timestamp and IP address, to enhance auditing and security protocols.

If the registration attempt fails:

- The system ensures no incomplete or duplicate accounts are created by logging all errors, thus maintaining robust data integrity and security measures.

Dana's scenario: A secure login**Initial Assumption**

The client is an active user who has an account with the online flight booking system. The user shall have entered the correct login details and the user is now authenticated. The option must allow the user to subsequently search for flights, book a ticket, or use any other service available in the system when logged in.

Normal Flow

After accessing the login screen, the client enters their registered email address and password and clicks the "Login" button. The system verifies the credentials by comparing them against the stored data in the database. If the credentials match, the client is redirected to their personal account dashboard. Simultaneously, the system updates the last login date and time and records the login attempt in the system logs. Once logged in, the client can access various services and features, such as searching for flights, booking tickets, managing account settings, or viewing past transactions.

Alternative Flow 1:Forgotten Password

On the login page, clients can click the "Forgot Password?" link if they are unable to recall their password. They are asked to enter their registered email address. Then, the system will dispatch a secure password reset link to the email address given. The client selects the link and is redirected to another secured page where they can set a new password. Once the client has reset their password successfully, they can log in normally.

Error Flow

If a client inputs an incorrect email or password, a message will display indicating, "Invalid email or password. Please try again." If you have five consecutive failed attempts, your account will be temporarily locked for security reasons. Later, you will be able to receive an e-mail that will indicate further steps to take in order to get into your account.

Furthermore, during the login process, the system will cease its operation and request that you connect with a secured network in case it identifies an insecure connection during the process.

System State on Completion

After the login validation, the system logs all the activities of the user; the time stamp of the session and the IP address among them. In the event of failure, it records the number of invalid attempts and the security steps that were taken (for instance account freezing or alarm messages sent). Also, to maintain security for the users' data, the system remains in a locked state, and the user is notified that the system is unlocked.

2.4 Effort/Time estimation calculation

Lead: [Roa]; contributors: [Mohammad: reviewing], [Yara: reviewing],[Dana: discussion]

UR	Estimated no of developer	Estimated effort	Total effort(for one developer)
UR1	3	2pw	6pw
UR2	2	2pw	4pw
UR3	1	1pw	1pw
UR4	3	3pw	9pw
UR5	1	2pw	2pw
UR6	2	1pw	2pw
UR7	2	1pw	2pw
UR8	1	1pw	1pw
Total effort/avg	15/8=1.875	13pw	27pw
Schedule time 30%		13*1.30=17w(min time to complete)	27*1.30=35w(max time to complete)
cost		Avg salary/w =330\$	330*35=11550\$
Profit margin (Min =12%, Max 30%)		Min cost	12,936\$
		Max cost	15,015\$

We have 5 employees and all of them can work in parallel.

1. Base Parameters

Total Effort = 27 person-weeks (pw)
 Number of Employees = 5 people
 Weekly Salary per Person = \$330

2. Time Calculation

Basic Time:

$$\begin{aligned}\text{Basic Time} &= \frac{\text{Total Effort}}{\text{Number of Employees}} \\ &= \frac{27 \text{ pw}}{5 \text{ people}} \\ &= 5.4 \text{ weeks}\end{aligned}$$

Schedule with Buffer:

$$\begin{aligned}\text{Actual Time} &= \text{Basic Time} \times 1.30 \\ &= 5.4 \text{ weeks} \times 1.30 \\ &= 7.02 \text{ weeks}\end{aligned}$$

3. Cost Calculation

Weekly Team Cost:

$$\begin{aligned}\text{Weekly Team Cost} &= \text{Weekly Salary} \times \text{Number of Employees} \\ &= \$330 \times 5 \\ &= \$1,650 \text{ per week}\end{aligned}$$

Total Base Cost:

$$\begin{aligned}\text{Total Base Cost} &= \text{Weekly Team Cost} \times \text{Actual Time} \\ &= \$1,650 \times 7.02 \\ &= \$11,583\end{aligned}$$

4. Profit Margin Calculation

Minimum Profit (12%):

$$\begin{aligned}\text{Min Price} &= \text{Total Base Cost} \times 1.12 \\ &= \$11,583 \times 1.12 \\ &= \$12,973\end{aligned}$$

Maximum Profit (30%):

$$\begin{aligned}\text{Max Price} &= \text{Total Base Cost} \times 1.30 \\ &= \$11,583 \times 1.30 \\ &= \$15,058\end{aligned}$$

Conclusion

The project should be priced between **\$12,973** and **\$15,058**, assuming:

- Perfect task distribution
- No dependencies between tasks
- Equal skill levels across team
- Minimal coordination overhead

2.5 ACTORS analysis

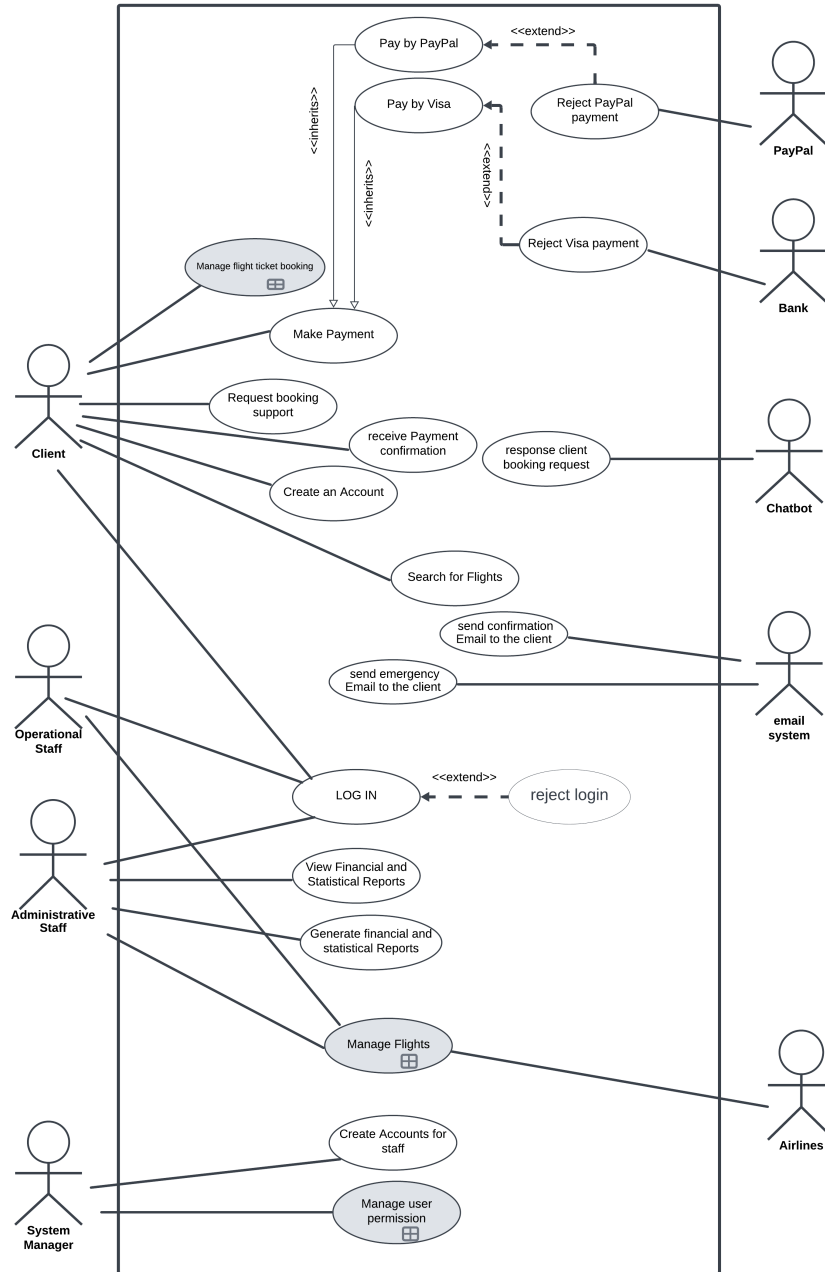
Team Lead: [Dana]. Contributing members: All members (Review, Edit).

Actor	Description
Client	Represents a user of the system who can create accounts, search for flights, manage bookings, make payments, and request support.
Operational Staff	Represents staff responsible for sending emergency emails and handling operational procedures.
Administrative Staff	Represents staff responsible for managing flights (e.g., adding flights, modifying schedules, updating ticket prices and available seats) and generating detailed financial/statistical reports, such as revenue analysis, flight occupancy rates, and sales trends.
System Manager	Represents the user responsible for creating staff accounts, managing permissions, and overseeing system configurations.
PayPal	Represents the external payment gateway through which clients can make payments using PayPal.
Bank	Represents the external payment gateway that handles Visa payments for flight bookings.
Chatbot	Represents an automated assistant that interacts with clients to provide basic support or respond to inquiries.
Email System	Represents the external system responsible for sending email notifications (e.g., booking confirmations, emergency emails).
Airlines	Represents the external entity that provides flight data, schedules, and availability, which administrative staff can update.

