Universidad Autónoma de Ciudad Juárez

Campus Ciudad Universitaria - IIT

Subject:

Métodos Formales en Especificaciones y Diseño de Software

Group: A

Project: BeFit

Professor: Benito Alan Ponce Rodriguez

Student Names:

Gabriel Alberto Vilchis Ríos - 215460

Diego Alejandro Jasso Fernández - 222804

Anna Lizbeth Barajas Sandoval - 222835

Luis Ubaldo Flores Pineda - 223046

Ricardo Rodríguez Ponce - 223229

Deadline: /2025

SRS Document

1. Introduction

The current environment of physical wellness and sports supplementation presents a complex dynamic where consumers are no longer guided solely by immediate need, but by a set of internal motivations that respond to their identity, purpose, and perception of health.

People are aware of their bodies and the benefits of functional nutrition, and they tend to seek products that represent an extension of their lifestyle, rather than simple supplements. This behavioral evolution reflects a transition from rational consumption to emotional consumption, where trust and authenticity become decisive factors (Solomon, 1994).

The fitness industry has a particular psychological makeup, where their purchasing behavior is determined by three fundamental approaches. According to authors Schmitt (1999) and Tajfel & Turner (2004), they mention that self-efficacy, coherence between brand and personal values, and the search for recognition and belonging are factors that drive emotional behavior. Brand relationships were measured by the feeling of being underwood rather than persuades, which reinforces loyalty and a sense of purpose (Hutmacher & Apple, 2023).

The field of supplements and holistic health products has experienced sustained expansion, driven by the growing interest in self-care, conscious nutrition, and physical performance optimization (Euromonitor International, 2024). However, market saturation has generated a competitive environment where differentiation no longer depends on price or composition, but rather on the psychological and emotional discourse that the brand conveys. In this context, trust emerges from brands that manage to connect their products with authentic experiences and values of genuine well-being, consolidating deeper and lasting relationships with consumers (Kotler, Kartajaya & Setiawan, 2021).

Another critical factor for this type of audience is consumer psychology, where a shift in interest from the material to the symbolic is observed. The product ceases to be an end or a necessity and becomes a means of self-realization. Personalization, underwood as a form of digital empathy, generates a perception of support that reinforces the satisfaction of basic psychological needs: autonomy, competence, and relatedness (Deci & Ryan, 2000).

1.1 Purpose

The purpose of this project, BeFit, is to provide an efficient and reliable e-commerce platform dedicated to the sale of fitness related products, including nutritional supplements such as protein powders, creatine, vitamins, collagen and pre-workout formulas, and more. The system aims to simplify the purchasing experience and amplify it by integrating a positioning test, which will allow the users to purchase products based on their actual needs, specifically tailored to them.

The purpose of this document is to formally define the software requirements for the BeFit e-commerce, in accordance with the IEEE 830-1998 standard. It serves as a reference for all stakeholders, including but not limited to: developers, designers, testers, and clients. The document establishes a shared understanding of the system's scope, objectives, and constraints. It also serves to ensure consistency throughout the project lifecycle, supporting the design, implementation, and maintenance of the system according to the specified requirements.

1.2 Scope

The scope of this project includes the design, development, and implementation of a specialized e-commerce platform tailored to the fitness and sports nutrition sector, with a focus on personalized product recommendations based on user profiles, and the creation of a loyal customer ecosystem through experiences related to their previous purchases. The platform will be designed to address the specific needs of fitness enthusiasts and health-conscious consumers, providing a solution for the acquisition of fitness-related products. Key elements within the scope include:

- Product Catalog and Personalized Recommendations: Development of a comprehensive product catalog that includes various supplements, multivitamins, protein shakes, and sports nutrition products. The platform will implement an intelligent recommendation engine based on user profiles, fitness goals, dietary preferences, and purchase history to provide personalized product suggestions.
- User Profiling and Positioning Test: Implementation of an onboarding questionnaire system that identifies user objectives (muscle gain, fat loss, energy maintenance, etc.), activity levels, dietary restrictions, and personal preferences. This data generates a personalized nutritional and training profile that drives product recommendations throughout the platform.
- Subscription Management System: Implementation of a flexible subscription model that allows users to receive personalized monthly product deliveries. Users can subscribe, pause, resume, or cancel their subscriptions at any time. The system automates recurring orders based on user profiles and preferences, ensuring continuous engagement.
- Order Management System: Implementation of an efficient order management system that supports the entire purchasing process, from order creation to payment processing and fulfillment tracking. This includes both subscription-based orders and individual one-time purchases.
- Shopping Cart and Inventory Management: Integration of a shopping cart system with real-time inventory tracking, ensuring that customers can make informed purchasing decisions. The system filters products by category, fitness objective, and physical activity level.
- Loyalty and Rewards Program: A comprehensive loyalty system that offers points, discounts, and rewards for recurring purchases and referrals. Users can track their progress within the loyalty program and redeem rewards directly in the store, fostering long-term customer retention.
- Communication and Notification System: Implementation of automated email services and push notifications for order confirmations, subscription renewals,

shipping updates, promotional offers, and personalized product recommendations on web and mobile platforms.

- User Experience: Design and development of an intuitive and minimalist user interface that enhances the customer experience, allowing fluid navigation and efficient use of the platform for fitness-conscious consumers. The design will be fully responsive, optimized for both web browsers and mobile devices.
- Administrative Dashboard: Development of an internal administration panel that allows administrators to manage products, view sales statistics, track active subscribers, generate automatic reports on revenue and costs, configure personalization tests, and monitor system performance.
- Direct Brand Partnerships: Integration of a partnership model that connects BeFit directly with certified supplement manufacturers and brands. This eliminates intermediary costs and marketing expenses, allowing competitive pricing while maintaining quality standards. Partner brands receive priority positioning on the platform.

1.3 Definitions, acronyms, and abbreviations

Terms	**Definition**
API	An API (Application Programming Interface) is a set of protocols that enable different software components to communicate and transfer data.
AWS	Amazon Web Services is a comprehensive and widely adopted cloud computing platform offered by Amazon. It provides a vast array of on-demand services such as computing power, storage, and databases, allowing businesses to scale and manage their IT resources efficiently.
Database	Database is a systematic collection of data. Databases support storage and manipulation of data. Databases make data management easy.
Deploying	The process of moving software code or updates from a development environment to a production environment where it becomes accessible to end users.
EC2	Elastic Compute Cloud.

Terms	Definition
Fitness Profile	Fitness Profile in the context of this project refers to the collection of data relating to the user's fitness goals, health lifestyle, and dietary preferences, which impact the product recommendation.
GPDR	General Data Protection Regulation.
Hosting	Service that allows individuals or organizations to publish a website or web application on the Internet.
HTTPS	Hypertext Transfer Protocol Secure
IAM	Identity and Access Management.
JSON	Java Script Object Notation is a lightweight and text-based data format for storing and exchanging information that is easy for both humans to read and machines to parse.
JWT	A JSON Web Token is an open standard for securely transmitting information between parties as a compact and self-contained JSON object.
LPDP	Ley de Protección de Datos Personales (Personal Data Protection Law).
RDS	Relational Database Service
Relational Database	A type of database that organizes data into tables with rows and columns, where data points are related to each other based on common attributes.
REST	A Representational State Transfer is an architectural style for designing networked applications. It is a set of constraints for building flexible, scalable, and lightweight APIs that use standard HTTP methods (GET, POST, PUT, DELETE) to communicate and exchange data between

Terms	Definition
	systems.
RESTful	Any artefact of software that follows the principles of REST.
S3	Simple Storage Service.
SRS	Software Requirements Specification
TLS	Transport Layer Security
UX	User Experience.
WCAG	Web Content Accessibility Guidelines

1.4 References

Deci, E. L., & Ryan, R. M. (2000). The "what" and "why" of goal pursuits: Human needs and the self-determination of behavior. Psychological Inquiry, 11(4), 227–268.

Euromonitor International. (2024). Health and Wellness Market Overview.

Hutmacher, F., & Appel, M. (2023). The psychology of personalization in digital environments: From motivation to well-being–a theoretical integration. Review of General Psychology, 27(1), 26-40.

Schmitt, B. (1999). Experiential marketing. Journal of marketing management, 15(1-3), 53-67.

Solomon, M. R., & Behavior, C. (1994). Buying, having and being. London: Prenticle Hall.

Tajfel, H., & Turner, J. C. (2004). The social identity theory of intergroup behavior. In Political psychology (pp. 276-293). Psychology Press.

Kotler, P., Kartajaya, H., & Setiawan, I. (2021). Marketing 5.0: Technology for humanity. John Wiley & Sons.

1.5 Overview

This present document serves as the formal Software Requirements Specification for the proposed project, BeFit. Its purpose is to offer a clear and comprehensive understanding of

the system’s requirements and behaviour, to both technical and non-technical readers as it guides them through the specifications of the system’s development and implementation.

The document is divided into 4 sections.

- Section 1 establishes the foundation of the project, offering context and general overview of the remainder of the document.
- Section 2 provides a general description of the system itself, including its main features, characteristics, and constraints.
- Section 3 details the project’s functional and non-functional requirements, specified according to the project’s objectives and needs.
- Finally, section 4 adds additional information regarding the project’s development and other documentation.

2. Overall description

2.1 Product perspective

BeFit is an e-commerce platform designed for the sale of fitness-related products, such as supplements, complements, shakes, and multivitamins. The project's strength lies in its target audience: a consistent consumer who prioritizes quality and functionality over price. Therefore, the system seeks to offer an environment where the user feels that their preferences lead them to the products they need.

The system is distinguished from other e-commerce sites by two key factors: personalization, which offers a warm approach by generating accurate recommendations based on a user profile (obtained from a placement test); and price competitiveness, which maintains low costs without sacrificing quality through direct partnerships with brands, reducing marketing and intermediary expenses.

For secure online payments, BeFit is affiliated with several online payment platforms, including PayPal and Stripe, in addition to accepting card payments, which guarantees the security of all transactions.

2.2 Products functions

2.2.1 General use case diagram

The use cases are divided into two main actors and the activities of each one, these will be users and administrators.

For users, the following activities are contemplated:

- **Sign up:** This can be done through two ways: the first one is an Email or a third party actor (in this case, Amazon Cognito) that can be used to create the account, and the second way is by manually entering the required information to register.

