Project Track1 Checkpoint2 Discussion

Team008
Muzi Peng(muzip2), Weilong Li(weilong3), Rutuja Narwade(narwade2), Zhuofan Zeng(zz115)

1 Implementation Challenges

1.1 Detailed Feature Requirements

The first challenge we meet is how to design real-life functionalities that satisfy requirements of Transaction (2 advanced queries), Procedure (2 advanced queries) and Trigger. Our application satisfies all these requirements, as is clearly shown follows:

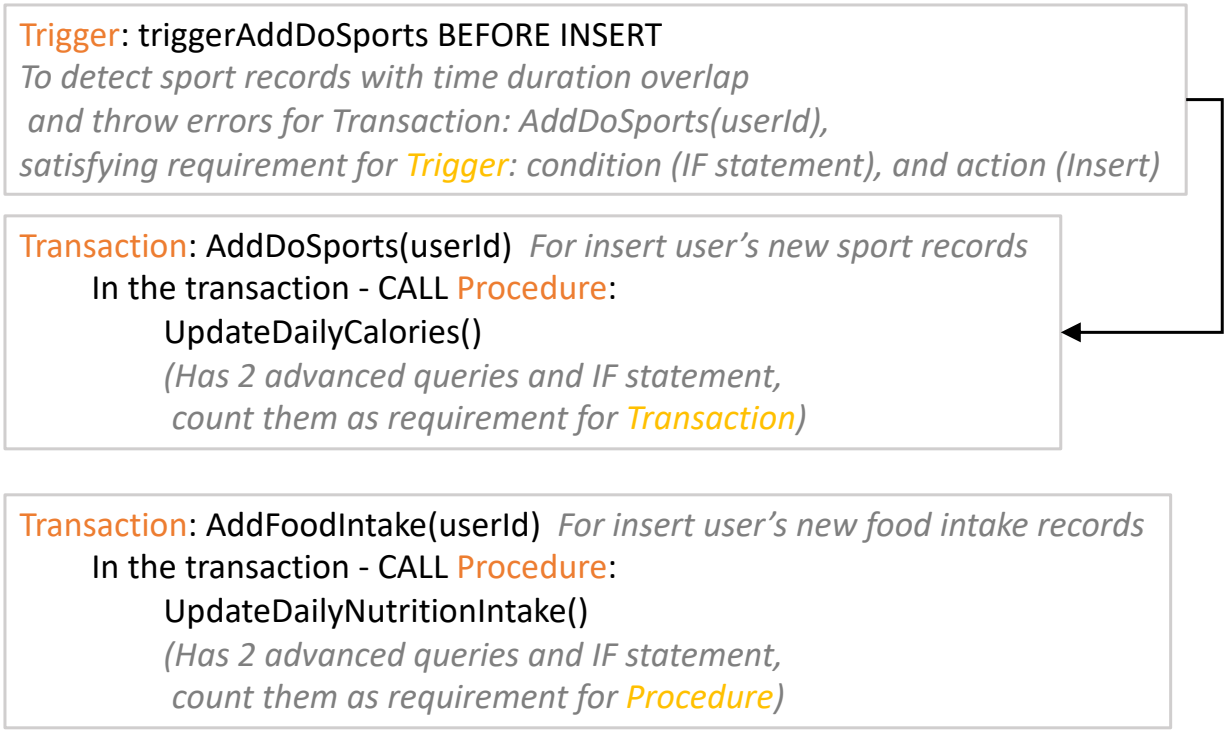

Figure 1: Outlines of our trigger, transactions and procedures, which satisfy all detailed feature requirements.

1.2 Creative Components

The second challenge is to make creative components work well. We have two creative components: 1) Dynamic pie chart to summarize user's intake for different food categories in a chosen period. 2) Dynamic line chart to track user's calories intake and burning in a chosen period. All these two creative components require advanced queries, and require backend to keep maintaining user's daily nutrient intake, category intake and sports tables, specifically, to UPDATE these tables whenever there are new records of intake and sports INSERT. Besides, these two creative components require complex transmission data types, and user-friendly interfaces.

