



# SOFTWARE ENGINEERING

## Group 2



# Group team

Group number 2

Project manager

- Yara Darabumukho
- 1211269



Technical architect

- Mohammad Manasrah
- 1211407



Programmer

- Dana Assad
- 1211452



Requirement Engineer

- Roa Daraghmeh
- 1221349



# ABOUT THE BUSINESS

---

It is an online platform for flight reservation that is easily accessible using a website. It has created a way whereby customers can easily follow their preferred airlines and search for flights that suit their specifications in terms of destination, dates, and so on. By providing links to the different airlines, we make it easy for users to compare the flight price, times with layovers, and the flight duration.



## Create an Account Scenario



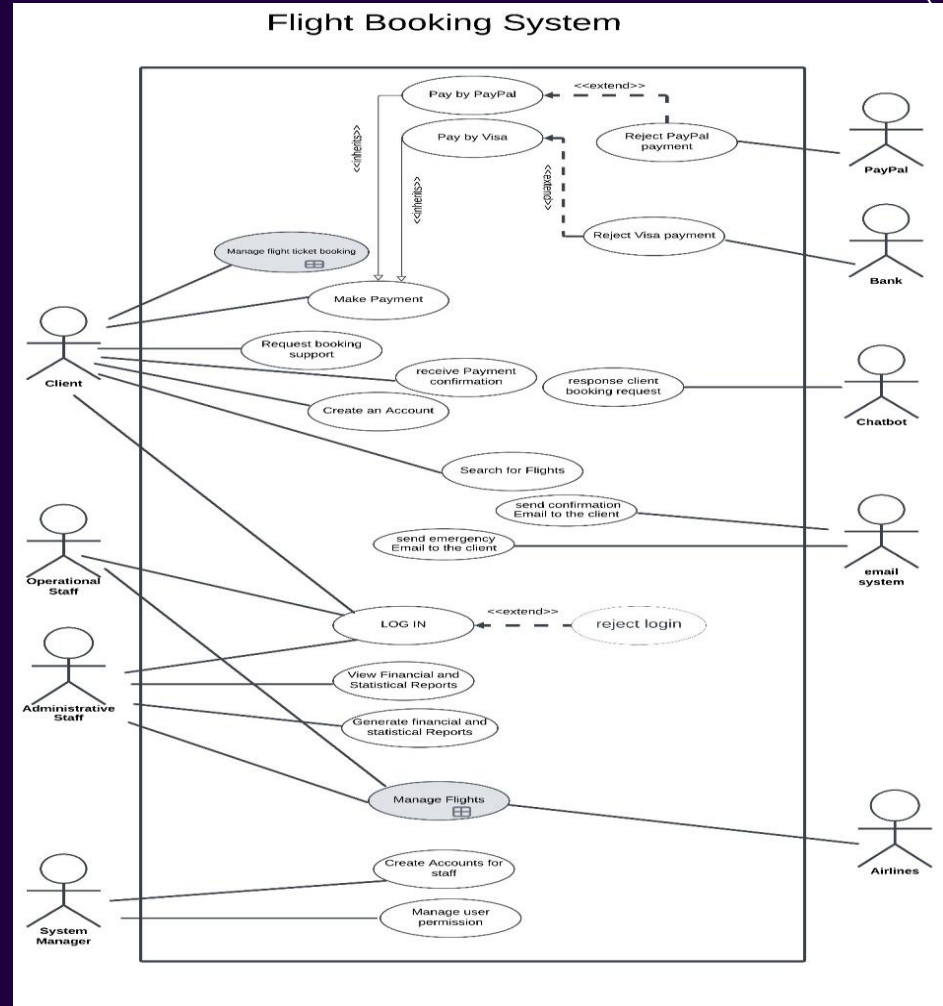
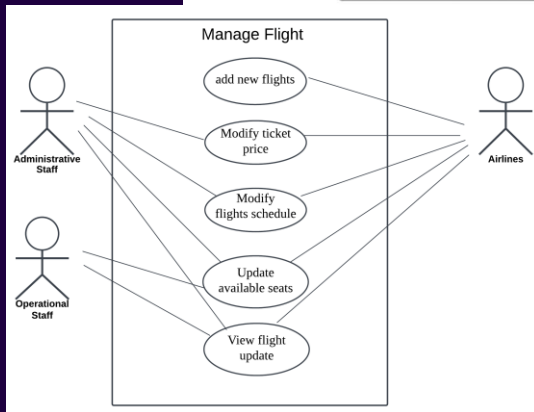
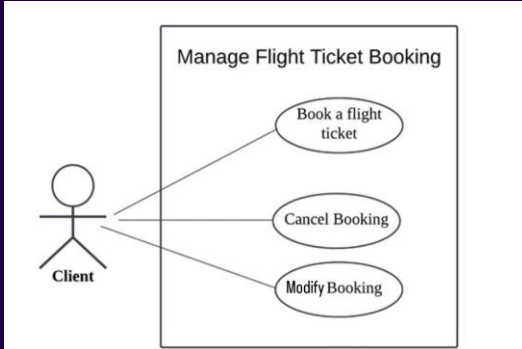
+  
+

