你可能需要修改一下文字的格式，有一些nfr和soft goal 有些混淆，但是没有

文字的排版需要根据你的要求修改

S4

Use case 1

Primary Actor: mobile user

Precondition: net connection with app downloaded and minimum Android/Apple version

Trigger: mobile user want to show friend his society ranking

**Main Success Scenario**

1. Mobile user click the ACME run app.

2. The server (ACME system) receives the request header.

3. Upon receiving the message, the user is asked if he/she has hidden his/her personal information.

4. If it is not hidden, the mobile user's personal information is sorted in the community ranking.

5 Community ranking is displayed to the user and after calculating the user's preferences, the ranking screen is displayed

6. The ranking screen displays the user's preferences, the community ranking and encourages users to share.

**Secondary Scenarios**

1.1. Discover that a user's login have expired and request a new login

3.1. Find that no information about the user can be found in the system.

4.1. The system storing the community rankings crashes due to too many requests.

4.2. After the information has been collected and the report has been created, the user is logged out, or the user did not receive the information due to the network.

**Success Postcondition**: The user receives the ranking screen or the information is closed after sharing is complete.

This scenario is very important, and is the most important part of improving the community function requirement, which satisfies the main needs of social types of users, because for many beginners of sports, their personality is **exogenous**, because a large number of athletes, especially beginners, need to accomplish their goals by comparing themselves with others. A lot of sports exercise software, go in trash because over complicated social system. Community type ranking information and Synchronized sharing with friends and family with various permissions, should not be separated into 2 part. So I would like to simplify the social system. Preferably with different sharing setting and layout for different community platforms

A diagram of a program

Description automatically generated

Use Case 2

Primary Actor: Mobile user

Precondition: net connection with app downloaded and minimum Android/Apple version Trigger: The mobile user want jogging activity and record the information.

Main Success Scenario:

1. The mobile user opens the ACME run app on their device.

2. The user navigates to the "WORKOUT" tab within the app.

3. The app prompts the user to confirm the start of the jogging activity.

4. The device's sensors(watch or band or cell phone), such as GPS and accelerometer, track and record relevant information

5. As the jogging activity progresses, real-time updates on distance and time are displayed on the user's device.

6. Upon completion of the jogging activity, the user stops the recording within the app.

7. The app processes the recorded information and updates the user's personal activity history with the details of the completed jogging activity.

8. The recorded jogging activity contributes to the user's overall health and workout statistics within the app.

Secondary Scenarios:

2.1. If the device loses GPS signal during the activity, the app may prompt the user to confirm the end of the activity based on available data.

3.1. In case of a sudden app crash, the system could have an auto-save feature that preserves recorded data up to that point, allowing the user to resume or recover the activity log.

Success Postcondition: The user's jogging activity information is successfully recorded and reflected in their personal workout history within the ACME run app, contributing to their overall fitness statistics.

S1

1 User Profile System: The component in charge of customer personal profiles (in global server), enhancing the app's ability to offer customized running plans(offer plan) and challenges tailored to individual fitness levels and preferences(modify plan).

2 Society System (Social Networking): The component in charge of society system. By allowing users to connect with friends (with position character system to create local rank) join running groups, and share achievements, this system fosters a supportive community and motivates users through social engagement and healthy competition.

3 Workout System: The component in charge of scientific arrangement and connect with social system. Offering customized training plans and adaptive running programs in global rank,(global rank) this system assists users in improving running skills and achieving their personal fitness goals effectively.

4 Tracking and Analytics System: The component in charge of testing training achievement and react with S1.2 S1.3. This system tracks running activities and provides analytical insights. (track and analysis)Enabling users to monitor their performance, set personal benchmarks, and track their progress. Partial configured messages will be automatically shared with friends who have access to them.

5 Notification System: The component in charge of team work with friends and mutual supervision(react with notification system) With personalized reminders and motivational messages, this system keeps users engaged and informed, encouraging a consistent and disciplined approach to their running routines.

The User Profile System customizes running plans and challenges based on user data stored globally, directly enhancing the app's personalization capabilities. The Friend Society System leverages social networking to build community, motivate through local rankings, and encourage competition. The Workout System aligns scientific training with global rankings, aiding users in advancing their running proficiency and achieving fitness goals. Tracking and Analytics monitor and analyze performance, allowing for progress sharing with friends, while also reacting to workout and profile data. Lastly, the Notification System sends tailored reminders and messages, promoting consistency in running routines through collaborative and supervised

S2

S21

User Profile System

Functional Requirements:

1 The system allow users(mobile and pc) to register and create a personal profile with details such as age, weight and fitness goals.