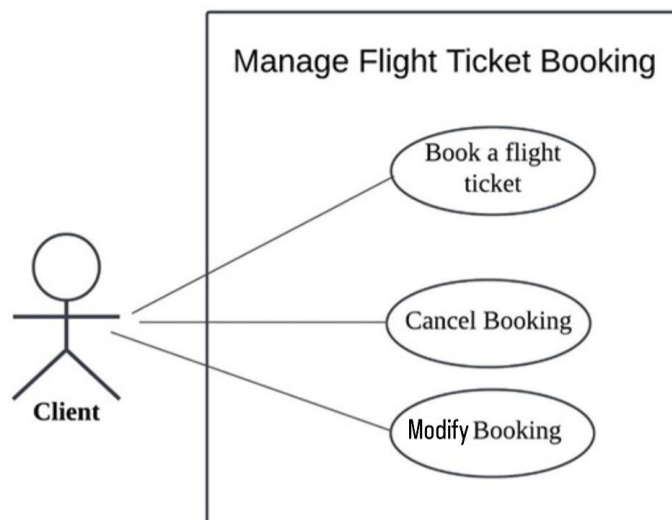
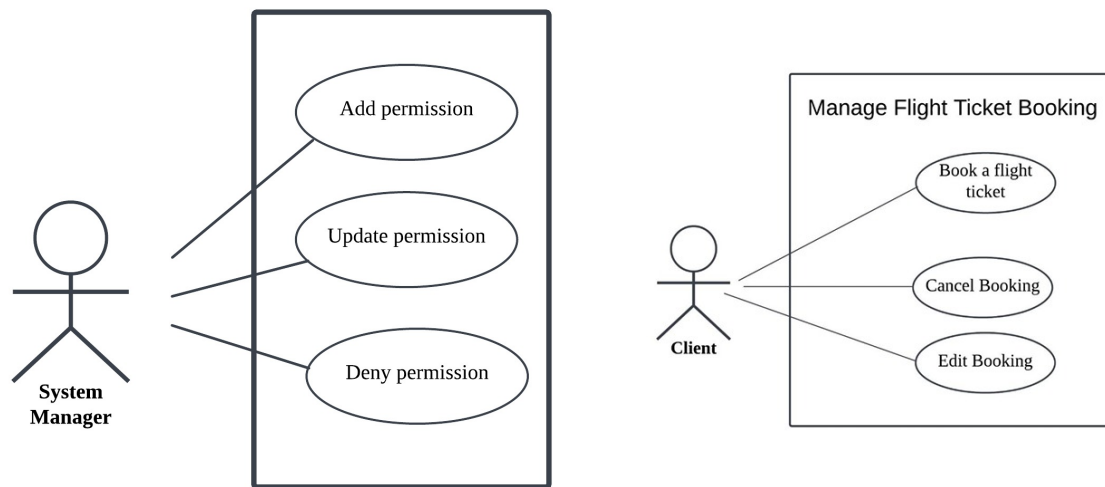
2.6 USE-CASE diagram

Lead: [Mohammad];Contributing members: All members (Review, Edit).

Flight Booking System



Manage user permission



2.7 Description of key USE-CASES

Mohammad's Use Case Description: Make Payment

System Name	Flight Booking System
Use Case Title	Make Payment
Description	A client may make a payment for their flight booking through the system. The process includes validation of payment details and transaction processing as mandatory steps, with an option to cancel the payment. After a successful payment, a confirmation is received.
Actors	<ul style="list-style-type: none">• Primary Actor: Client• Secondary Actors: Chatbot , Payment Gateway (PayPal, Bank)
Data	<ul style="list-style-type: none">• Payment amount• Payment method details• Booking reference• Transaction status• Client information
Stimulus/Trigger	Client selects "Make Payment" after booking a flight.
Pre-Conditions	<ul style="list-style-type: none">• Client has successfully logged into the system.• Client has a valid flight booking.• System is in a valid state to process payments.

Workflow OR Sequence/Flow Events	<ol style="list-style-type: none"> 1. Client initiates payment for flight booking. 2. System displays available payment methods (Visa, PayPal). 3. Client selects a payment method. 4. System prompts the user to enter required payment details: <ul style="list-style-type: none"> • For card payments: Client enters the card number, expiry date, and CVV code. • For PayPal: Client is redirected to PayPal's authentication page. 5. System asks if the client wants to save their payment information for future transactions. 6. If the user opts to save their payment information, the system encrypts all payment data using RSA algorithm. 7. System securely processes the transaction and validates the payment details. 8. Upon successful payment: <ul style="list-style-type: none"> • A detailed transaction record is generated. • A confirmation notification is sent to the user. • Payment history is updated.
---	---

Alternative Flows	<p>Alternative Flow 1: Unsupported or Invalid Payment Method</p> <ul style="list-style-type: none"> • If the selected payment method is unsupported or invalid (e.g., unsupported card type, invalid PayPal account), the system displays an error message. • The user is prompted to select a different payment method. • The system returns the user to the payment selection screen. <p>Successful Output? - Yes.</p> <p>Alternative Flow 2: User Chooses Not to Save Card Information</p> <ul style="list-style-type: none"> • The system processes the payment without storing card details. • The transaction is successfully completed. <p>Successful Output? - Yes.</p>
Error Flow	<ul style="list-style-type: none"> • If payment processing fails due to reasons such as: <ul style="list-style-type: none"> – Invalid card details. – Network failure. – System timeout. – Bank authorization failure. • The system halts the transaction within 10 seconds and ensures that no funds are deducted. • An error message is displayed, prompting the user to retry or select a different payment method. <p>Successful Output? - No.</p>
Post-Conditions / Response	<ul style="list-style-type: none"> • Payment confirmation received by client. • Booking status updated. • Transaction record stored in the system. • Payment history updated.

Other Activities	<ul style="list-style-type: none"> • Transaction logs are maintained. • Payment details are encrypted using RSA. • System monitors processing time to ensure compliance with a 10-second response time limit.
System State on Completion	<ul style="list-style-type: none"> • User remains in an authenticated session. • A payment record is created in the database containing: <ul style="list-style-type: none"> – Transaction ID. – Payment amount. – Payment method used. – Transaction timestamp. – Encryption status. – User preferences for card storage. • A confirmation notification is sent to the user's registered contact information.

Yara's Use Case Description: Manage flight ticket booking

System Name	Flight Ticket Booking System
Use Case Title	Manage flight ticket booking
Description	Client has the option to book a flight ticket using this system, which allows the client to look up specific flights, select and book a seat. Once the client confirms their booking, the seat status will automatically change to <reserved>.
Actors	Client, and Email system.
Data	Flights information.
Stimulus/Trigger	Client enters flight information then chooses a flight.
Pre-Conditions	The client has an account and logs into the system.

Flow of Events	<ol style="list-style-type: none"> 1. The client enters the flight main information. 2. The system displays all available matching flights. 3. The client chooses a flight. 4. The system asks the user to enter their personal information. 5. The client enters the personal information. 6. The system moves to the special requirements part. 7. The client enters the special requirements. 8. The system displays the seats by customary aircraft layout. 9. The client chooses their seats. 10. If the selected seat is no longer available, the system displays updated seat availability and asks the client to select a new seat. 11. The client chooses a new seat. 12. The system displays the baggage options. 13. The client enters the baggage options. 14. Confirm the booking. 15. Receive a confirmation email. 16. If the client needs to modify their booking before the last 48 hours, the system allows changes. 17. The system updates the booking summary accordingly. 18. If the client needs to modify their booking within the last 48 hours, the system displays the restriction message "updates do not allow". 19. If the internet connection is lost during the booking process, all entered data will be lost. So, the user will restart the booking process when the connection comes back.
-----------------------	---

Post Conditions/Response	<ol style="list-style-type: none"> 1. Number of flight tickets updated. 2. The client receives a confirmation email. 3. Booking status updated. 4. Seats aircraft layout updated.
Comments	Client confirms their booking by making a payment.

Roa's Use Case: Create an Account

System Name	Online Flight Booking System
Use Case Title	Create an Account
Description	A new client registers by entering personal data, including a valid email address, password, and other required information. Before proceeding, the system ensures the client is not already logged in. Once verification is successful, the system validates the provided details to ensure they meet age, security, and uniqueness standards before activating the account.
Actors	Client
Data	Full Name, Email, Password, Date of Birth, Phone Number
Stimulus/Trigger	The client clicks the "Create an Account" button on the flight system's interface and submits the required registration information.
Pre-Conditions	<ol style="list-style-type: none"> 1. The system is online, fully operational, and ready to access its database and email verification service to handle client registration requests. 2. The client is not logged into any existing account, ensuring the registration process can be initiated without conflicts. 3. The system has an active and stable connection to its database, enabling real-time validation of email addresses and duplicate checks. 4. The age verification mechanism is functional and configured to validate the client's date of birth against the minimum age requirement.