The account creation process enables clients to register by providing their email, full name, password, date of birth, and phone number. The system validates the email format and ensures it is unique, displaying a message if it is already registered. Passwords are checked for strength and confirmation, with real-time feedback provided if corrections are needed. The date of birth is verified to ensure the client is at least 18 years old; otherwise, the process halts with an error message. Upon successful validation, the system saves the client's information, logs the IP address and account creation timestamp for security, and sends a verification email. Clicking the link activates the account, granting full access, while unverified accounts are canceled to prevent duplicates.



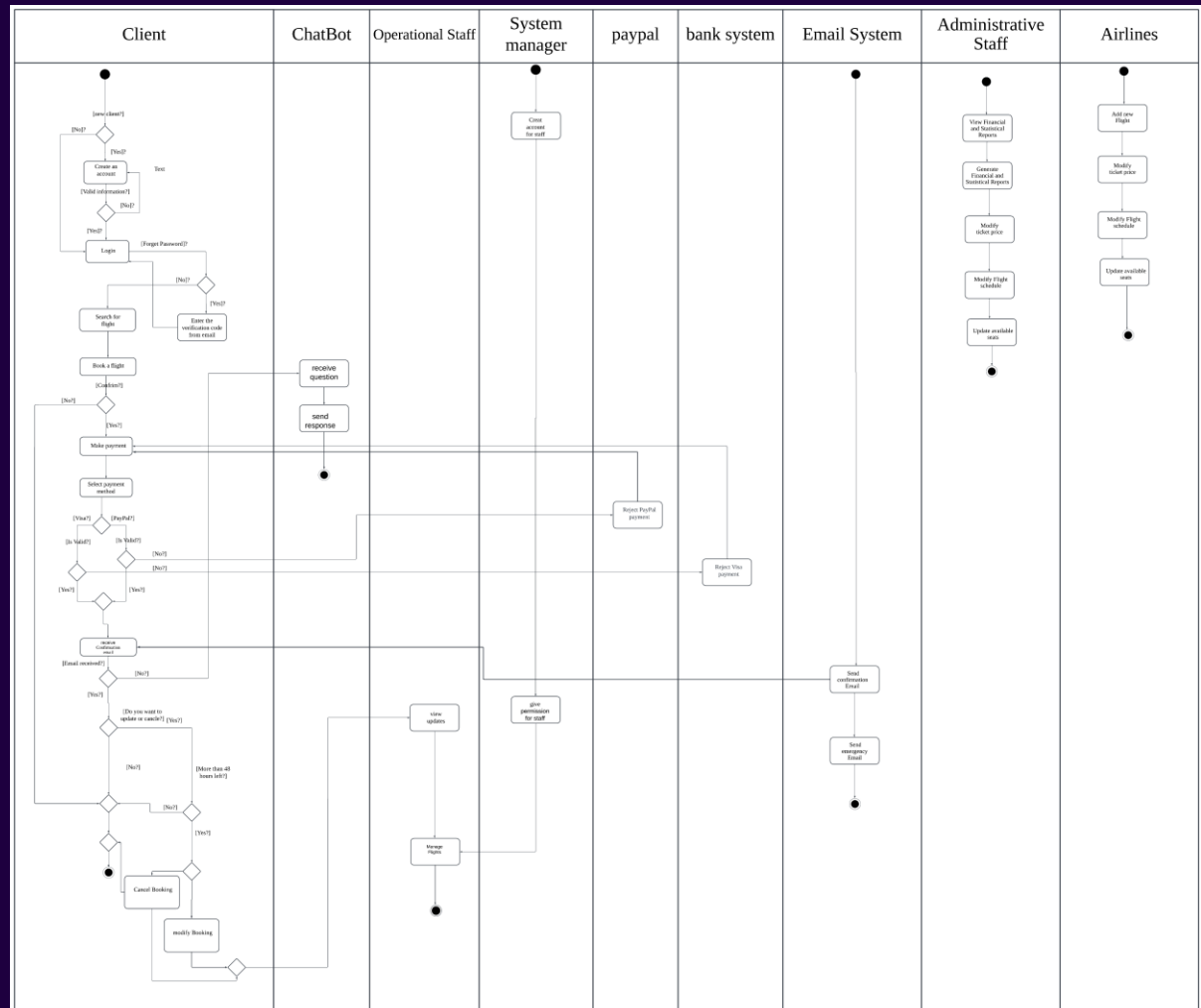
+  
+

## ➤ Overall Use case diagram

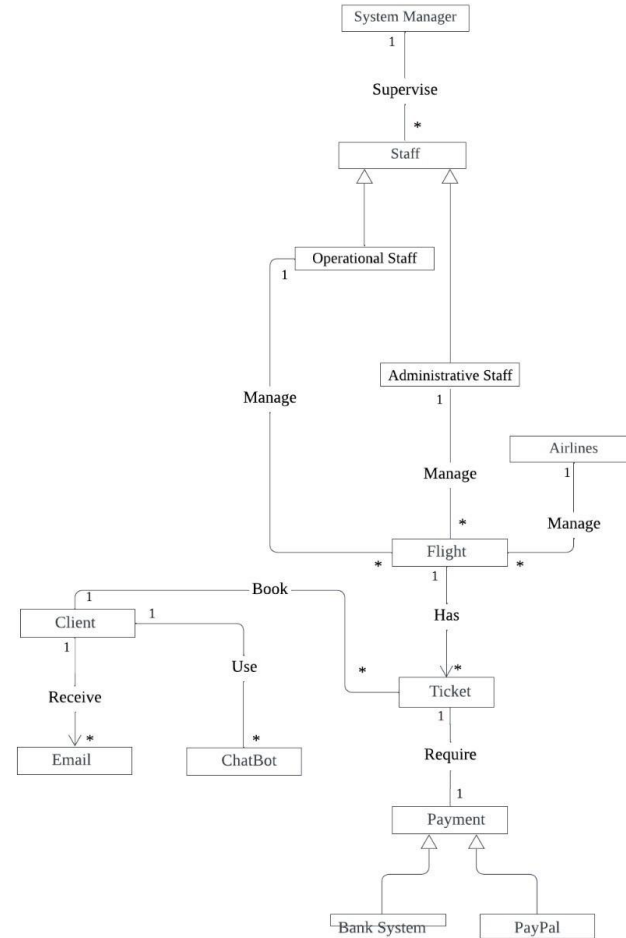


## Detailed description of “Create an Account use case”

System Name	Online Flight Booking System
Use Case Title	Create an Account
Description	A new client registers with personal details (email, password, etc.). After verification, the system validates the data for age and security before activating the account.
Actors	Clients
Data	full name, Email, Password, Date of Birth, Phone Number
Stimulus/Trigger	The client clicks 'Create an Account' and submits registration details.
Pre-conditions	1. The system is online with stable database and email verification. 2. The client is logged out to start registration. 3. The system validates emails and checks duplicates in real-time. 4. Age verification ensures compliance with the minimum age requirement.
Workflow OR Sequence/Flow of Events	1. Initiate Registration: The client clicks "Create an Account" and enters full name, email, password, phone number, and date of birth. 2. System Validation: The system validates the email (format and uniqueness), password (strength and match), and date of birth (age ≥18). If the email is registered, the system displays: "This email is already registered" and prompts the client to use a valid phone number as an alternative. Weak or mismatched passwords prompt corrective feedback. Underage clients see: "You must be at least 18 years old." 3. Account Verification: A verification email with an activation link is sent. Clicking the link activates the account, while failing to do so cancels the process. 4. Final Result: The account is either successfully activated or canceled.
Post-conditions/ Response	1. Upon successful verification, the account becomes "active," granting full system access. 2. Client data, including IP address and creation timestamp, is securely stored per privacy protocols. 3. Unverified accounts are canceled to prevent duplicates or incomplete entries.
Comments	The system must ensure data security, validate inputs, and prevent duplicate or incomplete account creation.

