Contents

[Introduction: 2](#_Toc132841599)

[Requirements and Planning: 2](#_Toc132841600)

[Goals and Objectives of the Project 2](#_Toc132841601)

[Activity-On-Node (AON): 2](#_Toc132841602)

[Critical Path: 3](#_Toc132841603)

[System Design: 4](#_Toc132841604)

[Actors: 5](#_Toc132841605)

[Requirements 5](#_Toc132841606)

[Functional Requirements: 5](#_Toc132841607)

[Non-Functional Requirements: 6](#_Toc132841608)

[Usability Requirements: 6](#_Toc132841609)

[Performance Requirements: 7](#_Toc132841610)

[Reliability Requirements: 7](#_Toc132841611)

[User stories of the system: 7](#_Toc132841612)

[Work breakdown Structure based on user stories 7](#_Toc132841613)

[Use Cases 9](#_Toc132841614)

[Use case: Create new account 9](#_Toc132841615)

[Use case: Record fitness activities 10](#_Toc132841616)

[Use case: View workout history and statistics 10](#_Toc132841617)

[Use case: Update subscription plan 11](#_Toc132841618)

[UML Diagram Use Case: 12](#_Toc132841619)

[Development Team Requirement: 12](#_Toc132841620)

[Completion of Application: 13](#_Toc132841621)

[Conclusion 13](#_Toc132841622)

[Reference: 13](#_Toc132841623)

# Introduction:

The proposed product is named "Fitness App" and its goals include finding a means to schedule training sessions with trainers, offering exercise regimens, and managing a healthy lifestyle. The purpose of this document is to provide comprehensive explanations of the specifications for the Fitness software application. It will outline the system's whole goal and functionality. The restriction of its interface and interaction with external apps is also explained in this document. This paper serves as a guide for the development team building the software systems as well as its stakeholders. The system's scope comprises a client-server system with a mobile app for activity tracking, goal selection, customer management, and subscription management.

The basic functionality of the system is customers may set a goal using the system, such as weight reduction, muscle building, endurance, etc., and optional extras like coaching sessions and diet programs may also be selected.

# Requirements and Planning:

People experience stress from their jobs and personal lives in the contemporary world, and they also lead unhealthy lifestyles, which contribute to health problems like obesity. It's crucial to maintain excellent health and a healthy physique. People may better control their health by using the fitness app. In order to measure users' progress and health, fitness monitoring systems need them to log their workouts on a website or a mobile app.

## Goals and Objectives of the Project

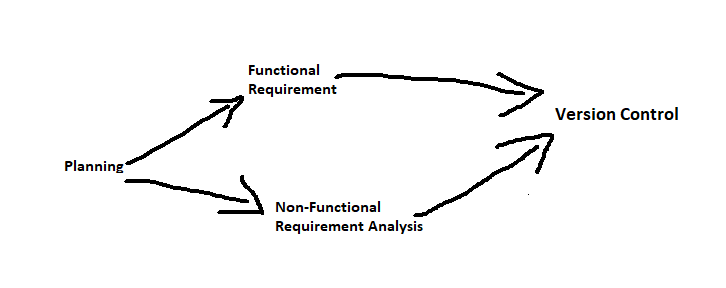
The Fitness App will be available for the two most popular mobile operating systems (iOS and Android). Users of the gym will be able to schedule sessions with personal trainers and tailor their exercise routines to achieve specific objectives (such weight loss, muscle gain, general fitness, etc.) via a dedicated app. Customers may tailor their fitness app to include dietary tracking. Achieving one's ideal physical form is now within the reach of every gym-goer who takes this step. Those who want to acquire or reduce weight strategically may find this software very useful. Anyone looking for a dependable resource where they can get a detailed strategy with videos to help them lose, maintain, or gain weight would benefit from reading this paper.

## Activity-On-Node (AON):

A causal network depicting processes as nodes and arrows as their connections. AON diagrams depict four different kinds of connections, as opposed to only one kind in Activity-on-arrow diagrams, which are relationships from end to start.

