CS225 Software Development

Maintenance Team: Baheeja, Jordan, Gabi, Valerie, Brad, Andrew, Gabriel, Lucas

CS225 Project 4 Maintenance Report

Bugs🪲

"find and document bugs"

Bracket.java

- ➔ Removed private void add(int position, String s){
  bracket.add(position, s);
  - ◆ Never Used*

BracketPane.java

- ➔ Removed ArrayList<GridPane> gridPanes = new ArrayList<>();
  - ◆ Never used*
- ➔ Pane createFinalFour does not have the hover effect of BracketNode
  - ◆ Does not get the cyan background when hovering over the winning team pick of the final two teams
  - ◆ Also named createFinalFour but only creates the final two teams

MarchMadnessGUI.java

- ➔ Changed private TableView table to private TableView<Bracket> table;
- ➔ Needed parameters*

Missing Features❌

"list missing features that can make the system more in-line with the description of the project"

GUI Responsiveness:

The graphical user interface (GUI) should exhibit dynamic resizing capabilities. When the user adjusts the frame size, all UI elements should automatically adjust their dimensions and layout to maintain a consistent and user-friendly interface.

Time Estimate: 2– 4 hours

Difficulty: Medium

- Requires setting up layout managers like VBox, HBox, GridPane, or AnchorPane properly, possibly with binding properties to allow dynamic resizing.

Username Management via Dropdown:

Upon the user's completion of filling out a bracket, the system should capture the username and store it in a dropdown menu. This feature will allow users to retrieve and load their previously filled brackets by selecting their username from the list.

Time Estimate: 1–2 hours

Difficulty: Easy

- Capturing a String and adding it to a ComboBox is straightforward. Storing associated data might involve a map or a small file/database.

Bracket Display with Username Selection:

Upon selection of a username from the dropdown list, the corresponding user's bracket should be displayed. This ensures the system can retrieve and showcase any stored bracket associated with the chosen user.

Time Estimate: 3–5 hours

Difficulty: Medium

- You need to load user-specific data dynamically, display the bracket, and ensure that the layout updates correctly based on the data retrieved.

Score Display for Each Game:

In the user’s individual bracket, the score for each game should be clearly displayed. This will provide transparency and ensure that users can easily comprehend the rationale behind each team's victory or defeat based on the final scores.

Time Estimate: 2–3 hours

Difficulty: Medium

- Requires updating the bracket UI to show game scores next to or within game nodes, and storing that score info.

Simulation Results & Points System:

Following the simulation, the system must display the final score for each game alongside the number of points the user earned based on the accuracy of their predictions. This will incentivize users to refine their predictions and track their progress throughout the simulation.

Time Estimate: 4–6 hours

Difficulty: Hard

- You need to simulate outcomes, compare them to user predictions, and assign points based on custom logic.

Instructions Button for User Guidance:

Once the user has logged in, an "Instructions" button should be readily available. This button will lead to a concise and user-friendly guide, outlining the steps necessary to navigate and utilize the software effectively. The instructions should be as simple and intuitive as possible to ensure ease of use and avoid user confusion.

Time Estimate: 1 hour

Difficulty: Easy

- A basic dialog or pop-up with instructional text. Very little logic involved.

Bracket Comparison with 2017 Tournament:

The user's bracket should be automatically compared to the actual tournament results from 2017. Correctly predicted teams should be highlighted in green text, providing visual feedback on the accuracy of the user's choices. Each correct prediction will be visually distinguished to reflect the user's success in forecasting the outcome of individual games.

Time Estimate: 3–5 hours

Difficulty: Medium to Hard

- You'll compare user predictions with the 2017 results and update the UI accordingly, which requires both logic and visual updates.

Incorrect Predictions in Red Text:

Teams that were incorrectly predicted by the user should be highlighted in red text. If a user erroneously selects a team to win, the name of the losing team should be displayed in red, providing clear visual feedback of the incorrect prediction.

