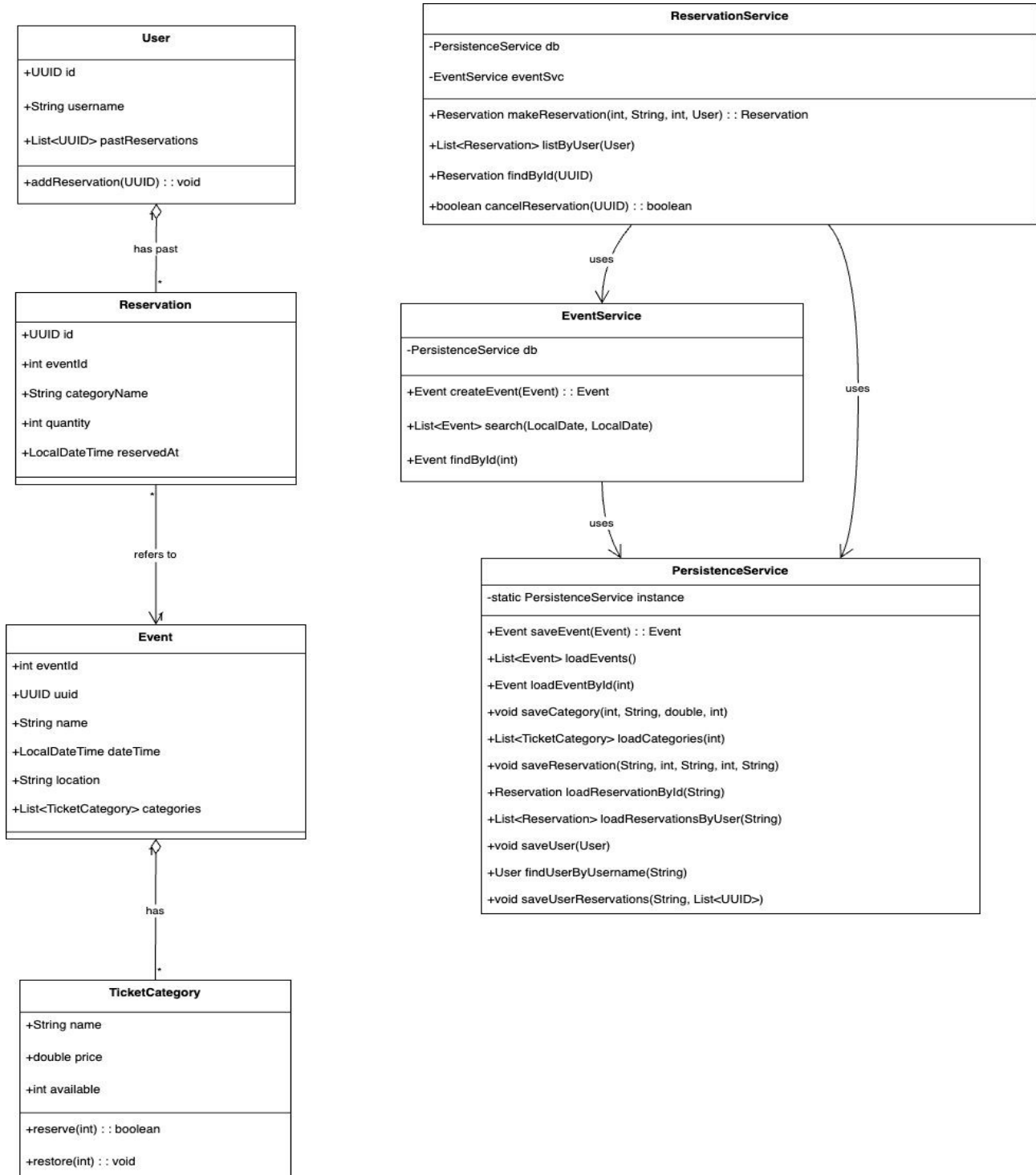# SEDS 504 – HW2 - Project Report: Online Event Ticket Reservation System

Team Members and Contributions

Name	Contribution
Umut Akın	Designed the overall system architecture. Implemented the main classes and core functions.
Toprak Zeybek	Enhanced and extended the initial implementation. Replaced UUIDs with internal ID management.
Yasin Türk	Wrote and executed unit tests. Verified application functionality and prepared the project report.

