

CS3354 Software Engineering

Final Project Deliverable 2

Gym Rats

Poorna Bharanikumar, Yvonne Hsiao, Mollik Rahman, Naufal Sajjad, Muhilan Selvaa, Temi
Badmus, Jacy Grossman

1. Delegation of Tasks

Poorna Bharanikumar - Project Scheduling, Presentation, UI design

Yvonne Hsiao - Staffing, Presentation, UI design, Deliverable 1 Content

Mollik Rahman - Comparison of project with similar projects & reference page

Naufal Sajjad - Comparison of project with similar projects & reference page, Presentation

Muhilan Selvaa - Ran & documented results of test plan, conclusion

Temi Badmus - Created test plan, Function Point Estimation

Jacy Grossman - Cost estimation, Effort estimation

2. Project Deliverable 1 Content:

1. Final Project Draft Description

Group members: Poorna Bharanikumar, Yvonne Hsiao, Mollik Rahman, Naufal Sajjad, Muhilan Selvaa, Temi Badmus, Jacy Grossman

What are we doing: Fitness App/website

Description:

Many people do workouts that aren't properly balanced leading to injury and not working out the whole body. They also find it hard to be consistent when traveling or following online workout plans that use equipment they don't access too. Therefore, an app that makes custom workouts based on users' goals and what equipment they have access to. The app continuously changes and adapts based on users progress by tracking your workouts.

Motivation:

Our motivation as fitness aficionados, an app that provides a variety of utility would be greatly appreciated, as such a comprehensive app does not exist on the market.

Task Delegation:

Poorna - I will do anything related to our backend including anything related to the actual software implemented in a project of this kind. Additionally I will help with the research and analysis of our product to ensure the data we collect is reflective of our goals with our app.

Naufal - I would like to work on UI/UX and Frontend design, while helping with market research, requirement analysis, and ensuring the app meets certain standards.

Mollik - I will be doing anything related to the research needed to develop a fitness app that meets the suggested requirements. I can also help anywhere else that is needed in the development of this project.

Temi - I will be doing anything related to research and implementation of fitness, muscle groups, and workouts based on the user. I will also assist in backend design for the app.

Yvonne - I will be doing anything related to software demo (UI/UX) as well as research includes user research and competitive research.

Muhilan - I will be working on diagram creation as well as doing the J Unit unit testing for this app. I will also assist in cost analysis and planning.

Jacy - I would like to contribute to the front-end and/or back-end development, wherever additional personnel would be most helpful. I can also assist with requirement analysis.

Scholar Paper: Maybe, still figuring out commitment.

Feedback Received: Great project topic with a fringe benefit, as it will help with maintaining a healthy lifestyle. In the final report, please make sure to include comparison with similar applications -if any- and make sure that you differentiate your design from those and explicitly specify how. Please share this feedback with your group members. You are good to go. Have fun with the project and hope everyone enjoys the collaboration.

Addressing feedback: The feedback we received was to look into similar applications, compare them, and differentiate our design while explaining how it is different. In order to do this, we are looking into similar workout and diet apps such as Strong and Sworkit, in order to see what features they provide. We will then add new features or combine features from multiple apps in ways that have not been done before in order to create a more comprehensive app. Firstly, the Strong app only contains certain preset workouts/routines, which makes our app unique, since we are generating a workout for the user based on available equipment and preferences. Next, the Sworkit app is a really great app which provides a variety of plans, however, it only provides workout plans for at home workouts. This problem is solved in our app, which incorporates body weight and resistance training with weights into the workout. Lastly, many nutrition apps, such as MyFitnessPal, do not include workout plans, so our app encompasses workouts combined with nutritional information. Currently, there is not an app as comprehensive as ours in terms of overall health, as they either specialize in strength training, at home training, or nutrition, making our app a great addition to the category.

Title: The title of our project is Gym Rats. It is a mobile app that creates workout plans based on user preferences and available equipment.

2. Delegation of Tasks

Jacy - Software requirements (functional and non-functional), Architectural design pattern

Naufal - Title, Delegation of tasks, Addressing project feedback, Software process model,

1.3-inviting everyone to the github

Mollik - Title, Delegation of tasks, Addressing project feedback, Software process model, 1.5 commit project_scope file

Temi - Sequence diagram, Class diagram

Muhilan - Sequence diagram, Architectural design pattern

Poorna - Software requirements (functional and non-functional), Use case diagram. 1.4-first commit to the repository, readme file

Yvonne - Use case diagram, Sequence diagram

3. Software Process Model

The software process model employed in our app is the spiral model. The spiral model works well here because of its inherent flexibility and adaptability. The spiral model combines the iterative aspects of prototyping with the controlled nature of the waterfall model. The model is evolutionary, meaning there are a series of releases, like prototypes, throughout the process. This allows for changing requirements to be added and tested thoroughly, increasing the quality of the software. The requirements of our fitness app are subject to change based on user feedback, emerging trends, and technological advancements. The spiral model's iterative nature allows for continuous refinement and enhancement of the app throughout its lifecycle. It allows us to gradually incorporate new features, accommodate evolving user preferences, and address any issues. The model also emphasizes risk management, which is important for maintaining data security and user privacy in our app. By iterating through planning, design, build, and test phases while assessing risks, the spiral model ensures that our fitness app can evolve with the ever-changing health and fitness scene. Overall, the spiral model is the model that best fits our app because it offers great flexibility to changing requirements and plenty of risk management throughout the development process, which are important for mobile applications.

4. Software Requirements

Functional Requirements:

1. User Registration and Profile Management:

Users should be able to create an account and provide essential information such as age, gender, fitness level, and fitness goals.

Users should have the option to edit and update their profiles as needed.

2. Personalized Assessment:

The app should conduct an initial assessment of the user's fitness level, taking into account factors like BMI, medical conditions, and dietary restrictions.

3. Workout Customization:

Users should be able to select their preferred workout type (e.g., strength training, cardio, flexibility) and specify their target areas (e.g., upper body, lower body, core).

The app should recommend workouts tailored to the user's preferences and fitness goals.

4. Progress Tracking:

Users should be able to track their progress over time, including weight, measurements, and workout performance.

The app should use this data to adapt and adjust the workout plan accordingly.

5. AI Workout Recommendations:

The AI should suggest workout routines based on the user's goals, fitness level, and progress.

It should consider factors like exercise intensity, frequency, and duration.

6. Exercise Library:

The app should provide a comprehensive library of exercises with video demonstrations, instructions, and variations.

Users should be able to filter and search for exercises based on their preferences and equipment availability.

7. Nutritional Guidance:

Provide dietary recommendations and meal planning based on the user's goals (e.g., weight loss, muscle gain, maintenance).

Allow users to log their daily food intake for tracking purposes.

Non-Functional Requirements:

Product Requirements

Usability Requirement:

The software shall have user interface elements that are clear and explicit.

Efficiency Requirements

Performance Requirement:

The software shall have a two second maximum response time.

The software shall support a minimum of 250,000 concurrent users.

Space Requirement:

The software shall take up no more than 150MB on mobile devices.

Dependability Requirement:

The server shall be available to clients 99.5% of the time.

Security Requirement:

The software shall not expose personally identifiable information to unauthorized parties.

Organizational Requirements

Environmental Requirement:

The software shall maintain a unique directory to avoid interference with other software.

Operational Requirement:

The client shall run on Android.

The client shall run on iOS.

The client shall run on WearOS.

Development Requirement:

Development of the software shall take no more than two years.

External Requirements

Regulatory Requirement:

Parts of the software that handle transactions shall be PCI compliant.

Ethical Requirement:

The software shall not recommend unsafe exercises.

