Aqualure

WATER TRANSPORTATION SYSTEM

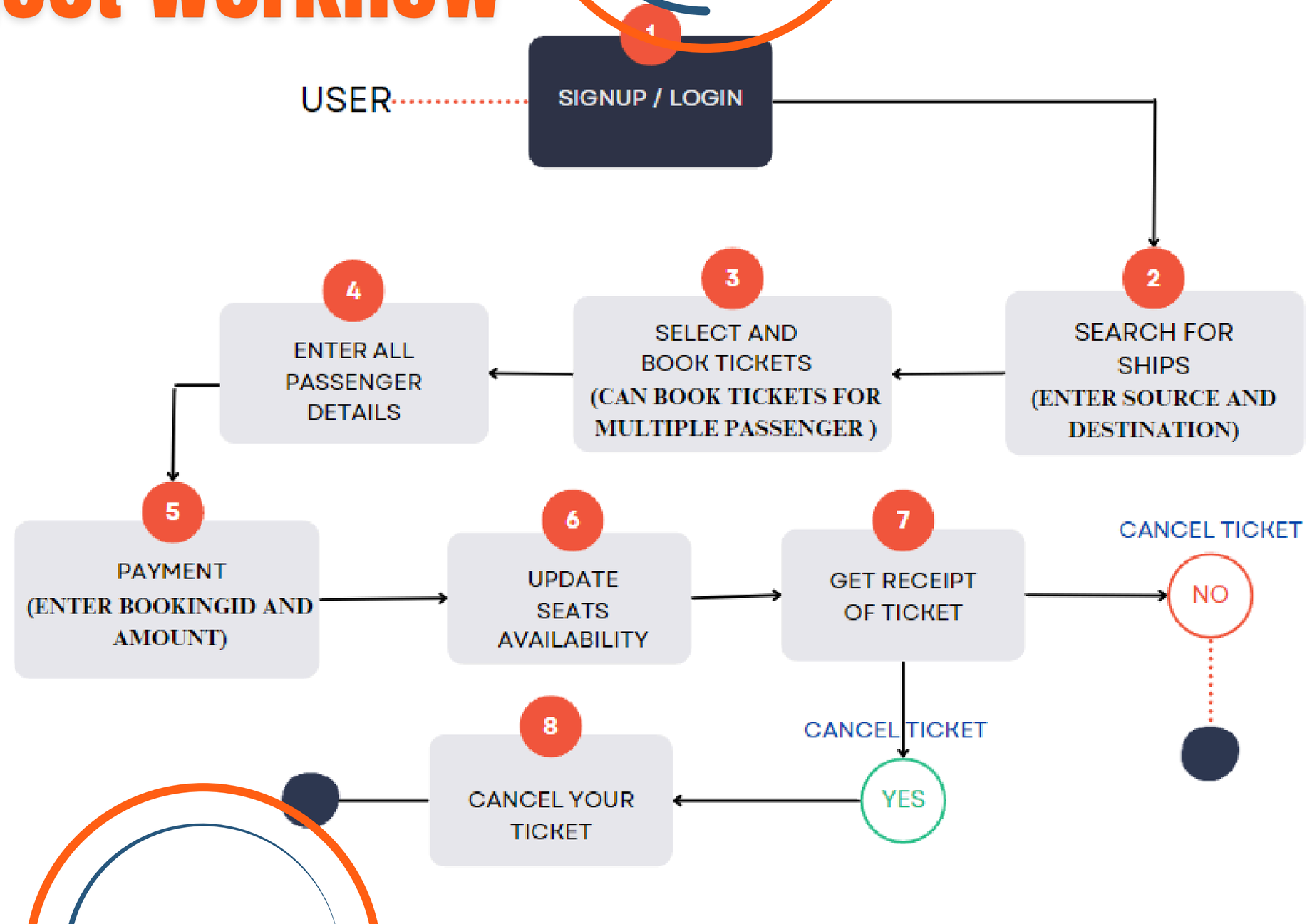
Presented by TEAM GAMMA-

Ujjwal Patil
Sakshi Kalyankar
Vedant Shrirao
Aditya Baheti
Maitrey Chamnikar

CONTENTS

- Project Workflow
- Class Diagram
- Logical Data Model (LDM)
- Jira Dashboard
- SOLID principle usage
- OOPs concept integration