2 The system allow users to modify their fitness plans and goals based on their performance data and preferences.

Non-Functional Requirements:

1 User data must be encrypted and securely stored to protect privacy(Confidentiality).

2 The management interface should be accessible on various devices.

S22

Society System (Social Networking)

Functional Requirements:

1 The system allow friend requests, group creation and simple chat(without long video and large zip bag)

2 The system provide a local rank system based on users' running activities.

Non-Functional Requirements:

1 The system must be able to handle high traffic and data throughput to ensure real-time updates.

S23

Workout System

Functional Requirements:

1 The system adapt workout(achievement) plans based on User Profiles and Tracking and Analytics System.

2 The system adapt with the social system to allow for group challenges and shared achievement (in time or delayed).

Non-Functional Requirements:

1 The workout generation algorithm needs to perform efficiently, providing quick responses to user inputs.

2 The system present an easy-to-understand and clear format(shared pdf) for users of all levels.

S24

Tracking and Analytics System

Functional Requirements:

1 The system accurately track user activities, including distance, time, pace, and calorie.

Non-Functional Requirements:

1 The system must be accurate and reliable, with minimal downtime.

2 The system presented in an intuitive manner, with the capability for users to customize reports.

S25

Notification System

Functional Requirements:

1 The system send personalized notifications for workouts system, Tracking and Analytics System, and Society System.

2 It should allow users to set reminders for upcoming events or training sessions.

Non-Functional Requirements:

1 The system must deliver without significant delays.

2 The system should minimize the risk of sending redundant or irrelevant notifications to avoid user fatigue.(avoid overlap notification)

S3

1

The api system will prevent the access of other source of request and only respond to the nano source of auth request.

The given sentence suggests the implementation of an API system with a stringent access control mechanism. This system is designed to block requests originating from any source other than the specified "nano source" that possesses the appropriate authentication credentials. Essentially, the API is configured to only acknowledge and respond to requests that meet the specific criteria associated with the nano source. By this method we can ensure a highly secure and restricted access environment.

2

The user system will have 4 basic down tab. HEATH which show the recent body activity the second tab is WORKOUT show recent exercise or running cycling activity you chosen the third is DEVICE show the band or phone we use and information we sync recently. The forth tab is WE LIFE and the social system and setting of application.

+-------------------------------------------------------+

| User System |

| |

| +------------------+ +------------------+ |

| | HEALTH | | WORKOUT | |

| | - Recent Body | | - Recent Exercise| |

| | Activity | | or Running | |

| | | | Cycling | |

| +------------------+ +------------------+ |

| |

| +------------------+ +------------------+ |

| | DEVICE | | WE LIFE | |

| | - Band or Phone | | - Social System | |

| | Information | | - Settings | |

| | Synced | +------------------+ |

| +------------------+ |

|  |
| --- |
|  |

S4

Critical Functions:

(S.2.1) User Profile System serves as a foundational and fundamental component of other part. It stores and manages user information, including personal details, preferences, and activity history. This system enables personalized experiences by tailoring content and features based on individual user profiles. Authentication mechanisms are integrated into the User Profile System to ensure secure access and protect user data.

USE CASE 2 (S.4.1): The WORKOUT behavior is the accurate recording of body index and exercise activities. This includes functionalities like real-time tracking, data precision, and seamless integration with device sensors. SO The highest priority is given to the workout system, emphasizing functionalities like accurate activity tracking, exercise logging, and real-time feedback during workouts. This ensures that users can effectively use the app for their primary fitness activities.

Important Functions:

USE CASE 2 (S.4.2) While social sharing interfaces are crucial, interfaces supporting additional workout features, like guided workouts or training plans, are also important. Certainly, leveraging the popularity of the sharing page to make the app more engaging, fun, and supportive can significantly contribute to the app's success.

Good to have

(S.2.4) Gamification Elements: Introduce gamification features like badges, achievements, or challenges to motivate users and make the app more entertaining.

(S.2.5) Notify and Personalized Recommendations: Provide personalized workout recommendations based on user preferences, goals, and historical activity data. Notifications serve as reminders for upcoming workouts, encouraging users to stay consistent with their fitness routines.

S.6

User Profile System (S21)

1 Test user registration and profile creation on both mobile and PC platforms. Ensure that age, weight, and fitness goal details are accurately captured and stored.

2 Modify fitness plans and goals for a user based on performance data. Confirm that changes are reflected in the user's profile and impact subsequent workout plans.