Workflow OR Sequence/Flow Events	Se- of
	<ol style="list-style-type: none"> 1. The client clicks "Create an Account," and the system checks if the provided email is registered. <ul style="list-style-type: none"> • If registered, the client is prompted to log in. Upon successful login, they are redirected to their dashboard; otherwise, an error message is shown. • If the email is not registered, the system displays a registration form prompting the client to enter their full name, email, password, phone number, and date of birth. 2. The system validates the entered information as follows: <ol style="list-style-type: none"> (a) Validate Email: <ul style="list-style-type: none"> • The system checks if the email format is valid and ensures it is not already registered. • Alternative Flow: If the email is registered, the system displays: "This email is already registered. Please log in or use a different email." Alternatively, the client can register using a phone number with OTP verification. (b) Validate Password: <ul style="list-style-type: none"> • The system checks the password strength and ensures it matches the confirmation. • Alternative Flow: If the password is weak or mismatched, the system displays feedback to correct it and allows the client to proceed once fixed. (c) Validate Date of Birth: <ul style="list-style-type: none"> • The system verifies if the client meets the minimum age requirement (18 years). • Error Flow: If the client is under 18, the system halts the process and displays: "You must be at least 18 years old to create an account." 3. If all validations are successful, a verification email with an activation link is sent, and a confirmation message is displayed. 4. Verification Process: If the client clicks the verification link, the account is activated successfully. If the client does not click the link, the account creation process is canceled. 5. The process concludes with either a successful account creation or cancellation.

Post-Conditions/Response	<ol style="list-style-type: none"> 1. The client's account is transitioned to an "active" state after successful verification, enabling full access to all system services. 2. The system securely stores all relevant client data, including the IP address and account creation timestamp, in compliance with privacy and security protocols. 3. If verification is not completed, the system cancels the account creation process, ensuring no incomplete or duplicate accounts remain in the system.
Comments	The system must ensure data security, validate inputs, and prevent duplicate or incomplete account creation.

Dana's Description: Client Log In

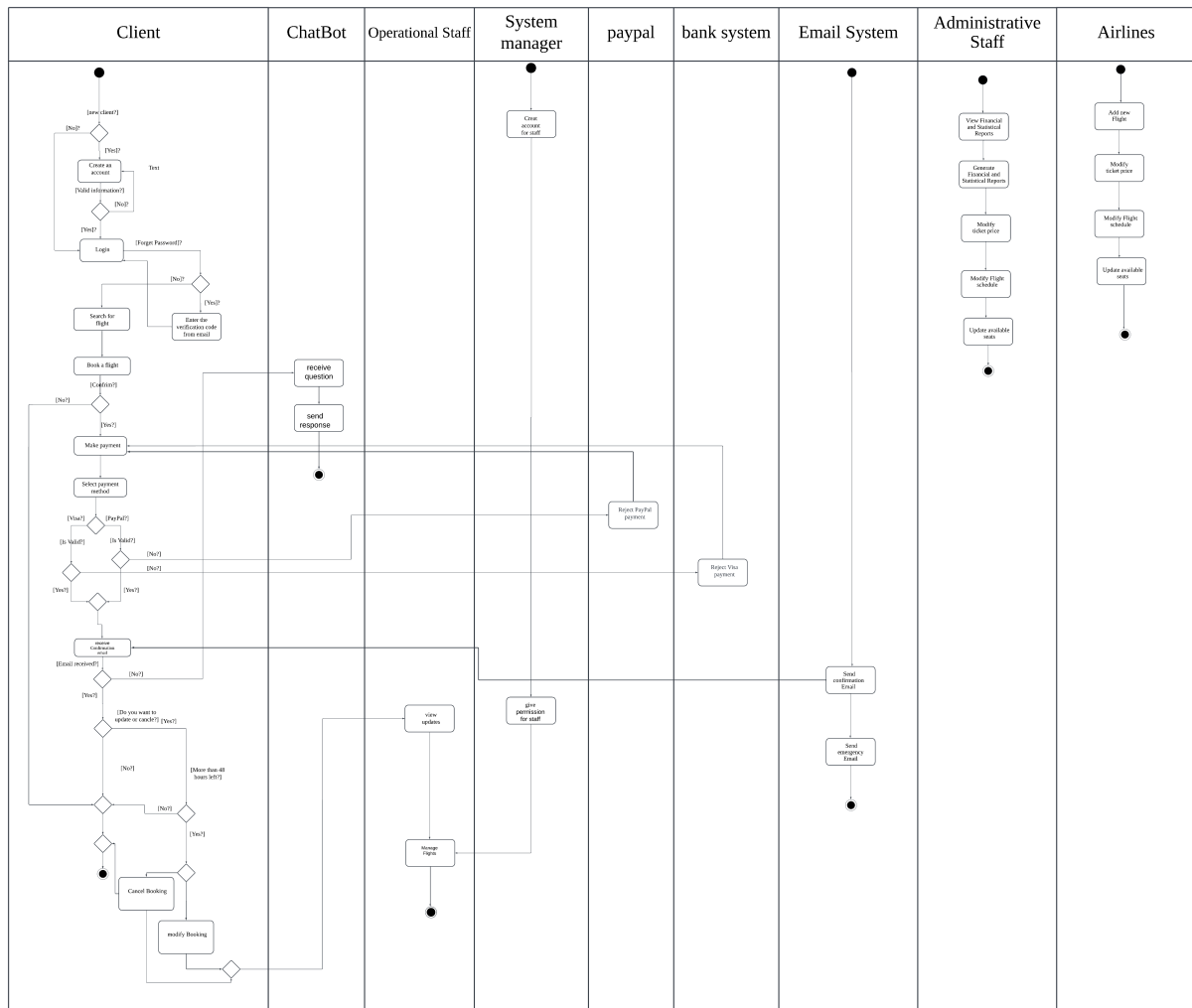
Use Case Title	Log In
Description	In order to log into the site, the clients must provide the username or registered email address and the appropriate password. The system ensures that the entered password matches the one stored during account creation. If the password is forgotten, the system allows the client to reset it via an email verification link. Clients must enter valid credentials to access booking, search, and management features.
Actors	Client
Data	Username, Email Address, and Password
Stimulus/Trigger	The client enters login credentials (username and password) and sends the login request by pressing the "Login" button.
Pre-conditions	<ol style="list-style-type: none"> 1. The client must have an active account in the system. 2. The system must be functional and accessible, with the client on the login page where the login interface is displayed, and the client is aware of their registered email address and password.

Workflow OR Sequence/Flow Events	Se- of
	<ol style="list-style-type: none"> 1. The client accesses the login page of the online flight booking system. 2. The client enters their username and password. 3. The client submits the login request by pressing the "Log In" button. 4. The system checks whether the provided login details match a registered account. 5. If the credentials are correct, the system grants access and redirects the client to the homepage. 6. If the credentials are incorrect, the system returns an error message prompting the client to try again or register. 7. If the client enters incorrect login credentials more than five consecutive times, the system blocks the account for security reasons and sends an email with the account retrieval process. 8. If the client has forgotten their password, the system sends a password reset link to the registered email. 9. The system logs all login attempts for security monitoring.

Post-conditions/Response	<ul style="list-style-type: none"> • Successful Login: <ol style="list-style-type: none"> 1. The system updates the last login time and records the successful login information. 2. The system establishes a secure session, allowing the client to access their account and use available services. • Unsuccessful Login: <ol style="list-style-type: none"> 1. The system records the failed attempt and returns an error message. 2. If the client's account is locked due to multiple failed attempts, the system sends an email with account recovery instructions. 3. If an unsecure network is detected, the system blocks login attempts and warns the client to switch to a secure connection. 4. The system remains on the login page until valid authentication is completed or further actions (e.g., password reset) are initiated.
Comments	<ol style="list-style-type: none"> 1. The system should support future additions such as Two-Factor Authentication for enhanced security. 2. Login credentials should be encrypted during transmission and storage. 3. The system should work properly across all devices and browsers. 4. CAPTCHA should be used after multiple incorrect login attempts to distinguish human users from bots. 5. Error messages should not specify whether the email or password is incorrect to prevent data leakage. 6. Clients should be able to see their last login time, device, and location to detect unauthorized access.