Legislative Requirements

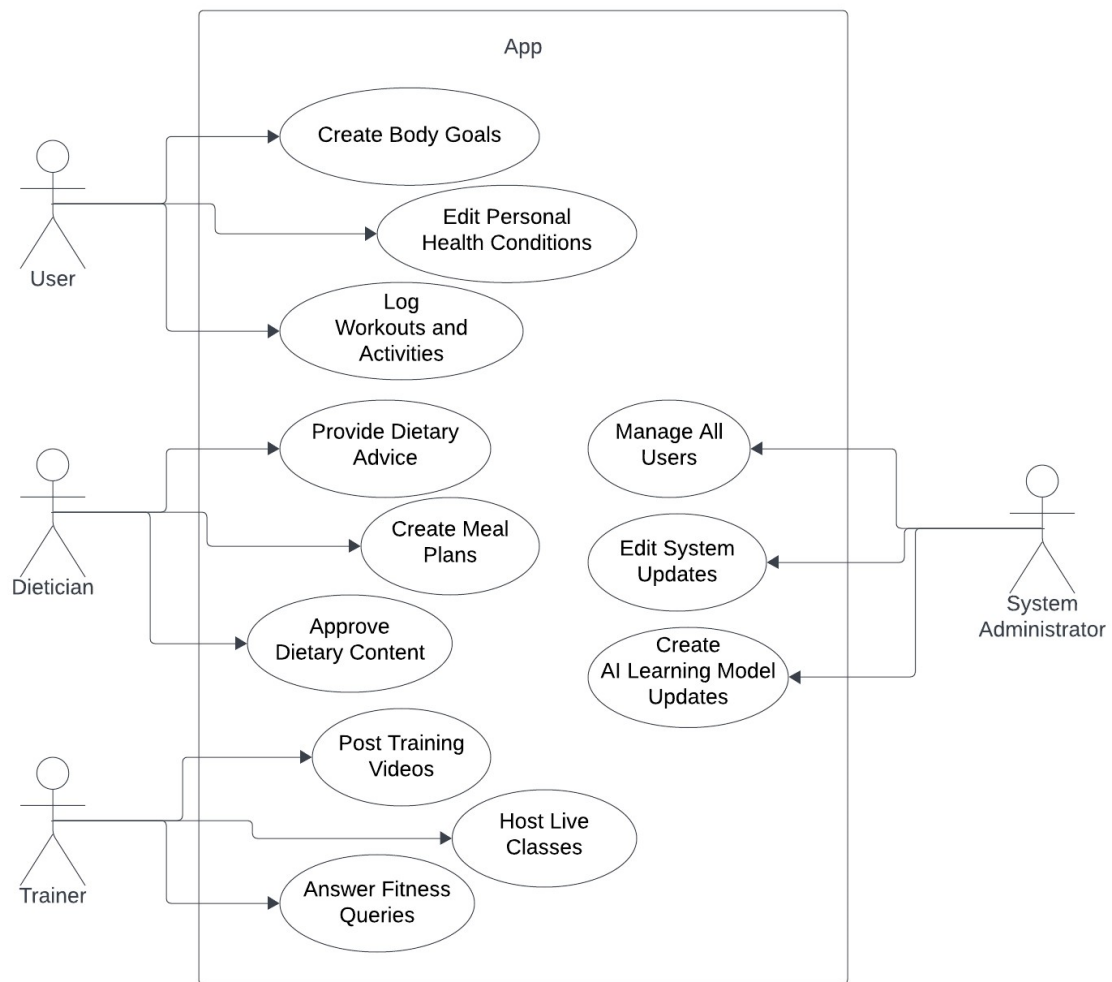
Accounting Requirement:

The software shall be compliant with the Sarbanes-Oxley Act.

Safety/Security Requirement:

The software shall not offer features subject to legislative safety requirements.

