Softies Milestone 2

CSCI 3308 Group 107-3 Development Project
Madison Yost, Daniel Koris, Xinguan Tang, Dominic Pontious, Wyatt Richards

Project Features List

- User Sign-In/Sign-Up
    - Landing screen where the user either logs into their existing account or chooses to create an account
- Sign-Up onboarding
    - Get basic information from the user to set up their account
- Predetermined Workout Plan Template (Weekly)
    - Provides several sets of instructions for popular workouts
- Customized Workout Plan Template (Weekly)
    - Allows users to create and revisit individualized workout plans
- Weekly Activity Tracker (Calendar)
    - Display this calendar week's planned/completed workouts in a calendar format
- Awards
    - Users earn badges and earn associated points when they accomplish a goal
- Leaderboard
    - Based on the points earned from awards users see where they stack up against other users
- Workout Summary
    - User can see a graph of their fitness progress over monthly or yearly timeframes
- User Personal Info
    - Keeps track of weight, height, age calculates BMI
- Kilogram/lbs converter
    - Converts kg to lbs or lbs to kg based on users request
- UI-customization (colors etc)+
    - Allows the user to select a different site theme
- Push notifications (email/text/google calendar)
    - Sends a notification to the user at the beginning of the week showing them their upcoming workouts or the day of a scheduled workout

Requirements

Feature: Sign In/Up

Functional:

Takes user input and checks for an existing fitness profile. If one doesn't already exist, it will prompt the user to go to the sign-in onboarding feature to input a username, password, email, age, and weight in order to create one. Users will be able to revisit this profile by signing in with their username and password.

Non-Functional:

The provided user information will be stored in a user database which will be checked against the name and encrypted password. If there is no such user name in the database the user will be prompted to go to a form where they will create an account. The form will include validation with the database to ensure a unique username.

Feature: Weekly activity tracker

Functional:

Displays information about the current weeks workout information in a calendar view. Lets the user see the workout details and whether or not the workout has been completed. The user can mark workouts as completed or add new workouts to their schedule.

Non-Functional:

Exercise information will be stored in a database table with the date and the information about the workout. Using the dates, this data will be accessed and displayed in the UI in a card on the date that the user plans to do that workout. When the user marks a workout as complete the database will be updated with this new information.

Feature: Merit/award system

Functional:

Rewards the user with badges for completing pre-set goals and assigns points based on those goals that are applied to the user's account and help them level up in the leaderboard.

Non-Functional:

All of the possible awards will be stored in a database and when the requirements for that reward are met by the user the award gets added to their account via a boolean value. On the awards page the user's earned awards and possible awards are displayed.

Feature: Predetermined Workout Plan Template (Weekly)

Functional:

Lets the user choose from different workout styles or focuses and shows them different predetermined workouts that they can choose to add to their workout schedule.

Non-Functional:

Several workouts will be created and stored in a workout database, and accessible to every user. This will be displayed on the front-end to make selection of the generic exercise routine easy.

Feature: Customized workout plans

Functional:

Lets the user input their own workout name and exercises that they can add to their weekly workout schedule.

Non-Functional:

There will be a button or link on the front-end which will prompt the user to create their own workout. Each exercise will be either based on: repetitions x sets x weight, or time. Providing the information into the front-end prompt will save the information associated with the user into a table to the present the information to the user at a later time.

Feature: Monthly/Yearly Summary

Functional:

Displays information from past completed workouts in a graph view so that the user can see their improvement over time. The user can choose to see a monthly or yearly summary and filter based on workout/exercise type.

Non-Functional:

Pull information based on exercise name from the database to be compared against the same exercise name from dates with the same year or month marker.

Have the information sorted chronologically and then after pulling the information generate a graph to allow the user to visually compare their exercise progress.

Project Plan (Kanban link)

Phase One: Design

Sprint One: 2/26 - 3/5

- Create wireframes for UI design (Madi)
- Create database chart for initial database implementation (Dom)
- Milestone 2 Document
    - Comprehensive Feature List (Team)
    - Feature Functional list (Wyatt, Scott)
- Git / GitHub Documentation + Kanban Setup (Dan)

Sprint Two: 3/5 - 3/12

- Merit / Award System Documentation (Wyatt, Scott, Dan)
- Come up with a preliminary list of needed function calls for integration layer (Madi, Dan)
- Milestone 2 Document
    - Project Plan (Dan, Wyatt)
- Team will discuss present design implementations and make final decisions allow them to move completely into *Phase Two* (had to pre-emptively start phase 2 due to lab interview, see continued *Sprint Two* under *Phase Two: Proof of Concepts* section).