UML DIAGRAM

User
+UUID id

+String username

+List<UUID> pastReservations

+addReservation(UUID) : : void

has past

Reservation
+UUID id

+int eventId

+String categoryName

+int quantity

+LocalDateTime reservedAt

refers to

Event
+int eventId

+UUID uuid

+String name

+LocalDateTime dateTime

+String location

+List<TicketCategory> categories

has

TicketCategory
+String name

+double price

+int available

+reserve(int) : : boolean

+restore(int) : : void

ReservationService
-PersistenceService db

-EventService eventSvc

+Reservation makeReservation(int, String, int, User) : : Reservation

+List<Reservation> listByUser(User)

+Reservation findById(UUID)

+boolean cancelReservation(UUID) : : boolean

uses

EventService
-PersistenceService db

+Event createEvent(Event) : : Event

+List<Event> search(LocalDate, LocalDate)

+Event findById(int)

uses

uses

PersistenceService
-static PersistenceService instance

+Event saveEvent(Event) : : Event

+List<Event> loadEvents()

+Event loadEventById(int)

+void saveCategory(int, String, double, int)

+List<TicketCategory> loadCategories(int)

+void saveReservation(String, int, String, int, String)

+Reservation loadReservationById(String)

+List<Reservation> loadReservationsByUser(String)

+void saveUser(User)

+User findUserByUsername(String)

+void saveUserReservations(String, List<UUID>)

Usage Scenarios

1. **Guest Ticket Reservation**
   A user launches the application and selects the "Continue as Guest" option or User can sign

up. They search for concerts in December 2025 and find a suitable one. Without signing up, they reserve 2 VIP tickets. A unique reservation number is shown for tracking.

2. **User Login and Reservation History**
   A registered user logs in with their username. They search for sports events in the next month and book 3 regular tickets. They then view their current and past reservations using the "List My Reservations" option.

3. **Event Creation by Firm**
   A firm logs into the system backend and creates a new theatre event. They specify date, location, ticket categories, and ticket prices. The event immediately becomes available to users.

4. **Reservation Cancellation**
   A user who previously made a reservation logs in and chooses the "Cancel Reservation" option. They select the reservation ID to cancel, and the system confirms the cancellation.

Explanation of Inheritance

The system leverages inheritance to create a clean and maintainable hierarchy:

- **User Type Hierarchy**:
  A base class `User` is extended by specialized subclasses like `GuestUser` and `RegisteredUser`, allowing for differentiated behavior while sharing common functionality.

- **Command Pattern**:
  An abstract `Command` interface is implemented by concrete classes such as `SearchEventsCommand`, `CreateEventCommand`, and `ListMyReservationsCommand`. This enables dynamic execution of commands via polymorphism.

Explanation of Polymorphism

Polymorphism is employed primarily in the command execution mechanism. Each menu option corresponds to a class implementing the `Command` interface. The UI executes commands uniformly without knowing their specific type:

```
Command cmd = commandMap.get(userChoice);
cmd.execute();
```

This design promotes scalability and cleaner code, making it easy to add new commands without altering the central logic.

Explanation of Data Persistence

The system uses **a SQLite database (`.db` file)** for data persistence. The database schema includes tables for:

- `users` – to store login credentials
- `events` – event name, date, location, and metadata
- `ticket_categories` – for ticket type and price per event
- `reservations` – linked to users or anonymous reservations

Java interacts with the SQLite database using the `sqlite-jdbc` driver. Queries are executed via JDBC to insert, retrieve, and update data. All operations (login, search, reservation, cancellation) result in real-time updates to the `.db` file, ensuring data consistency and permanence between sessions.

