MuscleMate - Fitness Tracking App

Created by: Allie Miller

5/05/2024

CSC 3150–Systems Design
System Proposal
Instructor: Andy Cameron
University: Seattle Pacific University

Image generated by ChatGPT on 05.05.2024

Table of Contents

Executive Summary

MuscleMate is a proposed fitness-tracking application tailored for advanced lifters, including bodybuilders and powerlifters. Created by Allie Miller, a student at Seattle Pacific University, under Professor Andy Cameron's guidance, this project addresses the limitations of current fitness apps, which often focus on beginners and need more features than experienced athletes.

1.0 Introduction and Overview

Problem Statement:

Advanced lifters, like bodybuilders and powerlifters, often need to pay more attention to the current fitness-tracking applications, which mainly cater to beginners. Most of them typically include extensive databases filled with exercise descriptions and tutorials. These features are for people who need guidance on performing exercises correctly. The features of MuscleMate must align with the needs of experienced lifters who already know what to do. Users need a specialized app that supports straightforward workout logging without unnecessary features that can slow down their workout. MuscleMate provides a customizable platform that allows users to log workouts, monitor their progress over time, and easily access their data without the clutter of unnecessary features. MuscleMate ensures advanced lifters can focus on their workouts rather than navigate unnecessary features.

Project Vision:

MuscleMate aspires to redefine fitness tracking for advanced lifters by providing an intuitive app tailored to their needs. Our app enhances our users' training experience without the clutter of beginner-focused features.

Scope and Boundaries:

In response to this vision, we will develop MuscleMate to include essential features that allow for:

- Quick entry of exercises, sets, reps, and Rate of Perceived Exertion (RPE).
- Customization and reusability of workout templates tailored by the user.
- Progress tracking for users, coaches, and clients to monitor and analyze personal fitness goals and achievements over time.
- A clean, distraction-free environment that respects the user's familiarity with their workout routines.
- Ability to export workouts and progress graphs to others.

MuscleMate will strictly avoid beginner-oriented content, such as exercise tutorials or pre-defined workout plans, ensuring it remains highly relevant and efficient for its target user base. MuscleMate's primary environment will be iOS devices.

Requirements Summary:

The main business requirements for MuscleMate come from the needs and preferences of advanced lifters. We aim to maximize the efficiency of their workout tracking. The requirements are:

- **Workout Logging**: Users must be able to log their workouts quickly, including exercises, sets, reps, and Rate of Perceived Exertion (RPE).
- **Customizable Templates**: The system should allow users to create, save, and reuse customized workout templates.
- **Progress Tracking**: Provide tools for users to track their progress over time, including graphs that display user strength changes.
- **Workout Customizability**: Ensure all logged data is easily accessible and editable so users can revisit and change past entries.
- **Cross-Platform Synchronization**: MuscleMate must synchronize data across multiple devices, allowing users to access their workout logs from any Apple device anytime.
- **Coaching Portal**: Coaches should be able to view clients' progress.
- **Exporting to Others**: Users can export their workouts and progress graphs directly from MuscleMate.

These requirements fulfill the vision of MuscleMate by creating a user-centric app that focuses on functionality and efficiency, tailored explicitly for experienced lifters.

Stakeholders and Their Interests:

- **Users (Bodybuilders, Powerlifters, Fitness Enthusiasts):** They want an efficient and customizable app for tracking their workouts without unnecessary complexity. They require a tool to log workouts, track their progress, and integrate seamlessly into their workout routine.
- **Project Sponsor (Andy Cameron):** He is concerned with ensuring the app aligns with user needs, meets quality standards, and achieves the intended benefits outlined in the project initiation report.
- **Fitness coaches/trainers:** They're interested in using MuscleMate to enhance their coaching, allowing more precise tracking of their client's progress and better workout planning.
- **Potential Collaborators (Athletic Wear Brands, Gyms, and Coaches):** These people want to explore partnerships with the app.
- **Administrators:** They're interested in managing and maintaining the app's performance and ensuring its smooth operation.

Expected Costs and Benefits:

This section outlines the intangible and tangible benefits expected from the development of MuscleMate and the projected costs involved in completing this project.

Business Benefits/Intangible Benefits:

- **Increased User Satisfaction:** By offering an app tailored to the needs of advanced lifters, MuscleMate will enhance user satisfaction.
- **Brand Strengthening:** High user satisfaction contributes to a positive brand image. This satisfaction can elevate the company's standing in the marketplace, increasing its popularity and respect.
- **Market Differentiation:** MuscleMate's focus on advanced lifters sets it apart in a market saturated with beginner-oriented fitness apps.
- **Collaborative Opportunities:** The app will open new doors for collaborations with fitness professionals, gyms, and athletic brands, enhancing the app's visibility and usage.

Tangible Benefits:

We expect MuscleMate to generate a steady revenue stream through subscription models and potential in-app purchases. For more information on tangible benefits, refer to section 2.0.

Costs:

Below, I summarize what our funds will go towards. If you want further information, you can find it in section 2.0.

- **Development Costs**: Significant investment will be required for the initial development of MuscleMate, covering areas such as software design, programming, testing, and launching. The development cost also includes the price of hiring skilled developers and purchasing necessary technology.
- **Marketing and Advertising**: Some of our funds go towards marketing strategies, including online advertising, social media campaigns, and partnerships with fitness influencers and gyms.
- **Operational and Maintenance Costs:** Ongoing costs will include customer support, app updates, and data management to ensure continuous and smooth operation.
- **Training and Human Resources:** We will invest in training staff to manage the app and interact with users effectively.
- **Apple Prices:** Apple charges a 30% commission on all app sales and in-app purchases for the first year of a subscription, which then reduces to 15% in the following years.

Constraints:

We will develop MuscleMate within certain constraints that may affect the final product development process. Here's a summary of these constraints, along with our strategies for mitigation:

1. **Budget Constraints**:
    - A limited budget may restrict the scope and quality of our initial features, which could lead to delays or the exclusion of "nice to have" features. We will prioritize core functionalities, like workout logging and progress tracking. We will also use agile development methods to improve the app based on user feedback.
2. **Time Constraints**:
    - A tight schedule might force us to take shortcuts in testing and quality checks, resulting in the app missing some features at launch and potentially turning users away. We plan to focus on a minimum viable product with thoroughly tested features.
3. **Technical Constraints**:
    - Focusing only on the iOS platform simplifies initial development but limits the app's availability to only Apple users. We will test on various iOS devices to cover all supported models to ensure a smooth launch.
4. **Resource Constraints**:
    - With limited employees, our app development process might slow down and affect the app's quality. We could hire independent contractors for specific jobs.
5. **Market Constraint**:
    - Targeting a niche market might limit the initial user base and affect the revenue. To counteract this, we will launch targeted marketing campaigns and offer trial periods to attract users who can spread the word.