5. Use Case Diagrams



6. Sequence diagrams

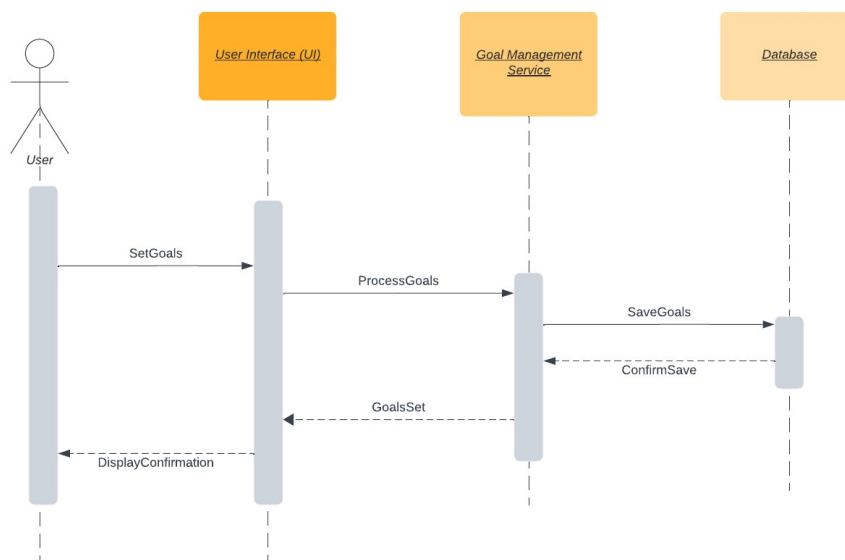


Figure 6.1 Sequence diagram for creating body goals

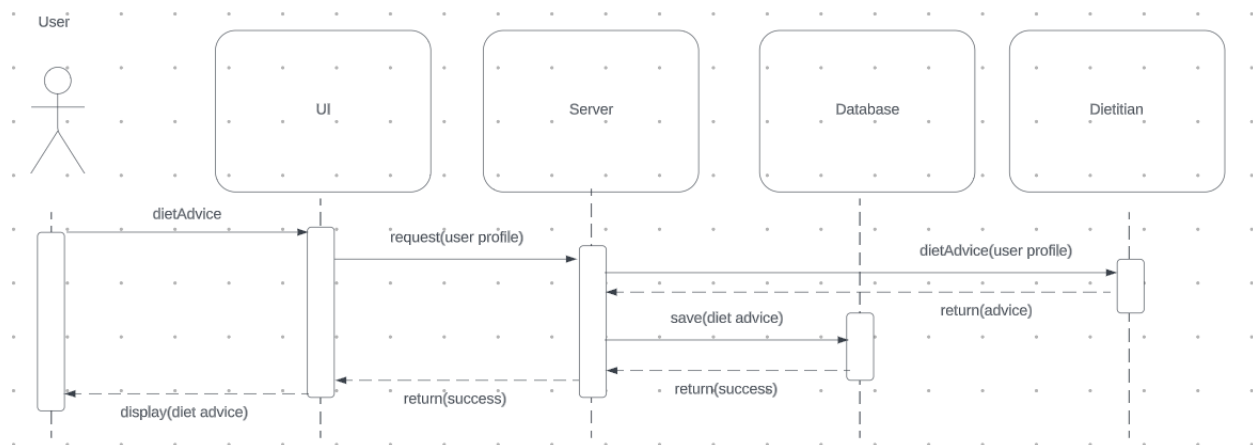


Figure 6.2 Sequence diagram for dietitian advice

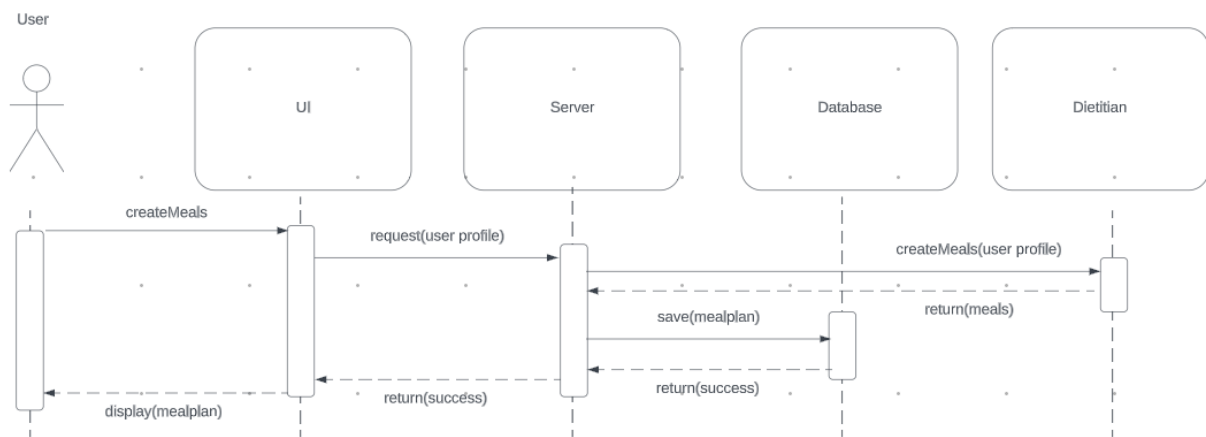


Figure 6.3 Sequence diagram for requesting meal plan

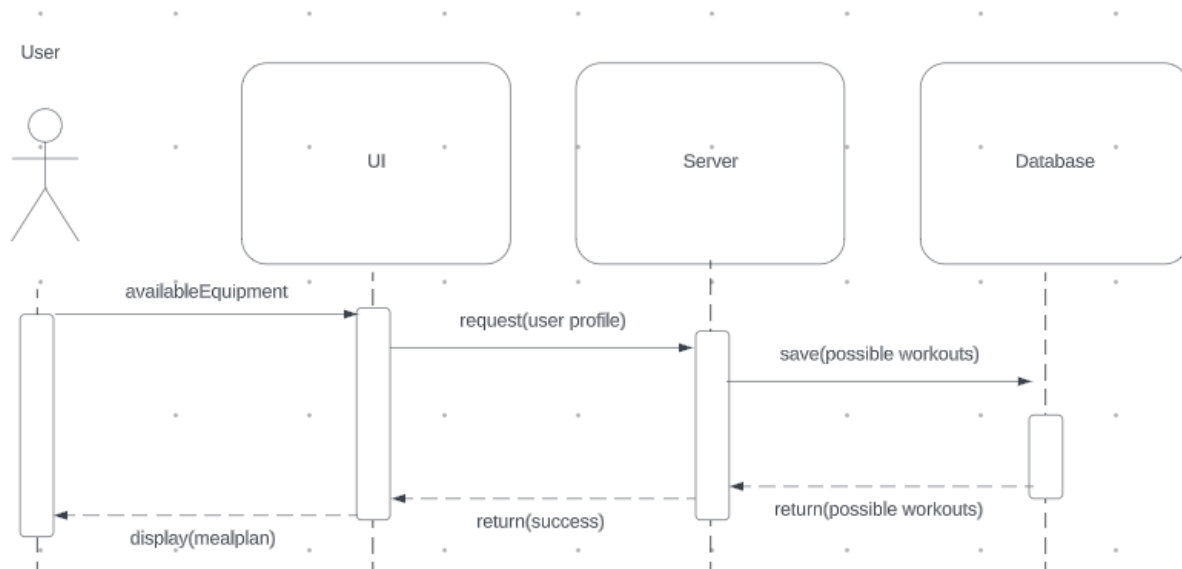


Figure 6.4 Sequence diagram for editing workout equipment

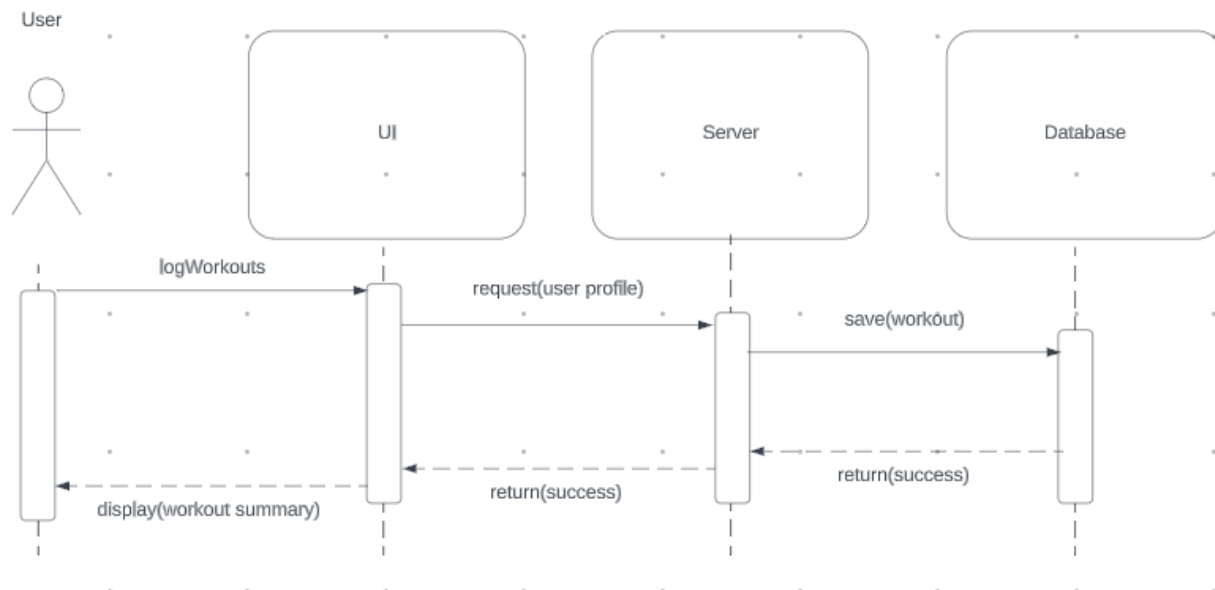


Figure 6.5 Sequence diagram for logging workouts and activities

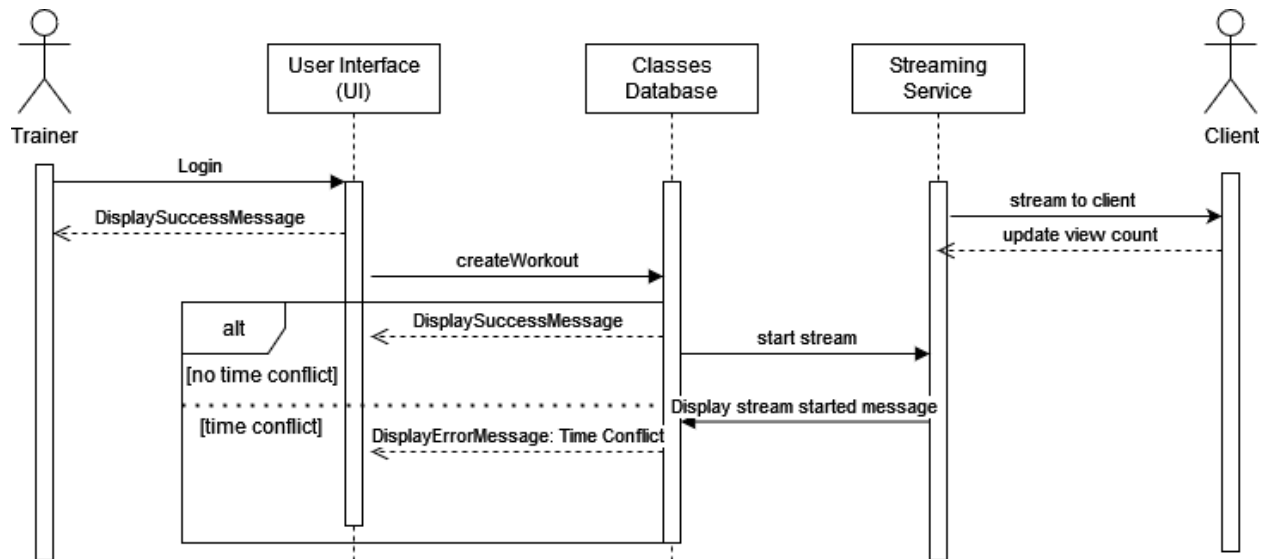


Figure 6.6 Sequence diagram for live classes