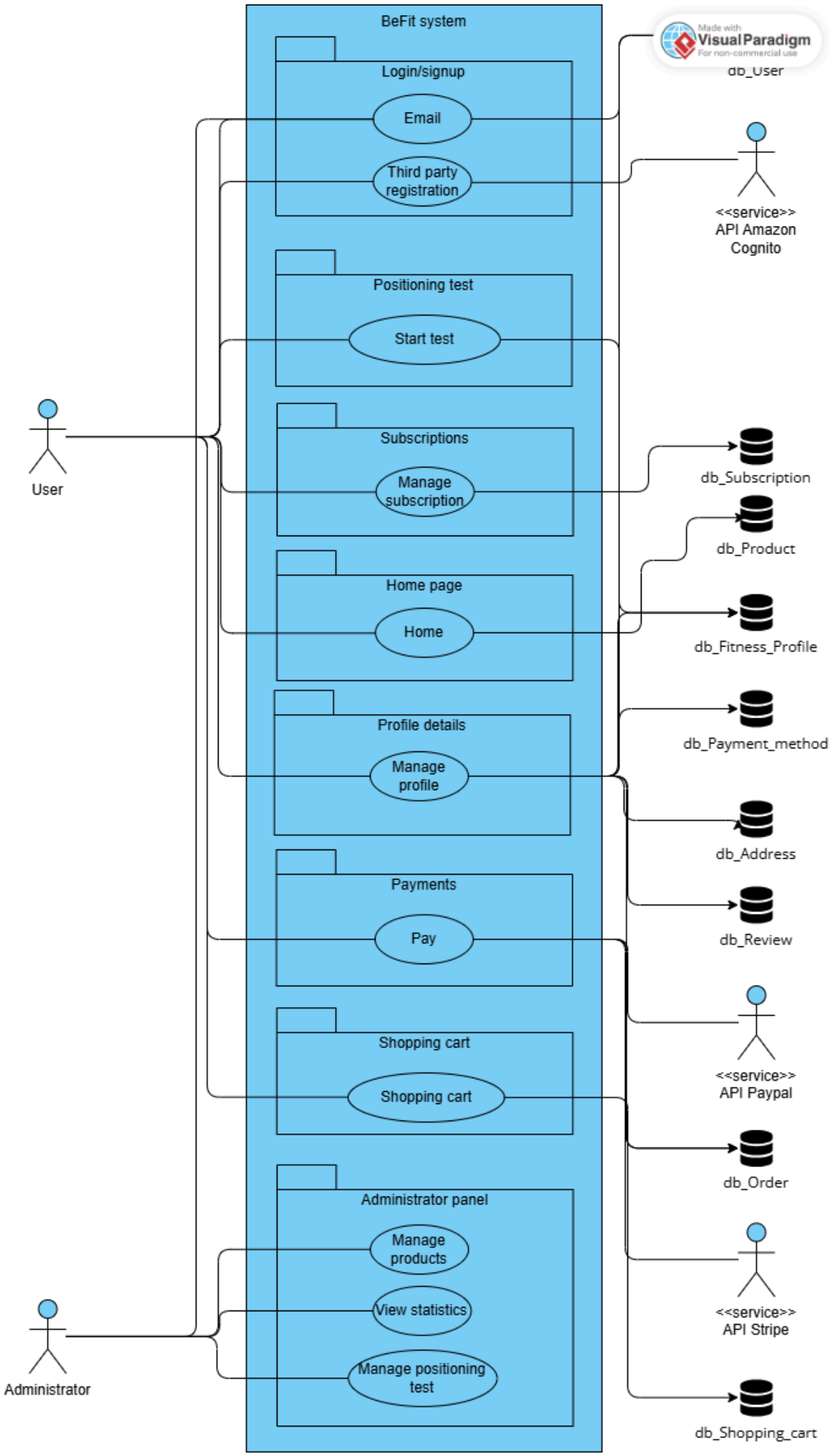
- **Log in:** Depending on the registration method used, the user will be able to enter their account. Additionally, a password recovery option is available in case the user forgets their password.
- **Take a positioning test:** The user is able to take a positioning test to receive recommendations and product suggestions based on their goals or answers.
- **Shopping cart:** The user can create a shopping cart, add the product in a shopping cart either by searching or using filters like a category, objective, and physical activity filters. And also, the user is able to delete a product and modify product quantities on their cart.
- **Home page:** The user can access the system´s main page, where they can view the product catalog, search for products, or search for products using category, objectives, and physical activity filters.
- **Select payment method at checkout:** The system has the option of using PayPal, or stripe to generate a voucher for the customer and pay for the product.
- **User profile:** The system has a profile for users, where they can modify their information, such as their first name, last name, date of birth, profile image, and gender. They can also select a payment method, which can be PayPal or credit/debit card. They also can modify their shipping information, including address, zip code, city, country, phone number and state.

  They will be able to check their purchase history of past orders, leave reviews of past purchases and the shipping status. They can also delete their account if they wish, and they can view their fitness profile to see the data given by the positioning test. Also, users can manage their subscription, they will be able to cancel, continue, or pause their subscription. Lastly, users can view their progress in the loyalty program to claim rewards in the system.

- **Manage subscriptions:** The user will be able to subscribe to a personalized monthly plan that will send specific products based on their profile.

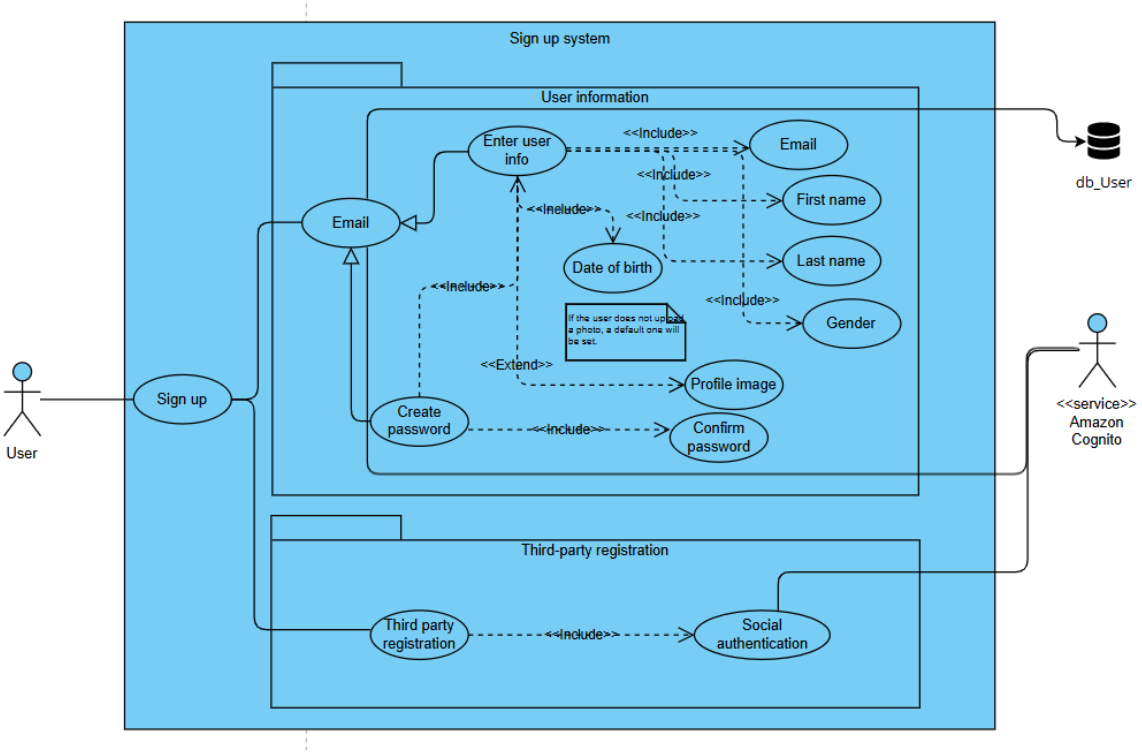
For administrator, the following activities are contemplated:

- **Sign up:** Unlike how the user creates the account, the administrator can only create it by entering data by data.
- **Log in:** The account creation parameters are followed, where the user enters their email and password to enter the site.
- **Administrator panel:** The system will display a panel where the administrator can manage products, the positioning test, and view statistics to generate reports of the system.


BeFit system
Login/signup
Email
Third party registration
Positioning test
Start test
Subscriptions
Manage subscription
Home page
Home
Profile details
Manage profile
Payments
Pay
Shopping cart
Shopping cart
Administrator panel
Manage products
View statistics
Manage positioning test
User
Administrator
db_User
<<service>>
API Amazon Cognito
db_Subscription
db_Product
db_Fitness_Profile
db_Payment_method
db_Address
db_Review
<<service>>
API Paypal
db_Order
<<service>>
API Stripe
db_Shopping_cart

2.2.2 Sign in

2.2.2.1 Sign in user



When a new user navigates into the account creation section of the system, they are presented with 2 options for signing up: using an email or a third party registration method:

Third party registration: If the user chooses to sign using a third party service, they can select two available options for signing up, Facebook or Google, each a different authentication flow managed through Amazon Cognito:

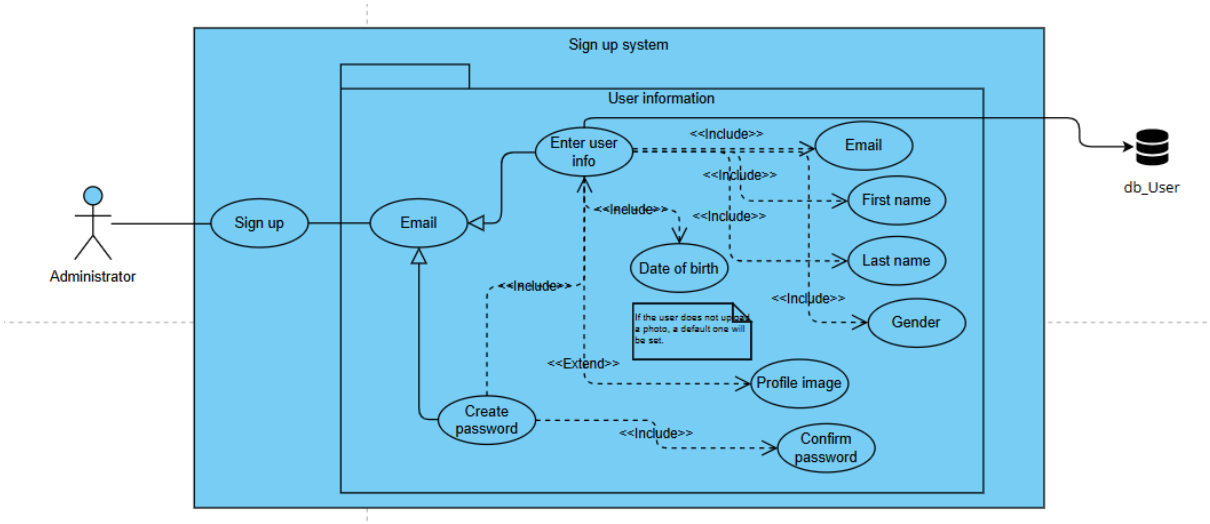
1. Sign in with Facebook.
2. Sign in with Google.