Time Estimate: 1–2 hours

Difficulty: Easy

- Reuses most logic from the previous point, just changes the color based on result comparison

---

Leaderboard Table with Sorting Functionality:

A leaderboard table should be implemented, consisting of the following columns: "User Name," "User Points," and "User Winning Team." This table should be dynamically sorted by the "User Points" column in descending order, with the user accumulating the highest points appearing at the top. The user with the most points will be recognized as the winner of the tournament, offering a competitive and transparent ranking system

Time Estimate: 3–4 hours

Difficulty: Medium

- Setting up a TableView, populating it with users and their scores, and allowing sorting (especially custom sorting) requires attention to JavaFX table configuration.

Summary Table📑

Feature	Purpose / Description	Time Estimate	Difficulty
GUI Responsiveness	UI elements resize with window for consistent layout	2–4 hours	Medium
Username Dropdown Management	Save and select usernames to retrieve brackets	1–2 hours	Easy
Bracket Display by Username	Show stored bracket based on selected username	3–5 hours	Medium
Score Display for Each Game	Clearly show final scores in user's bracket	2–3 hours	Medium
Simulation & Points System	Show simulation scores and reward points for correct picks	4–6 hours	Hard
Instructions Button	Display step-by-step user guide	1 hour	Easy
Correct Predictions in Green	Highlight accurate picks in green for feedback	3–5 hours	Medium–Hard
Incorrect Predictions in Red	Highlight incorrect picks in red for clarity	1–2 hours	Easy
Leaderboard with Sorting	Rank users by points with sortable table	3–4 hours	Medium

Improvements✅

"list improvements that would make it more user friendly"

The following features are notable improvements that can be made to the program:

Renaming the 'ScoreBoard' Button:

The button labeled "ScoreBoard" should be renamed to "Scoreboard" to maintain proper capitalization and improve the consistency of the UI language.

Time Requirement: 0.5–1 hour

Difficulty: Easy

- This is a simple text change in the UI, likely requiring minimal effort.

Simulated Bracket Layout Issue:

The simulated bracket is currently misaligned, appearing in the top-left corner instead of being centered both horizontally and vertically. The bracket should be repositioned to ensure it is displayed in the center of the screen for optimal visual presentation.

Time Requirement: 1–2 hours

Difficulty: Moderate

- This task involves UI layout adjustments, which can take time depending on how the bracket is implemented (e.g., using grid systems or absolute positioning).

Missing Score Tooltips and Display:

The simulated bracket should include score tooltips or directly display the scores for each game. This is essential to provide transparency and allow users to easily track game outcomes and the progress of their bracket predictions.

Time Requirement: 2–4 hours

Difficulty: Moderate to Difficult

- You'll need to implement logic to display game scores either as tooltips or as part of the game display. It involves working with data visualization and potentially altering the bracket structure to accommodate this feature.

Improving Documentation Presentation:

The program's documentation could be enhanced in terms of appearance and structure. A more polished format, such as exporting the documentation to PDF, will improve accessibility and readability.

Time Requirement: 2–5 hours

Difficulty: Moderate

- This requires generating a PDF version of the documentation, which may involve integrating libraries and ensuring the formatting is appropriate.

Missing 'User Winning Team' Column in Scoreboard:

The "User Winning Team" column is currently absent from the scoreboard. This column should be added to provide a clearer view of which team the user selected as their predicted winner.

Time Requirement: 1–2 hours

Difficulty: Easy

- Adding a new column to the scoreboard is a simple change that may require some updates to the database or data storage structure.

Renaming 'Total Points' Column to 'User Points':

The column titled "Total Points" in the scoreboard should be renamed to "User Points" for better clarity, ensuring the column accurately reflects the points earned by each user based on their predictions.

Time Requirement: 0.5–1 hour

Difficulty: Easy

- A simple rename operation for consistency, which should not require significant time or effort.

Display of Winner Prompt After Simulation:

After the simulation has run, the program should automatically display a prompt informing the user of the winner of the simulation. This will provide a clear and engaging conclusion to the game, offering immediate feedback to the user.