Society System (Social Networking) (S22)

1 Test the functionality of sending and accepting friend requests, creating groups, and engaging in simple chat. Ensure that chat functionality works without issues.

2 Test the rank system by simulating running activities. Confirm that the local rank system accurately reflects users' running activities.

Workout System (S23)

1 Modify user profiles and track the impact on generated workout plans. Confirm that the system adapts workout plans based on user profiles and achievements.

2 Test the integration with the social system by participating in group challenges and confirming shared achievements.

Tracking and Analytics System (S24)

1 Perform various activities and validate that the system accurately tracks distance, time, pace, and calories. Cross-verify tracked data with manual measurements.

Notification System (S25)

1 Trigger various events related to workout, tracking, and social activities. Confirm that personalized notifications are received promptly and accurately reflect the specified events.

2 Allow users to set reminders for events or training sessions. Confirm that reminders are delivered at the specified times.

3 Test the system with various scenarios to ensure that notifications are not redundant or irrelevant. Verify that the system intelligently manages notifications to prevent user over-fatigue.

P1

Project Manager

1 Lead the planning, execution, and closure of the project. Ensure tech team with goals and objectives.

2 Identify potential risks and modify plans. Monitor and manage risks throughout the project lifecycle.

3 Allocate resources efficiently to meet project objectives.vManage the project budget and timeline.

Product Owner

1 Clearly articulate the product's vision and goals. Define and prioritize features based on business value.

2 Engage with stakeholders to gather and understand requirements. Maintain a well-prioritized product backlog.

3 Make decisions on feature priorities and trade-offs. Define the acceptance criteria for user stories.

4 Participate in sprint planning, reviews, and retrospectives. Provide feedback on delivered increments.

Investor

Financial Support: PAY MONEY

Development Team (Developers, Designers, Testers)

Developers:

1 Write code based on requirements and design specifications.

2 Collaborate with the team to implement software solutions.

3 Conduct code reviews to maintain code quality.

Designers:

1 Create user interface and experience designs.

2 Ensure a visually appealing and user-friendly product.

3 Collaborate with developers to implement designs.

Testers:

1 Develop and execute test plans for software quality assurance.

2 Identify and document defects.

3 Collaborate with developers to resolve issues.

P2

Choose a cross-platform development framework that allows code sharing between mobile and web platforms.

React Native Leverages React for building user interfaces. Supports both iOS and Android mobile platforms and can be extended to web applications using tools like React Native Web.

P3

Week 1: Project Kickoff

Start the project, anticipate problems that may occur, and in order to prevent the plan from not being implemented successfully, it is important to clarify everyone's responsibilities. Define project scope, objectives, and success criteria.Identify key stakeholders and establish communication channels.

Week 2: Design and Architecture

Develop wireframes and user interface designs.Define the technical architecture and choose development tools.

Week 3-4: Backend Development

Implement user and database integration.

Week 5-7: Cross-Platform Framework Implementation

Using the chosen cross-platform framework, Integrate basic functionalities for user registration and profile creation.

Week 8-9: Feature Development

Implement core features such as workout tracking, social system functionalities, and notification integration.

Week 10-12: Refinement and Testing

Conduct thorough testing, including unit testing and user acceptance testing.

P4

Week 1: Project Kickoff

Task:

Project Initiation Hold a kickoff meeting to introduce the project.

Expected Outcome:

Clearly defined project scope, objectives, and success criteria.

Task:

Identify key stakeholders and their roles.

Expected Outcome: Comprehensive stakeholder list and communication plan.

Week 2: Design and Architecture

Task: Wireframes and UI Designs

Develop wireframes for user interfaces. Expected Outcome: Completed wireframes and initial UI designs.

Week 3-4: Backend Development

Task: Backend Infrastructure Setup. Set up the backend infrastructure.

Expected Outcome: Functioning backend infrastructure.

Task: User Integration and Database Integration Integrate the database for data storage.

Expected Outcome: Database integration for user-related data.

Week 5-7: Cross-Platform Framework Implementation

Task: Cross-Platform Framework

Expected Outcome: Framework ready for development.

Task: Implement basic functionalities for user registration and profile creation.

Expected Outcome: Users can register and create profiles.

Week 8-9: Feature Development

Task: Workout Tracking Implementation

Expected Outcome: Users can record and track their workouts.

Task: Social System Functionalities

Implement social features such as friend requests and group creation.

Expected Outcome: Functional social interaction features.

Week 10-12: Refinement and Testing

Task: Testing Plan

Expected Outcome: Documented testing plan.

Task: Conduct unit testing and user acceptance testing.

Expected Outcome: Identified and addressed issues, fully tested application.