Email: If the user wants to create an account using their email, the process involves entering the following information:

- First name
- Last name
- Gender
- Birth date
- Profile image
- Password (which must be confirmed by re-entering it).

Once this process is finished, the account creation process is finalized.

2.2.2.2 Sign in administrator



This case is similar to the administrador, when this actor goes into the account creation section of the system, they are presented with only one option, using an email address.

Email: The administrator to create an account using their email, the process involves entering the following information:

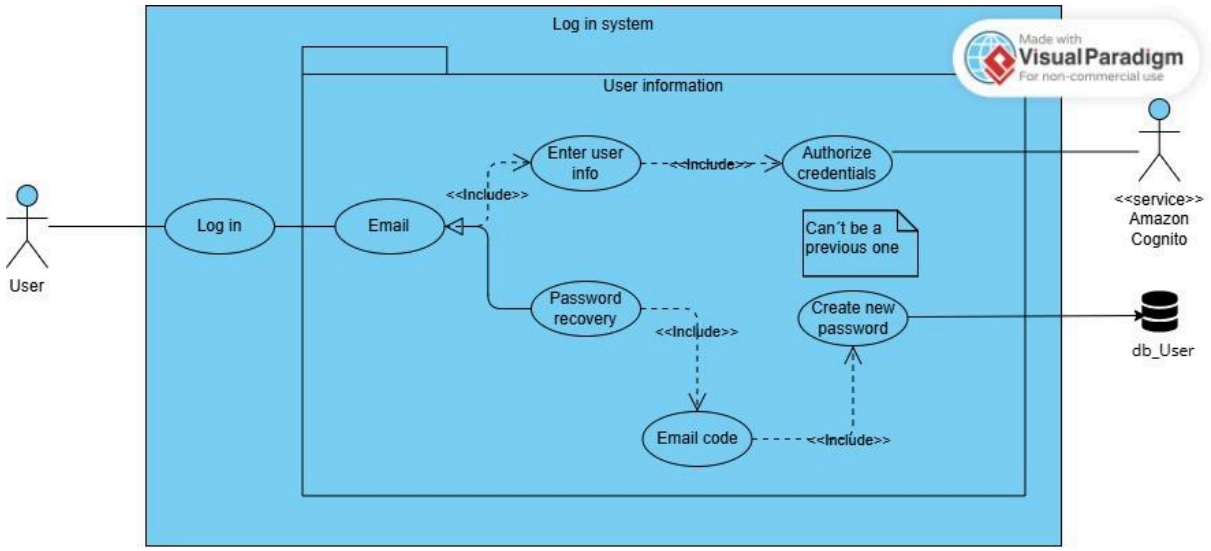
- First name
- Last name
- Age
- Gender
- Birth date
- Profile image
- Password (which must be confirmed by re-entering it).

As part of the registration process, the system communicates with Amazon Cognito, which handles the creation and storage of user credentials. Cognito validates and registers the user's and administrator identity, enabling future authentication and access management within the system.

After this interaction with Cognito is completed successfully, the account creation process is finalized.

2.2.3 Log in

2.2.3.1 Log in user



When a user navigates into the login section of the system, they have the option to log in using an email address and a third party method to log in. The login process for each process is as follows:

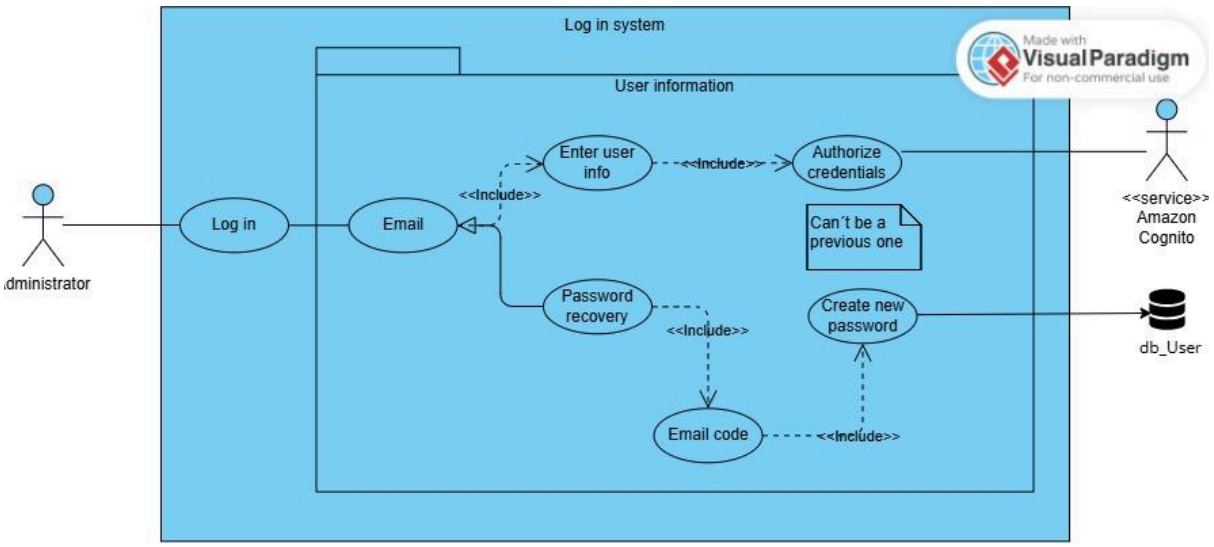
Email login: The user navigates to the login page and selects the email login option. They enter the email address and password that were used during account creation. If the credentials are correct, the user is logged into the system.

However, if the user has forgotten their password, they can recover it using the following process:

Reset password: The user, who has forgotten their password, navigates to the login page and clicks on the “¿Olvidaste contraseña?” link. They are redirected to the password reset page, where they are prompted to input their email address or their phone number associated with their account. A password reset link is sent to the provided email address. They are redirected to a page where they can enter a new password, it cannot be the same password entered before.

After entering and confirming the new password, they submit the form.

2.2.3.2 Log in administrator



When an administrator navigates into the login section of the system, they have the option to log in using an email address. The login process for each process is as follows:

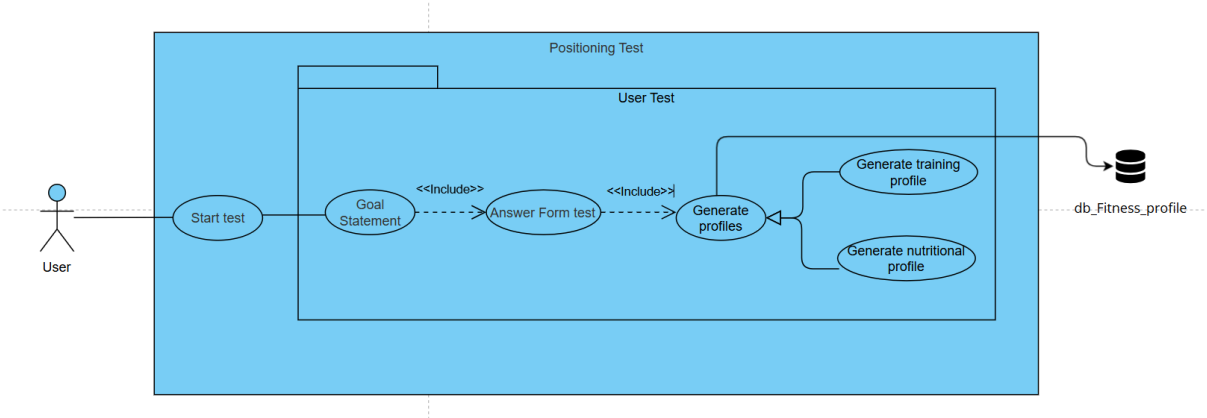
Email login: The user navigates to the login page and selects the email login option. They enter the email address and password that were used during account creation. If the credentials are correct, the user is logged into the system.

However, if the user has forgotten their password, they can recover it using the following process:

Reset password: The user, who has forgotten their password, navigates to the login page and clicks on the “¿Olvidaste contraseña?” link. They are redirected to the password reset page, where they are prompted to input their email address or their phone number associated with their account. A password reset link is sent to the provided email address. They are redirected to a page where they can enter a new password, it cannot be the same password entered before.

After entering and confirming the new password, they submit the form.

2.2.4 Positioning test



When a user accesses the “Positioning Test” section within the system, they are presented with an interactive questionnaire designed to collect relevant information about their physical condition, eating habits, and personal fitness goals.

1. **Access to the test:**

   The user logs into their account and navigates to the “Your BeFit Test” section.

2. **Questionnaire presentation:**

   The system displays a set of questions organized by category (habits, physical activity, rest, goals, dietary preferences).

   Questions are presented dynamically, with subsequent ones adapting based on previous responses (branching structure).

3. **Response recording:**

   As the user answers, the system temporarily stores the responses in the backend, linking them to the user’s ID.

   Once the test is completed, the data is permanently stored in the database.

4. **Result processing:**

   The backend sends all collected responses and metrics to the decision tree previously trained.

   The model analyzes the input and returns a user profile along with a personalized plan and product recommendation.