Project Workflow



User and Admin Actions

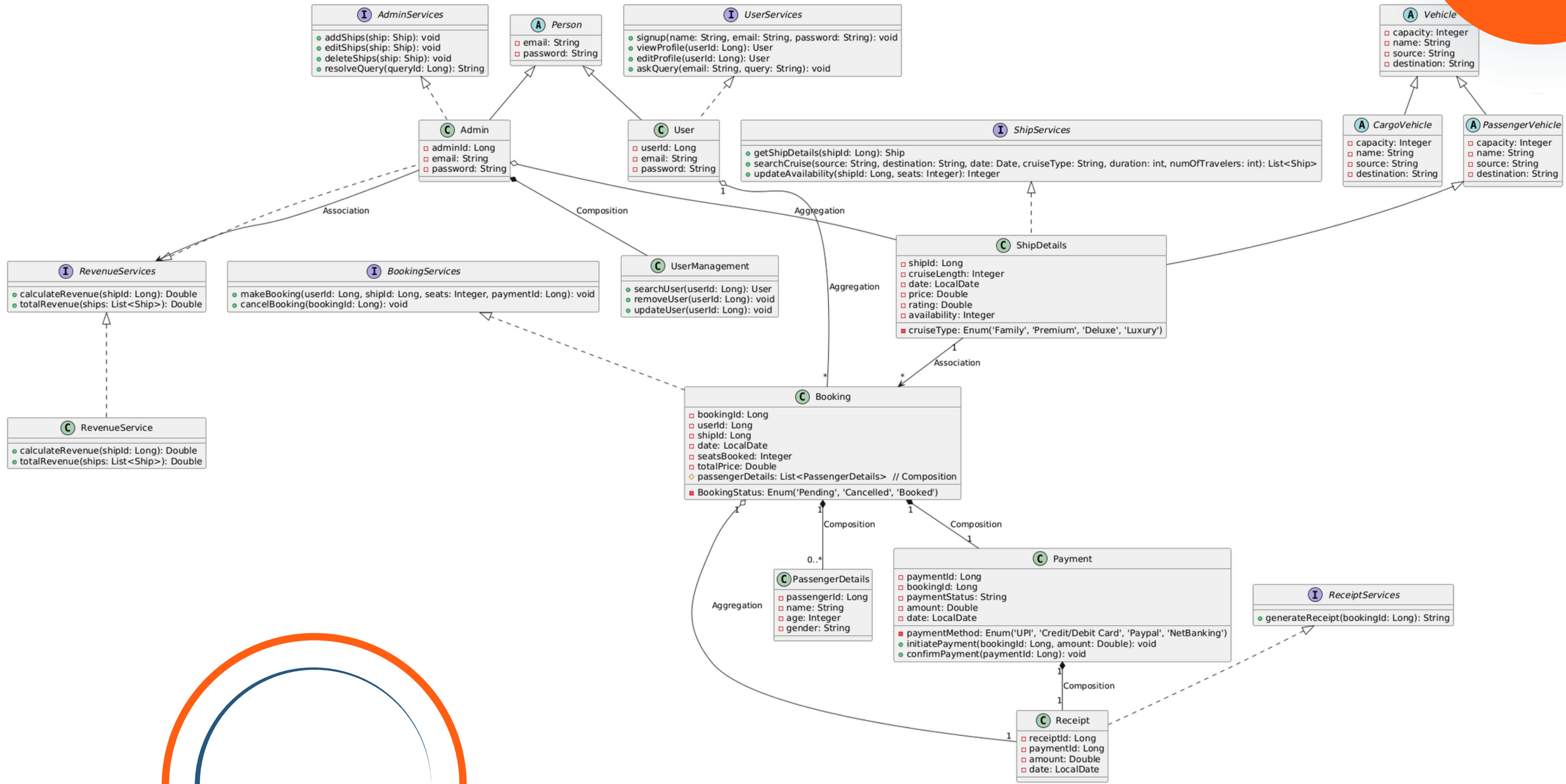
User Actions

Signup/Login
Search Ships
Book Ticket
Enter Passenger Details.
Make Payment
Cancel Ticket
Get Receipt
Ask Query