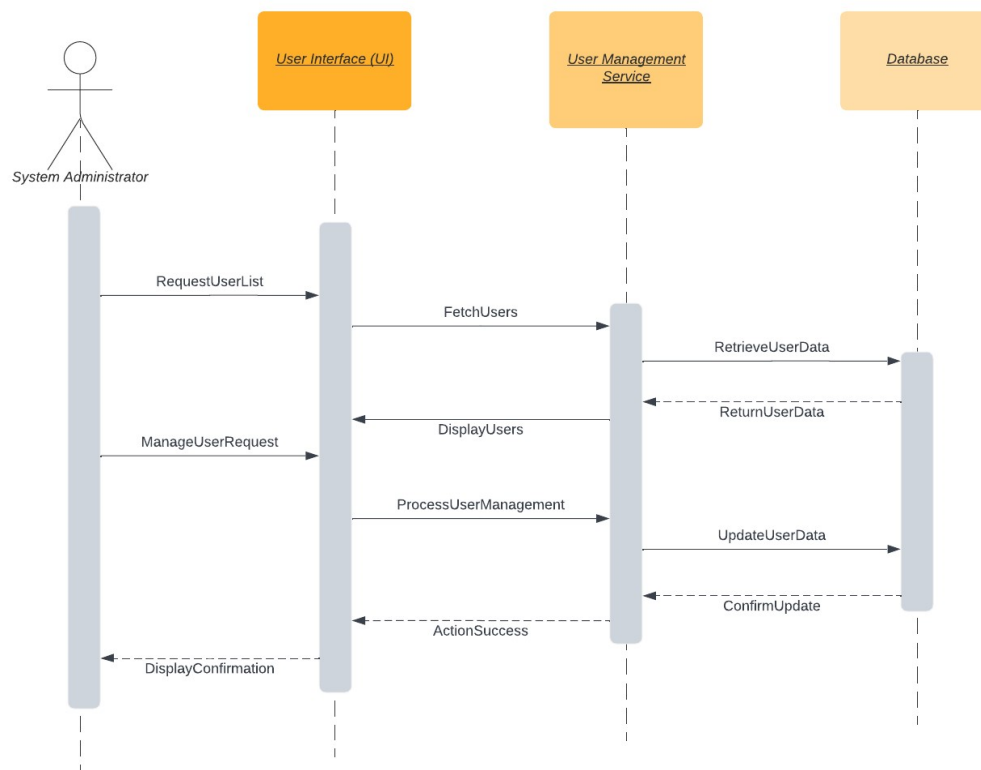


Figure 6.7 Sequence diagram for managing all users

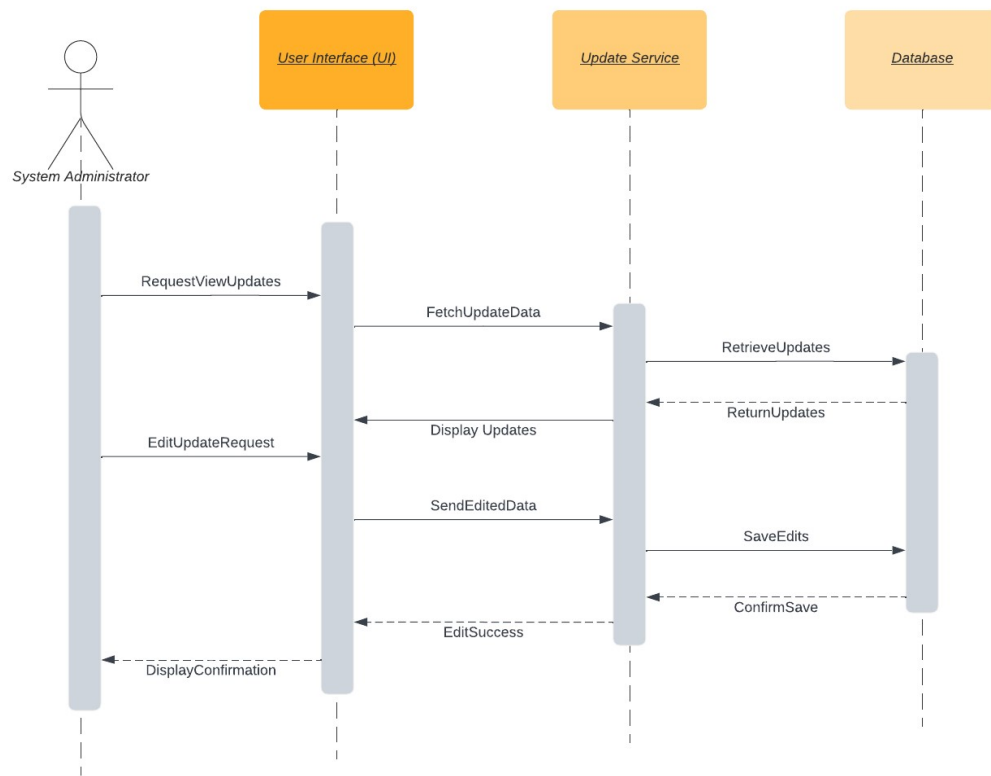


Figure 6.8 Sequence diagram for edit system update

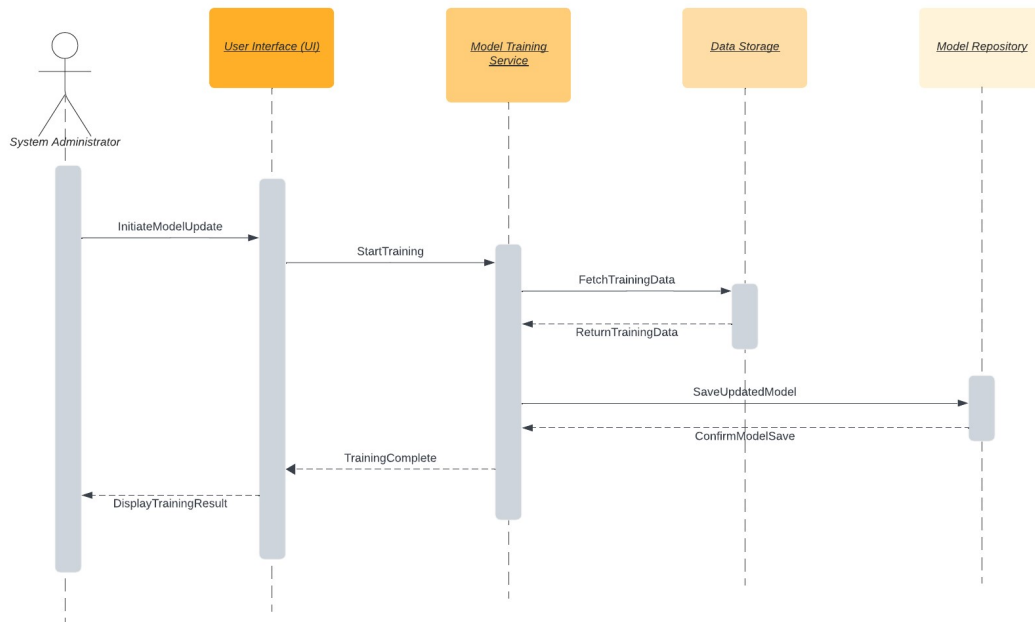


Figure 6.9 Sequence diagram for creating AI learning model updates

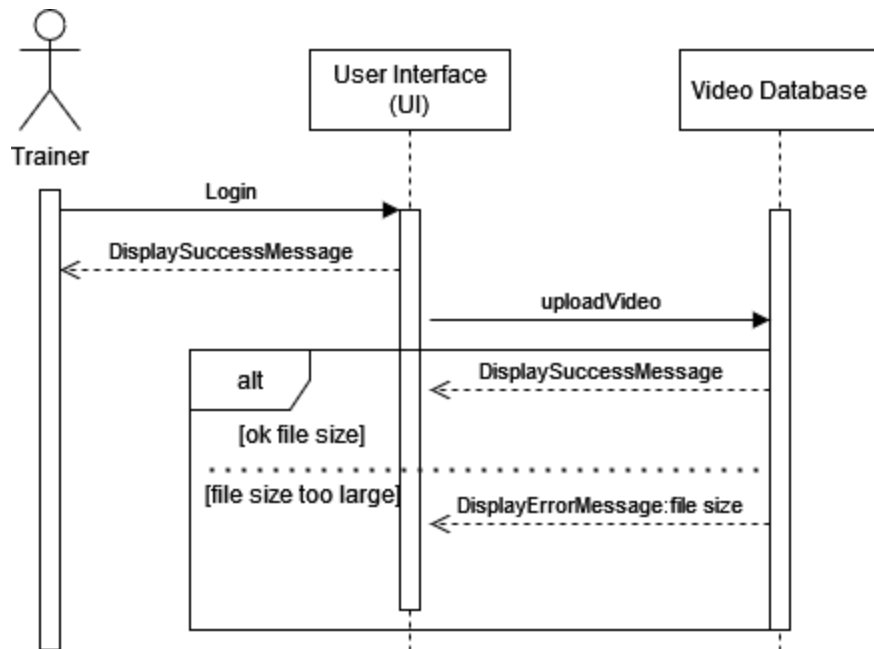


Figure 6.10 Sequence diagram for posting training video

7. UML Class Diagram

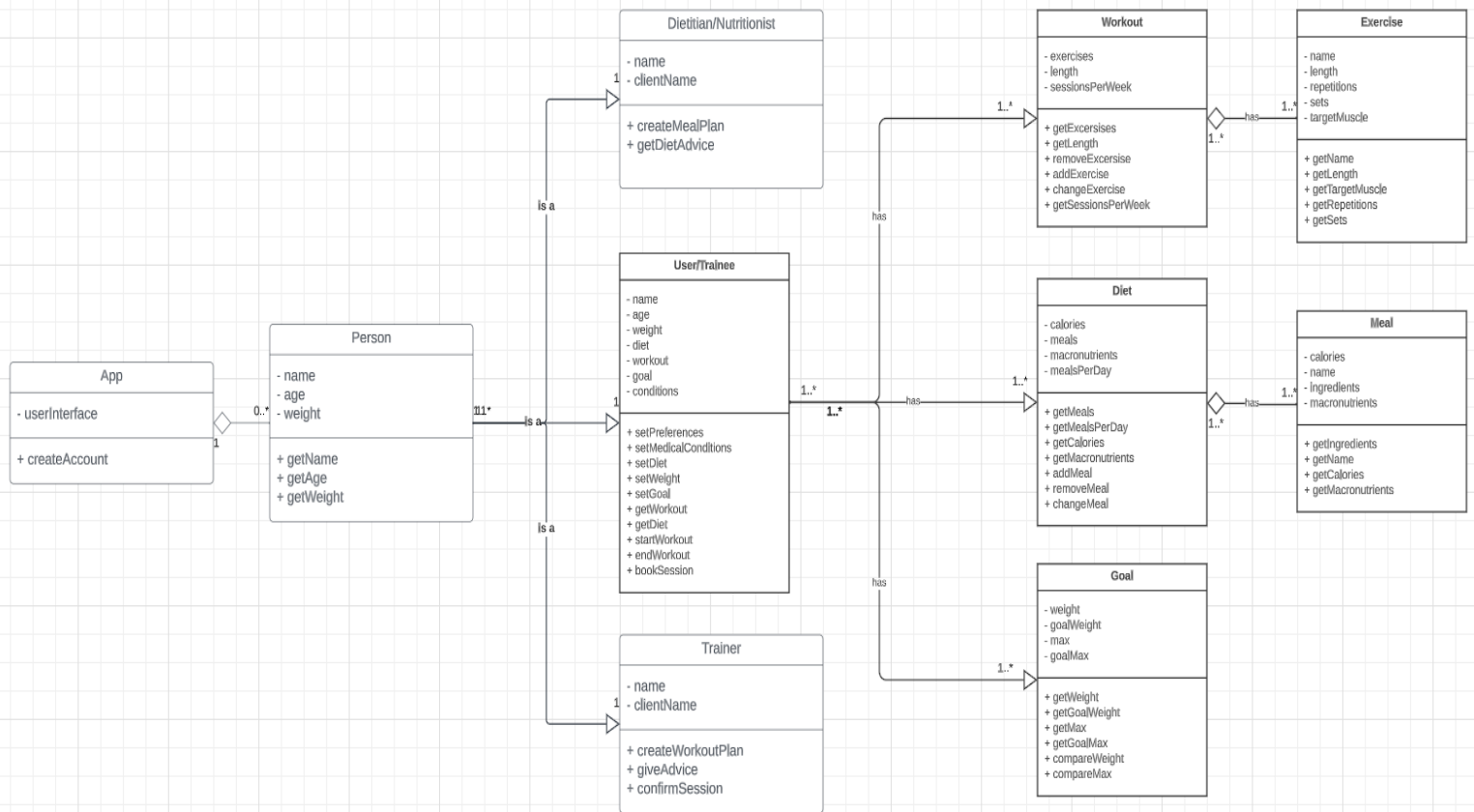


Figure 7.1 UML class diagram of the app

Link:

https://lucid.app/lucidchart/242f8d02-21a0-4140-8f90-69635128cf09/edit?viewport_loc=-2304%2C-1404%2C3025%2C1460%2C0_0&invitationId=inv_1f263d46-e7b0-400d-b08c-af79d8f86176

