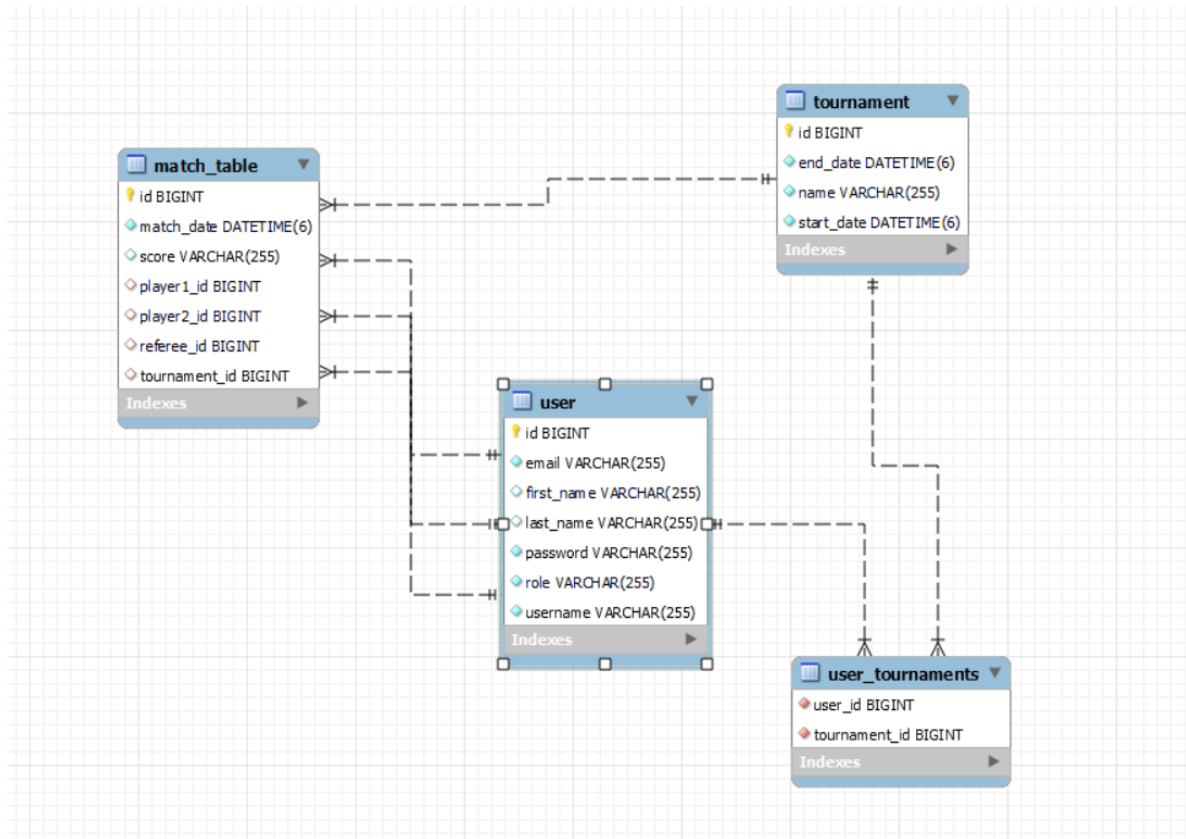


Software Design – Assignment 1

Tennis App

Database Schema



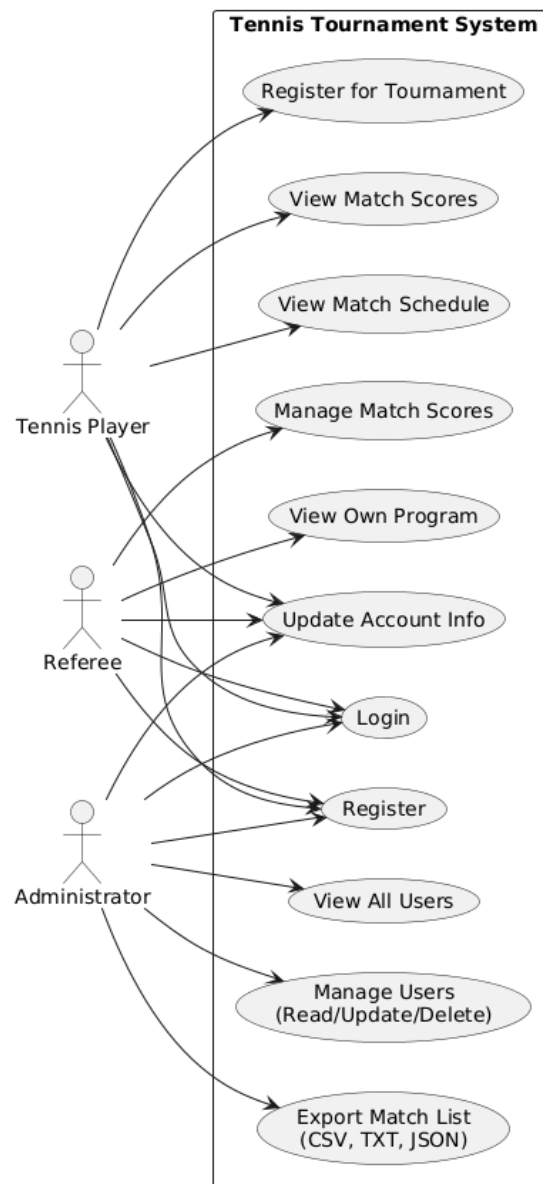
The database consists of three main entities: **User**, **Tournament**, and **Match**.

- **User**: Represents all types of users in the system, including players, referees, and admins. Each user has fields like id, username, password, email, first name, last name, and role.
- **Tournament**: Represents a tennis tournament with fields such as id, name, start date, end date, and registration deadline. Players can register for tournaments. The relationship between players and tournaments is many-to-many.
- **Match**: Represents a match that takes place within a tournament. Each match has an id, match date, score, and links to two players (player1 and player2), a referee, and the tournament it belongs to.