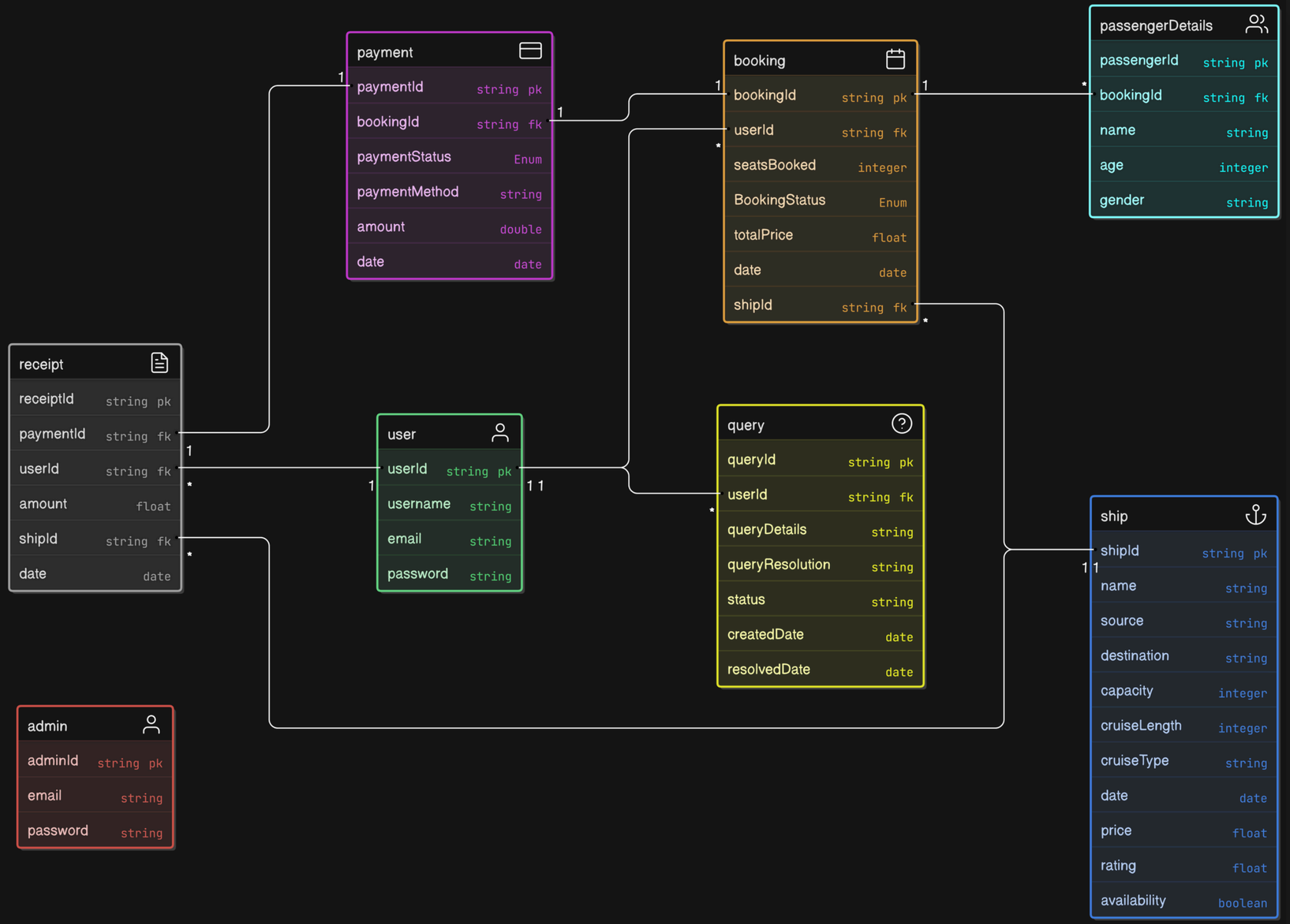
Admin Actions

Signup/Login
Add Ship
Edit Ship
Delete Ship
Monitor User Activities
Resolve Queries
Get Revenue Details
Update Admin Info
Get Admin Details

CIASS DIAGRAM



LOGICAL DATA MODEL (LDM)



JIRA DASHBOARD

Projects / Water-Transport-System

Backlog

Search | SK UP VS AB MC | Epic | Type | Insights | View settings

Epic

- Issues without epic
- Authentication (Login/Signup)
- User Dashboard
- Admin Dashboard
- Home page
- Booking ticket
- Ship Management(Admin)