7. Architectural Design

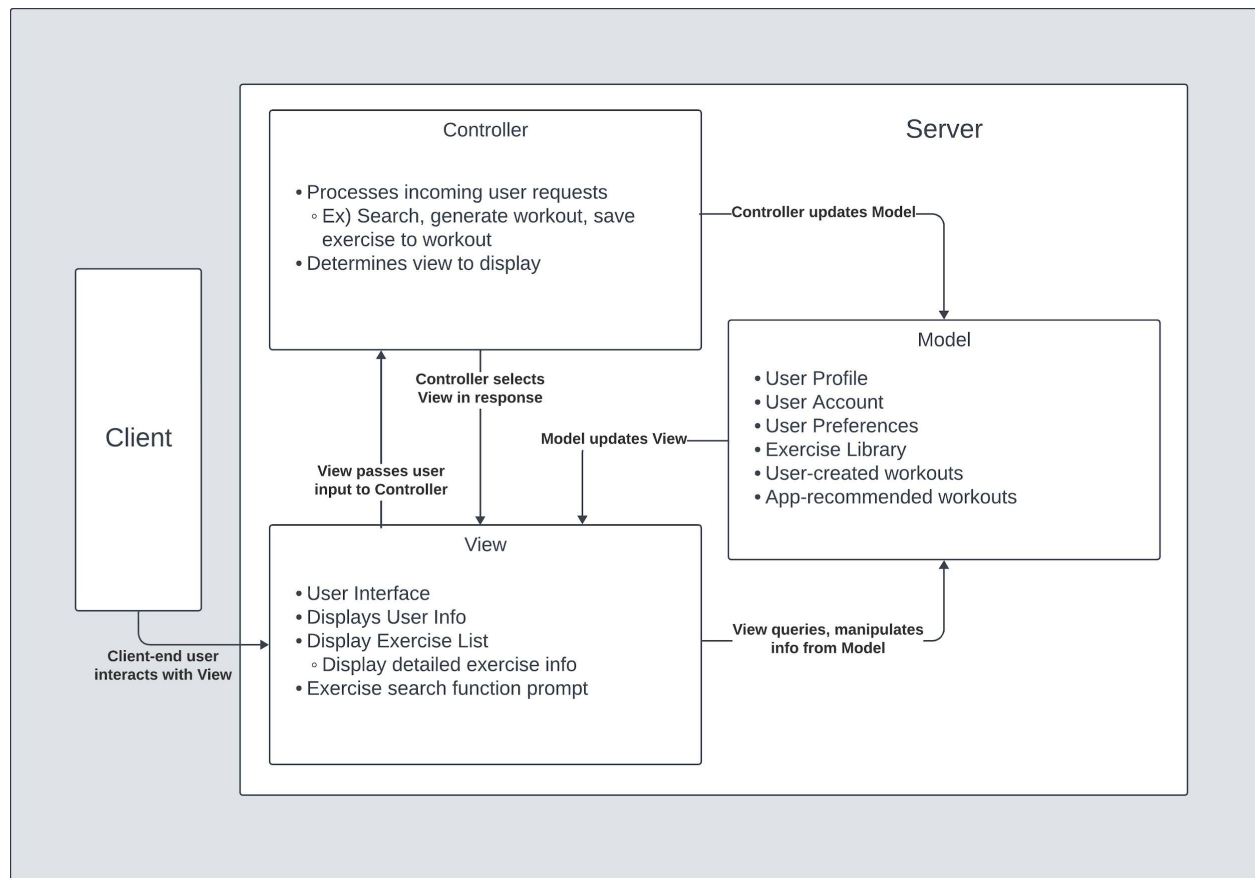


Figure 7.1 Architectural diagram of the app using MVC pattern

The Model-View-Controller pattern is the best fit for our application because it allows us to process and show the user's workout data in various ways easily. The main function of our app is to process the user's workout data to generate optimal workouts. The model does the processing and generation, the controller decides how to display workouts and suggestions and takes in user customization, and the view displays the workout and all the associated details.