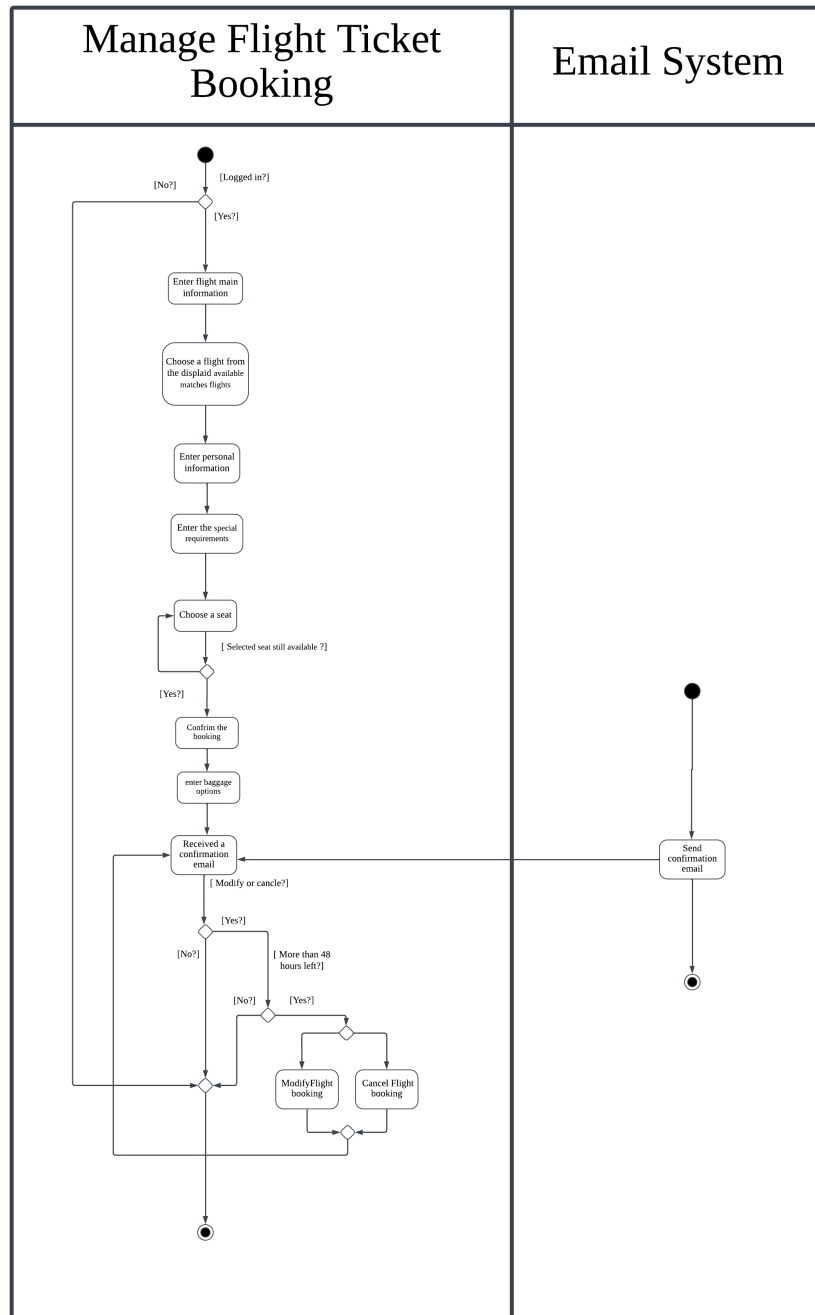
2.8 ACTIVITY diagram

Team Lead: [Yara]. Contributing members: All members (Review, Edit).

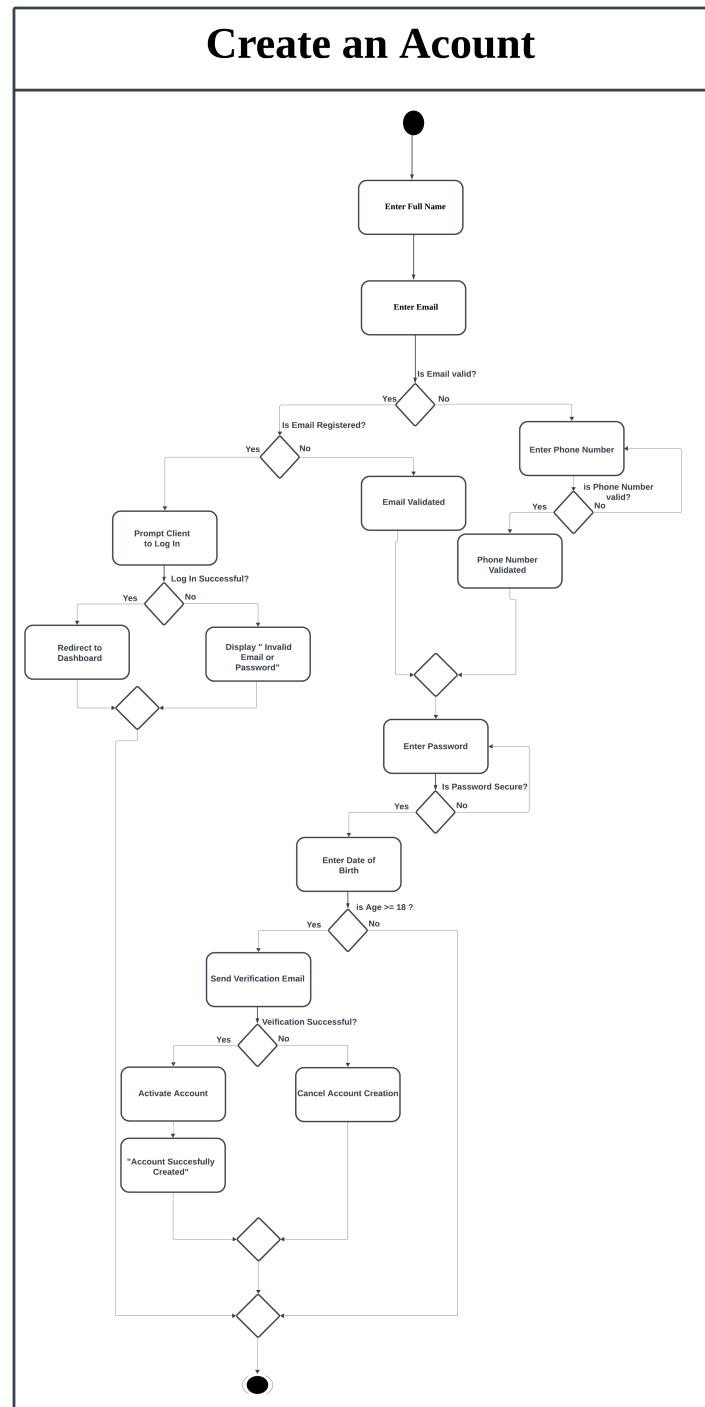


2.9 Instance Activity diagrams

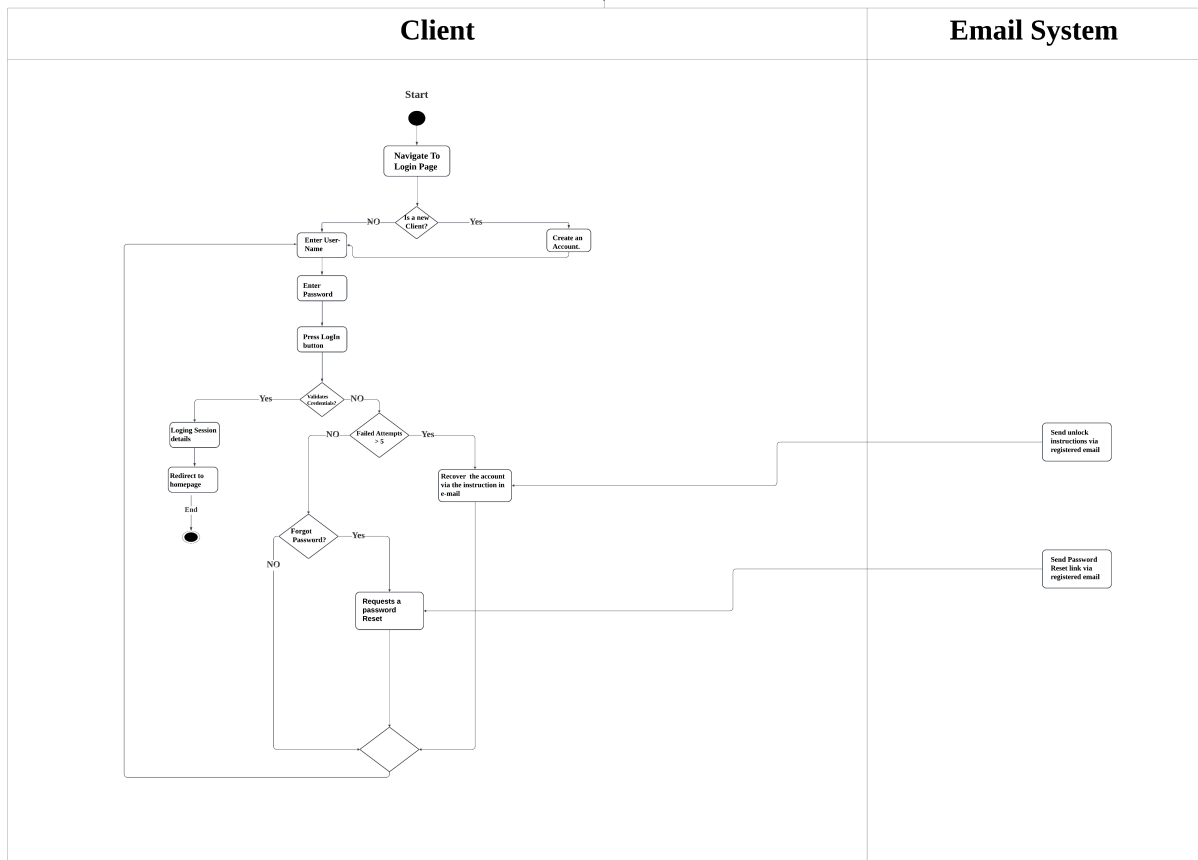
Yara's diagram: Manage flight ticket booking



