Project 4

March Madness Requirements

FURPS+

Table of Contents

1. Introduction

a. Purpose

i. The March MaDness Bracket Simulator is a standalone Java application that simulates the annual college basketball tournament known as March Madness. The system enables fans to create prediction brackets, earn points based on correct predictions, and compare their performance against other users. This document outlines the functional and non-functional requirements, extended features, and user interface expectations for the project.

b. Scope

i. The simulator will allow users to:
   1. Log in and access personalized brackets.
   2. Create and modify a bracket prediction for a tournament consisting of 64 teams.
   3. Finalize and save their bracket, triggering a tournament simulation.
   4. View detailed simulation outcomes with game scores and comparative rankings.
   5. Compare their predicted bracket with historical tournament results.

2. Functionality Requirements

a. Game Overview and Welcome Screen

i. Game Description:
   1. The application will display a comprehensive description of the tournament including its structure, the process for predictions, and the scoring mechanism upon startup.

ii. Welcome Message:

1. A welcome message will greet users on the landing page before login, setting the stage for a positive interactive experience.

b. User Authentication and Bracket Management

i. Login Interface:
   1. Users are required to log in with unique credentials.
   2. Upon login, the system will verify if a saved bracket exists for the user. If it does, the saved bracket is loaded; if not, an empty bracket is generated.

ii. User-Bracket Association:
   1. Each user's bracket will be saved in a serialized format.
   2. The system will maintain a dropdown menu (or equivalent UI component) that lists all logged in users to enable easy retrieval or switching between saved brackets.

c. Bracket Creation and Editing

i. Initial Bracket Loading:
   1. After login, a new or saved bracket containing 64 teams will be presented to the user.

ii. Interactive Prediction:
   1. The application will allow users to click on a team name to indicate the predicted winner for a game.
   2. When a prediction is made, the team will automatically be advanced to the corresponding slot in the next round.

iii. Cascading Update:
   1. When a team is removed or changed at any stage in the bracket:
      a. All subsequent selections that depend on the removed team will be automatically cleared.

b. This cascading propagation ensures consistency across the entire bracket.

d. Bracket Finalization and Tournament Simulation

i. Bracket Finalization:
1. The user will have the option to finalize their bracket once all predictions are complete.
2. Upon finalization, the system will serialize and store the bracket.
3. Finalizing the bracket will enable the “Simulate Tournament” functionality.

ii. Tournament Simulation:
1. The simulation engine will run all tournament games based on the finalized bracket.
2. Each game will have randomized scores within a defined range (50–125 points).
3. The score calculation will incorporate a ranking based modifier to offer higher ranked teams a statistical advantage.

iii. Scoring Mechanism:
1. Points are awarded for correct predictions, with later rounds awarding higher points.
2. A ranking table will display the user name, total points, and winning team for each bracket, sorted in descending order based on points.

3. Usability Requirements

a. Responsive User Interface

i. Scalable Layout:
1. The GUI will be designed to automatically adapt and resize based on the dimensions of the application window.

ii. Intuitive Navigation:

1. All controls, including buttons, form fields, and navigation elements, must be clearly labeled and arranged logically to minimize user confusion.

b. Reset and Edit Capabilities

i. Reset Function:
   1. A “Reset” button will be available to allow users to clear their current bracket and start over.
   2. Once the bracket is finalized, the “Reset” button will be disabled.

ii. Real Time Editing:
   1. Users will be permitted to continuously edit their bracket until it is finalized. Changes must be immediately reflected in all dependent parts of the bracket.

c. Detailed Results Display

i. Score Visualization
   1. Detailed game scores will be displayed within the bracket layout, ensuring users can understand the outcome of each matchup

ii. Feedback Mechanisms
   1. The system will provide visual indicators for score updates and bracket progress during the simulation.

4. Reliability Requirements

a. Data Integrity and Validation

i. Bracket completion check:
   1. Prior to finalization, the system will verify that every slot in the bracket has been filled.

2. Incomplete brackets will prompt the user to finish their selections

ii. Robust Exception Handling:
1. The application will implement comprehensive error handling for:
   a. File I/O issues (ex. missing or corrupted files)
   b. serialization/deserialization errors
   c. Invalid login credentials and other runtime exceptions

5. Performance Requirements

a. Multi-User and Simulation Performance

i. Multi-Users:
1. The application will efficiently support up to four players.

ii. Responsive Updates:
1. All simulations and UI updates will be optimized to ensure a smooth and responsive user experience.

6. Supportability Requirements

a. Editable Brackets and Continuous Improvement

i. Dynamic Editing:
1. Users will be allowed to change selections in their brackets as long as the bracket is not finalized.

ii. In App Documentation:
1. An “Instructions” or “Help” button will be provided post login.
2. This section will offer detailed, step by step guidance on using the application, making it accessible even for novice users.

iii. Maintainable Code and Documentation:
1. All code will be thoroughly documented, and technical documentation (such as UML diagrams and user manuals) will be maintained in the project repository.

7. Plus Features

a. Enhanced Simulation Based on Team Statistics

i. Realistic Game Outcomes:
1. The simulation will integrate team performance metrics (ex. offensive and defensive points per game) to give top-ranked teams a slight competitive edge, reflecting real-life tournament dynamics.

b. Historical Outcome Comparison

i. Comparative Analysis:
1. An optional module will allow users to compare their simulation results with historical tournament data (ex. the 2017 championship results), providing insights into prediction accuracy.

c. Tooltip and Visual Enhancements

i. Tooltips:
1. Detailed tooltips will be available for team elements, displaying key information on hover.

ii. Color-Coded Feedback:
1. Predictions will be visually distinguished:
    a. Correct predictions in green.
    b. Incorrect predictions in red.