5. **Result presentation:**

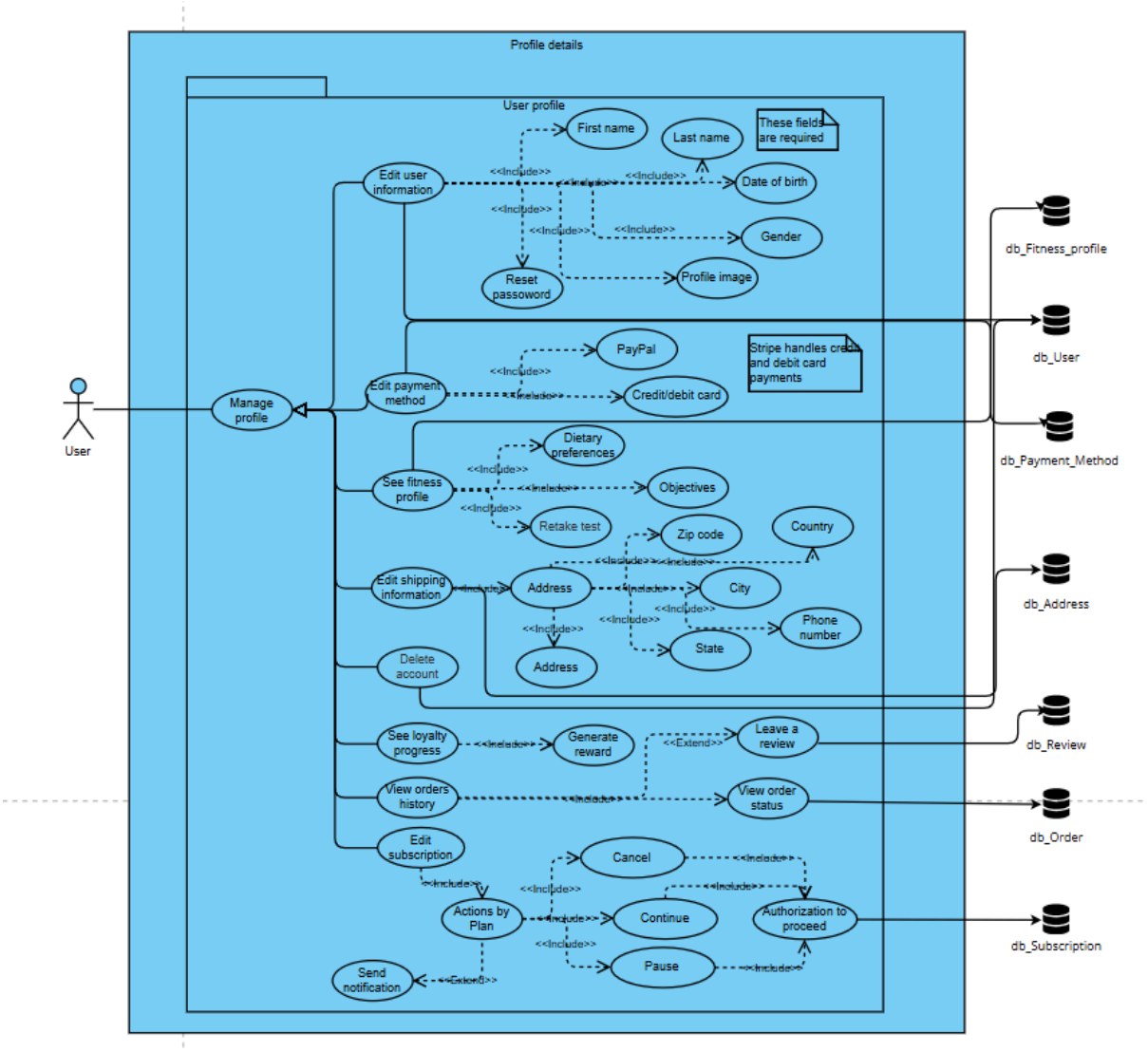
   The system displays a summary of the user’s profile (“Balanced,” “Muscle Mass,” or “Calorie Deficit”) and the recommended products.

The user can directly subscribe to the suggested plan or explore alternative recommendations and do the test again.

6. **Process completion:**

   The results are stored in the database, linked to the user's account, enabling future updates and model retraining.

2.2.5 Profile details

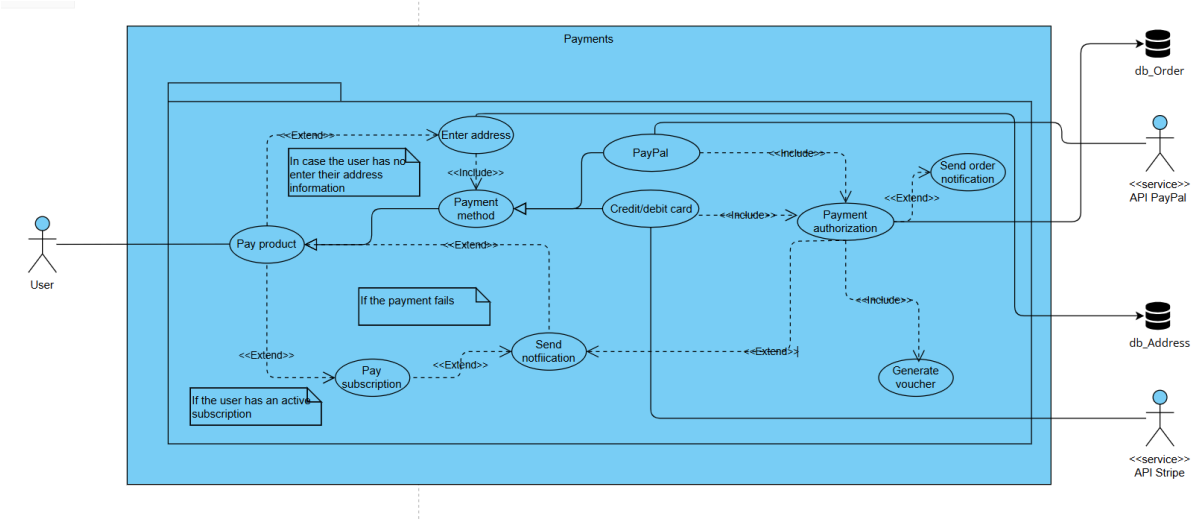


Profile details: When the user manages their profile ("Manage profile"), they can perform several actions within the "User profile" section:

1. **Edit user information:** The user can update their personal data. This process includes editing:
   - First name
   - Last name
   - Date of birth
   - Gender

    - Profile image
2. **Edit payment method:** The user can manage their saved payment methods. This includes:
    - PayPal
    - Credit/debit card
3. **See fitness profile:** The user can check their fitness profile and the option to retake or try again the positioning test. This includes:
    - Dietary preferences
    - Objectives
4. **Edit shipping information:** The user can edit their shipping addresses. This process includes:
    - Address
    - Zip code
    - City
    - State
    - Country
    - Phone number
5. **Delete account:** The user can delete their account.
6. **See loyalty progress:** The user can check their status or progress in the loyalty program. Also, users can generate their reward.
7. **View orders history:** The user can check the status of their orders, they can leave a review of a product purchased, and also see the history of the products that they have purchased.
8. **Subscription actions:**
   Within the panel, the user can perform the following actions:
    - Continue: keep the current plan active.
    - Pause: temporarily stop updates or shipments (e.g., vacation or training break).
    - Cancel: end the subscription and stop future deliveries.

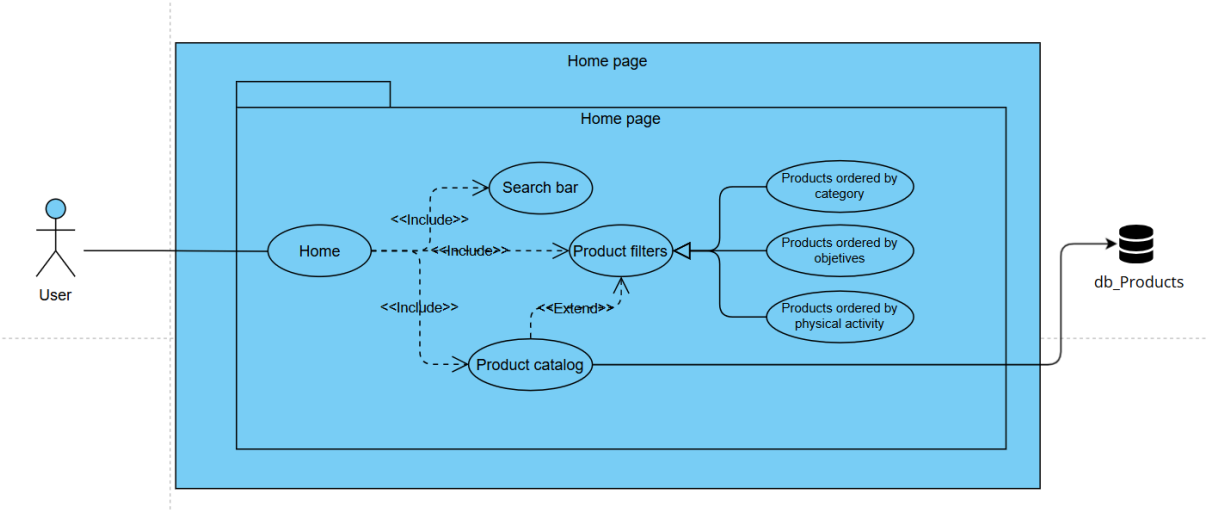
2.2.6 Payments



Payment: When the user i s ready to pay, they initiate the payment process.

1. **Pay product:** The user initiates the payment. This action automatically includes "Pay subscription" if the user has an active subscription. Optionally, "If the payment fails" process can be activated (which in turn can send a notification).
2. **Enter address:** Before entering the payment selection process. If the user has not entered their address information, they will need to enter it first to continue. Otherwise, they can proceed without any problems.
3. **Payment method:** The user must select a payment method. The options are:
    a. PayPal: The user chooses to pay with PayPal.
    b. Credit/debit card: The user chooses to pay with a credit or debit card.
4. **Payment authorization:** Regardless of the chosen method, the system must include the payment authorization. This process interacts with the PayPal and Stripe API.
5. **Send order notification:** After authorization, the system can send an order notification.
6. **Generate voucher:** The authorization process automatically includes the generation of a voucher, which interacts with the Stripe API.

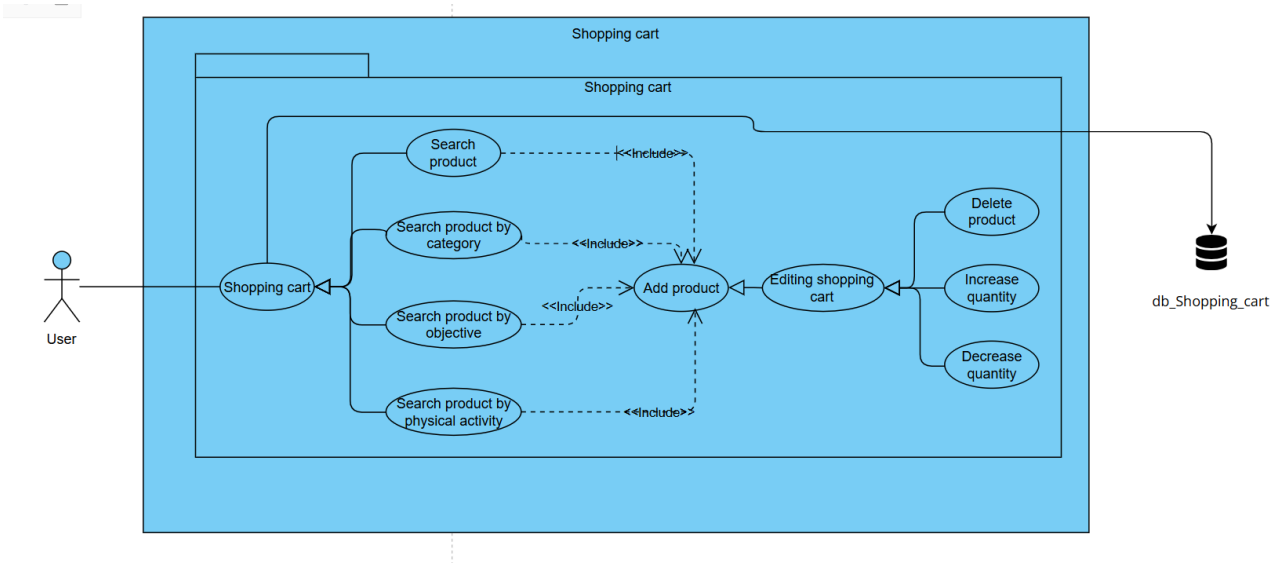
2.2.7 Home page



When a user accesses the system´s home page, they are presented with an online store interface designed to facilitate product searching and exploration through multiple navigation and filtering tools, they can view products in 3 ways:

- **Search bar:** Allows the user to perform direct product searches using keywords.
- **Product catalog:** Display the complete inventory of available products.
- **Product filters:** The user can organize products using three main criteria:
    - By category: Groups products according to their type.
    - By objectives: Filter products based on specific user goals.
    - By physical activity: Classifies products according to exercise or sport type.