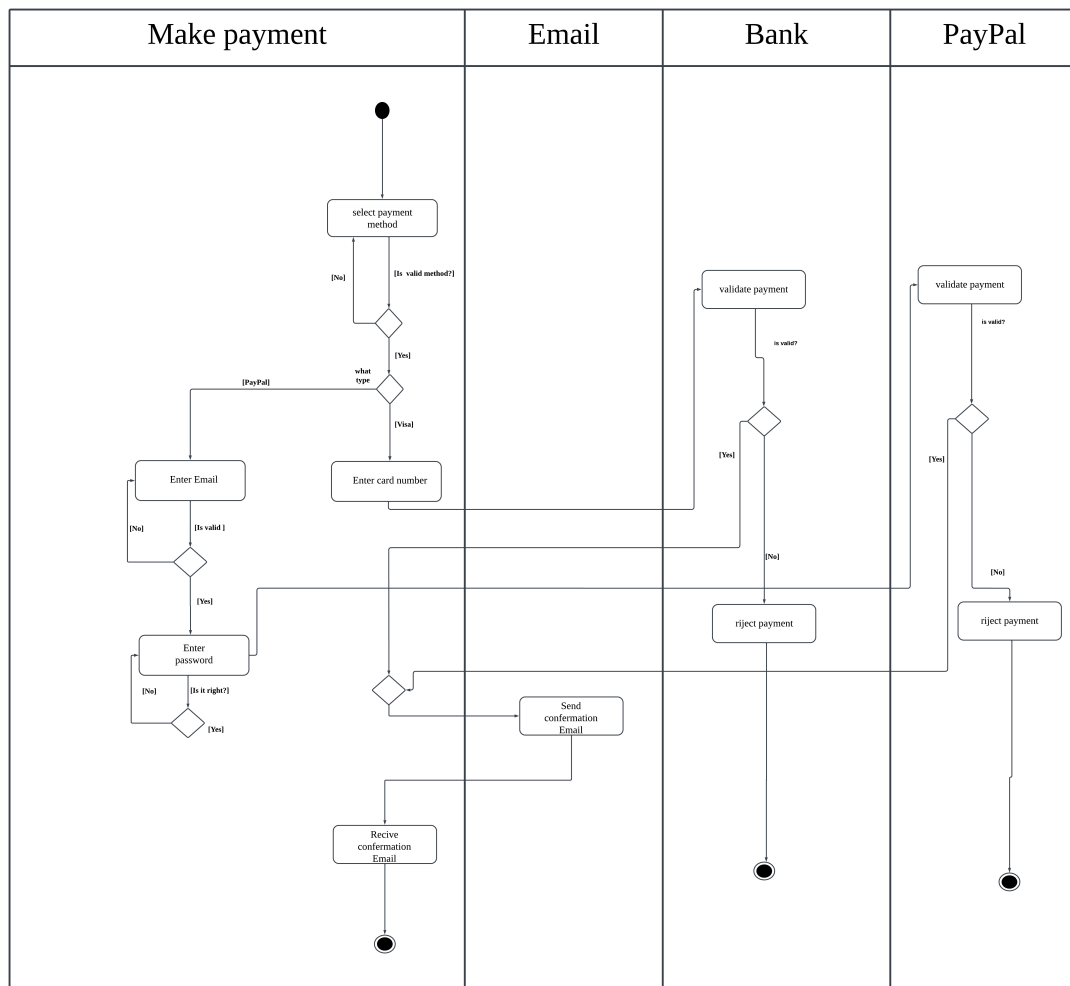
Roa's Diagram: Create an account



Dana's diagram: Client Log In



Mohammad's Diagram: Make payment

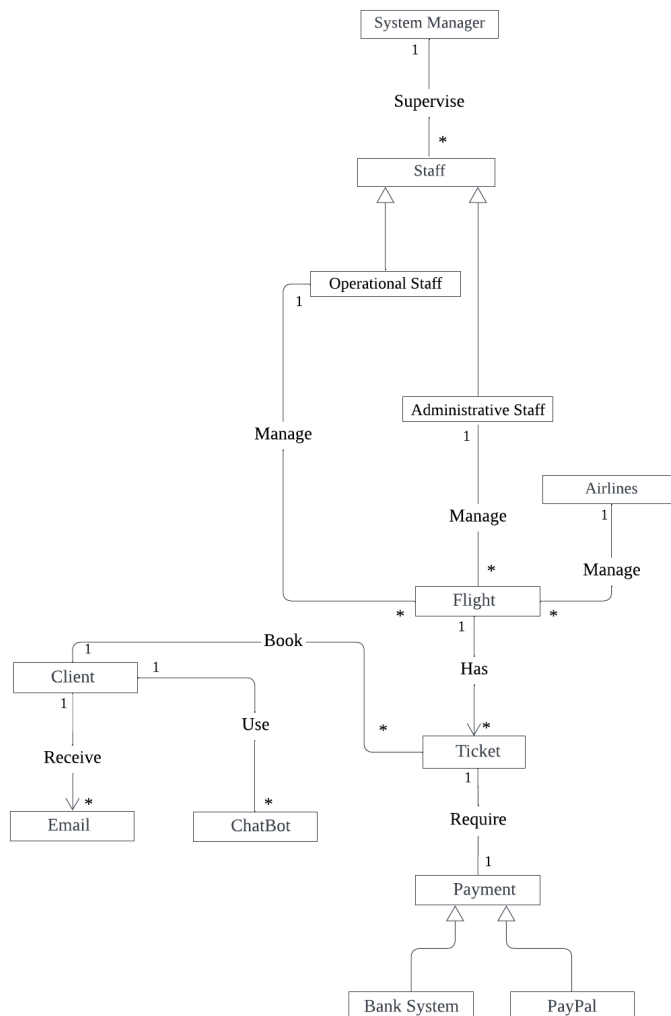


3 Chapter 3: System Analysis and Modelling

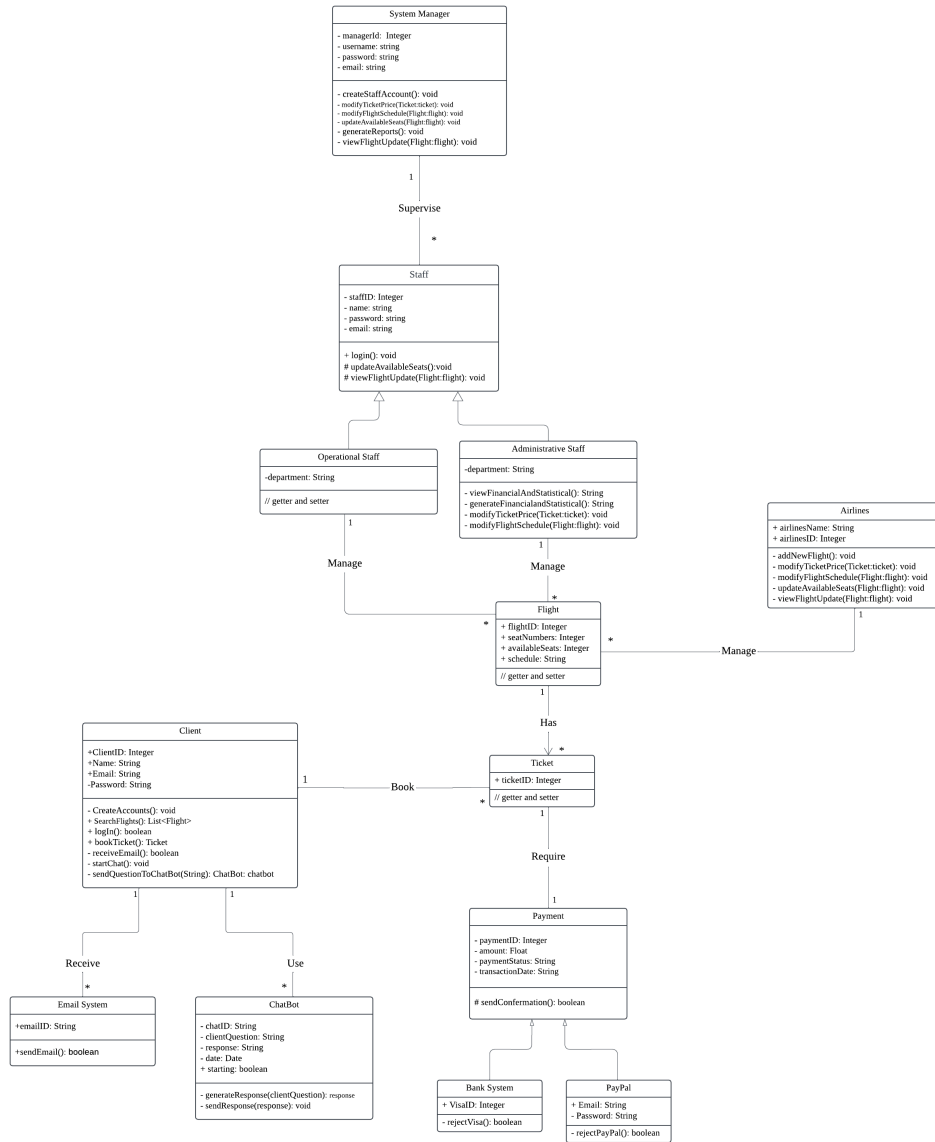
3.1 System CLASS Diagrams

Team Lead: [Mohammad]. Contributing members: All members (Review, Edit).

