Sprint O' Clock

Senior Design II

Cesar Frayre, Lauren Bryant Eyum, Kosuke Satake, Gin Sang, Shaheen Nijamudheen

UTA Department of Computer Science and Engineering

SCAN ME

Background

Sprint O' Clock is a mobile application designed for sprinters and fitness enthusiasts, offering a comprehensive platform to track, analyze, and improve running performance. The app is available on both iOS and Android platforms, providing a wide range of features that cater to runners of all levels, from beginners to experienced athletes. Users can set personalized goals and monitor their progress with detailed metrics. The app's robust features include real-time GPS tracking, which allows runners to accurately record their routes, distances, and pace. Users can customize settings, including enabling phone vibration and feedback audio. Sprint O' Clock also offers in-depth performance analytics, providing insights into key metrics such as distance covered, steps, and calorie burn. Users can visualize their progress over time through progress graphs, helping them identify strengths and areas for improvement. Sprint O' Clock also integrates with popular wearable devices and health apps, ensuring a seamless experience for users who rely on multiple tools to track their fitness journey. With regular updates and new features, Sprint O' Clock continuously evolves to meet the needs of its users, making it an essential tool for anyone serious about improving their running performance. Our app's back end also ensures seamless performance and security, handling data tracking, API management, and user data protection.

Key Requirements

1. Randomized Start Time Feature :
- Designed to test a sprinter's reaction time rather than using a traditional countdown timer.
- Start Button on Home Page: - Track the distance of the run as soon as the start button is pressed.
- Run stops either when the stop button is pressed or when the pre-set distance goal is reached.

2. Accurate Run Tracking :
- Calculate the correct distance covered during the run using GPS.
- Record accurate time in minutes and seconds.

3. Run History Recording :
- Store data for each run session, including start time, end time, and duration.
- Distance covered (tracked via GPS or user input) and store history of runs with respective finish times in a calendar interface.

4. Customizable Run Settings :
- Phone Vibration: - Start vibration as soon as the start button is pressed.
- Animation and Audio Sync: Sync animation with the audio cues "Ready… Set… Go".

5. User Sign-In and Profile Sync :
- Allow users to sign in and sync their profiles across both iOS and Android platforms.
- Store user profiles including preferences, such as vibration settings.

Design Constraints and Engineering Standards

Design Constraints

- **Cost**: The project budget is limited to $800, covering essential equipment and cloud storage. Careful planning and resource management are required to prioritize essential features and find cost-effective solutions within this budget.
- **Schedule**: The project must be completed between Spring 2024 and Summer 2024. A structured development plan with clear milestones and regular progress reviews is essential to avoid delays and ensure timely completion.
- **Aesthetics**: The app must have a visually appealing, user-friendly design with consistent branding and engaging visual elements. A well-designed interface enhances user experience and engagement without compromising functionality.
- **Satellite Accuracy**: The app's performance depends on accurate GPS satellite data for precise distance tracking. Ensuring reliable distance measurements despite variations in satellite data accuracy is crucial for data calculations.
- **Interoperability**: The app must function seamlessly across various mobile devices and platforms, including iOS and Android. Ensuring compatibility with different screen sizes, operating systems, and hardware configurations through extensive testing is essential for a consistent user experience.

Engineering Standards

- **Authentication & Security Standards**: The app must use secure authentication methods to protect user data, including encrypting sensitive information. Secure login procedures must be implemented to prevent unauthorized access.
- **Coding Standards**: Adherence to best practices in software development, such as consistent coding standards and thorough testing, ensures the app is reliable and maintainable. Following industry-recognized coding guidelines helps create clean, efficient, and bug-free code.
- **User Interface Design Standards**: The app should follow established UI/UX guidelines to create an intuitive and user-friendly interface. This ensures the app is accessible and easy to navigate, enhancing the overall user experience.
- **Data Privacy Compliance**: The app must comply with data protection regulations to ensure responsible handling of user data. Users should be informed about data usage, and consent must be obtained for data collection and storage.
- **Cross-Platform Compatibility Standards**: The app must be optimized to run efficiently on both iOS and Android platforms. Ensuring compatibility with different screen sizes, operating systems, and hardware configurations is essential for a consistent user experience across devices.

System Overview

Front-End Layer

- **User Interface:** The front-end is responsible for the user interface (UI) and user experience (UX) design. It includes visual elements and interactive components that users interact with, such as the "On your marks… Set… Go!" timer, run tracking screens, historical data displays, and settings menus. The front-end is developed using React Native to ensure cross-platform compatibility on both iOS and Android devices.

- **Input Handling:** The front-end captures user inputs, such as starting and stopping the timer, entering user details, and navigating through the app. It processes these inputs and communicates with the back-end to perform necessary actions and updates.

Back-End Layer

- **Business Logic:** The back-end handles the core functionality of the app, including the implementation of the randomized start timer, calculation of distance and time, and processing of user data. It ensures that all business rules and logic are correctly applied.

- **API Management:** The back-end provides APIs that facilitate communication between the front-end and the database. These APIs handle requests for user authentication, data retrieval, and data storage, ensuring secure and efficient data exchange.