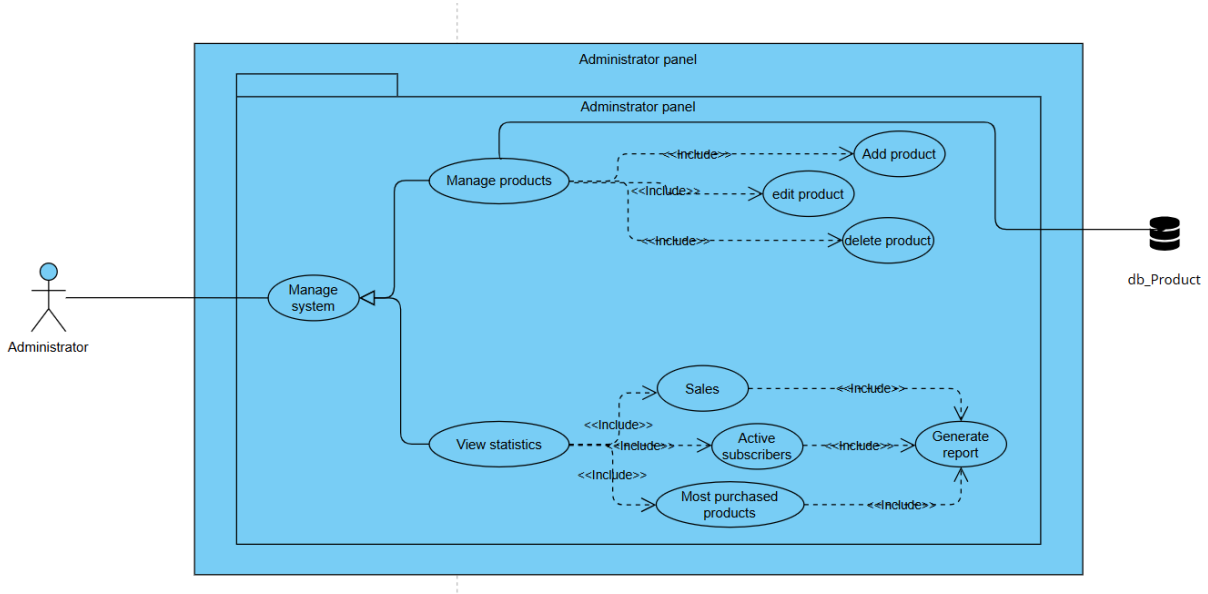
2.2.8 Shopping cart



Shopping cart: When the user has already created their shopping cart, they can add products by ways:

1. **Search product**: The user can browse the product catalog to select their desired product.
2. **Search product by category**: The user can browse the product catalog by filtering by category and select their desired product.
3. **Search product by objective**: The user can filter the product catalog based on a specific objective (e.g., muscle gain, weight loss) and select the product.
4. **Search product by physical activity**:The user can filter the product catalog based on the physical activity they are performing (e.g., running, crossfit, yoga) and select the product.
5. **Add product**: In either way, add the product to the cart.
6. **Editing shopping cart**:
    a. **Delete product**: The user can remove products from the cart.
    b. **Increase quantity**:The user can increase the quantity in the cart.
    c. **Decrease quantity**: The user can decrease the quantity in the cart.

2.2.9 Administrator panel



When the administrator logs into the system through a verified account, a centralized dashboard is displayed, providing access to the following main sections:

1. **Product management:**

   The administrator can add, edit, or delete products from the catalog.

   Each operation interacts directly with the db_Product database hosted on AWS RDS, ensuring real-time updates to the inventory.

2. **View statistics:**

   The panel allows the administrator to review key metrics about system performance:

   - *Overall sales and user purchasing behavior.*
   - *Number of active subscribers.*
   - *Most purchased products*

   All data is presented through interactive charts or downloadable reports generated from database records.

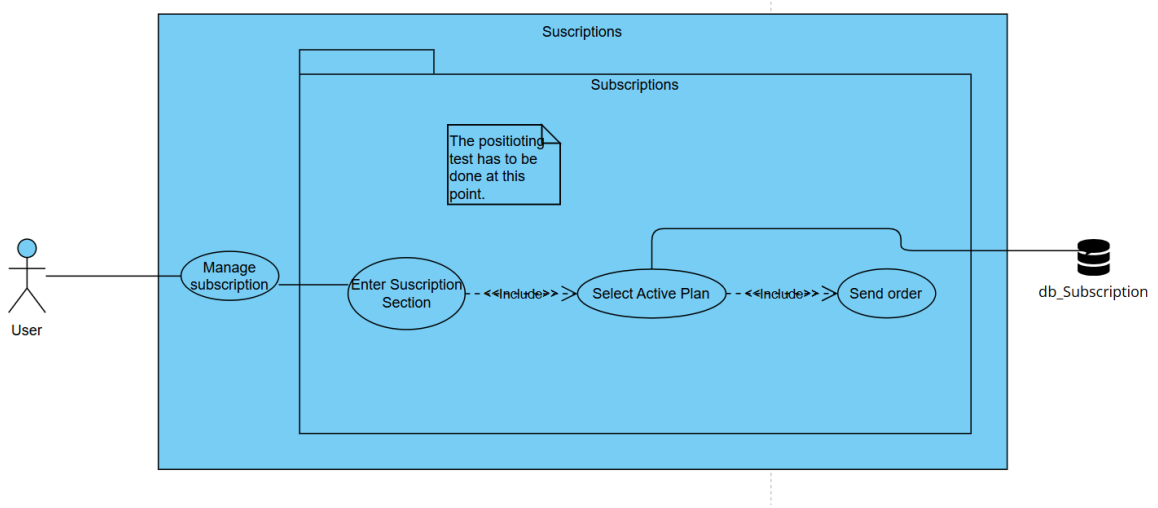
3. **Report generation:**

   The system can automatically or manually generate reports containing the most relevant monthly metrics.

   Reports can be exported in standard formats (**PDF, CSV**) for administrative or marketing purposes.

If an **error occurs** (e.g., connection loss or validation failure), the system alerts the administrator and prevents partial operations to preserve data integrity.

2.2.10 Subscriptions



When the user accesses the subscriptions section, a guided interface is displayed:

1. **Process initiation:**
   The user logs into the system and navigates to the Subscription section within their personal dashboard.
2. **Positioning test verification:**
   Before displaying subscription details, the system checks that the positioning test has been completed.
   This test determines the user's category (e.g., *Caloric deficit*, *Muscle gain*, *Maintenance*, *Body recomposition*).
   If not completed, the system prompts the user to finish it before accessing the management panel.
3. **View active plan:**
   The system displays the user's assigned plan, including:
   - Main goal (current objective).
   - Associated supplements or nutritional products.
4. **Send order:**
   The system validates the user's selection and will send an order based on a profile automatically monthly

2.3 User Characteristics

2.3.1 General Users

Users who browse the system without having an active account in use. They have limited interactions, mostly reserved to viewing the available products. In order to make purchases they must become registered users.

Characteristics:

- Can browse the catalogue.
- Can search for products.
- Can view products.

2.3.2 Registered Users

Users who have a registered account in the system. They can do all of the actions stated in the previous role, plus some additional features.

Characteristics:

- Can log in to their account with their credentials.
- Can view and edit their profile information.
- Can make purchases.
- Can view purchase history.
- Can receive suggestions based on their profile.
- Can subscribe to receive packages monthly.
- Can take part in the loyalty program.

2.3.3 Administrators

This role is reserved for the users who administrate the BeFit system. They have exclusive access to the administration panel, in which they can set up and maintain the system.

Characteristics:

- Can add, edit, and remove product listings.
- Can view sales and user statistics.
- Can generate status reports.

2.3.4 Payment System Interactors

These are users who interact with the payment processing features of the system, specifically when they are making payments using credit or debit card, PayPal, Stripe.

Characteristics:

- Payment information: Users must provide accurate payment details (card number, expiration date, billing address)
- Payment confirmation: Users receive real-time feedback on the success or failure of their payment transactions.
- Secure transactions: They rely on the system to securely handle and process payment information through integrated payment gateways.

2.3.5 External System Interactors

These are external entities and APIs integrated into the system to enhance its functionality, particularly in payment processing, authentication, health data from the user:

- AWS Cognito: Implements secure, scalable customer identity and access and access management, allowing users to sign in through social identities.

2.3.6 Database

The system relies on a centralized relational database hosted on AWS RDS to store, manage, and maintain integrity for all essential information, ensuring efficient data retrieval

and consistent relationships between different data elements. This includes user accounts, product inventory, order histories, user information and system settings.

2.4 Constraints

2.4.1 Technological Constraints

1. Defined languages and frameworks:
   The system must be developed exclusively using React (web frontend), React Native (mobile application), and FastAPI (Python) for the backend.

   No alternative frameworks or unstable versions of these technologies may be used.

2. AWS-based infrastructure:

   All system components (frontend, backend, database, and storage) must be hosted and deployed within the AWS ecosystem, using the following services:

   - AWS Amplify for frontend deployment.
   - AWS EC2 for backend (FastAPI) hosting.
   - AWS RDS (PostgreSQL) as the main relational database.
   - AWS S3 for static and media file storage.
   - AWS IAM for credential and access role management.
   - AWS CloudWatch for system monitoring and alert management.

3. Architecture and communication:

   The system shall follow a decoupled client–server architecture, communicating exclusively via RESTful APIs.
   Direct communication between frontend and database is strictly prohibited.

4. Browser and device compatibility:
   The web application must function properly on Google Chrome, Microsoft Edge, Mozilla Firefox, and Safari (latest versions).
   The mobile app must be compatible with Android 10+ and iOS 14+, developed solely in React Native.