3.1.1 Analysis class model

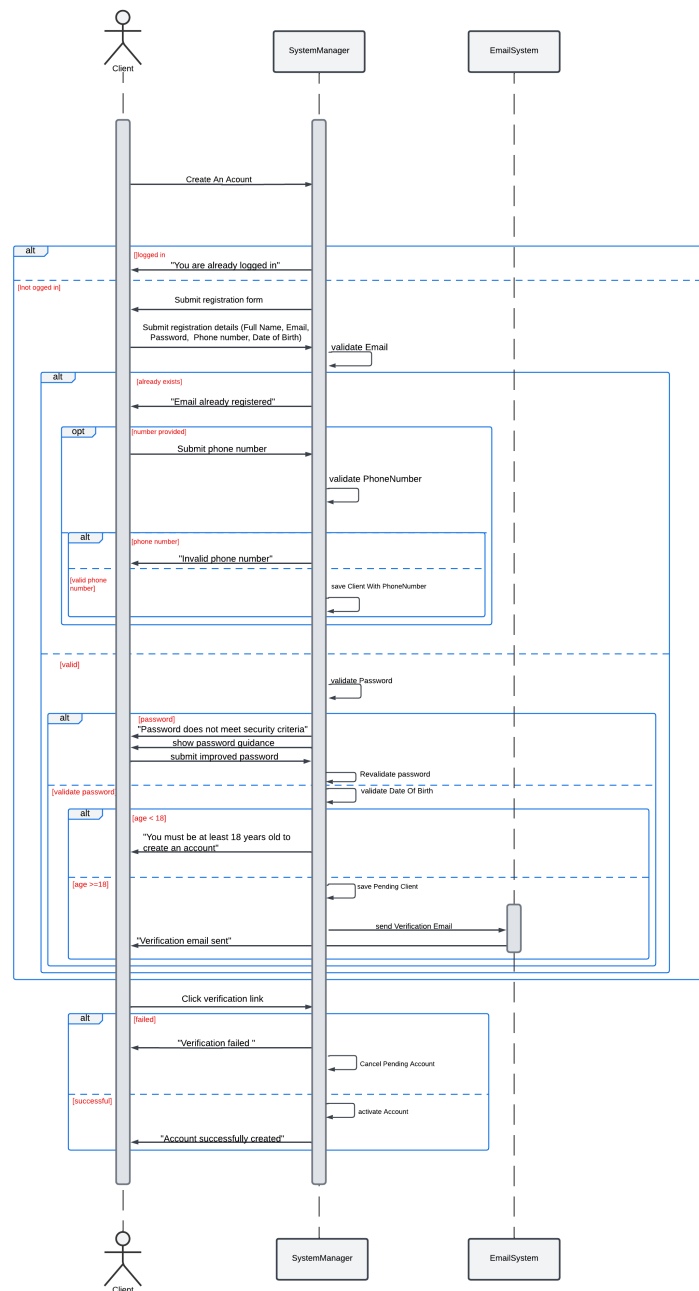


3.1.2 detailed class model

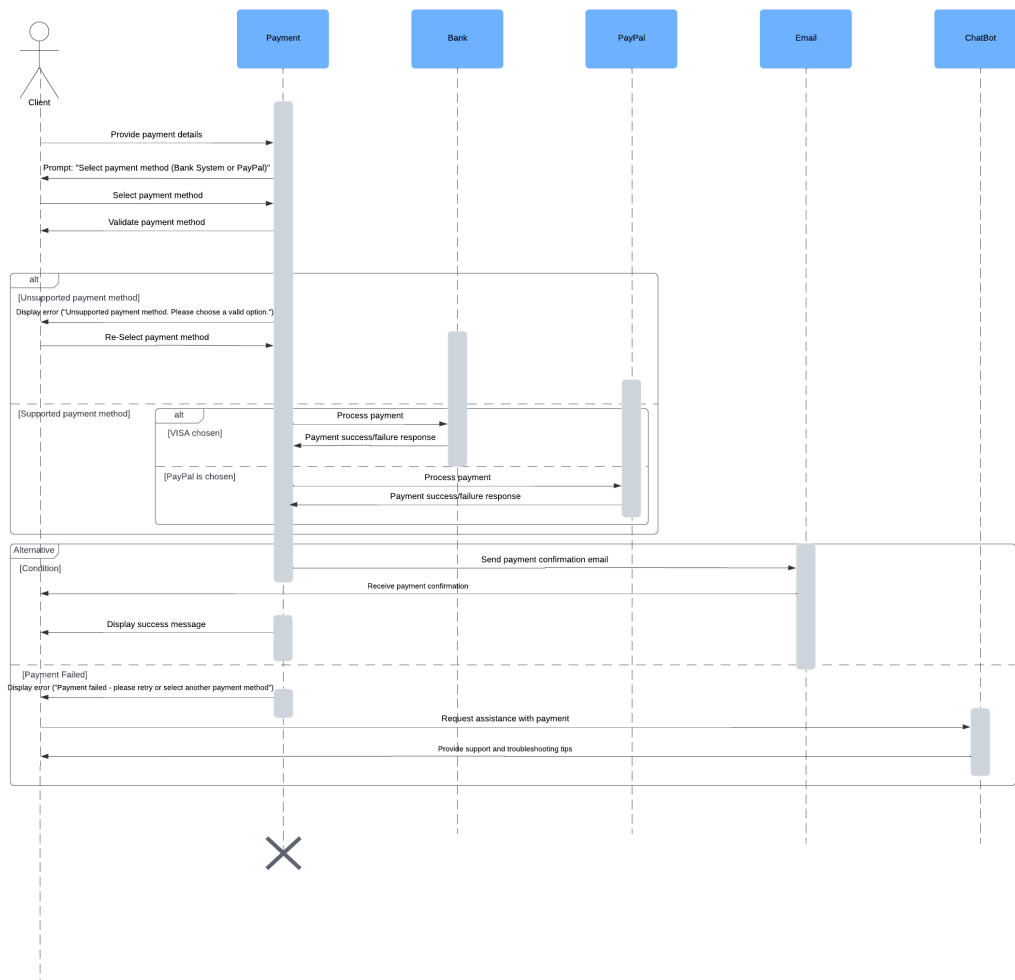


3.2 SEQUENCE Diagrams

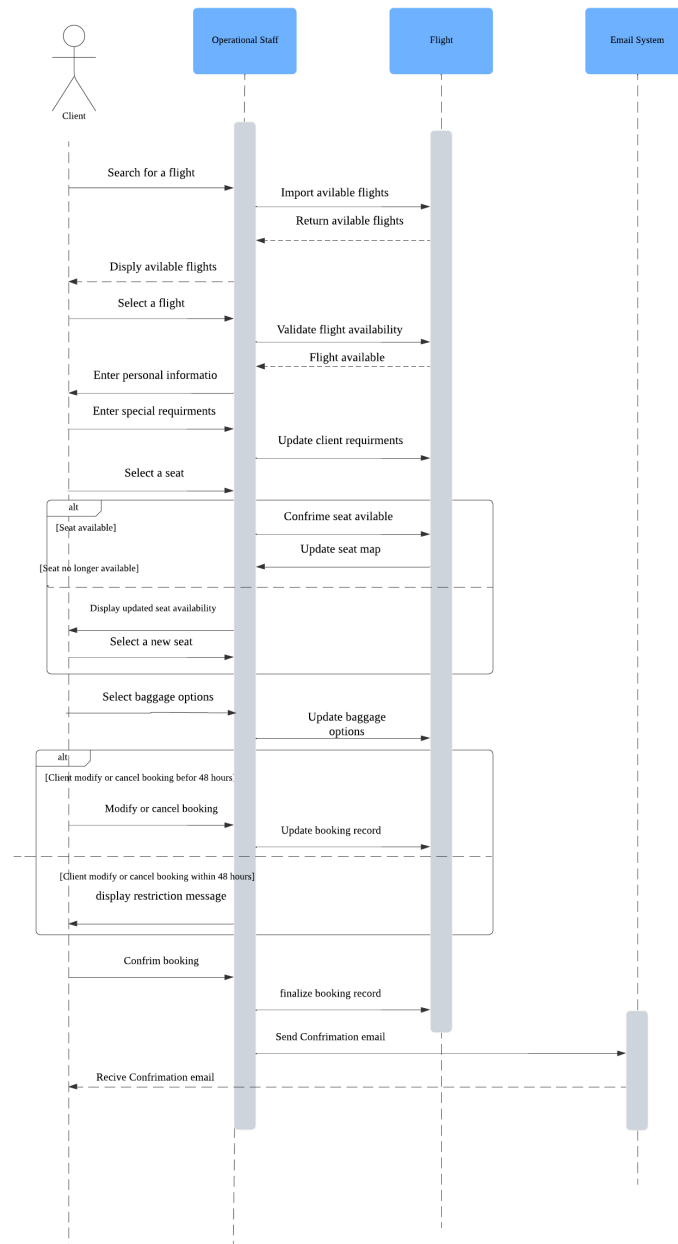
Roa's Diagram: Create an account



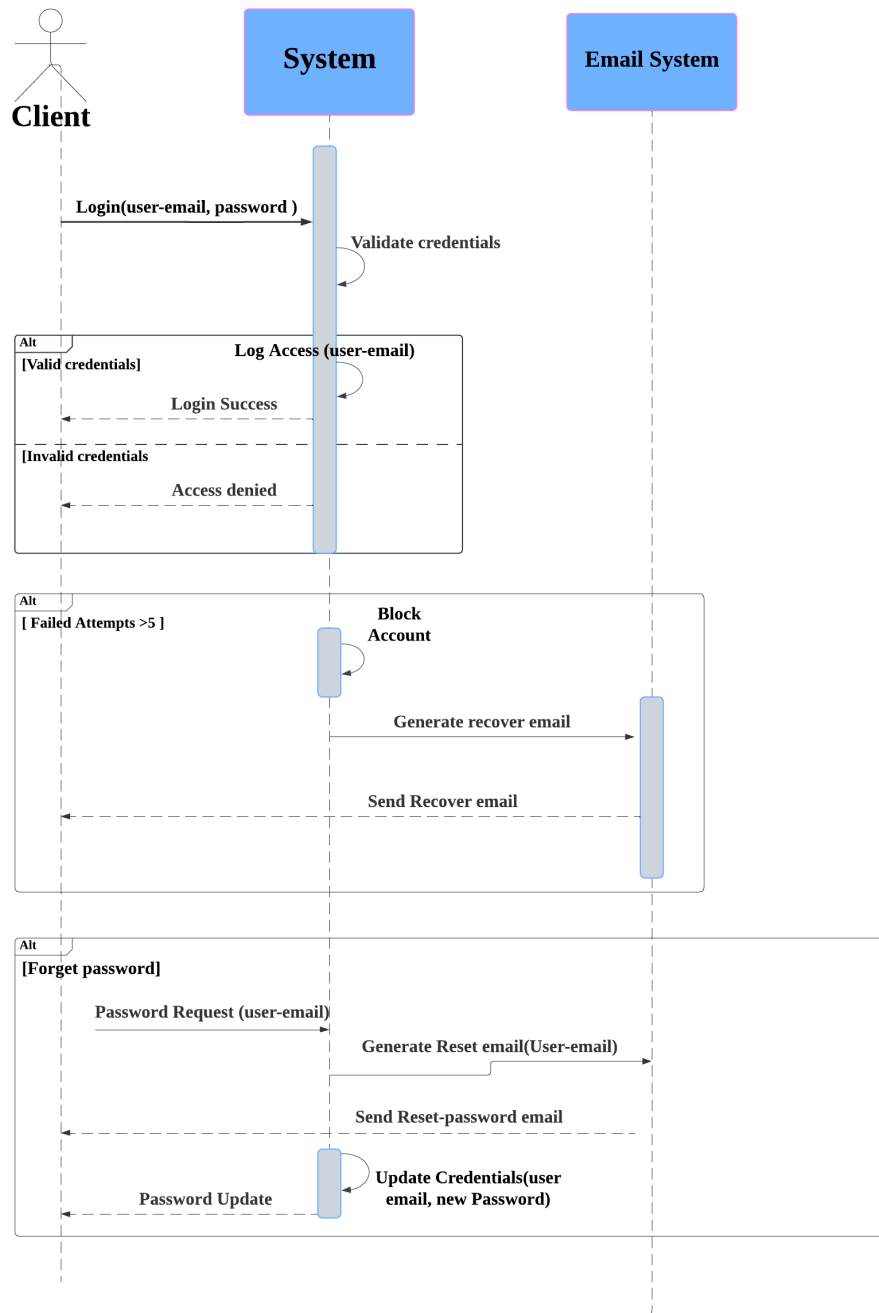
Mohammad's Diagram: Make payment



Yara's diagram: Manage flight ticket booking



Dana's diagram: Client Log In



4 Chapter 4: System Design and Modelling

4.1 Description of chosen Design Goals

Lead: [Roa], Contributing members: All members (Review, Edit)

I. Low Coupling

Objective: Reduce interdependencies between system components to improve flexibility, scalability, and maintainability.

Implementation: Design standard classes such as `Payment`, `Ticket`, and `Flight` to communicate through well-defined interfaces instead of direct dependencies.

How we achieve this?

1. **Implement Event-Driven Architecture:** This design model allows system components to communicate by producing and reacting to events rather than calling each other directly. For example, when a `Payment` is completed, a "Payment Successful" email is sent to the client. Meanwhile, the `Booking` module listens to this event and updates the booking status accordingly.
2. **Apply the Principle of Interface Segregation:** Create smaller, specific interfaces instead of one large interface for each module.

II. High Cohesion

Objective: Create structured system components that focus on single, well-defined functionalities, reducing overall complexity.