Recommendation:

Take the following steps to ensure the successful development and deployment of MuscleMate:

- Approval and Funding: We need to secure the approvals and funding from the project sponsor, Andy Cameron, and other stakeholders. Ensure the budget aligns with the projected costs outlined in the proposal, including development, marketing, and maintenance.
- Team Expansion: Consider expanding the development team to include specialists in user interface design, data security, and cloud infrastructure. The expansion will help minimize resource constraints.
- Development and Testing: We will start the development phase, focusing on core functionalities such as workout logging, customizable templates, progress tracking, coaching portal, and data synchronization. Implementing agile development methods, we will iteratively build and test these features while conducting testing across various Apple devices to ensure compatibility.
- User Feedback and Iteration: We plan to launch a beta version of the app to a select group of advanced lifters and fitness coaches. We will collect feedback from our users to identify areas for improvement.
- Marketing and Launch Strategy: Do social media campaigns, partnerships with fitness influencers, and collaborations with gyms to increase visibility and attract the target audience.

- Future Versions: Prepare for the subsequent phases of development, including introducing social media features, enhanced coaching portal functionalities, diet tracking, and integration with wearable devices like the Apple Watch.

By following these recommendations, MuscleMate can effectively meet the needs of advanced lifters and achieve a successful launch.

Document Overview:

The next part of this document organizes several key sections, each providing necessary information for the development, implementation, and management of MuscleMate. Here's an overview of what each section will be about:

- System Initiation: This section includes the Project Initiation Request (PIR), which showcases our project's information, business problem, justification, impact, importance, and benefits.
- Feasibility Assessment: This section evaluates the feasibility of MuscleMate across five key areas: Technical, Resource, Schedule, Organizational, Legal, and Contractual.
- Requirements Definition: This section outlines the functional, non-functional, and data requirements of MuscleMate.
- Requirements Model: This section includes a use case diagram and detailed use case descriptions. The diagram describes how users, coaches, and administrators interact with MuscleMate.
- System Evolution: This section discusses the future development of MuscleMate beyond the MVP release. Outlines planned features for future versions and recommended hardware/software upgrades.
- Conclusions and Recommendations: This section summarizes critical findings and recommendations for the next steps.
- Appendices: This section contains a survey, glossary, and bibliography.

2.0 System Initiation

Project Initiation Request (PIR):

PIR-00000 *[PIR Number to be assigned by the Project Office]* Project Initiation Request (PIR) – Level1 v6.0
Project Name: MuscleMate Student Name: *Allie Miller*

0. General Project Information

Project Name:	MuscleMate
Two Sentence Request Description:	MuscleMate is a fitness-tracking mobile app designed to help users log their workouts, track their progress, and achieve their fitness goals. It aims to provide a user-friendly platform for users to stay motivated and accountable during their fitness journey.
Requested Launch Date(s):	May 2025 (Flexible deadline)
Department(s) Affected By Project:	This project affects the health and fitness (subject expertise), technology (app development), and marketing (gain traction to the app) departments.
Project's Customers:	MuscleMate's users are bodybuilders, powerlifters, and fitness enthusiasts at the intermediate to advanced levels.
Date Request Submitted:	*4/16/2024*

1. Project Sponsor and Manager

Project Sponsor

Name:	Andy Cameron
Title:	Professor
Department:	Computer Science - SPU
Email:	acameron@spu.edu

Business Project Manager & Requestor

Name:	Allie Miller
Title:	Student/App Developer
Department:	Computer Science - SPU
Email:	millera31@spu.edu

2. Business Problem or Opportunity: The motivation for this request

MuscleMate aims to address the challenges intermediate and advanced bodybuilders and powerlifters face in effectively tracking and managing their workout routines. Despite the availability of fitness tracking apps, many users still resort to bringing a physical tracking journal to the gym because of the shortcomings of current fitness apps. However, carrying physical journals to the gym leads to inconvenience, the risk of loss or damage, and inefficient progress tracking.

The biggest problem in current fitness apps is their predominant focus on catering to beginners. These apps typically offer extensive exercise databases, instructional content, and pre-designed workout plans. However, this approach is inconvenient and time-consuming for advanced lifters, who already know the exercises they want to do in their workout and do not need help with their form. Users prefer not to search through extensive exercise lists to track their reps, sets, weight, and perceived exertion rate (RPE).

3. Justification, Impact, and Importance

Assumptions

- MuscleMate's target audience comprises approximately 10,000 active users within the first year of the launch.
- Our estimated average revenue per user is $0.99 per month. A user will pay $1.99/mo if they want additional features.

Competitive Landscape / Context

- Users might resort to the app "Sheets" since it is free on iOS.
- Beginner to intermediate-level users might enjoy using fitness apps like Fitbod, Workout Planner Muscle, SetGraph, and RepCount.

Tangible Return, Opportunity, or Value	One-Time	Ongoing
▪ The subscription is $0.99 a month to use the app.	$ 9,999/mo	$119,988/yr
▪ 2nd tier subscription: Users can only see progress graphs on squat, bench, and deadlift. If they want to see progress on other exercises, weight/body fat percentage, or track their food, they must pay $1.99/mo.	$ 9,900/mo	$ 238,800/yr

Intangible Benefits	Impact or Value
▪ The app allows future business collaborations with coaches and athletic wear brands. Collaborations might make more money per year than the subscription itself.	$N/A
▪ The company will have further exposure to new clients if MuscleMate becomes popular.	$ N/A

4. Product Requirements

4.1. Must Haves

4.1.1. MuscleMate must have a user-friendly interface for quick and easy workout logging.
4.1.2. **Comprehensive Workout Tracking:** The app will include a dedicated "log" tab where users can meticulously track sets, reps, weight, and rate of perceived exertion (RPE) for each exercise in their workout. Users will have the flexibility to edit entries seamlessly, even for past sessions.
4.1.3. **Workout Templates:** MuscleMate will provide a "template" tab, allowing users to save and reuse workout templates for future tracking.
4.1.4. **Progress Visualization:** The app will feature a "progress" tab where users can visually track their improvement in crucial lifts, including squats, bench presses, and deadlifts. The app displays progress graphically, which helps users visually see their strength gains.
4.1.5. **Workout History Tab:** MuscleMate's "history" tab will display a list of all past workouts, organized by name and date. Users can tap on any listed workout to view information like the exercises performed, sets, reps, and weights used. This feature allows users to revisit and analyze their past training sessions.
4.1.6. **Data Synchronization**: Ensure user data synchronizes across devices, allowing seamless access to workout logs and templates from multiple platforms.
4.1.7. **Coaching Portal:** Coaches can view clients' progress in the app.
4.1.8. **Exporting:** Users can export their workouts, goals, and progress graphs directly from MuscleMate.