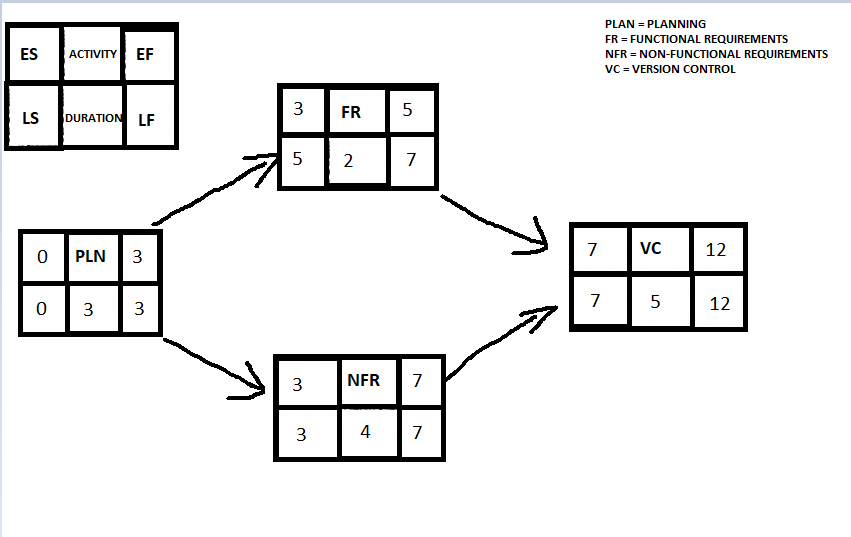
An AON or AOA graph incorporating duration estimates and dependencies.

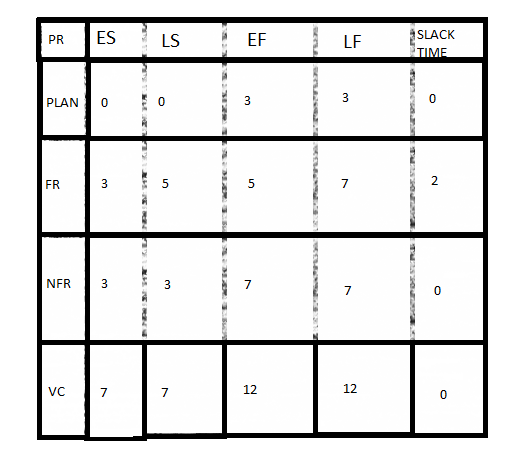
|  |  |  |
| --- | --- | --- |
| Activity | Predecessor | Duration (days) |
| Planning | --------- | 3 |
| Functional Requirement Analysis | Planning | 2 |
| Non-Functional Requirement Analysis | Planning | 4 |
| Version Control | Planning, Functional and non-functional Requirement Analysis | 5 |



## Critical Path:

Planning -> nonfunctional requirements -> version control





# System Design:

The following actors involved in the system.

## Actors:

1. Customer:

  It is the main actor in the system as it uses the application for fitness buying a subscription plan if there is any.

1. Administrator:

The actor who will manages the application, like adding or removing goals and managing customers’ accounts etc.

1. Fittrack smartwatch:

This will collect all user data and stores and send it to the connected mobile.

1. Payment Gateway:

This actor will handle all the payment related work like subscription plans activation or deactivation.

1. Third party service provider:

  These actors are going to provide additional services like nutrition plans or gym instruments etc.

# Requirements

## Functional Requirements:

1. The application allow the customer to select a goal like weight lose , muscle gain etc.
2. The application allows the customer to select optional extras like nutration plans and coaching classes
3. The application allows the administrator to add new goals remove existing goals
4. The application allows the customer to create and login to the account on application
5. The application sends periodic notifications to customer mobile/smartwatch
6. The application gives a functionality to allow customers to view their account details
7. The application allows the customer to Active/Cancel subscription plan at any time
8. The application should collect data timely and save from the FITTRACK smartwatch
9. All visualizations in the application should be made by consideration of color-blind friendly
10. The application allow administration to manage customers’ accounts

## Non-Functional Requirements:

## Non-functional requirements are those that focus on the overall behavior of the system rather than specific actions or outcomes. These are examples of non-functional requirements:

## There is a data-backup system.

## It is simple to maintain, or maintainability.

## Effectiveness and Rapidity of Action

## Ease of use for the intended audience

## Expansion potential: must be upgradeable or future-proof

## It must be risk-free to employ.

## Outstanding Efficiency

## User-Friendly

## Quicker responses

## Usability Requirements:

1.The application is simple and intuitive GUI so that user use it easily

2.the application should have a clear and concise instructions on how to use the application

3.The application should give feedback on every activity performance like login so system display that login successful

4.The application have a search bar functionality so that user easily find things

5.the application should give an easy and understandable message in case of any error by the user

## Performance Requirements:

1. The application able to handle large number of users account on concurrent time without being crash
2. The application should have a fast-loading time to provide a seamless user experience
3. The application is able to transfer large amount of data between application to server
4. The application have low latency to minimize delay
5. The application have the scalable infrastructure