5. Unified data model:
   All entities and relationships must be stored in a single PostgreSQL database managed through AWS RDS, with no dependency on local or external data sources.

2.4.2 Security and Privacy Constraints

1. Regulatory compliance:
   The system must comply with international data protection standards, including GDPR and LPDP.

2. Secure transmission:
   All data exchange between client and server must use HTTPS with TLS 1.2 or higher encryption.
3. Credential management:
   User passwords must be encrypted using secure hashing algorithms (bcrypt or Argon2).
   API keys and authentication tokens must be managed through AWS IAM Roles, avoiding static credentials in source code.
4. Authentication and authorization:
   All access to the system must be managed through JWT tokens and role-based permissions (user, admin, analyst).

2.4.3 Performance and Availability Constraints

1. Concurrent users:
   The system must support at least 1000 concurrent users without service degradation.
2. Response times:
   - REST API requests must respond in under 500 ms on average.
   - Web pages must fully load in under 2 seconds on a standard connection.
3. Minimum uptime:
   The infrastructure must maintain at least 90% monthly operational availability, monitored via CloudWatch.
4. Failure recovery:
   In case of downtime, the system must restore operations within 5 minutes using EC2 automatic restart or daily RDS backup recovery.
5. Caching system:
   Nginx or AWS CloudFront caching must be implemented to reduce backend load and improve response times.

2.4.4 Operational Constraints

1. Development and deployment environment:
   All stages of the software lifecycle (development, testing, deployment, and maintenance) must be executed using AWS and GitHub environments only—no local infrastructure is permitted.
2. Monitoring and logging:
   All logs, error reports, and operational events must be stored and monitored through AWS CloudWatch, ensuring full traceability and real-time auditing.
3. Maintenance policy:
   System updates must be planned and executed without disrupting service availability.
   AWS Amplify and EC2 must support parallel versioning and automatic rollback in case of deployment errors.
4. Backup policy:
   The database and system configuration must be automatically backed up daily, with a minimum data retention period of 7 days.

2.4.5 Design and Usability Constraints

1. Interface guidelines:
   The interface must follow a minimalist, intuitive, and functional design aligned with modern UX principles.
   Layouts, color palettes, and typography must maintain consistency with *BeFit's* corporate identity.
2. Responsive design:
   All views must adapt dynamically to different devices—desktop, tablet, and mobile.
3. Accessibility:
   The interface must follow WCAG 2.1 accessibility guidelines, ensuring usability for individuals with mild disabilities (contrast, typography, keyboard navigation).
4. System feedback:
   All notifications, warnings, and error messages must use clear, user-friendly language and consistent visual feedback..

2.4.6 Economic and resource constraints

1. Controlled scalability:
   System expansion (adding more instances or services) must occur only if projected costs remain within the team's allocated monthly budget.
2. Storage costs:
   Static files, logs, and backups stored in S3 must follow lifecycle policies that automatically delete obsolete files after 30 days to minimize costs.

2.4.7 Ethical and Legal Constraints

1. Data sharing or sale to third parties is strictly prohibited.
2. User consent:
   The system must obtain explicit user consent before accessing any sensitive or health-related information.
3. Algorithmic transparency:
   Recommendation outputs must be explainable, allowing users to understand which factors influenced their personalized suggestions.
4. Right to deletion:
   Users must be able to permanently delete their accounts and all associated data, in compliance with privacy laws and ethical data practices..

2.5 Assumptions and dependencies

- Users will provide accurate and valid personal information in accordance to their health profile and needs.
- Product information provided will be accurate and up to date.
- Web users will have access to a stable internet connection and an internet browser.

- Mobile users will have access to a stable internet and an application marketplace.
- The system will rely on the availability of the following external systems:
    - PayPal and Stripe for payment.
    - AWS for the infrastructure and hosting.
    - Google and Facebook as external authentication methods.
- The system's performance depends on the reliability and latency of the internet connection between client and server.

2.6 Apportioning of requirements

BeFit shall be a completely functional and operational system at the moment of release. However, some additional functionalities that were considered may be released in future versions and therefore will not be included in this document.

- The e-commerce may expand to a hybrid business model, integrating physical retail stores with the existing online platform.
- A machine learning model may be developed and integrated to analyze preferences and fitness goals in order to improve product recommendation.

In relation to requirement prioritization, the requirements were organized utilizing the MoSCoW technique. The details of this prioritization can be found in section 3.11.

3 Specific requirements

3.1 External interfaces

3.1.1 API Integration

API name	**API Description**	**Use**
Stripe	Secure payment management	Use the API to manage secure payments.
PayPal	Secure payment management	Use the API to manage secure payments.
AWS Cognito	Implement secure, frictionless customer identity and access management that scales.	Use the API to allow users to sign in through social identity.

3.1.2 AWS services

Service name	**Service Description**	**Use**
Amplify	A set of tools and services that simplify building, deploying, and hosting full-stack web and mobile applications on AWS.	Used to connect frontend apps with backend resources, manage hosting, and enable authentication.
Amazon EC2	Elastic compute cloud provides scalable virtual servers for running applications in the cloud.	Used to host and run backend processes, APIs, or application servers.
Amazon RDS	Relational Database Service that simplifies database setup, operation, and scaling in the cloud.	Used to store structured application data securely and efficiently.
Amazon S3	Simple Storage Service for storing and retrieving any amount of data at any time.	Used to store media files, documents, backups, and static assets.
AWS IAM	Identity and Access Management service that controls access to AWS resources securely.	Used to manage users, roles, and permissions for AWS service.
AWS Cognito	Implement a secure, scalable and personalized registration and login experience in minutes	Allow users to sign in through social identity.

AWS CloudWatch	A monitoring and observability service that collects and tracks metrics, logs, and events from AWS resources and applications.	Used to monitor application performance, detect operational issues, and set automated alerts to maintain system reliability.

3.2 Functions

3.2.1 Sign Up

Requirement ID	RF-01
Use case	Sign Up
Actor	General user
Pre-condition	The user must have a valid email or Gmail/Facebook account.
Main scenario	1. The user goes to the Sign Up page. 2. The user enters their credentials. a. The user enters a valid email, a password, and password confirmation. i. The password must contain at least 8 characters and include: a lowercase letter, an uppercase letter, a number, and a symbol. ii. Both password and password confirmation must match. b. The user selects “Continue with Gmail/Facebook”. i. The system redirects the user to the external authentication service to complete the sign-up process. 3. The system validates the input and creates the new account.
Alternative scenario	2b. The user enters invalid credentials. 3b. The system rejects the sign up attempt and displays an error message “Invalid email or password” or “Unable to authenticate with Gmail/Facebook”. 2c. The user enters an email or external authentication which is already in use for an existing account. 3c. The system displays an information message “There is already an account associated with this email/account”.

Post-condition	The user is redirected to the Profile Set Up page.

3.2.2 Log In

Requirement ID	RF-02
Use case	Log In
Actor	Registered user
Pre-condition	The user must be registered in the system with valid credentials in order to login.
Main scenario	1. The user goes to the Log In page. 2. The user enters their credentials. a. The user enters an email and a password. b. The user selects external authentication. 3. The system verifies the credentials and logs the user in.
Alternative scenario	2b. The user enters the wrong credentials. 3b. The system rejects the login attempt and displays an error message “Incorrect email or password” or “Unable to authenticate credentials”. 4b. After 5 failed attempts to log in, the system will automatically redirect to the Password Recovery page.
Post-condition	The user is redirected to the Home page.

3.2.3 Password Recovery

Requirement ID	RF-03
Use case	Password Recovery
Actor	Registered user
Pre-condition	The user must have an account registered in the system.
Main scenario	1. The user selects the “Forgot your password?” option in the Log In page. 2. The system redirects the user to the Password Recovery page.

	3. The user must enter the email associated to their account. 4. The system will send a recovery email to the email. 5. The user shall access the Reset Password link through the email sent.
Alternative scenario	3b. The user enters an email not associated with an account in the system. 4b. The system displays an error message “This email is not associated with an account” and displays a link to the Sign Up page. 3c. The user enters an invalid email address. 4c. The system displays an error message “Invalid email”.
Post-condition	The user is sent to the Reset Password page.

3.2.4 Profile Set Up

Requirement ID	RF-04
Use case	Enter user info
Actor	Registered user
Pre-condition	The user must have just finished the Sign Up process.
Main scenario	1. The user is redirected to the Profile Set Up page. 2. The user must enter their first name, last name, date of birth and gender. a. This data may be partially filled if the Sign Up process was done by external authentication. 3. The user can upload a profile picture from their device. a. If the user does not upload a picture, a predetermined one will be associated with the profile. 4. The user must press the “Save” button. 5. The system must save the information in the database
Alternative scenario	2b. The user does not enter one of the fields. 3b. The system displays a warning message “Please fill out this field”.
Post-condition	The user is sent to the Positioning Test.

3.2.5 Positioning Test

Requirement ID	RF-05
Use case	Positioning Test
Actor	Registered user
Pre-condition	The user must have an account.
Main scenario	1. The user is redirected to the Positioning Test page. 2. The user must answer a series of multiple choice questions related to their health lifestyle and fitness goals. 3. The system shall record the user's responses and process them using a decision-based algorithm to evaluate the user's Fitness profile. a. Based on the evaluation, the system shall determine attribute scores or categories corresponding to the user's needs and preferences. 4. The system shall store the test results in the user's Fitness profile.
Alternative scenario	2b. The user does not fill out the questionnaire or save their answers. 3b. The system shall discard any unsaved responses. 4b. The system will not generate personalized recommendations.
Post-condition	The user is sent to the Home page. The system will make recommendations and suggestions based on the user's goals and restrictions.

3.2.6 User Profile

Requirement ID	RF-06
Use case	Manage profile
Actor	Registered user
Pre-condition	The user must have an account.
Main scenario	1. The user navigates to the Profile page via the navigation menu. 2. The user can view and edit their information: first name, last

	name, date of birth, gender, profile picture. 3. The user can view the email or external account associated with the account. 4. The user can select the “change password” option, which will redirect them to the Reset Password page. 5. The user can select the “Edit payment method” option, which will redirect them to the Delivery and Payment Preferences page. 6. The user can view, and edit their address: address, zip code, state, city. 7. The user can view their Fitness profile, in which the information related to the Positioning Test will be displayed. a. The system shall display a “retake test” option, which will redirect the user to the Positioning Test. 8. The user can delete their account. a. The system shall display a confirmation message “Are you sure you want to delete your account?”. 9. The user can manage their subscription with the following options: Pause, resume, or cancel. a. The system shall send a notification in case the user wants to cancel their subscription.
Alternative scenario	2b. The user selects the edition option in one of the valid fields. 3b. The user leaves the field empty. 4b. The system shall display a warning message “Please fill out the field”.
Post-condition	If the user edited one of the valid fields, the system must update the information in the database.