Time Requirement: 1–2 hours

Difficulty: Moderate

- This task requires some post-simulation logic to evaluate and display the winner. Depending on the existing flow, it may require some UI updates and logic integration.

Ensuring UI Scales with Program Window:

The user interface should be fully responsive, ensuring that it automatically scales with the size of the program window. This will enhance the user experience, providing a seamless interaction regardless of the window's size.

Time Requirement: 3–6 hours

Difficulty: Moderate to Difficult

- This task involves making the UI responsive and ensuring that all elements adjust correctly when the window size changes. It may involve updating the layout logic and checking various screen resolutions.

Displaying Higher-Ranked Teams with Best Odds:

The UI should highlight the teams with the best odds, emphasizing those that are higher-ranked. This will guide users to better understand the likely outcomes of games and aid them in making more informed predictions.

Time Requirement: 3–5 hours

Difficulty: Moderate

- The logic for determining and displaying the best odds may require integration with external data or additional algorithm development. The UI will also need adjustments to properly display this information.

Account Creation Screen for New Users:

Instead of using the credentials from a failed login attempt to create a new account, the program should provide a dedicated account creation screen. This screen should be intuitive, allowing new users to easily register without confusion.

Time Requirement: 4–8 hours

Difficulty: Difficult

- This involves creating a full registration process, including data validation, form design, and potentially handling user data securely. This may require database integration, depending on the system's design.

Centering and Enhancing the Scoreboard Screen:

The scoreboard screen is not currently centered, and the UI design lacks visual appeal. The layout should be adjusted to center the content properly, while also improving the overall aesthetics to create a more engaging and visually pleasing user experience.

Time Requirement: 1–2 hours

Difficulty: Moderate

- Adjusting the layout of the scoreboard screen and improving visual appeal can be done by updating layout containers, but it may require testing to ensure it appears correctly across all screen sizes.

Displaying Specifics of Each Game:

After each game, the program should clearly display detailed information such as which team won, the final score, and any other relevant details. This transparency will help users understand the progression of the simulation.

Time Requirement: 2–4 hours

Difficulty: Moderate

- This involves showing specific game details and scores, which may require changes to how data is displayed in the UI and an update to the game logic to store and display relevant information.

Retry Option After Simulation:

Once the simulation is complete, the program should offer the user an option to try again. This prompt should allow the user to either log into another account or retry with the current account, providing flexibility and encouraging further engagement with the system.

Time Requirement: 2–4 hours

Difficulty: Moderate

- The user interface and backend logic will need updates to allow users to log in again or retry without restarting the program. This also involves tracking the session state.

Improving UI Aesthetics and Spacing:

The current UI feels too spaced out and lacks visual appeal. The design should be refined to ensure a more cohesive, aesthetically pleasing layout, making the program more enjoyable and intuitive for users to interact with.

Time Requirement: 4–8 hours

Difficulty: Moderate to Difficult

- Refining the visual layout and ensuring that it looks appealing involves a good understanding of design principles. This task may include reorganizing elements, changing font sizes, adjusting colors, and possibly integrating feedback from users to fine-tune the design.

Summary Table 📑

Improvement	Purpose / Description	Time Estimate	Difficulty
Rename "ScoreBoard" to "Scoreboard"	Corrects capitalization for UI consistency	0.5–1 hour	Easy
Fix Simulated Bracket Layout	Centers bracket for better visual alignment	1–2 hours	Moderate
Add Score Tooltips/Display	Show game scores on bracket for transparency	2–4 hours	Moderate–Difficult
Improve Documentation Format	Export to PDF for better readability	2–5 hours	Moderate
Add "User Winning Team" Column	Shows user's predicted champion in scoreboard	1–2 hours	Easy
Rename "Total Points" to "User Points"	Increases clarity of scoreboard column label	0.5–1 hour	Easy
Display Winner Prompt After Simulation	Informs user of simulation winner clearly	1–2 hours	Moderate
Make UI Responsive to Window Size	Ensures UI adjusts with window size	3–6 hours	Moderate–Difficult

