

Assignment One

Full Stack T app



April 10, 2024

Cot mihaita marian

Technical University of Cluj-Napoca

Table of Contents

1. **Database Diagram**
2. **Use-case diagram**
3. **Class diagram**

**Introduction**

Welcome to the documentation for our comprehensive Full Stack React application designed for the management of tennis matches and tournaments. This application serves as a powerful tool for tennis enthusiasts, tournament organizers, coaches, and players alike, streamlining the process of organizing, scheduling, and tracking tennis events efficiently.

With the integration of modern web technologies, including React.js for the frontend and a robust backend architecture, our application offers a user-friendly interface coupled with powerful functionalities to cater to the diverse needs of the tennis community.

In this documentation, you'll find comprehensive guides, tutorials, and reference materials to help you navigate through the features and functionalities of our application. Whether you're a seasoned developer looking to extend the application's capabilities or a novice user seeking guidance on basic operations, this documentation serves as your go-to resource for understanding and utilizing our Full Stack React application for tennis match and tournament management.

**Data-base diagram**

In this section, we provide a concise overview of the database schema, illustrating the relationships between different entities and their attributes. A database diagram serves as a visual representation of the database structure, offering insights into the organization of data and the connections between various components.

Throughout this documentation, you'll find detailed explanations of each entity and its attributes, as well as the relationships established between them. By visualizing the database schema, users gain a better understanding of how data is stored, retrieved, and manipulated within the system.

A computer screen shot of a black screen

Description automatically generated

**Use-case diagram**

In this section, we provide a brief overview of the use-case diagram, which serves as a powerful tool for understanding the interactions between system users and the functionality offered by the system.

A use-case diagram offers a visual representation of the various actions (use cases) that users can perform within the system, as well as the actors (users or external systems) involved in these interactions. By mapping out these use cases and actors, stakeholders gain a clear understanding of the system's capabilities and how it supports user workflows.

**A diagram of a company

Description automatically generated**

**Class diagram**

In this section, we introduce the class diagram, which provides a visual representation of the structure and relationships of classes within a system.

A class diagram serves as a blueprint for the system's object-oriented design, illustrating the classes, attributes, methods, and relationships between them. It offers a high-level overview of the system's architecture, helping developers and stakeholders understand the organization of code and the interactions between different components.

A diagram of a application

Description automatically generated