3.2.7 Reset Password

Requirement ID	RF-07
Use case	Reset password
Actor	Registered user
Pre-condition	The user must have an account that was created by email.
Main scenario	1. The user selects the “change password” option in their profile or is redirected by the Password Recovery email link. 2. The user must enter a new valid password. a. The password must contain at least 8 characters and include: a lowercase letter, an uppercase letter, a number, and a symbol.

	b. Both password and password confirmation must match. 3. The user confirms the password change. 4. The system must save the new password in the database.
Alternative scenario	1b. A user registered by external authentication selects the change password option. 2b. The system displays an information message “Users authenticated via external accounts must manage their credentials through the external provider used”. 2c. The password entered does not meet the criteria listed or does not match the password confirmation field. 3c. The system displays a warning message “Please enter a password with the following criteria: …” or “Passwords do not match”.
Post-condition	The system shall now only allow the user to log in with the new password.

3.2.8 Payment Method

Requirement ID	RF-08
Use case	Pay product
Actor	Registered user
Pre-condition	The user must have an account.
Main scenario	1. The user navigates to the payment Preferences through the User profile or at checkout. 2. The user can add a new delivery preference by selecting the “new address” option. a. The user must fill out the shipping information: Address line 1, Address line 2 (optional), Country, State, City, Zip Code, Name, Telephone number. b. The user can add an identification name for the address. 3. The user can add a new payment preference by selecting the “new payment method” option. a. The user must select the new payment method: PayPal account, Credit/Debit card. i. If the user selects PayPal, the system shall redirect the user to the PayPal authentication

	page. ii. If the user selects credit/debit card they must fill out their card information through a secure Stripe interface. 4. The system shall store the information accordingly in the database. 5. The user can view, edit and delete their previously saved preferences.
Alternative scenario	2b. The user does not fill out the required fields for the new address. 3b. The system displays a warning message “Please fill out all the obligatory fields”. 3c. The user does not complete the PayPal external authentication successfully. 4c. The system displays an error message “The process was unsuccessful, please try again”. 3d. The user does not fill out the required fields for the new card. 4d. The system displays a warning message “Please fill out all the obligatory fields”.
Post-condition	The user can choose stored delivery and payment preferences during their checkout.

3.2.9 Home

Requirement ID	RF-09
Use case	Home
Actor	General or registered user
Pre-condition	N/A
Main scenario	1. The user navigates to the Home page. 2. The system displays a series of product listings and promotional information. 3. The user can navigate the system through the Navigation Menu: a. General users can navigate through categorized

	listings, use the search bar, and have the option to sign up or log in. b. Registered users can: view categorized listings, use the search bar, view their profile, view their order history, view their cart, view the subscription panel and log out. 4. The user can select a product from the listing, which will open the Product Listing page of that specific product.
Alternative scenario	3b. A general user attempts to navigate to a registered user only page. 4b. The system shall redirect the general user to the Log In page.
Post-condition	N/A

3.2.10 Product Listing

Requirement ID	RF-10
Use case	Product catalog
Actor	General or registered user
Pre-condition	N/A
Main scenario	1. The user navigates to a specific product listing. 2. The system displays the following information: name, price, description, nutritional value, brand, and images. 3. The system will display the reviews related to the product and an average rating. 4. The user can add one or more (limited to the product's stock) of the product to their shopping cart.
Alternative scenario	4b. The user attempts to add more than the product's stock to their cart. 5b. The system displays an information message "There is no more stock of this item".
Post-condition	N/A

3.2.11 Product Review

Requirement ID	RF-11
Use case	Leave a review
Actor	Registered user
Pre-condition	The user has purchased the item in the past.
Main scenario	1. The user can give a star rating and write a review of a previously purchased product by selecting “Rate your order” in a completed order’s listing in the Order History page. 2. The user must select a rating from 1 to 5 stars and can optionally write a review of the product. 3. The user can see, edit and delete their reviews on the Reviews section in the Order History page.
Alternative scenario	2b. The user does not select a star rating and attempts to save their review. 3b. The system displays a warning message “Please select a star rating”.
Post-condition	The product’s average rating will be updated and the review will be shown in the product’s listing.

3.2.12 Search and Filter

Requirement ID	RF-12
Use case	Search bar, Product filters
Actor	General and registered users.
Pre-condition	N/A
Main scenario	1. The user can search for specific products, brands or other keywords using the search bar. 2. The system shall retrieve and display a list of products matching the search. 3. The system shall allow the user to apply filters to refine the search results based on specific criteria

Alternative scenario	1b. There are no products matching the search or filter. 2b. The system shall display an informative message “There are no products matching the search”.
Post-condition	N/A

3.2.13 Shopping Cart

Requirement ID	RF-13
Use case	Shopping cart.
Actor	Registered users.
Pre-condition	The user must have an account registered in the system.
Main scenario	1. The user adds a desired product to the shopping cart from the product catalog. 2. The user navigates to the shopping cart page to view its contents. 3. The system displays a list of all products in the cart, showing their name, price, quantity, and a subtotal. 4. The user can modify the contents of the cart: a. Increase or decrease the quantity of any product. b. Remove a product completely from the cart. 5. The system automatically updates the total price after any modification. 6. The user confirms the contents of the cart and proceeds to the payment process.
Alternative scenario	1. A product added to the cart goes out of stock before the purchase is completed. The system displays a notification message and prevents the user from proceeding to checkout until the item is removed. 2. The user attempts to increase the quantity of a product beyond the available stock. The system shows an error message and adjusts the quantity to the maximum available.

Post-condition	The user has a shopping cart ready to proceed to the checkout and payment process.

3.2.14 Order Processing

Requirement ID	RF-14
Use case	Payment authorization
Actor	Registered users.
Pre-condition	The user must be logged into an account and pay his shopping cart.
Main scenario	1. The user is on the checkout page, which displays a final summary of the items to be purchased. 2. The system displays a complete order summary for final review: items, shipping address, selected payment method, and total cost. 3. The user clicks a button like "Confirm and Proceed to Payment". 4. The user views their order.
Alternative scenario	1. The user enters incomplete or invalid shipping information. The system will display a warning message and prevent them from proceeding until the information is corrected. 2. The user decides to modify the order. The system provides a clear option to "Return to Cart" to edit its contents.
Post-condition	The order details are validated and confirmed by the user.

3.2.15 Order Payment

Requirement ID	RF-15
Use case	Generate voucher
Actor	Registered users.
Pre-condition	The user must be logged into an account and have at least one item in their shopping cart.

Main scenario	1. The user initiates the checkout process from the shopping cart page. 2. The system directs the user to the checkout page, displaying the items to be purchased. 3. The user selects a shipping address from their saved preferences or adds a new one. 4. The user selects a payment method from their saved options or adds a new one. 5. The system displays a final order summary, including the product list, shipping details, and total cost. 6. The user confirms all details and clicks the "Place Order" button. 7. The system processes the transaction through the selected payment gateway. 8. Upon successful payment, the system creates a new entry in the user's order history, updates product inventory, and sends an order confirmation notification.
Alternative scenario	1. The payment is declined by the external payment gateway (insufficient funds, invalid card information). The system will display an error message and allow the user to try again with a different payment method. 2. An item in the cart becomes out of stock during the checkout process. The system will display a notification and prevent the order from being finalized until the item is removed from the cart.
Post-condition	N/A

3.2.16 Order History

Requirement ID	RF-16
Use case	See orders history
Actor	Registered users.
Pre-condition	The user must be logged into an account.
Main scenario	1. The user navigates to the "Order History" page through their user profile. 2. The system displays a list of all the user's past orders, sorted

	from most recent to oldest. 3. For each order, the system shows a summary including the order number, date of purchase, total amount, and shipping status. 4. The user can select a specific order to view its full details. 5. The system then displays the detailed view, which includes the list of products purchased, quantities, individual prices, shipping address, and payment method used. 6. For orders with a "Delivered" status, the user will have an option to "Rate your order" to leave a review for the purchased products.
Alternative scenario	1. The user navigates to the Order History page but has not made any purchases yet. 2. The system displays an informative message, such as "You have no past orders."
Post-condition	The user has successfully reviewed the details of their past purchases and their current status.

3.2.17 Subscription

Requirement ID	RF-17
Use case	Manage Subscription.
Actor	Registered user.
Pre-condition	The user must be logged into an account and must have completed the Positioning Test.
Main scenario	1. The user navigates to the "Subscription" section from their profile or the main menu. 2. The system presents a recommended, personalized monthly plan, showing the specific kit of products selected based on the user's Positioning Test results. 3. The user reviews the products, the monthly cost, and is given a clear option to "Subscribe to this plan". 4. If the user accepts, they are directed to the payment process to confirm the subscription and authorize recurring payments. 6. The system confirms any changes with a notification.

Alternative scenario	1. The user navigates to the subscription section but decides not to subscribe. They can exit the section without any commitment. 2. The payment for the initial subscription fails. The system will display an error message, and the subscription will not be activated.
Post-condition	If the user subscribes, their account is updated with an active subscription.

3.2.18 Loyalty Program

Requirement ID	RF-18
Use case	See loyalty progress
Actor	Registered users.
Pre-condition	The user must be logged into an authenticated account.
Main scenario	1. The user earns loyalty points with each purchase made on the platform. 2. The user navigates to the "Loyalty Program" section in their user profile. 3. The system displays the user's current loyalty points balance and a list of available rewards (e.g., discounts, free products). 4. The user can select an available reward to redeem it. 5. The system validates if the user has enough points, applies the reward (e.g., generates a discount code), and deducts the corresponding points from the user's balance. 6. A confirmation of the redeemed reward is displayed to the user.
Alternative scenario	1. The user views the loyalty program section but does not have enough points to redeem any rewards. 2. The system will clearly show the available rewards as "locked" or indicate how many more points are needed to unlock them.
Post-condition	The user's loyalty points balance is updated, and the redeemed reward is made available for use in their next purchase.

3.2.19 Automatic Orders

Requirement ID	RF-19
Use case	Send order
Actor	Registered users.
Pre-condition	The user must have an active and valid subscription.
Main scenario	1. Following a successful monthly subscription payment, the system automatically generates a new order. 2. This order is populated with the specific kit of products corresponding to the user's personalized plan. 3. The system assigns a new order number and adds the order details to the user's "Order History" with a status like "Processing." 4. The system sends a notification to the user confirming that their monthly kit is being prepared for shipment. 5. The order is placed in the fulfillment queue to be shipped to the user's registered address.
Alternative scenario	1. If the scheduled recurring payment fails prior to this step, an automatic order will not be generated. 2. The system will instead notify the user of the payment issue and pause the subscription, preventing future shipments until the payment method is updated.
Post-condition	A new order for the current subscription cycle is created and ready for fulfillment, without requiring any additional payment or action from the user.