Implementation: Components are separated based on distinct functions. For example, the `Flight` module handles reservations, while the `Payment` module processes transactions. This modular design improves maintainability, scalability, and simplifies system management.

How we achieve this?

1. **Identify Core Responsibilities:** Break the system into distinct modules, each responsible for a specific functionality.
2. **Encapsulation:** Hide internal details of each module from other modules and expose only necessary methods or attributes.
3. **Test Each Module Independently:** Write unit tests for individual modules to ensure they function correctly in isolation.

Specific System Design Goal

I. User-Friendly Interface

Objective: Provide a clear and accessible system interface, ensuring that staff and managers can perform their tasks effectively with minimal training (less than or equal to two days).

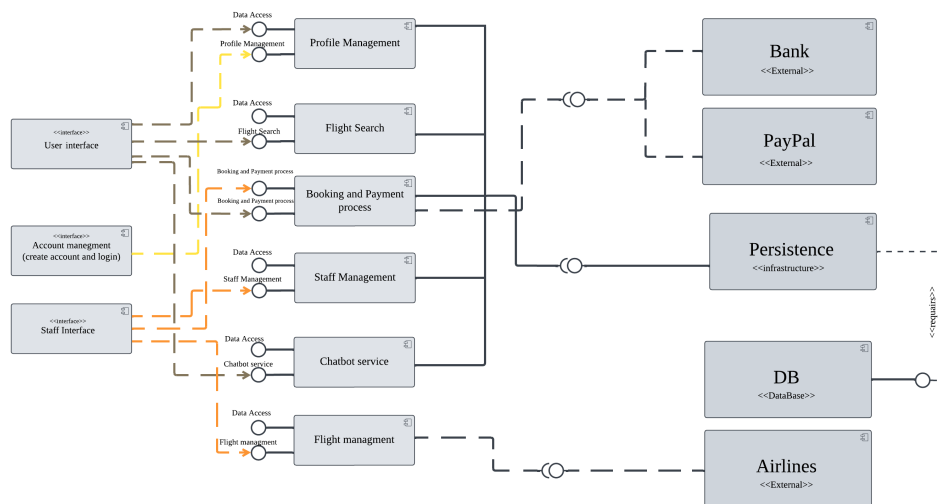
Implementation: Design a user-friendly interface based on UX research, implementing intuitive navigation and guided workflows. Client interfaces are simplified for usability, while staff dashboards are role-specific with built-in safeguards.

How we achieve this?

1. **Make UX Research:** Conduct user interviews with potential clients and staff to gather feedback on their needs, expectations, and challenges.
2. **Role-Specific Staff Dashboards:** The administrative dashboard manages flights, pricing, and reports, while the operational dashboard handles bookings and updates. Access permissions ensure that staff members can only use tools related to their roles.