WTS Sprint 2 1 Jan – 5 Jan (51 issues) — 0 0 57 — Start sprint

~~WTS-42~~	As a user, I want contact details so that I can clarify my doubts with the organization.	HOME PAGE	DONE	9	SK
~~WTS-40~~	As a user, I want to see offers and packages option so that I can book the journey budget ...	HOME PAGE	DONE	10	UP
WTS-93	HomePage	HOME PAGE	DONE	13	AB
~~WTS-37~~	As a user I want to plan my trip in premium class with more amenities so that i can compl...	BOOKING TICKET	DONE	-	UP
~~WTS-38~~	As a user I want to plan my trip in deluxe class with more amenities than premium class so...	BOOKING TICKET	DONE	-	UP
~~WTS-39~~	As a user I want to plan my trip in luxury class with best services so that i can have luxurio...	BOOKING TICKET	DONE	-	UP
~~WTS-80~~	As an admin, I want to provide the booking option as luxury so that user can get the luxur...	SHIP MANAGEMENT(A...	DONE	-	SK
~~WTS-68~~	As an admin, I want to provide the facilities available for the passengers so that user can p...	SHIP MANAGEMENT(A...	DONE	-	SK
~~WTS-48~~	As a user, I want to enter the different fields of details so that i can add single or multiple ...	BOOKING TICKET	DONE	-	VS
~~WTS-46~~	As a user, I want option like rating and review so that I can see the feedback of the availab...	BOOKING TICKET	DONE	-	MC

SOLID PRINCIPLE USAGE

1. Single Responsibility Principle (SRP)
   - Definition in Diagram: Each class in the diagram has a single well-defined purpose. For example:
     - The AdminServices class is responsible only for administrative operations like adding, editing, and deleting ships.
     - The BookingServices class handles booking-related functionalities.
     - The RevenueService class is dedicated to calculating revenue.
   - Benefits:
     - Each class can be changed independently if its specific responsibility changes.
     - Reduces coupling and improves maintainability.
2. Interface Segregation Principle (ISP)
   - Definition in Diagram: The system ensures that interfaces are specific to their use case. For example:
     - The AdminServices interface defines methods that are specific to admin-related operations, avoiding unrelated methods.
     - The UserServices interface focuses on user-related actions like sign-up and profile updates.
     - By segmenting responsibilities into specific interfaces, classes are not forced to implement unnecessary methods.
   - Benefits:
     - Avoids the "fat interface" problem.
     - Classes that implement these interfaces are simpler and more focused.

contract without worrying about implementation details.

- Benefits:
  - Encourages loose coupling.
  - Improves testability by allowing mock implementations.

SOLID PRINCIPLE USAGE

4. Liskov Substitution Principle (LSP)

- Definition in Diagram:
  - Classes such as PassengerVehicle and CargoVehicle are subtypes of Vehicle. These subclasses can replace Vehicle without altering the functionality, maintaining the behavior expected of the Vehicle class.
  - Similarly, Admin is a subtype of Person, ensuring it fulfills the contracts of Person.
- Benefits:
  - Enhances polymorphism.
  - Simplifies testing and maintenance by ensuring substitutability.

5. Dependency Inversion Principle (DIP)

- Definition in Diagram:
  - High-level modules like BookingServices depend on abstractions (Booking) rather than concrete implementations.
  - The ReceiptServices interface abstracts receipt generation, ensuring the Payment class can rely on its contract without worrying about implementation details.
- Benefits:
  - Encourages loose coupling.
  - Improves testability by allowing mock implementations.

OOPs CONCEPT INTEGRATION

Abstraction:

- Interfaces like UserService and AdminService define methods without specifying their implementation, ensuring a clear contract for services.
- Controllers serve as intermediaries, hiding complex business logic and providing simplified access for users.

Encapsulation:

- Private fields in entity classes (e.g., username, email) restrict direct access, and are securely accessed via getters and setters.
- Annotations such as @Column and the organized package structure ensure modularity and maintain security of the data.

Inheritance:

- Common attributes and behaviors (e.g., email, password) are inherited by specific classes like Admin and User from the Person class.
- Service implementations (e.g., UserServiceImpl, AdminServiceImpl) inherit functionality defined in their respective interfaces.

Polymorphism:

- Interfaces support multiple implementations, such as AdminServiceImpl and UserServiceImpl, for flexible service behavior.
- Method overriding allows customization of functionality while maintaining consistency with the parent class or interface.

Aqualure

THANK YOU