## Reliability Requirements:

1. The application has the backup and recovery functionalities
2. The application has failover functionality to ensure the high availability in case of the server or network failure
3. The application has an error monitoring and reporting functionalities to quickly and easily identify and resolve issue
4. Application has a data integrity to prevent the data loss

## User stories of the system:

1. As a customer I can create account to use the application Fit Track
2. As a customer I can select the fitness goals and view the personalized workout plans and nutrition's
3. As a customer I can record my fitness activities like running or weightlifting
4. As a customer I can view my progress toward my fitness goals metrics
5. As a customer I can view my workout history and statistics
6. As a customer I want to receive personalized feedback and suggestions for improvement based on my given fitness data
7. As an administrator I can manage customer accounts like their subscription plans
8. As an administrator I can add new fitness goals and extra features
9. As an administrator I can view system logs and data to monitor system performance
10. As a customer I am able to connect my fit track smartwatch to the system

## Work breakdown Structure based on user stories

1. Create new account
   * Design account registration form
   * Implement account registration form
   * Add account validation and error handling
2. Select fitness goal and view recommendations
   * Create database schema to store fitness goals
   * Develop algorithm to generate personalized workout plans and nutrition recommendations
   * Integrate algorithm with app to display recommendations to customers
3. Record fitness activities
   * Design activity tracking feature
   * Implement activity tracking feature, including integration with FitTrack smartwatch
   * Test activity tracking feature to ensure accuracy and reliability
4. View progress towards fitness goal
   * Create dashboard to display progress metrics, such as weight, body fat percentage, and activity levels
   * Implement data visualization to display progress over time
   * Incorporate feedback and suggestions for improvement based on progress data
5. View workout history and statistics
   * Design workout history feature
   * Implement workout history feature, including storage of data in database
   * Develop data analysis tools to generate statistics, such as average heart rate and calories burned per session
6. Receive personalized feedback and suggestions
   * Create algorithm to analyze fitness data and provide feedback
   * Integrate algorithm with app to display feedback and suggestions to customers
   * Test feedback and suggestion feature to ensure accuracy and effectiveness
7. Manage customer accounts and subscriptions
   * Design account and subscription management system for administrators
   * Implement account and subscription management system, including user interface and database integration
   * Add error handling and security features to ensure data privacy
8. Add new fitness goals and extras
   * Research industry trends and customer needs to identify potential new goals and extras
   * Design and implement user interface to add new goals and extras to the system
   * Test new goals and extras to ensure accuracy and effectiveness
9. Monitor system performance and identify issues
   * Develop system logs to track usage and performance data
   * Implement data analysis tools to identify potential issues or areas for improvement
   * Work with development team to address identified issues and implement improvements
10. Sync Fit Track smartwatch data

* Develop algorithm to sync data from smartwatch to app and server
* Integrate algorithm with app and server to ensure data accuracy and consistency
* Test data sync feature to ensure reliability and effectiveness

# Use Cases

## Use case: Create new account

* Actors: Customer
* Main flow:
  + Customer launches the FitTrack app
  + Customer selects "Create New Account" option
  + Customer enters their personal details, including name, email, and password
  + System verifies the email and password meet the requirements
  + System creates a new customer account and sends a verification email to the customer
  + Customer verifies their email by clicking on the verification link in the email
  + System confirms the account creation and logs the customer in

Extensions:

* + If the customer enters invalid or incomplete information, the system prompts them to correct it before creating the account
  + If the email is already associated with an existing account, the system prompts the customer to log in or recover their password

## Use case: Record fitness activities

* Actors: Customer
* Main flow:
  + Customer launches the Fit Track app
  + Customer selects "Record Activity" option
  + Customer selects the activity type, such as running or weightlifting
  + Customer enters the details of the activity, including duration, distance, and intensity
  + System records the activity and updates the customer's progress towards their selected fitness goal
* Extensions:
  + If the customer selects an invalid or unsupported activity type, the system prompts them to select a valid type
  + If the customer enters invalid or incomplete activity details, the system prompts them to correct it before recording the activity

## Use case: View workout history and statistics

* Actors: Customer
* Main flow:
  + Customer launches the FitTrack app
  + Customer selects "View Workouts" option
  + System displays a list of the customer's recorded workouts, including activity type, date, and duration
  + Customer selects a specific workout to view more details
  + System displays detailed workout statistics, including heart rate, calories burned, and distance covered
* Extensions:
  + If the customer has not recorded any workouts yet, the system displays a message indicating that there are no recorded workouts to view
  + If the customer selects a non-existent workout, the system displays an error message