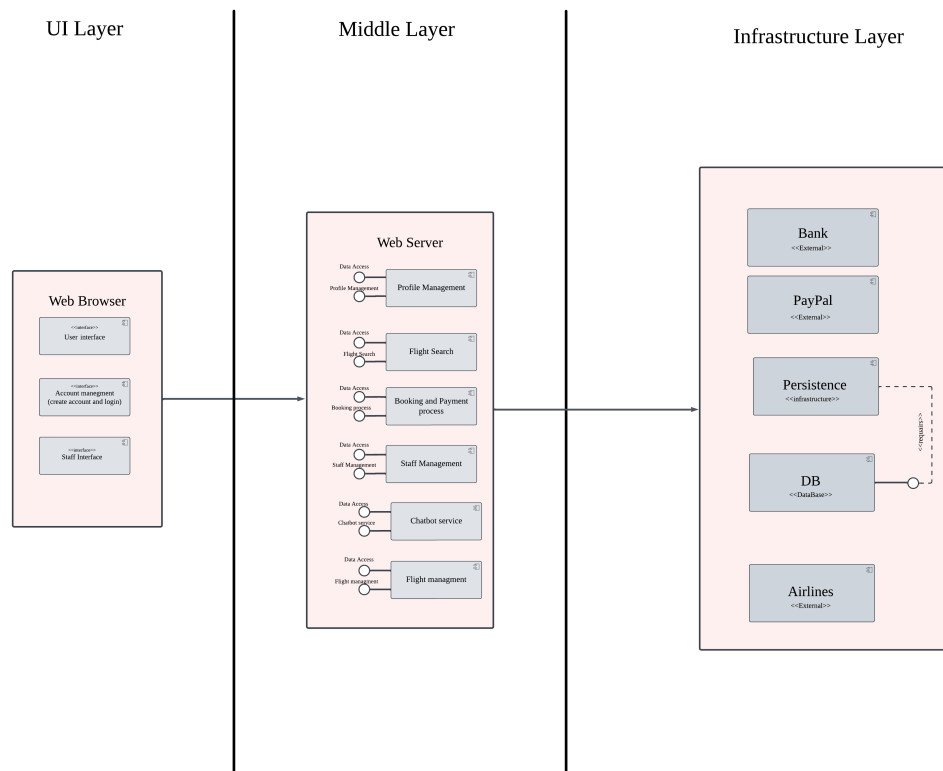
4.2 Component Diagram

Team Lead: [Yara], Contributing members: All members (Review, Edit)



4.3 Overall architecture diagram

Team Lead: [Mohammad], Contributing members: All members (Review, Edit). The Layered

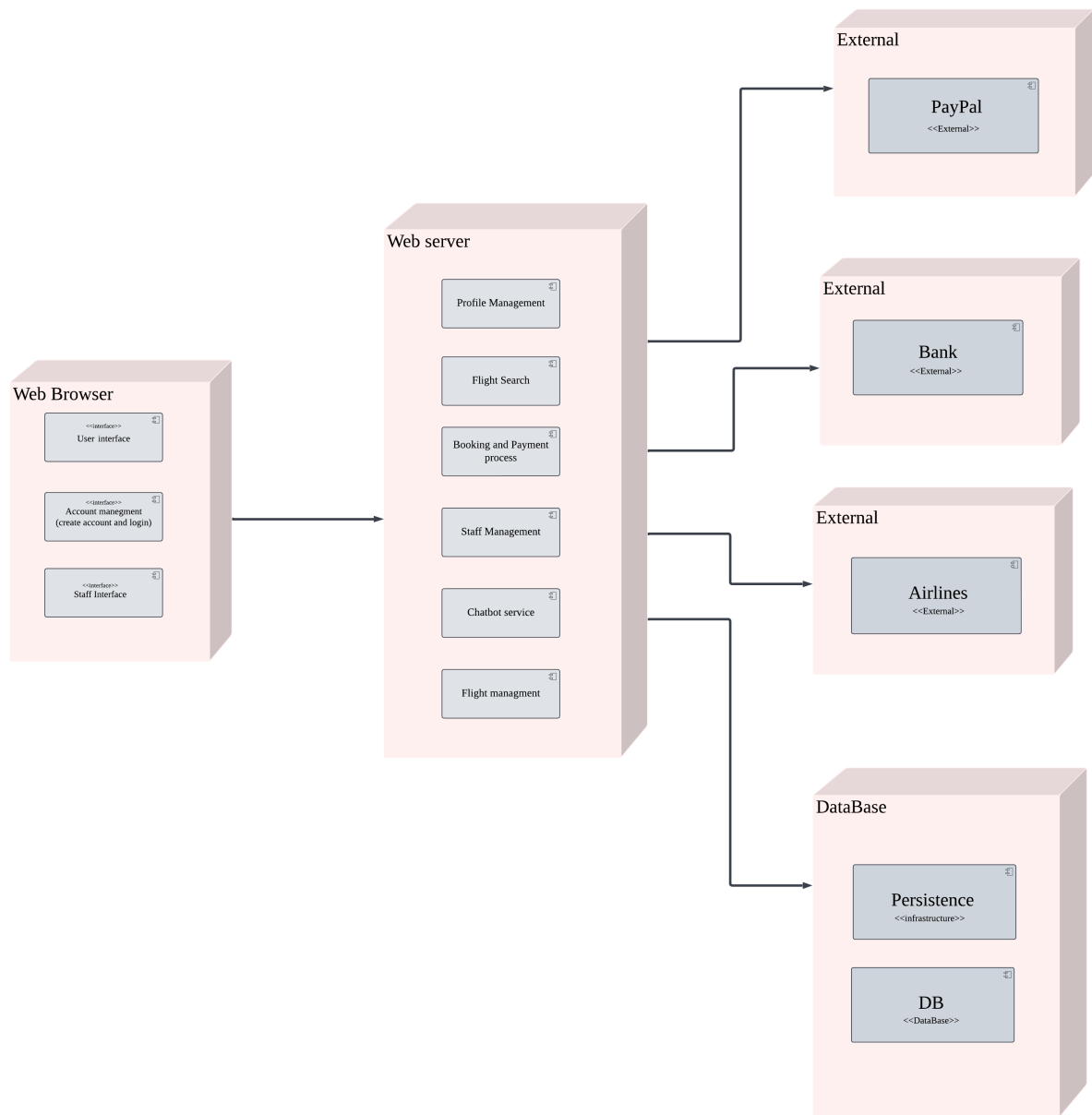


Architecture was chosen for this system because it effectively meets both design goals (low coupling, high cohesion, and user-friendly interface) and non-functional requirements (scalability, maintainability, and security). This architecture is structured into three layers: UI Layer, Middle Layer (Application/Web Server), and Infrastructure Layer. The UI Layer is responsible for handling user interactions through the web browser, ensuring a user-friendly interface with intuitive navigation and role-specific dashboards. The Middle Layer contains the core business logic, including profile management, flight search, booking and payment processing, staff management, chatbot services, and flight management, ensuring high cohesion by separating each function into distinct services. The Infrastructure Layer manages external services like banks, PayPal, airline systems, and the database, encapsulating data persistence and API integrations for enhanced security and maintainability.

This layered approach supports low coupling through an Event-Driven Architecture, where system components interact asynchronously. For example, once a payment is completed, an event triggers the booking module to update its status without direct dependencies. This design ensures that changes in one module do not affect others, making the system highly scalable and maintainable. Furthermore, the architecture enhances security by restricting direct data access and ensuring that the UI only communicates with the Middle Layer, reducing exposure to vulnerabilities. By isolating concerns, updates and feature additions (such as new payment providers) can be made without disrupting the system. This approach ultimately ensures that the system remains modular, efficient, and adaptable, making it the optimal choice for achieving the project's functional and non-functional requirements.

4.4 Deployment diagram

Team Lead: [Dana], Contributing members: All members (Review, Edit).



5 G2 team meetings

Date	Meeting Type	Start Time	Duration
02/11/2024	Online	12:05 PM	40 min
13/11/2024	Face-to-Face	1:00 PM	3 hours
17/11/2024	Online	7:48 PM	2 hours 30 min
11/12/2024	Online	7:00 PM	4 hours
13/12/2024	Online	5:30 PM	2 hours
17/12/2024	Online	7:30 PM	3 hours
20/12/2024	Online	5:00 PM	2 hours 30 min
21/12/2024	Online	8:40 PM	3 hours
25/12/2024	Face-to-Face	11:00 AM	2 hours
28/12/2024	Online	5:00 PM	3 hours 30 min
29/12/2024	Online	1:00 PM	2 hours 30 min
29/12/2024	Online	9:00 PM	1 hour
08/01/2025	Face-to-Face	3:00 PM	1 hour
12/01/2025	Online	7:00 PM	3 hours
15/01/2025	Online	7:15 PM	2 hours
16/01/2025	Face-to-Face	11:00 AM	1 hour

6 course feedback

The Software Engineering course was incredibly beneficial, providing us with valuable insights into how software projects are planned, developed, and executed in real-world work environments. Through this course, we gained a deeper understanding of project management, requirement gathering, system design, and implementation strategies, which are essential for professional software development.

We would like to express our gratitude to Professor Adel for his continuous support, guidance, and constructive feedback throughout the course. His valuable insights and encouragement played a key role in shaping our understanding of software engineering principles and improving the quality of our project. This course has not only enriched our technical and analytical skills but has also prepared us for the challenges of real-world software development.