4.2. Could Haves (Nice to Haves)

4.2.1.	Day and night mode.
4.2.2.	Each tab could have an icon next to its name. For example, the home tab could have a house icon next to it. The log tab could have a pencil icon next to it. The progress tab could have a graph icon next to it.
4.2.3.	Performance Tracking: It could include features for tracking additional performance metrics beyond the main lifts, like cardio progress, body measurements, and personal records.
4.2.4.	Coaching Portal Enhancements: Coaches could be able to create workout templates for their clients.
4.2.5.	Social Media Aspect: Users' could add friends and share their workouts/progress within the app.

4.3. Won't Haves (Don't Do's, aka Out of Scope)

4.3.1.	Pre-designed workout plans.
4.3.2.	List of exercises with form tutorials.

5. Project Costs (Operating and Capital: One-time and Recurring)

Labor Costs

Type	Team(s) Affected	Low (hrs)	High (hrs)
Analysis & Design	Project Manager, Designers	160	320
Development	Developers	800	1200
Testing and Quality Assurance	QA Engineers	200	400
Systems Integration	Developers	120	200
Deployment	DevOps	40	80
Support and Maintenance	Developers, Support Team	80	160
Sales and Marketing	Marketing team	120	240
Total		**1520**	**2600**

Comments: Labor costs are likely the most significant expense for this project, considering the extensive hours required for development, testing, and quality assurance.

Capital Costs (Equipment, Software, Licenses, …)

Description	Quantity	Cost ($)
Apple charges a 30% fee for apps and in-app purchases on the iOS, iPadOS, watchOS, and macOS App Store, affecting the overall revenue generated from the app.		$0
Apple charges 30% of revenue from subscriptions for the first year, then drops to 15% for subsequent years.		$ 35,996
Total		$35,996

Maintenance Costs (Costs after the product is live)

Type	Hours / Month Low	Hours / Month High
System / User Support	0	0
Business / Process Support	0	0
Total Support & Maintenance	**0**	**0**

Comments: MuscleMate is a workout tracker app, so changing things around might confuse the users. It's not a social media app, so there won't be many updates in that area. However, maintenance costs could accumulate over time, especially if the user base grows, requiring increased support and ongoing updates.

3.0 Feasibility Assessment

Introduction:

This section of the System Proposal evaluates the feasibility of our app MuscleMate across five key areas: Technical, Resource, Schedule, Organizational, Legal, and Contractual. We will assess each category to determine the project's feasibility. Each area of feasibility receives a rating on a scale from "High" to "Medium" to "Low" risk.

Feasibility Analysis:

Technical Feasibility: Medium Risk

- **Users' familiarity with the Application (Low Risk):** The primary users of MuscleMate are coaches and advanced lifters. They know about workout routines and what they need from a fitness-tracking app. The users are mainly in the age range of 18-50 years old. So, most people should understand how to navigate the app, mainly because the user interface will be easy to learn. This makes this area **feasible**.
- **Development Team's Familiarity with the Application Area (Medium Risk):** Our development team is familiar with mobile application development. The user interface should be easy to execute, although they will probably need training for security and storage aspects. Therefore, feasibility in this area is **feasible**.
- **Project Size (Medium-High Risk):** MuscleMate's size is moderate. Since the app is explicitly for advanced lifters, this simplifies some aspects of development. When developing MuscleMate, our team will focus on the MVP, which includes logging a workout, creating a workout history log, template creation, simple progress tracking, a coaching portal, and the ability to export workouts/progress graphs. Further along, our team will put out other versions that could expand on our original features. We plan to incorporate more complex progress reports and integrate external devices. Since we will implement subsequent app versions, the **feasibility is risky.**
- **Project Structure (Medium Risk):** We have stated the critical elements of the app and expect them to remain the same. Implementing future requirements could be complex, especially when integrating external devices and social media. The feasibility **is risky**.

Resource Feasibility: Low Risk

- **Team Composition (Low-Medium Risk)**: Given the project's scope, our small but capable team should be sufficient. Our team has experience in app development, Swift, and UI/UX design. The only risk is that most of our teams' developers have 2-5 years of experience. The feasibility in this area is ideal.
- **Software and Tools (Low Risk)**: The development team will use Xcode, which provides all the tools for iOS app development. The IDE should be enough for both coding and testing. We will use SQLite for database management. The feasibility of this area is ideal.
- **Hardware (Low Risk):** Since MuscleMate only caters to IOS devices, developers must use a Mac to run Xcode. However, testing devices should cover a range of iOS devices to ensure the app runs well on multiple devices. Xcode has a built-in feature to test on different devices. The feasibility in this area is ideal.
- **Environment (Low-Medium Risk)**: The development environment is digital. We must have software licenses and adequate cloud storage for backup. We must also get the Apple developer program membership to publish the app.

Schedule Feasibility: Medium Risk

- **Likelihood of Meeting Timeframes (Medium Risk):** MuscleMate should be ready for launch within nine to fifteen months, which includes the initial development, test, and preparation for launch. Given the scope of the app, it is **feasible** that the MVP will be on time.
- **The Flexibility of Completion Dates (Medium Risk):** The deadlines are moderately flexible. While the goal is to have the app ready by May 2025, there's some grace to extend this to ensure we put out the highest quality product possible.
- **Consequences of Missing the Deadline (Medium Risk):** Any delay could affect initial user engagement, especially if marketing comes out with an expected deadline that is not met. Being late to the launch date could also dampen our stakeholders' trust in us and escalate project costs. However, a positive aspect is that a delayed launch might cause a higher-quality app, enhancing user satisfaction.
- **Availability of Resources (Low Risk):** MuscleMate depends on a small team and tools like Xcode and SQLite. Which we already have access to. We might have to hire or contract specialized experts to ensure we meet our development goals in time.