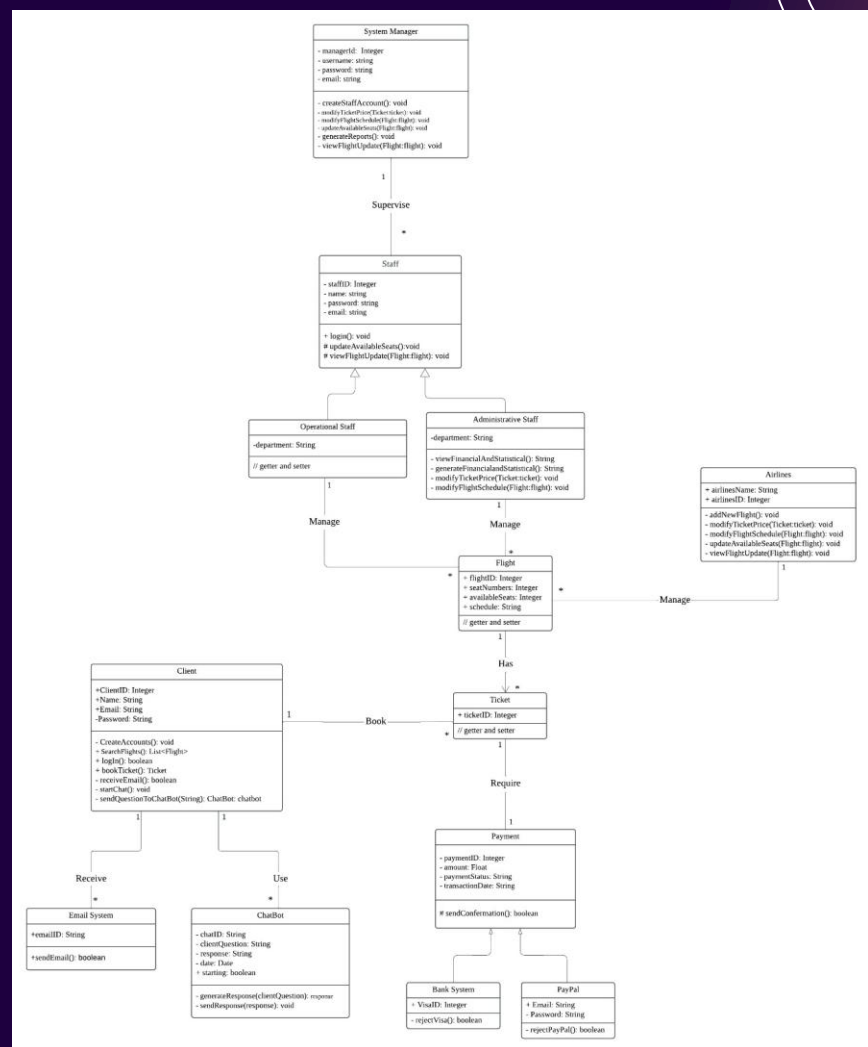


# Overall Analysis Class Model



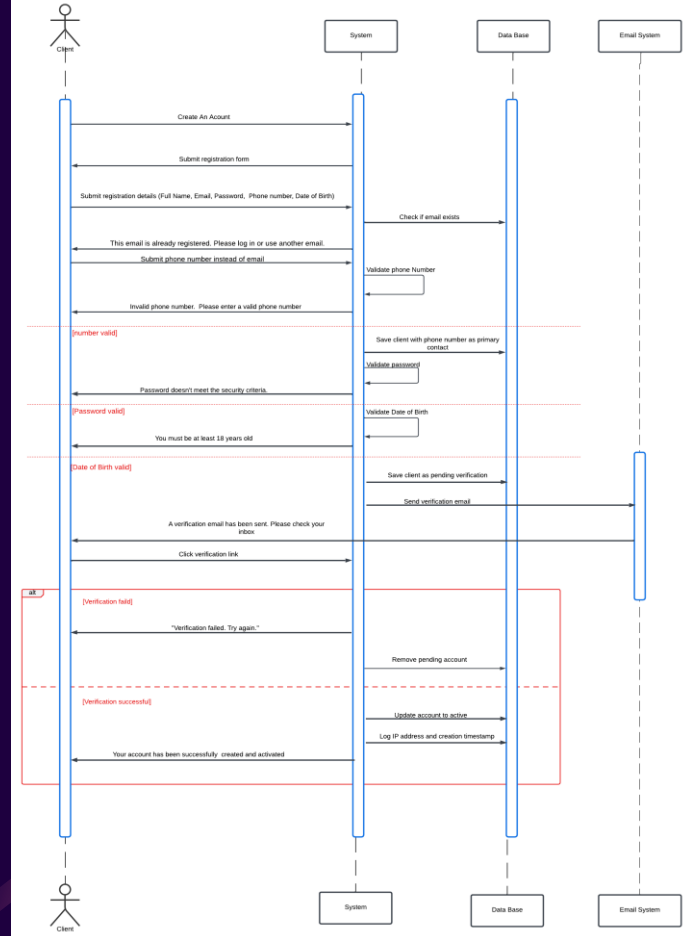


# Overall detailed Class Model



# Sequence diagram

## “Create an