Phase Two: Proof of Concepts

Sprint Two(continued): 3/5 - 3/12

- Implement a non-functional version of UI (Madi)
    - Plan is to use HTML, CSS/Bootstrap, and possibly JQuery to create…
        - Login Page
        - Registration Page
        - Landing Page (comes after either login or registration)
    - These pages don't need to be functional (connecting to database, etc).
    - Pages will give perspective on our user experience and give an in-browser perspective of how our app will be to use.
- Implement database init scripts using bash and postgres (Dom)
    - Postgres script will drop any existing database and upload a clean version

    - Bash script will use calls to the postgres, and above init scripts, to pull out user specific data, run database init scripts, then re-import the sensitive user specific data.

Sprint Three: 3/12 - 3/19

- Program in JavaScript / NodeJS the basics of our integration layer as outlined in Phase one (Dan, Wyatt, Scott)
    - Each developer on this task will take a subset of functions to write
- Milestone Documentation 4 (due: 3/16):
    - Revised Features List (Wyatt, Scott)
        - Team believes that it will require minimal effort to revise our already brilliant list of features.
    - Architecture Diagram (Dan)
        - This diagram will be done and labeled using a single PowerPoint slide exported as png.
    - Front End Design (Madi)
        - Team is hopeful this will only be require minor tweaking and digitizing of our already excellent wireframes from *Sprint One*
    - Web Service Design
        - No plans to implement as of now. However, the document should clearly state this fact in the appropriate section.
    - Back End Design (Dom)
        - Team is hopeful this will only require minor tweaking and digitizing of our already amazing database diagrams from *Sprint One*
- Examine and tweak possible UI and Database interactions and testing strategies (Madi, Dom)
    - This will be primarily manual analysis and notes that will feed into *Phase Three*.
    - The ideas on testing plan will directly feed into Milestone 5 (due: 4/7)

Phase Three: Connecting Phase

Sprint Four: 3/19 - 3/26

- Implement our UI using NodeJS (Madi, Dan)
- Begin hooking present integration layer and database together (Dom, Wyatt, Scott)
    - Implement the login / registration feature as a means of making sure our NodeJS routing path design works
- Use notes from *Sprint Three* to begin writing system acceptance scripts (Madi, Dan, Dom, Wyatt, Scotti)

Phase Four: First Release

Sprint Five: 3/26 - 4/9 (Two weeks)

- Implement our minimum feature requirements within the UI framework (Team)
    - Each team member will be responsible for doing their bits of the HTML to support the feature they take complete of for this phase.
- Finish database implementation that supports our present 6 features (Team)
    - Each team member will be responsible for doing their bits of SQL to support the feature they take ownership of for this phase.
- Write supporting javascript for features
    - Predetermined workout javascript calls (Wyatt)
    - Weekly activity javascript calls (Scott)
    - Merit / Award system (Dan)
    - Customized Workout Plan (Dom)
    - Monthly / Yearly Summary (Scott, Wyatt)
- Milestone 5 (due 4/7)
    - Architectural overview of the testing plan (Madi)
        - When tests will run.
        - What software will be used to run tests.
    - Outline specific features to be tested and test cases (Dan)

Sprint Six: 4/9 - 4/16

- Systems Level Testing (Testing software TBD)
    - Implement testing for Login / Register system (Madi)
    - Implement testing for Weekly activity tracker (Scott)
    - Implement testing for Merit / Award system (Dan)
    - Implement testing for Customized Workout Plan (Dom)
    - Implement testing for Monthly / Yearly summary (Scott, Wyatt)
- Begin working on Milestone 6, the PowerPoint presentation (due: 4/26)
    - Template design for presentation (Madi, rest of team, but really Madi)
    - Title slide (Madi)
    - Team member slide (Madi)
    - Tools slides
        - HTML/CSS/Bootstrap/JQuery (Madi)
        - Kanban (Dan)
        - Git and Github (Dan)
        - NodeJS (Wyatt)
        - Postgres (Dom)
        - Unit/Integration Testing (Scott)
    - Challenges encountered slide (Scott)
- Begin practicing Demoing the App (Scott)

Sprint Seven: 4/16 - 4/23 (The Final Sprint)

- Fix any bugs found during testing from *Sprint Six* (All team members)
- Finish Milestone 6 and practice for demoing the application that was started during *Sprint Six*
    - Same assignments as *Sprint Six*