Organizational Feasibility: Rating Low Risk

- **Competitive Analysis**: Many fitness apps are available, but most cater to beginner lifters. A significant competitor is Google Sheets.
- **Competitive Advantage**: MuscleMate differentiates itself by focusing only on the needs of advanced lifters, like bodybuilders and powerlifters. It is the closest app to a standard workout journal. The user doesn't have to go through endless databases of exercises.
- **Current Problems with Other Fitness Apps**: Current apps overwhelm intermediate and advanced lifters with unnecessary tools like tutorials on form and preset workout plans. They often cannot offer the customization experienced lifters require, which can frustrate lifters who have specific training routines and goals.
    - **New System Benefits:** By making workout templates and logging customizable, MuscleMate focuses directly on advanced lifters' needs, hopefully increasing user engagement and retention.
    - **Necessity:** There is a significant demand for a specialized workout app catering to advanced lifters' rigorous and specific training routines.
- **Consequences of Non-Development:** Without MuscleMate, advanced lifters will have to resort to using a journal in the gym, Google Sheets, beginner workout apps, or potentially paying much more money for a more advanced catered app.
- **Integration into Ongoing Operations:** MuscleMate will integrate smoothly into its users' everyday routines since it is compatible with IOS devices, which many lifters use.
- **Stakeholders' Perspectives:** Stakeholders, including fitness trainers and gyms, see the app as a tool to enhance their client services.
- **Pricing Strategy:** Users will most likely be willing to pay for a tiered subscription model because the app will start off providing basic features for free and then will charge for advanced analytics and app features.

Legal and Contractual Feasibility: Low Risk

- **Compliance Requirements (Low Risk):** Follow financial reporting related to app sales/revenue. Ensure that the app meets ADA standards and is accessible to all users.
- **Legal Concerns (Low Risk):** Address copyright issues, like libraries and APIs.
- **Privacy Concerns (Low Risk):** The user must sign up to use the app providing their information, such as a username, password, address, and phone number.

Additional Comments:

Our team plans to continue monitoring our app's feasibility. We will also ask our users' opinions on the app to continue improving.

Conclusion:

We assess the feasibility of MuscleMate as a medium risk. While certain risks are associated with the project, we are confident in our ability to mitigate them effectively. Key strengths include user familiarity with fitness tracking needs, simplifying the user interface design, and our development team's proficiency in mobile app development, mainly using Xcode and SQLite for iOS development. Necessary tools and environments are already accessible, reducing the risk of resource constraints.

However, we must carefully manage challenges like meeting tight deadlines and ensuring seamless integration of advanced features. By using agile development practices, prioritizing user feedback, and planning for regular updates, we are confident in delivering a high-quality product that meets the needs of advanced lifters.

4.0 Requirements Definition

Introduction:

This section covers the functional and non-functional requirements of MuscleMate. Functional requirements are services in the app that fulfill the users' needs. Data Requirements are the specific information that needs to be collected, stored, and managed by the app. Non-functional requirements specify how the app works and outline the constraints that dictate its operation.

Functional Requirements:

1. **User Registration and Authentication**
   - 1.1. Users must be able to register for a new account using an email address, username, and password. (Must Have)
   - 1.2. Users must be able to log into their accounts using their email, username, and password. (Must Have)

2. **Workout Logging**
   - 2.1. Users must be able to log their workouts, including their exercises, sets, reps, weights, and RPE. (Must Have)
   - 2.2. Users must be able to edit/change their workouts. (Must Have)
   - 2.3. Users must be able to choose to either log their workout freely or use a template.

3. **Workout Templates**
   - 3.1.Users must be able to create, save, and reuse customized workout templates. (Must Have)

4. **Progress Tracking**
   - 4.1. The app must create progress reports for the user's key lifts. (Must Have)

5. **Workout History Access**
   - 5.1. Users must be able to view their past workouts, organized and displayed with the workout's name and dates. (Must Have)

6. **Data Synchronization (Must Have)**
   - 6.1. User data should be consistent across IOS devices. (Must Have)

7. **Manage Users/App Maintenance**
   - 7.1. Administrators should manage user accounts. (Should Have)
   - 7.2. Administrators should perform maintenance tasks like software updates and system monitoring and acknowledge user feedback. (Should Have)

8. **Coach Portal**
   - 10.1. View Client Progress (Must Have)
   - 10.2. Provide Feedback (Could Have)

9. **Social Interaction**
    9.1. Users can directly export their workouts, goals, and progress graphs from MuscleMate to social media. (Must Have)
    9.2. MuscleMate could have a social media aspect where users can add each other as friends and share their progress/workouts. (Version 2–Could Have)

10. **Future Enhancements**:
    10.1. Integration with the Apple Watch. (Version 2–Must Have)
    10.2. Implement diet tracking features. (Version 2-Could Have)

11. **Pre-Designed Workout Plans (Won't Have)**

12. **List of Exercises with Form Tutorials (Won't Have)**

Data Requirements:

1. **User Data**
    1.1. Personal information, including name, email address, age, and gender.
    1.2. Account details like username, securely stored password, and user type (Coach or Lifter).
2. **Workout Data:**
    2.1. Workout details, including date, time, and workout name.
    2.2. Exercise details, including exercise name, sets, reps, weight, and RPE.
    2.3. Workout templates, including template name and exercise details.
3. **Progress Data:**
    3.1. Track progress on users' lifts (date, weight, reps).
4. **Interaction Data:**
    4.1. Monitor user interactions, including app navigation patterns, usage frequency, and actions taken within the app.
    4.2. Collect feedback like feature feedback, bug reports, and suggestions.
5. **Analytics Data:**
    5.1. Gather usage metrics on active users, retention rates, and user engagement.
    5.2. Assess feature effectiveness like workout logging success, template popularity, and progress tracking usage.
6. **Security Data:**
    6.1. Ensure data protection (encryption, secure storage, security audits) and privacy compliance (user consent management, transparent data practices).

Non-functional Requirements:

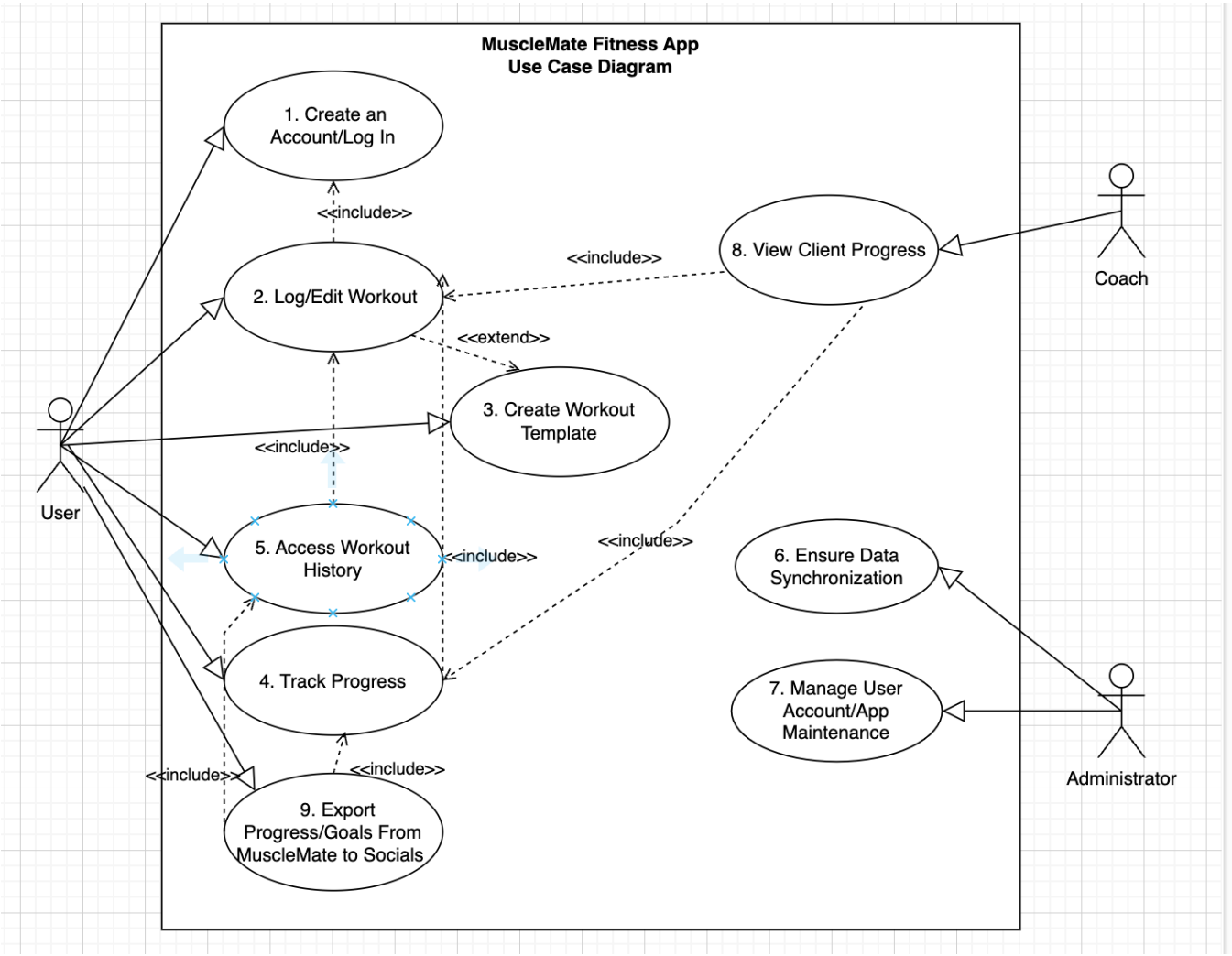
1. **Operational Requirements**
    1.1. **Device Support:** MuscleMate must be compatible with all current versions of iOS to ensure accessibility across different Apple devices.
    1.2. **Integration:** The app could integrate seamlessly with Apple Health.
    1.3. **Documentation:** Comprehensive documentation should accompany all aspects of the system to simplify maintenance and future enhancements.
2. **Performance Requirements**
    2.1. **Response Time:** The app should load user data and display the workout log within 2 seconds.
    2.2. **Concurrent Users:** The system should handle at least 10,000 users without bad performance, ensuring standard operation even during peak usage.
    2.3. **Scalability:** The system's backend should scale automatically to handle sudden increases in demand without affecting performance.
    2.4. **Resource Optimization:** We should optimize resources like memory usage to handle growth in user databases.
3. **Security Requirements**
    3.1. **Data Encryption:** We will encrypt user data to protect against data breaches.
    3.2. **Authentication:** The app should support multi-factor authentication to enhance security during user login.
4. **Cultural and Political Requirements**
    4.1. **Accessibility:** MuscleMate should comply with accessibility standards to accommodate users with disabilities
    4.2. **Languages:** The user should be able to change the app's language.

5.0 Requirements Model

Introduction

This section includes a use case diagram along with use case descriptions. The use case diagram shows how users, coaches, and administrators interact with MuscleMate. The use case descriptions go into more detail. In them, I go over things like the flow of events, a brief description, special requirements, and triggers.

Use-Case Diagram:



https://drive.google.com/file/d/13n7NVVQZNihhLxkNhFtny-37iWIaCOmO/view?usp=sharing

Use-Case Descriptions:

Use Case Name: User Registration and Authentication**:** Register and Login	**ID:** UC-1	**Importance**: Must Have (in MVP)
Primary Actor: User	**Use Case Type**: Detail, Essential	
Supporting Actors: An Administrator, if necessary		
Stakeholders and Interests: Users: Interested in a straightforward login process to access the fitness tracking features. Project Manager (Allie) and Administrators: Interested in securing the app.		
Brief Description: This use case describes how a user registers or logs into an account on Musclemate.		
Trigger: The user downloads MuscleMate from the app store and clicks the option to register or log in from the app's main screen. **Type** (mark one): _X__ External ___ Temporal (rare)		
Relationships: **Association**: User **Include**: N/A **Extend**: N/A **Generalization**: This use case is an instance of the broader user management use case.		
The Normal Flow of Events: Register: 1. The user selects either the 'Register as Coach' or 'Register as Lifter" option. 2. The user then enters their email address. 3. Once the user has passed the authentication, the user creates a username and password. 4. The system creates a new account and confirms the user's registration. Login: 1. The user selects the "Log In as Coach" or "Log in as Lifter" option. 2. The user enters their username and password. 3. The system validates the credentials. 4. The system grants the user access to their account.		
Sub-flows:		

2.1 Email Verification: After The user enters their email, the system sends a verification email. The user must click the link to verify their email.

Alternate/Exceptional Flows:

In step 2 of "Log In", if the user forgets their username/password, then the user can select "Forgot Password?". The user will enter their email address to receive a password reset link. Then, the user can set a new password and login.

Special Requirements:

Performance

1. The system should process registration and login requests promptly.

User Interface

1. The registration/login process should be accessible to Apple products.

Security

1. Store user passwords securely.

To do/Issues:

1. Review password complexity requirements to ensure they meet security standards.
2. Do multi-factor authentication to enhance the app's security.