USAGE SCENARIOS WITH SCREEN SHOTS

Login and Create an Event

```
yasin@yasinturk:~/Desktop/seds504-ticket-reservation$  /usr/bin/env /home/yasin/.vscode/extensions/redhat.java-1.42.0-linux-x64/jre/21.0.
-linux-x86_64/bin/java @/tmp/cp_600nz45scm5mh1nhevw621g4y.argfile com.myorg.ticket.App
Welcome to the Ticket Reservation System!

--- Authentication ---
1) Login
2) Sign Up
3) Continue as Guest
0) Exit
Choose: 1
Username: toprakzeybek
Welcome back, toprakzeybek!

--- Menu ---
0) Exit
1) Create Event
2) Search Events
3) Reserve Tickets
4) View Reservation
5) List My Reservations
6) Cancel Reservation
Choose: 1
Event name: IYTE Concert
Date & time (yyyy-MM-ddTHH:mm): 2025-12-12T14:00
Location: IYTE
Add ticket category? (y/n): y
 Category name: VIP
 Price: 10
 Available tickets: 100
Add ticket category? (y/n): n
Created event with ID: 6

--- Menu ---
0) Exit
1) Create Event
2) Search Events
3) Reserve Tickets
4) View Reservation
5) List My Reservations
6) Cancel Reservation
Choose: []
```

Search for Events

```
--- Menu ---
0) Exit
1) Create Event
2) Search Events
3) Reserve Tickets
4) View Reservation
5) List My Reservations
6) Cancel Reservation
Choose: 2
Start date (yyyy-MM-dd): 2020-12-12
End date   (yyyy-MM-dd): 2025-12-31
ID: 1 | Test Concert @ 2025-12-31T12:00 | Izmir
ID: 2 | Duman Concert @ 2025-12-31T12:15 | Istanbul
  - Private: $1000.00 (100 available)
ID: 3 | Yasin Concert @ 2025-03-03T12:30 | Izmir
  - public: $19.00 (100 available)
ID: 4 | Test Event @ 2025-01-01T12:00 | Karabaglar
  - VIP: $10.00 (100 available)
ID: 5 | Unit test event @ 2023-12-12T01:00 | Roof Kafe
  - VIP: $102.00 (0 available)
ID: 6 | IYTE Concert @ 2025-12-12T14:00 | IYTE
  - VIP: $10.00 (100 available)

--- Menu ---
0) Exit
1) Create Event
2) Search Events
3) Reserve Tickets
4) View Reservation
5) List My Reservations
6) Cancel Reservation
Choose: []
```

Reservation and List Reservations

```
--- Menu ---
0) Exit
1) Create Event
2) Search Events
3) Reserve Tickets
4) View Reservation
5) List My Reservations
6) Cancel Reservation
Choose: 3
Event ID: 6
Category name: VIP
Quantity: 10
Reserved! Your Reservation ID = 62f9f7b2-1597-4a89-a707-61c00f7620f3

--- Menu ---
0) Exit
1) Create Event
2) Search Events
3) Reserve Tickets
4) View Reservation
5) List My Reservations
6) Cancel Reservation
Choose: 4
Reservation ID: 62f9f7b2-1597-4a89-a707-61c00f7620f3
Reservation 62f9f7b2-1597-4a89-a707-61c00f7620f3: event 6, category VIP, qty 10, at 2025-05-27T11:21:18.359268965

--- Menu ---
0) Exit
1) Create Event
2) Search Events
3) Reserve Tickets
4) View Reservation
5) List My Reservations
6) Cancel Reservation
Choose:
```

Cancellation

```
--- Menu ---
0) Exit
1) Create Event
2) Search Events
3) Reserve Tickets
4) View Reservation
5) List My Reservations
6) Cancel Reservation
Choose: 6
--- Your Reservations ---
ID: 62f9f7b2-1597-4a89-a707-61c00f7620f3 | For Event: 6 | Category: VIP | Qty: 10
ID: d99de3e3-c4a2-4a8d-ba07-33f26900a41c | For Event: 5 | Category: VIP | Qty: 3

Enter ID of reservation to cancel: d99de3e3-c4a2-4a8d-ba07-33f26900a41c
Reservation cancelled successfully.

--- Menu ---
0) Exit
1) Create Event
2) Search Events
3) Reserve Tickets
4) View Reservation
5) List My Reservations
6) Cancel Reservation
Choose:
```