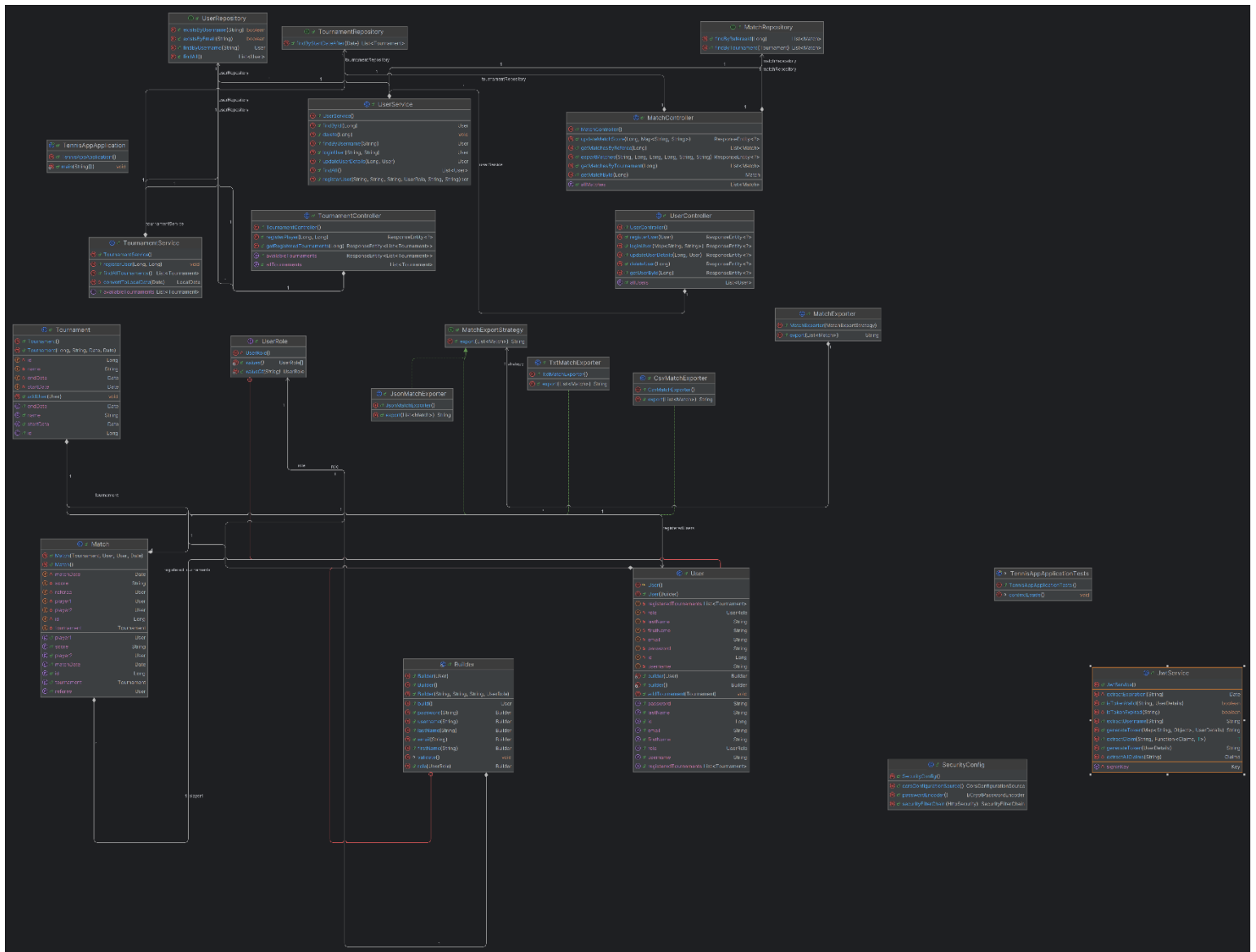
The relationships between entities are:

- Users and tournaments have a many-to-many relationship through a join table for registrations.
- A tournament has many matches.
- A match is linked to two players and one referee, all of whom are users.

Use Case Diagram



Class Diagram



1. Entities

- **User:** Represents a person using the system. Users have a role, username, email, and are associated with matches and tournaments (both registered and refereed).
- **Tournament:** Represents a tennis competition with details such as name, startDate, endDate, and registrationDeadline. It has a many-to-many relationship with User.
- **Match:** Represents a match in a tournament. It includes player1, player2, referee, score, and a reference to the Tournament it belongs to.

2. Repositories

- **UserRepository, TournamentRepository, MatchRepository:** Interfaces that handle database operations for their respective entities using Spring Data JPA.

3. Controllers

- **UserController:** Manages user-related endpoints like registration, login, and updates.
- **TournamentController:** Handles tournament-related operations such as viewing, registering, and listing available tournaments.
- **MatchController:** Provides endpoints to retrieve match data, update scores, and export match information.

4. Services

- **UserService:** Handles logic for user operations such as registration, login, and update.
- **TournamentService:** Manages tournament registration and filtering.
- **MatchExportStrategy:** Interface for match exporting strategy.
 - **CsvMatchExporter, JsonMatchExporter, TxtMatchExporter:** Implementations for exporting matches in different formats.
- **MatchExporter:** Uses the strategy pattern to export matches through one of the format-specific exporters.

5. Utilities & Configuration

- **SecurityConfig:** Defines Spring Security rules and authentication behavior.
- **JwtService:** Utility for generating and validating JWT tokens.
- **TennisAppApplication:** Main Spring Boot entry point.
- **TennisAppApplicationTests:** Contains test configurations.

6. Builder

- **Builder class:** Provides custom object construction, likely for the User or another entity using method chaining.