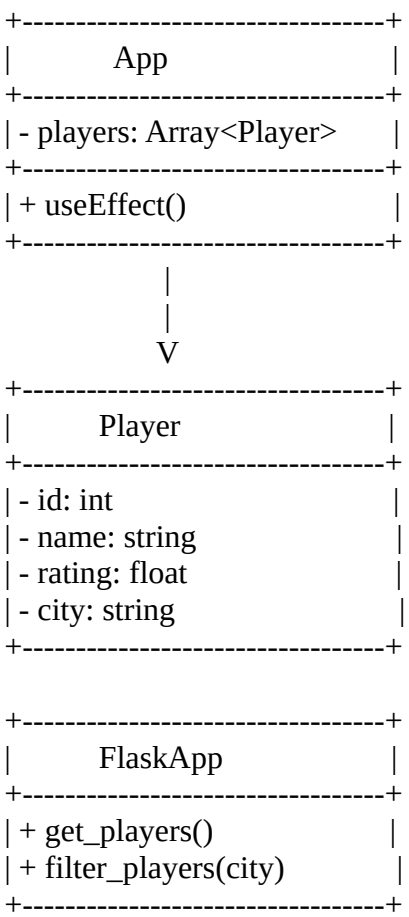
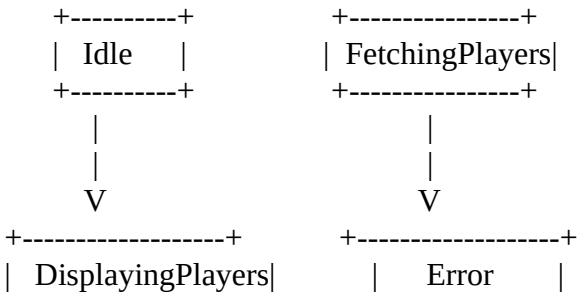
UML diagram

Class Diagram:



Explanation:
- `App`: Represents the main component of the React application. It contains a state variable `players` which holds an array of `Player` objects. The `useEffect()` method is responsible for fetching player data from the Flask API.
- `Player`: Represents the structure of a player object with attributes `id`, `name`, `rating`, and `city`.
- `FlaskApp`: Represents the Flask application with methods `get_players()` to retrieve all players from the API and `filter_players(city)` to filter players by city.

State Machine Diagram:



+------------------+ +------------------+

Explanation:
- `Idle`: Initial state before the application starts.
- `FetchingPlayers`: State when the application is fetching player data from the Flask API.
- `DisplayingPlayers`: State when the application successfully retrieves and displays player data.
- `Error`: State when an error occurs during the data fetching process.

Transitions:
- `Idle` to `FetchingPlayers`: Triggered when the application starts.
- `FetchingPlayers` to `DisplayingPlayers`: Triggered when player data is successfully fetched.
- `FetchingPlayers` to `Error`: Triggered when an error occurs during data fetching.