8. Github Repo:

<https://github.com/NSPolloWaffle/3354-GymRats>

End of deliverable 1

3. Project Scheduling, Cost, Effort and Pricing Estimation, Project duration, and staffing

3.1. [5 POINTS] Project Scheduling. Make an estimation on the schedule of your project. Please provide start date, end date by giving justifications about your estimation. Also provide the details for:

- Whether weekends will be counted in your schedule or not
- What is the number of working hours per day for the project

Project Scheduling:

Start Date: January 16, 2024

End Date: March 8, 2024

Duration: Approximately 7 weeks

Justifications:

1. AI Model Development: Developing and training AI models is time-consuming and involves iterative testing and refinement. Allocating sufficient time for this phase ensures the quality and efficiency of the models.
2. Data Gathering: Collecting relevant data is critical to AI development. The time allocated for data gathering takes into account the need for a comprehensive and diverse dataset.
3. Research on Client Wants and Needs: Understanding the client's requirements is crucial for delivering a product that meets expectations. This research phase involves communication with the client, analysis of market trends, and refining the project scope.
4. Front-End Development: Building the user interface and ensuring a seamless user experience requires careful consideration. Allocating time for front-end development allows for thorough testing and refinement.
5. Marketing and Deployment: Successfully launching and marketing the app requires strategic planning. Allotting time for marketing activities and ensuring a smooth deployment process contributes to the overall project's success.

Work Schedule:

- Weekends: Excluded from the schedule to allow for a balanced work-life approach and to accommodate potential unforeseen delays.
- Working Hours per Day: ~5 hours per day. This schedule is designed with the understanding that the person coordinating the project is a full-time developer. It allows for focused, productive work while considering other work commitments and meetings.

Considerations:

- Flexibility: While the schedule provides a roadmap, it's essential to remain flexible. Unforeseen challenges or opportunities may arise, requiring adjustments to the timeline.
- Regular Updates: Regularly reviewing and updating the schedule ensures that the project stays on track. Milestones can be adjusted based on progress and any evolving requirements.

By incorporating these elements, your project scheduling estimation becomes more comprehensive and adaptable to potential changes.

3.2. [15 POINTS] Cost, Effort and Pricing Estimation. Describe in detail which method you use to calculate the estimated cost and in turn the price for your project. Please choose one of the two alternative cost modeling techniques and apply that only:

Function Point Estimation:

User Inputs: Current Overall Fitness Level, Desired Overall Fitness Level (Goals), Name, Age, Weight, Height

User Outputs: Workout Summaries, Achievements, Tailored Diet Plan, Tips/Advice, Trainer Dietitian

User Inquiries: changeDiet, changeWorkout changeTrainer, changeDietician changeGoals checkProgress

Data/Table Files: Workout History Database, Diet Plan Database, Trainer Database, Us, Database, Dietitian Database

External Interface Files: Smartwatch Workout Data, User Database, Food Database, AI Model

| | Function Category | Count | Simple | Average | Complex | Count ✖ Complexity |
|---|----------------------|-------|--------|---------|---------|--------------------|
| 1 | Number of User Input | 6 | 3 | 4 | 6 | 24 |

| | | | | | | |
|---|-------------------------------|---|---|----|------|-----|
| 2 | Number of User Output | 6 | 4 | 5 | 7 | 42 |
| 3 | Number of User Queries | 6 | 3 | 4 | 6 | 24 |
| 4 | Number of Data Files & Tables | 5 | 7 | 10 | 15 | 35 |
| 5 | Number of External Interfaces | 4 | 5 | 7 | 10 | 40 |
| - | - | - | - | - | GFP: | 165 |

Determine Processing Complexity (PC)

| | |
|---|---|
| 1) Does the system require reliable backup and recovery? | 5 |
| (2) Are data communications required? | 5 |
| (3) Are there distributed processing functions? | 4 |
| (4) Is performance critical? | 3 |
| (5) Will the system run in an existing, heavily utilized operational environment? | 1 |
| (6) Does the system require online data entry? | 5 |
| (7) Does the online data entry require the input transaction to be built over multiple screens or operations? | 3 |
| (8) Are the master files updated online? | 5 |
| (9) Are the inputs, outputs, files, or inquiries complex? | 3 |
| (10) Is the internal processing complex? | 4 |
| (11) Is the code designed to be reusable? | 2 |
| (12) Are conversion and installation included in the design? | 2 |
| (13) Is the system designed for multiple installations in different organizations? | 0 |
| (14) Is the application designed to facilitate change and ease of use by the user? | 4 |

$$PCA = 0.65 + 0.01(5+5+4+3+1+5+3+5+3+4+2+2+0+4) = 1.11$$

$$FP = GFP \cdot PCA = 183.15$$

$$E = FP / \text{productivity} = 183.15 / 30 \approx 7 \text{ person-weeks}$$

3.3. [5 POINTS] Estimated cost of hardware products (such as servers, etc.)

The price of server subscriptions for mobile apps can range from \$5 - \$250 per month. A reasonably generic AWS EC2 is estimated to cost \$162.79 monthly or \$1953.48 yearly.

- Developing for iOS requires Apple hardware. This can be fulfilled by acquiring Mac devices for developers or a subscription to a service like MacInCloud. The former would cost \$599 per developer, while the latter would cost about \$49 per month and lack capability for physical IO devices (such as if we implemented features for syncing health data over a USB rather than wirelessly).
- With a team of four developers, we will spend roughly \$4,349 on hardware. This assumes we acquire Mac Minis for developers, and pay for a year of server hosting.

3.4. [5 POINTS] Estimated cost of software products (such as licensed software, etc.)

- Developing for iOS requires a subscription to the Apple Developer Program, which is \$99/year per user.
- For a team of four developers, we will spend about \$396.

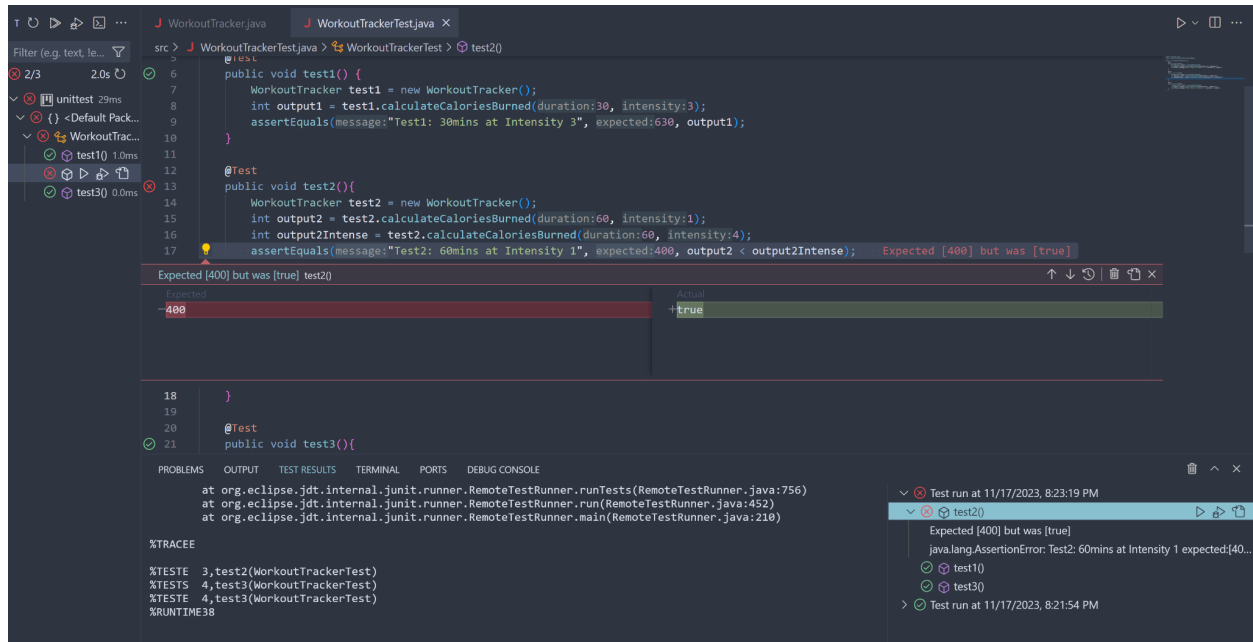
3.5. [5 POINTS] Estimated cost of personnel (number of people to code the end product, training cost after installation)

- Assuming a developer's yearly salary is \$150,000, or 12,500/month, and considering that our app is not especially large or complex, a small team of four would be appropriate for our purposes. The most complicated aspect is the AI algorithm, which would receive most of the workforce, while the creation of the app/UI would benefit from a smaller team to limit complexity and the need for communication.
- Assuming researchers and/or market researchers' yearly salary is \$60,000, we should expect to spend at least \$5,000/month per researcher.
- For a 7-week project, we can expect to spend roughly \$120,000 on personnel.