Highlight Best Odds Teams	Visual emphasis on likely winners	3–5 hours	Moderate
Create Account Registration Screen	Simplifies user onboarding with a separate screen	4–8 hours	Difficult
Center & Style Scoreboard Screen	Improves visual layout and UI alignment	1–2 hours	Moderate
Display Game Details (Winner, Score)	Adds transparency after each game	2–4 hours	Moderate
Add Retry Option After Simulation	Lets users easily retry or switch accounts	2–4 hours	Moderate
Improve Overall UI Aesthetics/Spacing	Makes layout cleaner and more user-friendly	4–8 hours	Moderate–Difficult

---

🧭EXTRA Project Roadmap to Recovery: Feature Implementation & UI Improvements

---

🔹 Phase 1: Foundational UI & Usability (Week 1)

Focus: Set up the base features that improve user experience and consistency.

Task	Est. Time	Notes
✅ Rename “ScoreBoard” to “Scoreboard”	0.5–1 hr	Quick UI fix

✅ Rename “Total Points” to “User Points”	0.5–1 hr	Improves clarity
✅ Username Dropdown Management	1–2 hrs	Enables user tracking
✅ Bracket Display by Username	3–5 hrs	Ties into saved data
✅ Add “User Winning Team” Column	1–2 hrs	Expands leaderboard info
✅ Instructions Button	1 hr	Help users get started easily

📌 Goal: Basic usability, input tracking, and visual clarity.

🔹 Phase 2: Core Simulation Features (Week 2)

Focus: Build the actual functionality that powers the game logic.

Task	Est. Time	Notes
✅ Score Display for Each Game	2–3 hrs	Adds game result transparency
✅ Display Specifics of Each Game	2–4 hrs	Deeper info per game
✅ Simulation Results & Points System	4–6 hrs	Core simulation + scoring logic
✅ Display Winner Prompt After Simulation	1–2 hrs	Engaging feedback
✅ Retry Option After Simulation	2–4 hrs	Improves user flow

📌 Goal: Get the simulation engine and feedback loop fully functional.

🔹 Phase 3: UI Responsiveness & Layout Fixes (Week 3)

Focus: Make the application feel polished on different screens and improve layout alignment.

Task	Est. Time	Notes
✅ GUI Responsiveness	2–4 hrs	Layouts resize dynamically
✅ Ensure UI Scales with Program Window	3–6 hrs	Restructure containers
✅ Fix Simulated Bracket Layout	1–2 hrs	Bracket centered visually
✅ Center and Enhance Scoreboard Screen	1–2 hrs	Clean up final leaderboard view

📌 Goal: Improve layout presentation and adaptability.

🔹 Phase 4: Advanced UI Feedback (Week 4)

Focus: Add visual feedback and dynamic insights to help users make better decisions.

Task	Est. Time	Notes
✅ Highlight Correct Predictions in Green	3–5 hrs	Instant visual validation
✅ Highlight Incorrect Predictions in Red	1–2 hrs	Visual differentiation
✅ Add Score Tooltips or In-Game Score Display	2–4 hrs	Optional hover-based or direct score view
✅ Highlight Best Odds Teams	3–5 hrs	Requires ranking logic or data integration

📌 Goal: Inform users with dynamic, insightful visual cues.

🔹 Phase 5: Aesthetic & Final Touches (Week 5)

Focus: Polish the design, documentation, and onboarding flow.

Task	Est. Time	Notes
✅ Improve Overall UI Aesthetics and Spacing	4–8 hrs	Final design pass
✅ Improve Documentation & Export to PDF	2–5 hrs	Helpful for users/instructors
✅ Add Account Creation Screen	4–8 hrs	Optional: Requires new form + validation

📌 Goal: Final polish + better onboarding and documentation.

📅 Suggested Weekly Plan

Week	Focus
Week 1	UI labels, dropdowns, bracket selection, instructions
Week 2	Game simulation, score displays, retry + winner prompt
Week 3	Responsive layouts, centering, scoreboard polishing
Week 4	Correct/incorrect prediction feedback, odds highlighting
Week 5	Final aesthetics, documentation, optional account creation

---