Database Layer

- **Data Storage:** The database layer is responsible for storing all user-related data, including run history, user profiles, and performance statistics. It uses a secure and scalable database solution to manage this data.

- **Data Retrieval:** The database layer retrieves data as requested by the back-end, providing necessary information for display and analysis on the front-end. It ensures data consistency and integrity across the app.
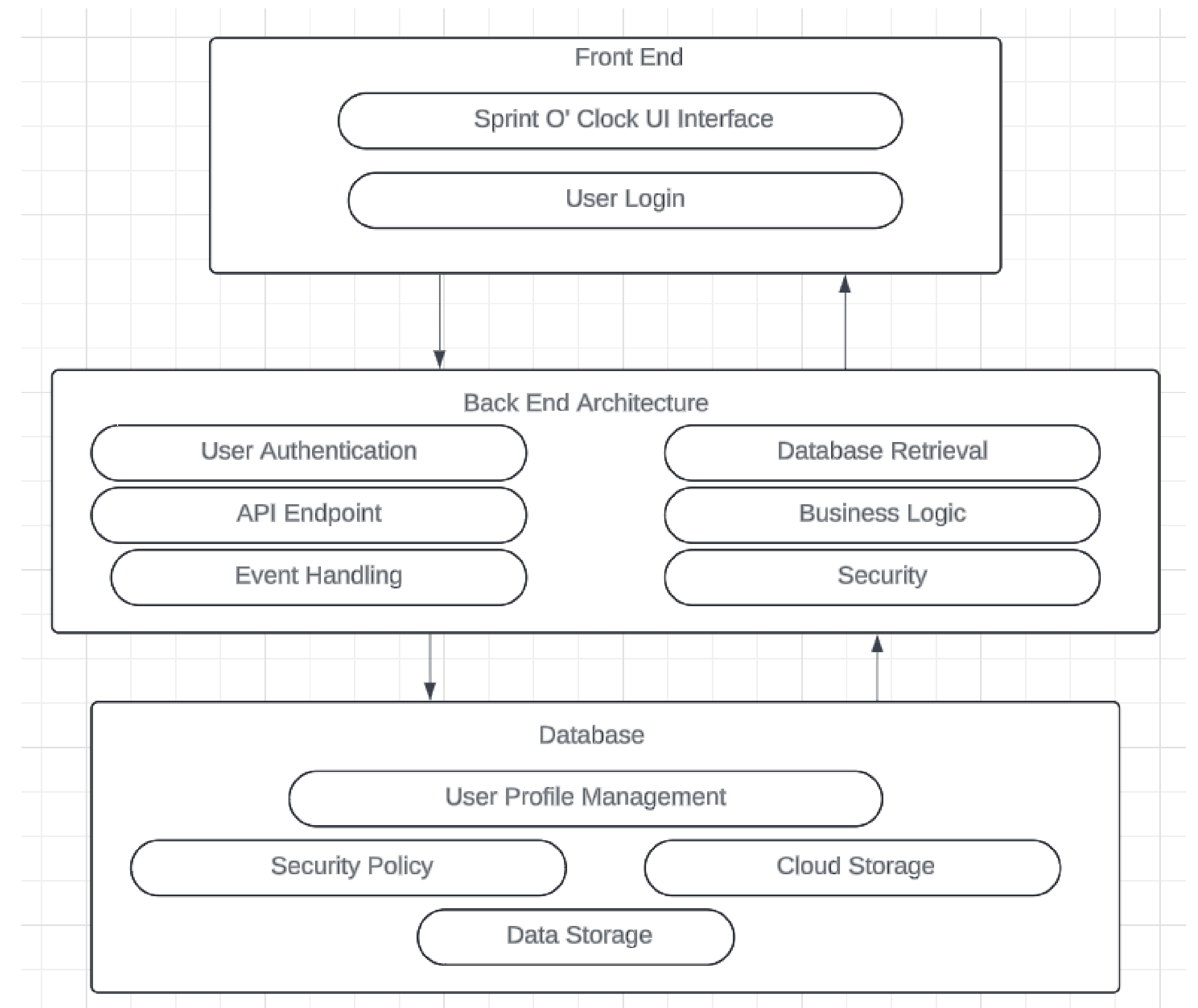
Conclusions and Future Work

Due to time constraints, we were unable to implement a smartwatch feature for distance tracking, necessitating the use of a phone instead. Additionally, the competition feature of the app remains incomplete, and the distance tracking accuracy is limited to approximately 10-20 meters due to satellite limitations. However, we were able to meet all our key requirements for the app. In the future, we plan to develop the smartwatch component to enhance data collection, including heart rate monitoring. This will provide a more convenient experience for runners, allowing them to wear a smartwatch instead of carrying a phone.

References

- National Coordination Office for Space-Based Positioning, Navigation, and Timing.
  "GPS Accuracy." *GPS.gov*, 3 Mar. 2022, https://www.gps.gov/systems/gps/performance/accuracy/.
- Meta Platforms, Inc.
  "Showcase." *React Native*, 2024, https://reactnative.dev/showcase.
- Ahmed, Kamran.
  "Developer Roadmaps." *roadmap.sh*, 2024, https://roadmap.sh/.