4. A test plan for your software

This test plan aims to verify the functionality of the `calculateCaloriesBurned` method within the `WorkoutTracker` class of our fitness app. Three test cases have been created to assess the accuracy of the method. The first case evaluates the computation of calories burned for a valid input, ensuring the result is a positive integer within a reasonable range. The second case evaluates the impact of intensity by comparing the calories burned during a workout of the same duration, but with different intensity levels. This case ensures higher intensities yield higher values. The third case evaluates the method's behavior when provided with 0 for the duration, expecting the calories burned to be 0 as well.

The following screenshots of testing the `calculateCaloriesBurned` method indicate the method's successful functionality in computing calories burned based on the given duration and intensity.



5. Comparison of your work with similar design

Our project essentially combines two different types of apps: nutritional intake loggers and fitness/workout loggers. We have researched apps in both of those fields for the purpose of comparing them to our project to see what similarities they have and what improvements could be made to those apps that we have implemented in our own project.

Strong - Workout Logger

Part of the functionality for our Gym Rats app involves allowing users to create workout plans and log their workouts based on their preferences. This is similar to what the app Strong does. Strong allows users to log workouts and track their weightlifting progress. The app itself has built in rest timers during your workout duration and a database of exercises to log. Our app improves on the foundation that Strong has built by using both fitness trainers registered on the app and an AI Model that aids users in creating their own workout plans that they can then log into the app to track their progress, and it takes into account the user's current fitness level, goals, and available workout equipment. We also include the functionality of rest timers in our workouts, taking the best from the strong app and improving upon the preset workouts with the help of AI.

Sworkit - Home workouts

Another app we can compare ours to is Sworkit, which is an app that provides preset workouts that are usually able to be done at home. The shortcomings of the app are that it does not really have plans for people that want to build strength in the gym using compound exercises

and machines. The home workouts it provides are great for more casual users, however, our app aims to provide workouts for users of all experience levels. This problem is solved in our app, which incorporates body weight and resistance training with weights into the workout. Our app aims to combine good at home workouts for more casual users and good strength workouts at gyms in order to satisfy the needs of all our users and help them meet all of their goals.

MyFitnessPal - Calorie Tracker

The other part of our Gym Rats app involves tracking users' daily nutritional needs. MyFitnessPal is an app that lets users track their daily caloric intake in regards to their goals with a large and extensive database of foods with their nutritional information. The app also includes additional features such as scanning barcodes of food items to view the nutritional information of the item. Our app intends to take this design a step further and allow users to enter their current weight and goal weight and create a diet or nutritional plan for the user using both dietitians registered on the app and our AI Model. Lastly, many nutrition apps, such as MyFitnessPal, do not include workout plans, so our app encompasses workouts combined with nutritional information.

Currently, there is not an app as comprehensive as ours in terms of overall health, as they either specialize in strength training, at home training, or nutrition, making our app a great addition to the category. Overall, with our app, people of any experience level can get a custom tailored workout designed specifically for their goals and needs so that they can be the best versions of themselves.

References will be in the IEEE format in the references section.

6. Conclusion

We believe that our team had an innovative and necessary idea for an app and did an adequate job in analyzing the software architecture and implementation timelines and details of the app. The app will be a comprehensive tool for all aspects of fitness and will not be too difficult to create. Our original idea was a more focused fitness app that only created workouts based on your goals, preferences, and available equipment. However, the idea evolved to include human personal trainers and dietitians that can provide even better personalized advice and add a human touch. We also decided to include calorie tracking as it was a supplemental service that most users would need.

7. References

Works Cited

[1]A. Stassen, “The 10 Best Fitness Apps to Download in 2020,” Verywell Fit, Dec. 21, 2021. <https://www.verywellfit.com/best-fitness-apps-4173707>

[2]“The 9 Best Free Fitness Apps [2022] - Boomfit,” www.boomfit.com.
<https://www.boomfit.com/en/blog/best-fitness-apps-b22.html>

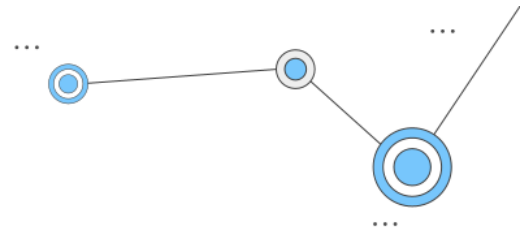
[3]“The 13 Best Health and Fitness Apps for Home and the Road,” Greatist, Dec. 26, 2018.
<https://greatist.com/fitness/best-health-fitness-apps#The-best-health-and-fitness-apps>

[4]“Home - Sworkit,” Sworkit, 2019. <https://sworkit.com/>

[5]“Strong Workout Tracker Gym Log App,” www.strong.app. <https://www.strong.app/>

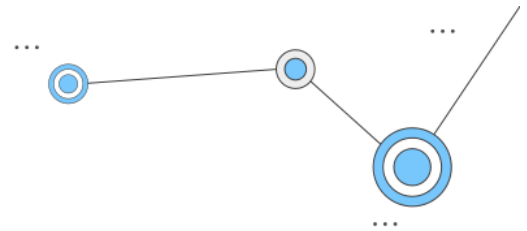
[6]MyFitnessPal, “MyFitnessPal,” Myfitnesspal.com, 2022. <https://www.myfitnesspal.com/>

8. Presentation Slides

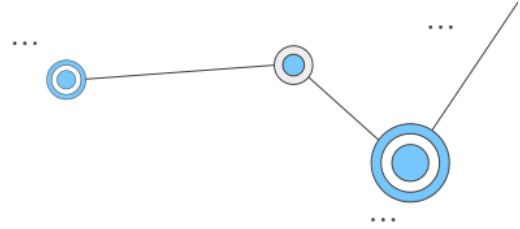


GYM RATS

Poorna Bharanikumar, Yvonne Hsiao, Mollik
Rahman, Naufal Sajjad, Muhilan Selvaa, Temi
Badmus, Jacy Grossman

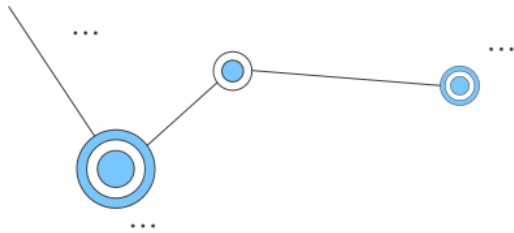


Imagine

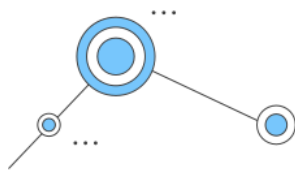


Our Solution

GYM RATS

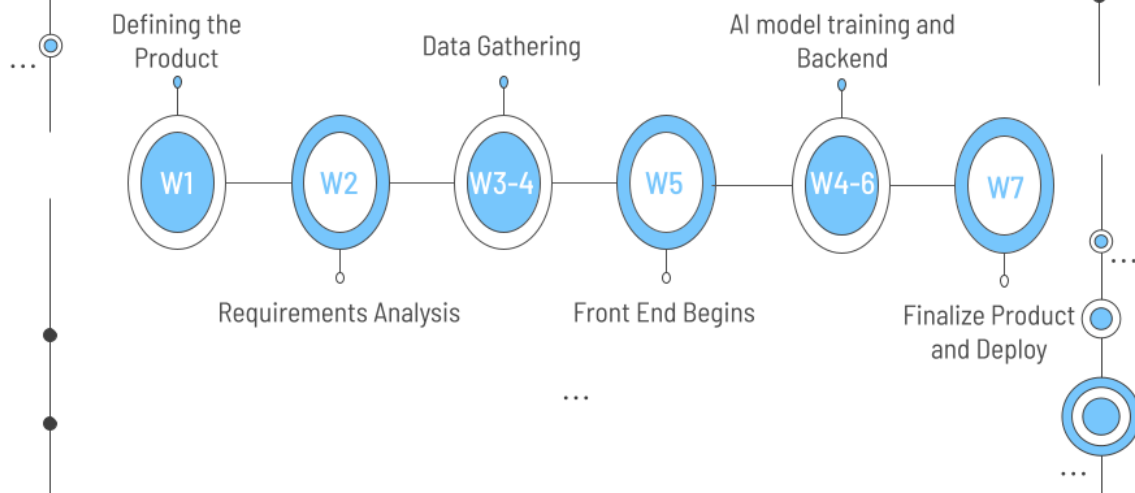


Cost Estimation



| | Unit Cost | Number of Units | Cost \times Units |
|--------------------------------------|---------------------------------|-----------------|-----------------------------|
| Personnel | | | |
| Developer | \$12,500 _{/month} h | 4 | 50,000 \times 2 months |
| Researcher | \$5,000 _{/month} | 2 | 10,000 \times 2 months |
| Required Hardware | | | |
| Server for hosting | \$1953 _{/year} | 1 | \$1953 |
| Mac Devices | \$599 | 4 | \$2,396 |
| Required Software | | | |
| Apple Developer Program subscription | \$99 _{/year} | 4 | \$396 |
| | | Total | \$124,745 |