Use Case Name: Workout Logging: Log and Edit Workouts	**ID:** UC-2	**Importance**: Must Have (In MVP)
Primary Actor: User	**Use Case Type**: Detail, Essential	
Supporting Actors: N/A		
Stakeholders and Interests: Users: Interested in a way to log their workouts. Fitness Coaches: Interested in using the app alongside training their users.		
Brief Description: This use case allows users to enter and track their workouts, including their sets, reps, weights, and rate of perceived exertion for each exercise. Users can also edit these entries.		
Trigger: The user selects "Log Workout". **Type** (mark one): __X_ External ___ Temporal (rare)		
Relationships: **Association**: User **Include**: Create an Account/Login: A user must have an account before logging their workout. **Extend**: N/A **Generalization**: N/A		
The Normal Flow of Events: Log Workout (Without Template) 1. The user navigates to the "Log" tab. 2. The user selects the date and time of the workout 3. The user names their workout 4. The user adds exercises, specifying details for each exercise (sets, reps, weights, RPE). 5. The system saves the workout details and confirms the successful logging to the user. 6. The user can edit the exercise names, weights, number of reps, and number of sets if necessary. Log Workout (With Template) 1. The user navigates to the "Log" tab. 2. The user selects the date and time of the workout 3. The user selects the option to "Use a Template." 4. The user selects the workout template they want to use, and the app automatically logs the workout for that date. 5. The user can edit whatever they need to fit their workout for that day. For example, if the number of reps changed in their workout, the user can change them in the template to reflect that.		

Sub-flows:
5.1 When a user opts to log a workout using a pre-defined template but wants to include extra exercises not in the template, then the user can do one of two things:
1. The user can add the exercise to their workout on top of the template. This won't change the template; it'll only add the exercise to your day's workout.
2. The user can navigate to the "Template" tab and edit the workout. The app will now include this exercise in the template, which users can use when logging their workouts.

Alternate/Exceptional Flows:
3.1. If the user enters incomplete exercise details, the system flags the incomplete fields and prompts them to complete them. Once the user fills them out, the system saves the exercise.

Special Requirements:
Performance
1. The app should process workout log entries in under three seconds.
User Interface
1. The workout logging interface must be easy to navigate on any Apple device.
Security
1. Save each workout on the workout log.

To do/Issues:
1. Where do we store all the exercises and workouts without taking up much space?

Use Case Name: Create Workout Template	**ID:** UC-3	**Importance**: Must Have (in MVP),
Primary Actor: User	**Use Case Type**: Detail, Essential	
Supporting Actors: N/A		
Stakeholders and Interests: Users: Interested in quickly logging repetitive workouts.		
Brief Description: This use case allows users to create customized workout templates, simplifying the process of workout logging.		
Trigger: The user selects "Create New Template" under the "Template" tab. **Type** (mark one): _X__ External ___ Temporal (rare)		
Relationships: **Association**: Users **Include**: N/A **Extend**: This use case extends the base functionality of "workout logging." **Generalization**: N/A		
The Normal Flow of Events: 1. The user selects "Create Template" from the template tab. 2. The user enters a name for the new template. 3. The user adds exercises to the template, specifying the exercise, sets, reps, and RPE. 4. The user saves the template stored in their list of templates.		
Sub-flows: N/A		
Alternate/Exceptional Flows: 3.1. If the user enters incomplete exercise details, the system flags the incomplete fields and asks them to complete them. The user needs to fill them out to save the exercise.		
Special Requirements: Performance 1. The system should save the workout template within three seconds. User Interface 1. The interface for creating templates should be simple to use. Security 1. User-created templates should be secure and only be accessible to the user and whoever else they share them with.		
To do/Issues: We should collect user feedback on the template creation process to help make it more intuitive.		

Use Case Name: Track Progress	**ID:** UC-4	**Importance**: Must Have (in MVP)
Primary Actor: User	**Use Case Type**: Detail, Essential	
Supporting Actors: Fitness Coach		
Stakeholders and Interests: Users: Interested in monitoring their progress toward set goals. Fitness Coaches: Interested in using progress reports to adjust training plans.		
Brief Description: This use case makes it easier for users to see their progress in the gym. The system will generate reports on critical lifts.		
Trigger: The user selects the "Track Progress" tab. **Type** (mark one): _X__ External ___ Temporal (rare)		
Relationships: **Association**: Users **Include**: "Log Workout"–required to track progress on logged workouts. **Extend**: N/A **Generalization**: N/A		
The Normal Flow of Events: 1. The user navigates to the "Track Progress Tab". 2. Upon accessing, the user views three graphs displaying 1 rep max estimations over squat, bench, and deadlift dates based on their logged sets. 4. The user can click on any graph point to see the exact 1 rep max estimation and the actual weight lifted for the number of reps. 5. The graphs automatically update as the user logs new workout data.		
Sub-flows: 2.1. If the user wants to see progress on other lifts, they need to pay $1.99/month. 4.1. The user can customize which data points (i.e., date ranges) to display on the graphs.		
Alternate/Exceptional Flows: 4.1.1. If the user finds that the graph is showing inaccurate data, the user can report the issue, and customer support can follow up.		
Special Requirements: Performance		

1. Graphs should automatically update after a user logs their workout.

Security

1. The system must process subscription transactions through a secure payment gateway.

User Interface

1. Graphs must be easily interpretable, with options to customize color and scale for better visibility

To do/Issues:

There's most likely a better way to track the progress than displaying one rep max. We should listen to the feedback from users. Some users may want to see their total volume displayed instead of one rep max. We could look into doing them both.

Use Case Name: Access Workout History	**ID:** UC-5	**Importance**: Must Have (In MVP)
Primary Actor: User	**Use Case Type**: Detail, Essential	

Supporting Actors:
N/A

Stakeholders and Interests:
Users: Interested in reviewing their workout history to monitor their progress and consistency.

Brief Description:
This use case allows users to access and view the history of all their past workouts, including dates and the specific exercises in each workout.

Trigger:
The user selects the "Workout History" tab within the app.

Type (mark one): _X__ External ___ Temporal (rare)

Relationships:
Association: User
Include: "Log/Edit Workout"- The user needs to log a workout to create a "Workout History".
Extend: N/A
Generalization: Coaches have a general relationship since they will track their client's progress.

The Normal Flow of Events:
1. The user selects the "Workout History" tab in the app.
2. The system displays a list of past workouts, each entry labeled with the date and name of the workout.
3. The user can select any workout to view its information, including exercises, sets, reps, and weights used.
4. Users can edit past workouts and delete them.

Sub-flows:
N/A

Alternate/Exceptional Flows:
2.1. If no workout history is available (new user), the app displays a message saying, "No workouts found."

Special Requirements:
Performance
1. The system should update the workout history log immediately after a user records a new workout.