Account use case”



## ➤ Design Goals

### 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 the direct dependencies.

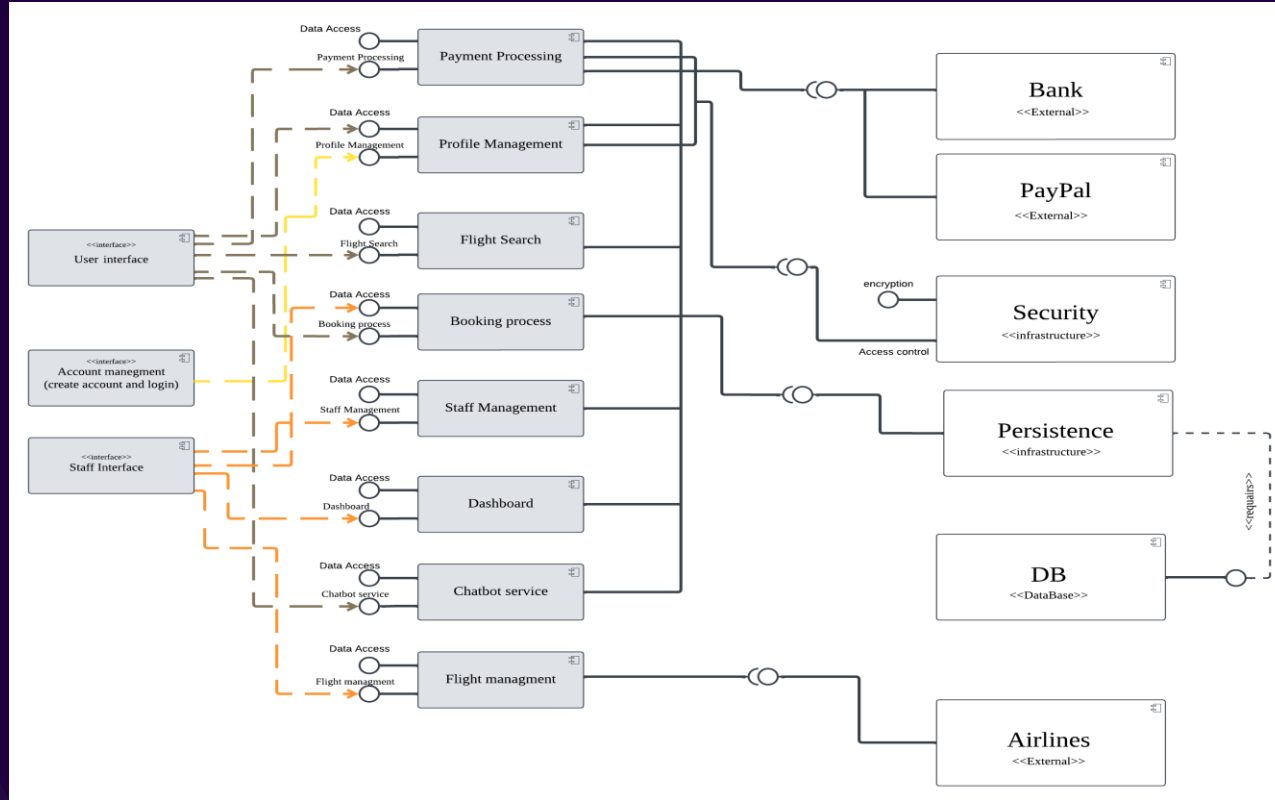
### II. High Cohesion:

- Objective: Create structured system component to focus on single, well-defined functionalities and reducing overall complexity.
- Implementation: Components are separated by distinct functions. For example, the Flight module handles reservations, while the Payment module processes transactions. This modular design improves maintainability, scalability, and makes the system easier to manage.

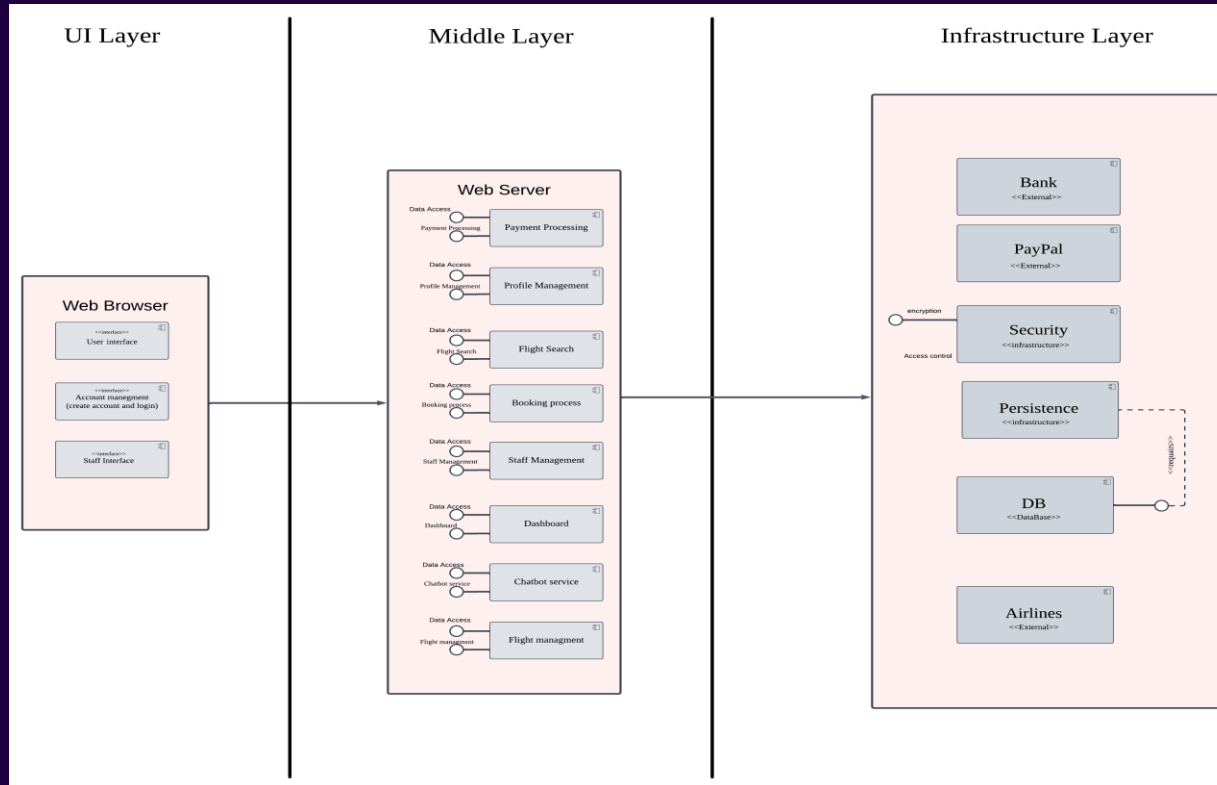
### III. User-Friendly interface:

- Objectives: Provide an obvious and accessible system interface, to make sure that the staff and the manager can perform their tasks effectively with minimal training “less or equal to two days”.
- Implementation: Design a user-friendly interface based on UX research, implementing features intuitive navigation and guided workflows for client, and Staff dashboards are role-specific with built-in safeguards.

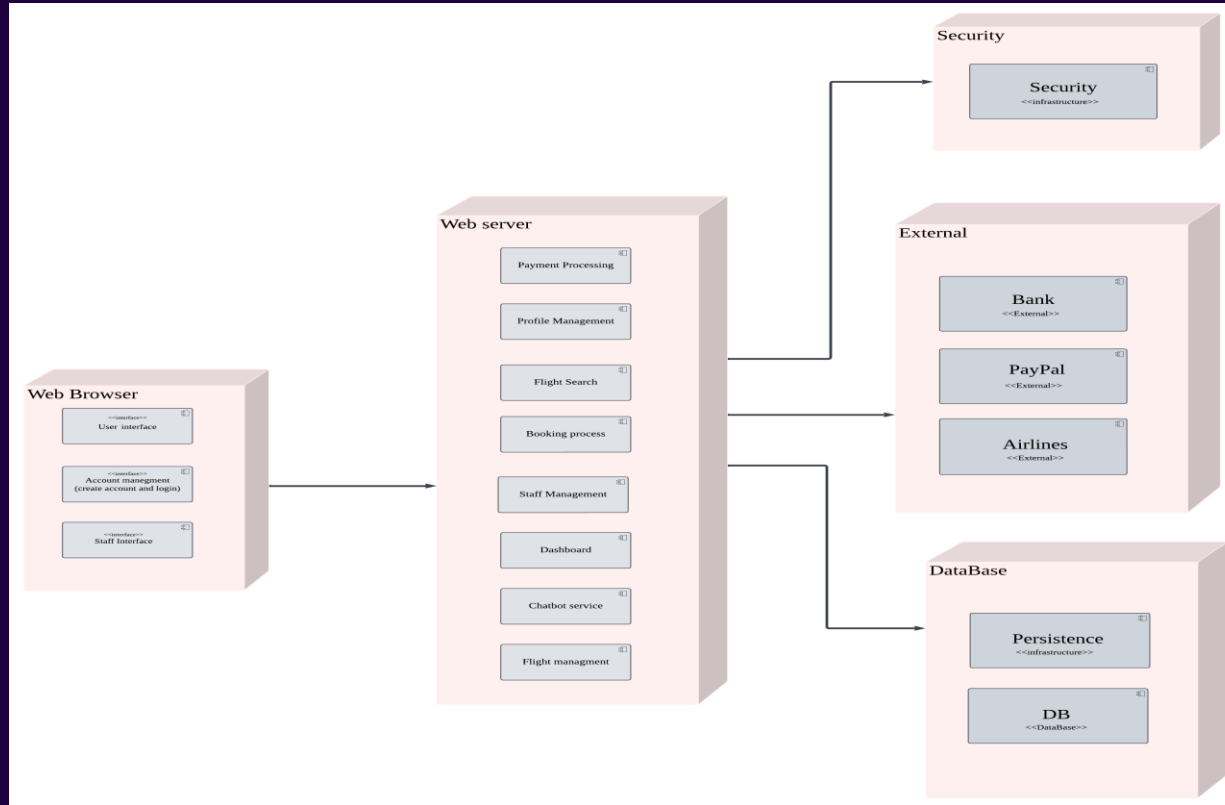
# Overall architecture “Component diagram”



# ➤ System and architectural Design “Layered Diagram”



# Deployment modelling





THANKS!

Do you have any questions?