Project Timeline



Functional Requirements

- User Registration and Profile Management
- Personalized Assessment
- Workout Customization
- Progress Tracking
- AI Workout Recommendations
- Exercise Library
- Nutritional Guidance

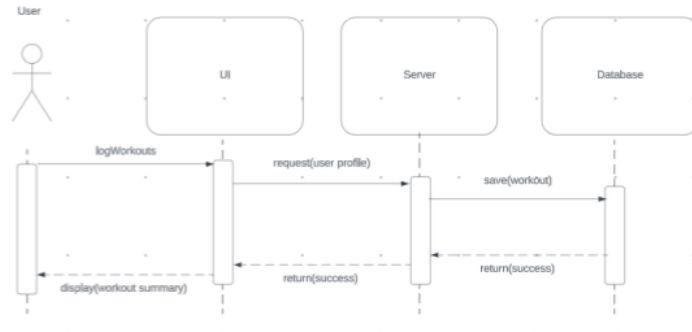
Non-functional Requirements

- Efficiency Requirements
- Performance Requirement
- Space Requirement
- Dependability Requirement
- Security Requirement
- Operational Requirement
- Development Requirement
- Ethical Requirement

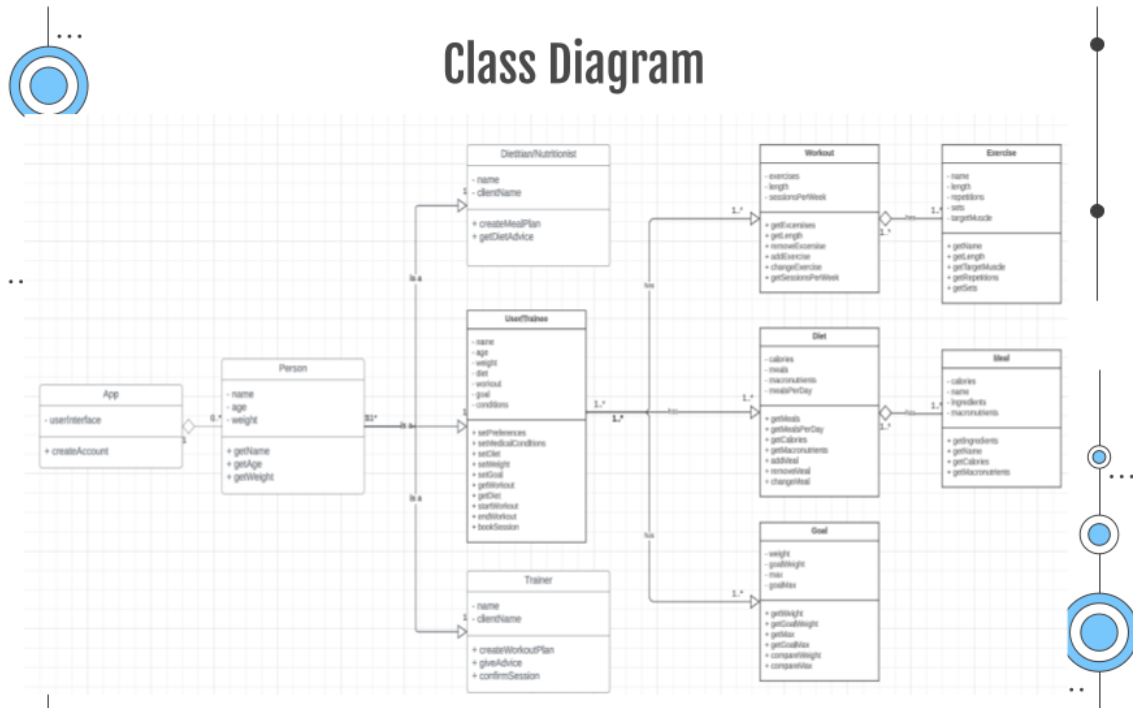
Use Case Diagram



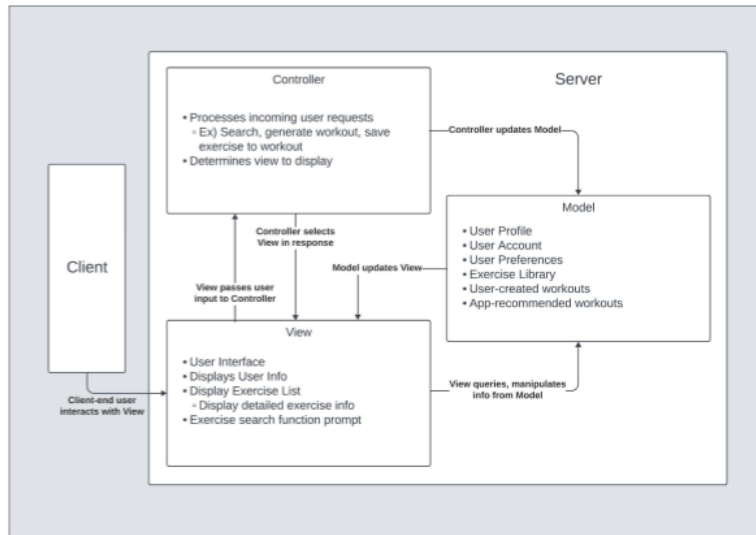
Sequence Diagram for Log Workouts and Activities



Class Diagram

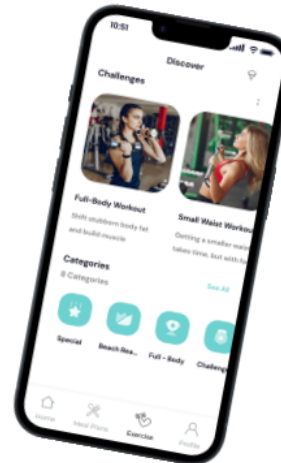


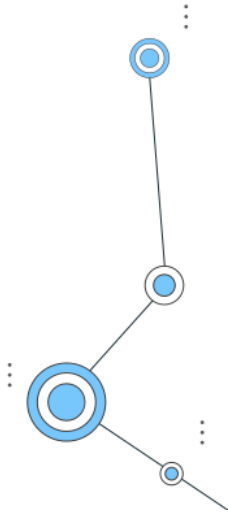
Architectural Design



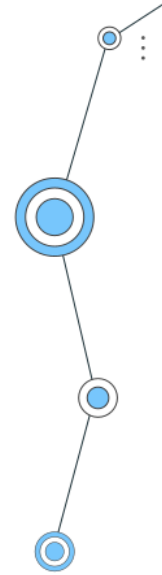
GYM RAT Mockup

<https://www.figma.com/file/VmD2rN77H9eygFAQeL5x8Qj/Fitness-App?type=design&node-id=1%3A3&mode=design&t=gawmkg2pgxQk5uKH-1>





Thanks!



10. [5 POINTS] GitHub requirement: Make sure at least one member of your group commits everything for project deliverable 2 to your GitHub repository, i.e.

- Your final project deliverable2 report
- Unit test code for a sample unit of your project
- Implementation code (if you have implemented your project)
- Presentation slides

Still, one member of your team should also submit the required project deliverable 2 materials to eLearning.