User Interface

1. The workout history should display the most recent workouts at the top.

To do/Issues:

1. We should think about making a "search" so it's easier to find specific workouts.

Use Case Name: Synchronize User Data	**ID:** UC-6	**Importance**: Must Have
Primary Actor: Administrator	**Use Case Type**: Detail, Essential	
Supporting Actors: N/A		
Stakeholders and Interests: User: Interested in the app's data staying consistent across their different Apple devices.		
Brief Description: This use case ensures that all user data related to workouts and progress synchronizes across any iOS device.		
Trigger: Data changes on one device (for example, a user logs a new workout). **Type** (mark one): _X__ External ___ Temporal (rare)		
Relationships: **Association**: Administrators make sure data is synchronizing correctly. **Include**: **Extend**: **Generalization**: Users interact with the app across multiple devices.		
The Normal Flow of Events: 1. A user logs a workout or changes settings on one device. 2. The system automatically detects the change and starts a synchronization process. 3. The system should transfer the updated data to a cloud server. 4. All other devices connected to the user's account get the latest data from the cloud server.		
Sub-flows: N/A		
Alternate/Exceptional Flows: 2.1 If the synchronization fails, then the system will try again. If it continues to fail, it'll ask the user to try again later. (Maybe the user can't connect to Wi-Fi)		
Special Requirements: Performance 1. If the network connection is fine, the data synchronization process should happen briefly after data changes. Security 1. Data needs to be encrypted		
To do/Issues: Test the synchronization before pushing the app out to ensure it works smoothly.		

Use Case Name: Administer User Accounts and App Maintenance	**ID:** UC-7	**Importance**: Should Have
Primary Actor: Administrator	**Use Case Type**: Should Have	
Supporting Actors: N/A		
Stakeholders and Interests: Administrators: Interested in efficiently managing user accounts and maintaining app stability. Users: Depend on reliable system performance and responsive support.		
Brief Description: This use case outlines the responsibilities of administrators in managing user accounts, performing system maintenance, and interacting with user feedback to ensure the smooth operation of MuscleMate.		
Trigger: There is a request for user account modifications, system updates, or user feedback. **Type** (mark one): _X__ External ___ Temporal (rare)		
Relationships: **Association**: < Administrators need to interact with various backend systems. **Include**: N/A **Extend**: N/A **Generalization**: Part of administrative duties		
The Normal Flow of Events: 1. Admin access a panel to create, change, or delete user accounts based on requests. 2. Admin schedules and performs software updates to address functionality or security issues. 3. Admin review user feedback and address it.		
Sub-flows: 1.1. If a user can't log in to their account, the admin can help them.		
Alternate/Exceptional Flows: 2.1. If a system update fails, retry. If it keeps failing, get technical support.		
Special Requirements: Security 1. Log and monitor administrative actions for unauthorized access or changes.		
To do/Issues: 1. Admin needs to be trained		

Use Case Name: View Client Progress–Coaching Portal	**ID:** UC-8	**Importance**: Should Have
Primary Actor: Coach	**Use Case Type**: Detail, Essential	
Supporting Actors: The database		
Stakeholders and Interests: Coaches: Interested in monitoring client workouts and progress to help their clients. Clients: Benefit from personalized coaching based on their progress.		
Brief Description: This use case enables coaches to access client progress details.		
Trigger: A coach logs into their portal to view clients. **Type** (mark one): __X_ External ___ Temporal (rare)		
Relationships: **Association**: Coaches **Include**: "Log/Edit Workout" and "View Progress"- both need to be included so a coach can view a client's progress. **Extend**: N/A **Generalization**: Part of a coach's job.		
The Normal Flow of Events: 1. The coach logs into the coaching portal and selects a client to view detailed progress reports. 2. Depending on what the user allows to be shared, the coach can view a client's workout history and 1RM (rep max) estimation graphs.		
Sub-flows: 2.1. If the user allows, the coach can see all the details within a workout, including the number of sets, reps, and weights they did for an exercise.		
Alternate/Exceptional Flows: 2.1. If a coach wants to view progress on other lifts, they must pay $0.99/month. 2.2. The coach needs to contact technical support if their client's data cannot load.		
Special Requirements: Performance 1. The client's data should update after the client logs each workout Security 1. Ensure that all client data is accessible only to their coach with strict data access codes.		
To do/Issues: **1.** Develop and integrate the feedback module for the coach portal if it's not initially available. 2. The coach portal isn't apparent yet, so it might be better to launch in another version.		

Use Case Name: Export workouts/Progress Graphs	**ID:** UC-9	**Importance**: Should Have
Primary Actor: User	**Use Case Type**: Detail, Essential	
Supporting Actors: N/A		
Stakeholders and Interests: Users: Want to share their fitness achievements. Potential New Users: Friends of the user who share their progress or workouts might feel inspired to use MuscleMate.		
Brief Description: This use case enables users to share specific workout logs and progress graphs on social media (iMessage, Instagram, Facebook), including a link to download the MuscleMate app.		
Trigger: The user selects the "Share" icon after logging a workout or viewing their progress graph. **Type** (mark one): _X__ External ___ Temporal (rare)		
Relationships: **Association**: User **Include**: "Track Progress" and "Access Workout History."- To export a graph or workout, the user needs to have a workout history. **Extend**: N/A **Generalization**: N/A		
The Normal Flow of Events: 1. The user chooses a workout or a progress graph to share. 2. The app converts the selected data into an image, incorporating the app's branding and a link to download the app. 3. The user selects the platform (Facebook, Instagram) and sends the image.		
Sub-flows: N/A		
Alternate/Exceptional Flows: 3.1. If there are technical issues with sharing, the system alerts the user and suggests troubleshooting steps (ex., check network connection). The user can retry or save the image to their camera roll to share later.		
Special Requirements: Performance 1. Optimize the image for viewing on different platforms. Security 1. Ensure that sharing respects user privacy settings and data protection standards.		
To do/Issues: We should collect feedback to improve the process.		

6.0 System Evolution

Introduction

As MuscleMate grows beyond its MVP release, our roadmap includes introducing innovative features designed to enhance user experience, expand functionality, and ensure scalability to meet increasing user demands.

Features for Future Versions

- **Social Interaction Enhancements:** Future versions of MuscleMate will enhance social interaction capabilities, allowing users to connect with friends, share workouts, and engage in fitness challenges within the app. These features aim to build a tight fitness community within the app, enhancing user engagement and motivating our users.
- **Coaching Portal Improvements:** We plan to upgrade the Coaching Portal, adding functionalities that allow coaches to create personalized workout templates for their clients.
- **Diet and Progress Tracking:** A significant addition will be the integration of diet tracking and detailed progress analytics focused on fat loss and muscle gain. This feature is a premium option.
- **Wearable Device Integration:** Ensuring compatibility with the Apple Watch will allow users to sync their workouts and health data seamlessly, enhancing the app's utility and convenience.