Because of limited demo time duration, and an unexpected accident when we show delete function, we record a 1-minute supplementary video to show our delete function and our second creative component. Here is the video link. We would really appreciate if you could see this short video!

1.3 Connecting Backend APIs with Frontend

The third challenge is to properly connect backend APIs with frontend. Especially, during the initial stage of our development, we frequently meet problems of backend cannot receive or parse values from frontend, or the frontend cannot get response from the backend. Fortunately, we gradually find an efficient workflow. After we develop a backend API, we will firstly use Postman API platform to test it, ensuring it respond properly when receive requests. After that we develop the frontend based on exact data transmission results shown in Postman. We find this workflow is much more efficient than briefly determin which data will be transferred and then let members develop frontend and backend separately.

2 Design Deviations

Initially, we plan to design more features to facilitate the use of healthTrack, some of them are implemented while some of which are not, but we add some amazing functions that were not initially envisioned.

- **nutriTrack**: The current implementation of nutriTrack is quite complete, we have given calculations on various foods, nutritional intake to help users track their nutritional intake, and added creative components to further visualize the results of nutriTrack. These features are due to our continuous exploration and refinement during the development process, we stand on the user's point of view to think about how our features should be designed, how can we be more minimalist. Back to the initial design concept is just to realize the record of food eaten by the user and nutritional quantity calculation, the function is still a bit limited and not flexible enough. However, in our final version, we implemented multiple components to display the results, and realized more features about the nutritional content of the food, as well as a very beautiful page.
- **Sport Search**: Regarding sportTrack's functionality, we have now implemented the addition of sports, time tracking and the use of line graphs to visualize the user's movements. We have high expectations for this feature and hope to continue to improve the interface so that users can automatically add sports to the track by timing them while they are exercising to make the process easier. The original idea was to allow the user to enter the type of exercise and time to help track their workout history. As the development progressed, we expanded and refined the functionality to show a wide range of exercise types, as well as the amount of calories burned per hour of exercise to help users understand their exercise consumption in more detail. By calculating the amount of exercise consumed, users can have a clear reference to what they should be supplementing with, which is a very meaningful feature. By combining exercise consumption with the CaloriesTrack feature, users have a clear picture of how much they have consumed in a given period of time, which complements the functionality of the entire program.
- **scheduleTrack**: One of the first features we designed was scheduleTrack, but after careful consideration, we decided to remove it. While we provide a great schedule for users to help them plan their nutrition, exercise, and routine, adherence was an issue and we couldn't have a direct impact on users through this feature alone. We also decided to remove this feature for the time being because it was causing users to feel stressed and uncomfortable when using our product.

3 Potential NoSQL Integration

Integrating NoSQL solutions into our HealthTrack database could offer significant flexibility and performance advantages, particularly for handling semi-structured or unstructured data, and for scenarios where read and write speeds are critical.

- **Document Stores for User and Activity Data:** A NoSQL document store like MongoDB could be used to manage user, sportTrack, and nutriTrack data. These document stores excel in handling varied and complex data structures which can evolve over time. For instance, each user document could embed the details of sports activities and nutritional intake directly, simplifying queries related to a user's daily or historical activities.

- **Wide-Column Stores for Nutritional Data** A wide-column store like Cassandra could be effective for handling the food and contains tables, where each row can store a vast number of dynamic columns. This would be useful for foods that have a wide range of nutrient measurements, allowing for efficient storage and querying of these items based on nutrient content.
- **Graph Database for Relationships:** Relationships such as those between users, their diets (nutriTrack), and their physical activities (sportTrack) can be efficiently modeled with a graph database like Neo4j. This would allow complex queries on relationships, like finding connections between users' dietary patterns and their physical activity levels or the influence of specific foods on health outcomes.
- **Key-Value Stores for Session Management:** Redis, a key-value store, could be used for managing user sessions and temporary data with high availability and fast access. This would be particularly useful for features like tracking real-time updates during user activities or managing quick access to user preferences.