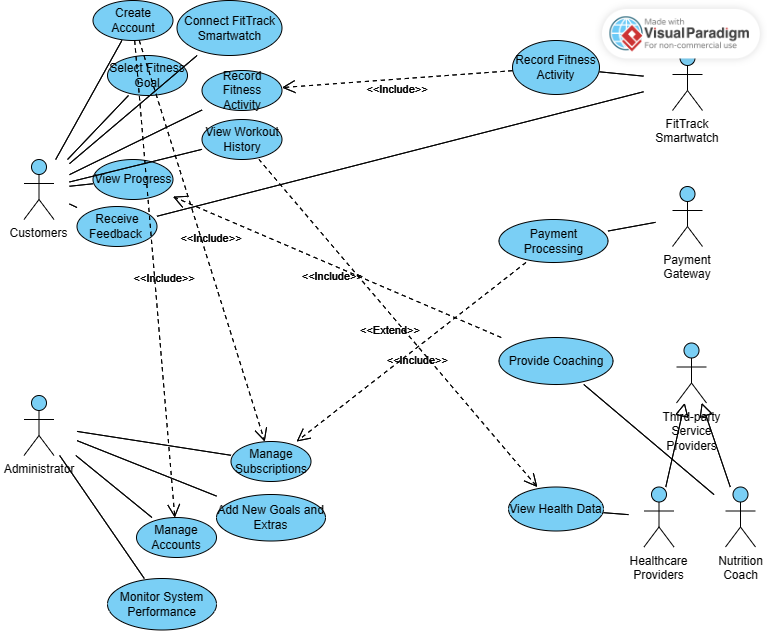
## Use case: Update subscription plan

* Actors: Customer
* Main flow:
  + Customer logs into the FitTrack website
  + Customer selects "Subscription" option
  + Customer selects "Update Plan" option
  + Customer selects the new subscription plan they want to switch to
  + System verifies that the customer is eligible to switch to the new plan based on their current subscription and payment status
  + System updates the customer's subscription plan and payment information

Extensions:

* + If the customer has any outstanding payments, the system prompts them to pay them before switching to a new plan
  + If the customer is not eligible to switch to the new plan, the system displays an error message explaining why

## UML Diagram Use Case:



(UML Diagram of FitTrack App, Source: Created using visual paradigm)

# Development Team Requirement:

The development team will require a master branch and a development branch. The master branch will include code that is stable and ready for release, while the development branch will have new features and bug fixes.

## Completion of Application:

The team followed the method outlined in the previous section, generating many branches for specific features and milestones. The main branch is the master branch, which contains the most current stable version of the code. The team has also created a dev branch for the creation and testing of new features.

After reviewing the code changes, they may be merged into the relevant branch. To guarantee a seamless merging procedure, stick to the branching method.

Changes should be pushed to a remote repository: The modifications should be pushed after testing.

Conclusion:

We developed this application to maintain the record of the customers and we must protect this application from hackers in order for fitness monitoring systems to function, users must input their data into a computer or mobile app. Customers may choose a goal from a list that includes weight reduction, muscle building, endurance, and more, and then customize their experience with the system by adding features like dietary planning and coaching sessions. Integration with a FitTrack smart watch for step and heart rate monitoring is also desirable. The administrative tasks associated with running a company must also be supported by the system in question. This includes, but is not limited to, client administration, keeping track of available objectives and extras, etc. We will update this application as per requirement of the customers and protect their privacy.

# Reference:

* + - 1. Collado-Borrell, R., Escudero-Vilaplana, V., Villanueva-Bueno, C., Herranz-Alonso, A., & Sanjurjo-Saez, M. (2020). Features and functionalities of smartphone apps related to COVID-19: systematic search in app stores and content analysis. *Journal of medical Internet research*, *22*(8), e20334.
      2. Mohanty, B., Chughtai, A., & Rabhi, F. (2019). Use of Mobile Apps for epidemic surveillance and response–availability and gaps. *Global Biosecurity*, *1*(1).
      3. Alfawzan, N., Christen, M., Spitale, G., & Biller-Andorno, N. (2022). Privacy, Data Sharing, and Data Security Policies of Women’s mHealth Apps: Scoping Review and Content Analysis. *JMIR mHealth and uHealth*, *10*(5), e33735.
      4. Tsui, F., Karam, O., & Bernal, B. (2022). *Essentials of software engineering*. Jones & Bartlett Learning.
      5. Laplante, P. A., & Kassab, M. (2022). *What every engineer should know about software engineering*. CRC Press.
      6. Ludewig, J. (2003). Models in software engineering–an introduction. *Software and Systems Modeling*, *2*, 5-14.