Recommended Hardware/Software Upgrades

- **Server Infrastructure:** If the app gains significant traction, an upgrade to our server infrastructure will be necessary to support the added features while maintaining performance.
- **Security Enhancements:** As we integrate more social features and connect with third-party devices, strong security measures will be necessary to protect user data and maintain trust.

Conclusion

Our commitment to delivering a good user guides our strategic planning for these enhancements. By advancing these new features, MuscleMate aims to meet and exceed our users' expectations, fostering long-term engagement. User feedback will play a key role in refining these features, ensuring they effectively meet the needs of our community.

7.0 Conclusions and Recommendations

Conclusion:

We expect MuscleMate to revolutionize the fitness app market, especially for advanced lifters. Our system proposal shows that MuscleMate fulfills a specific need through a robust technical and strategic approach. The app enhances the workout experience by focusing on key features, like workout logging, template creation, and progress tracking, without the unnecessary clutter of beginner-oriented features.

Medium-risk factors related to the project's size and technical complexities are manageable. Through strategic planning and a skilled development team, we are confident that MuscleMate will stand out by offering a service that gym enthusiasts need and want.

Recommendations:

To ensure MuscleMate's success, follow these recommendations:

1. **Validate and Enhance Requirements:**
   1.1. Continuously review and update the functional and non-functional requirements to align with user needs and project goals.
2. **Strengthen Data Management:**
   2.1. Develop a comprehensive data strategy to support all app functionalities.
3. **Mitigate Feasibility Risks:**
   3.1. Address risk factors identified in the feasibility analysis by allocating resources strategically and planning for potential technical challenges.
4. **Enhance Security Measures:**
   4.1. Implement advanced security protocols, including data encryption and multi-factor authentication, to protect user information and maintain data integrity.
5. **Monitor and Adjust Project Scope:**
   5.1. Regularly assess project scope to meet stakeholder expectations.
6. **Plan for Scalability:**
   6.1. Prepare the infrastructure to scale efficiently as the user base grows, ensuring the app remains fast and responsive.
7. **Integrate with External Platforms:**
   7.1. Explore opportunities for integration with Apple Health.
8. **Establish a Continuous Feedback Loop:**
   8.1. Set up a way to receive ongoing user feedback to improve the app based on user preferences.

Appendices

MuscleMate App Survey:

Purpose: To understand what advanced lifters want from a workout app.

Demographic Information

Age:
Pronouns:
Training Experience (in years):

Current Fitness App Usage

- Do you currently use any fitness-tracking apps?
  - Yes
  - No
- If yes, which app(s) do you use?
- What do you like most about the current app(s) you use?
- What do you dislike about the current app(s) you use?

Workout Logging

- How important is the ease of logging workouts when choosing a fitness app?
  - Not important
  - Somewhat important
  - Very important
  - Essential
- What information do you consider essential to log for each workout? (Select all that apply)
  - Exercises
  - Sets and reps
  - Weight used
  - Rest periods
  - Total session duration
  - Notes or observations

Features and Functionality

- Which features would you value most in a fitness-tracking app? (Rank the features from most to least important)
  - Visual progress tracking (i.e. Graphs)
  - Workout templates (you create)
  - Social sharing capabilities
  - Data synchronization across devices
  - Form tutorials
  - Pre-made workout templates (we create for you)
- Would you prefer a fitness app that offers extensive guidance and tutorials or focuses on straightforward workout tracking?
  - Extensive guidance and tutorials
  - Straightforward workout tracking
  - No preference

Customization

- How important is personalizing the app's appearance and functions to fit your workout preferences? (For example, you can choose to display weights in pounds (lbs) or kilograms (kg) or change the language settings)
    - Not important
    - Somewhat important
    - Very important
    - Essential

Additional Feedback

- What other features or improvements would you suggest for MuscleMate?
-

Contact Information (Optional)

- Would you be interested in participating in a beta test of MuscleMate?
    - Yes
    - No
- **Email Address:**

Glossary

- **Agile Development:** A method of software development that uses an iterative process. Small, manageable phases of work, called sprints, help developers be flexible to change.
- **Authentication:** Verifies the identity of a user or device.
- **Cross-Platform Synchronization:** Allows using an app or system on multiple devices.
- **Customizable Templates:** Pre-designed workouts that users can change and apply across multiple workouts.
- **Data Encryption**: Converting data into code to prevent unauthorized access.
- **MVP (Minimum Viable Product):** A product with just enough features to satisfy early customers.
- **RPE (Rate of Perceived Exertion):** An exercise intensity scale. An RPE of ten would mean that the lifter has gone to failure. An RPE of five would mean the lifter is 5 reps from failure.
- **U.I. (User Interface):** U.I. refers to the screens, buttons, and other visual elements you interact with when using an application.
- **U.X. (User Experience):** U. X. Includes the users' complete engagement with an application.
- **Wearable Devices:** Electronic technologies that users wear. For example, people use the Apple Watch for tracking physical activity and health metrics.
- **Resource Constraints:** The limitations on the availability of resources needed to complete a project.
- **Bodybuilder:** A person who trains to gain muscle.
- **Powerlifter:** A person who trains to gain strength.
- **Set:** A group of reps performed without resting.
- **Reps:** The number of times a person performs a specific exercise without stopping.
- **Intangible Benefits:** A type of benefit that isn’t quantifiable.
- **Tangible Benefits:** A type of benefit that is quantifiable.
- **Database:** A collection of data that is stored by computer systems.
- **Competitive Analysis: Identifying** and comparing competitors to your products.
- **Competitive Advantage:** The features that you have against your competitors.
- **Technical Feasibility:** This refers to how well our team can successfully design, develop, and implement the system.
- **Resource Feasibility** Refers to the availability of necessary resources to complete a project.
- **Schedule Feasibility:** This checks if a project can meet proposed timelines.
- **Legal and Contractual Feasibility:** This checks if our project conflicts with legal or contractual requirements.

Bibliography

- Larman, C. (Year). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative*. Third Edition.
- Statista. (2023). *Revenue split of leading digital content stores worldwide as of August 2023*. Retrieved from https://www.statista.com/statistics/975776/revenue-split-leading-digital-content-store-worldwide/
- Cameron Andy. CSC 3150 – Systems Design. "Feasibility and Risk"
- "Tag: product management" | Kiran Johns.

  Retrieved from https://kiranjohns.com/tags/product-management/3/
- "Is it free to publish an app on app store?"–thecubanrevolution.com.

  Retrieved from https://thecubanrevolution.com/news/is-it-free-to-publish-app-on-app-store/
- *Intelligent diagramming*. Lucidchart. http://lucidchart.com/