3.2.20 Admin Panel

Requirement ID	RF-20
Use case	Manage system.
Actor	Administrator.
Pre-condition	The user must be logged into the system through a verified account

	with administrator privileges.
Main scenario	1. The administrator logs into the system and accesses the centralized dashboard. 2. The panel displays several management sections. 3. The administrator selects "Product Management" to add, edit, or delete products from the catalog. 4. The administrator selects "View Statistics" to review key performance metrics, such as overall sales, the number of active subscribers, and the most purchased products.
Alternative scenario	1. The administrator is performing an action, but an error occurs due to a connection loss or a validation failure. 2. The system alerts the administrator and prevents the partial operation from completing to preserve the integrity of the data.
Post-condition	The administrative changes are successfully saved in the database and are reflected across the system for all users.

3.3 Non functional requirements

3.3.1 Performance requirements

- The system must ensure that all REST API requests respond in less than 500 milliseconds on average.
- The frontend must load primary page content in under 2 seconds under standard internet conditions.
- The platform must maintain stable operation with up to 1,000 concurrent users without performance degradation.
- Static content such as images, scripts, and styles must be delivered through AWS S3 and CDN integration for faster access.
- System scalability must be automatic, allowing new AWS EC2 instances to activate when traffic demand increases.

3.3.2 Security requirements

- All communication between frontend and backend must occur over HTTPS secured with TLS 1.2 or higher.
- Authentication and access management must use AWS IAM roles and policies instead of hardcoded credentials.
- The system must comply with GDPR and LPDP data protection regulations for user privacy.

- The system must never store raw biometric or sensitive medical data, only anonymized aggregates.

3.3.3 Reliability requirements

- The system must guarantee a minimum monthly reliability rate of 90% uptime.
- AWS CloudWatch must continuously monitor server health, request load, and performance metrics.
- The database (AWS RDS – PostgreSQL) must perform automated daily backups for data preservation.
- All user data and transactions must remain consistent and traceable during and after recovery processes.

3.3.4 Maintainability requirements

- All API endpoints must be automatically documented.
- Maintenance operations must not disrupt user access or ongoing processes (e.g., subscription tracking).
- Error logs and update histories must be stored for future debugging and post-deployment validation.
- The deployment process must be managed through AWS Amplify (frontend) and EC2 (backend)

3.3.5 Portability requirements

- The web application must function properly on Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
- The mobile application must be developed in React Native, supporting Android 10+ and iOS 14+.
- Data and configurations must be easily transferable between development, staging, and production environments.
- All dependencies and configurations must be clearly defined in environment files for seamless migration.

3.3.6 Availability requirements

- The system must maintain a minimum operational uptime of 90% per month.
- AWS CloudWatch alarms must detect unusual activity or service interruptions in real-time.
- Automatic recovery mechanisms must restart EC2 instances or reroute traffic during outages.
- The system must be accessible 24/7, including peak hours for fitness-related activity.
- Scheduled maintenance must be announced in advance to minimize user disruption.
- Database and user sessions must remain persistent and recoverable after any restart event.

3.4 Business rules

Payments:

- **Prices in local currency:** All product prices must be displayed in Mexican pesos (MXN).
- **Accepted payment methods:** The system accepts credit or debit cards, PayPal, and Stripe as valid payment methods.
- **Payment authorization requirement:** All payments must be authorized through secure payment gateways (PayPal or Stripe) before order confirmation.
- **Payment failure protocol:** If a payment fails, the system must notify the user immediately and prevent order completion until payment is corrected or an alternative method is selected.
- **Subscription payment processing:** Recurring subscription payments are processed automatically on a monthly basis for active subscriptions.

Users:

- **Minimum age to register:** Users must be at least 18 years old to create an account and purchase products.
- **One account per person:** Each person can only have one account in the system to maintain data integrity and prevent abuse.
- **Account creation methods:** Users can register using email or third-party authentication (Google, Facebook). Administrator accounts can only be created via direct email registration.
- **Password security requirements:** Passwords must contain at least 8 characters including one lowercase letter, one uppercase letter, one number, and one special symbol.
- **Password change restrictions:** Users authenticated via third-party providers (Google, Facebook) must manage their passwords through their external provider.
- **Account deletion rights:** Users have the right to permanently delete their accounts and all associated data at any time.

Product sales:

- **Stock management:** The system must not allow the sale of out-of-stock products. Users cannot add more units to their cart than are currently available.
- **Product categorization requirement:** All products must be associated with at least one category and can be filtered by category, fitness objective, and physical activity type.
- **Inventory updates:** Product stock must be automatically updated after each successful purchase to maintain accurate availability.
- **Cart limitations during checkout:** If a product goes out of stock during the checkout process, the system must prevent order completion until the item is removed from the cart.

Orders:

- **Shipping address requirement:** Users must provide a valid shipping address before completing any order.
- **Payment method requirement:** Users must select and validate a payment method before order confirmation.
- **Order modification policy:** Once payment is authorized, orders cannot be modified or cancelled if they have entered the fulfillment stage.
- **Order confirmation notification:** Users must receive automated email and push notifications for order confirmations, payment failures, and shipping updates.
- **Automatic order generation for subscriptions:** The system automatically generates monthly orders for active subscriptions after successful recurring payment, containing products based on the user's fitness profile.

Subscriptions:

- **Positioning test requirement:** Users must complete the positioning test before accessing or subscribing to personalized monthly plans.
- **Subscription activation:** Subscriptions can only be activated after successful payment authorization for the initial billing cycle.
- **Subscription management flexibility:** Users can pause, resume, or cancel their subscriptions at any time through their user profile.
- **Paused subscription behavior:** Paused subscriptions do not generate automatic orders or charges until resumed by the user.
- **Failed payment suspension:** If recurring payment fails, automatic order generation is suspended and the user is notified to update payment information.
- **Subscription cancellation policy:** Cancelled subscriptions immediately stop all future deliveries and recurring payments.

Reviews:

- **Purchase requirement for reviews:** Only users who have purchased a product can leave a review for that specific product.
- **Mandatory star rating:** All reviews must include a star rating between 1 and 5 stars.
- **Review management:** Users can edit or delete their own reviews at any time through the order history section.
- **Review display:** Product reviews are displayed on product listing pages along with the average rating calculated from all reviews.

Loyalty program:

- **Points accumulation:** Users earn loyalty points with each completed purchase. Points are awarded only after successful payment and order confirmation.
- **Points redemption validation:** The system must validate that users have sufficient points before allowing reward redemption.
- **Automatic points deduction:** Redeemed points are immediately and automatically deducted from the user's balance upon successful redemption.
- **Available rewards:** Users can redeem accumulated points for discounts, free products, or other rewards as defined by the system administrators.

3.5 Process diagram

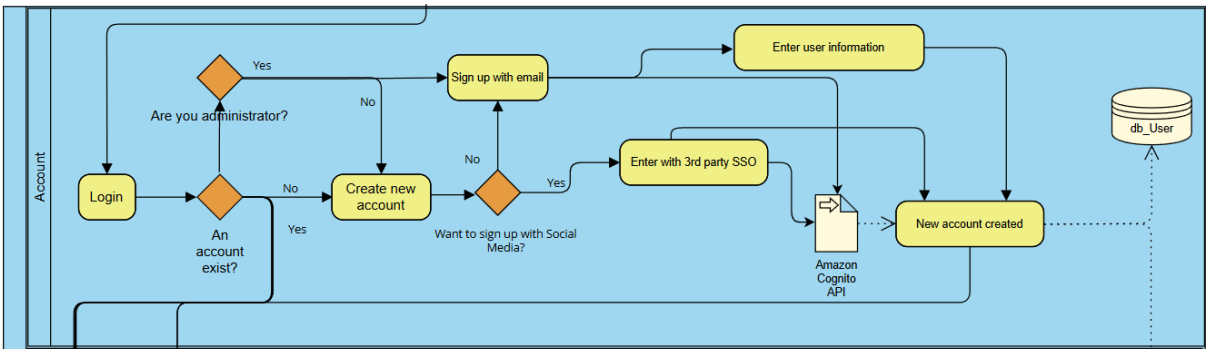
Process Diagram link: BPMN-MFDS.png

3.5.1 Description of process diagram

3.5.1.1 Start of the process:

- The user or client has access to the web/mobile application.

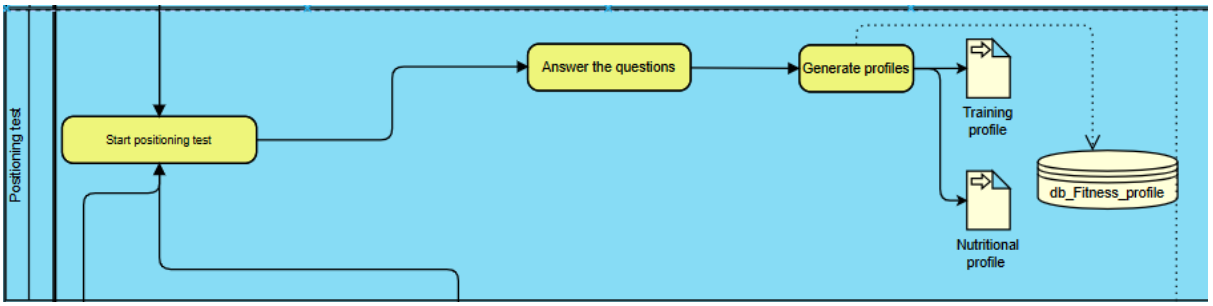
3.5.1.2 Account:



- **Purpose**: The user will be able to create or log in to their account.
- **Flow:**
  - **Administrator verification:**
    - The system asks: "Are you an administrator?"
    - **If Yes** → Navigate directly to "Login" for administrator access.
    - **If No** → Proceed to account verification.
  - **Account existence check:**
    - The system asks: "Does an account exist?"
    - **If Yes** → Navigate to "Login" to access an existing account.
    - **If No** → Proceed to "Create new account" process.
  - **Account creation method:**
    - The system asks: "Want to sign up with Social Media?"
    - **If No** → Navigate to "Sign up with email":
      - The user proceeds to "Enter user information" (first name, second name, email, gender, date of birth, profile image).
    - **If Yes** → Navigate to "Enter with 3rd party SSO":
      - Users authenticate using social media platforms (Facebook, Google, etc.).
  - **Account validation and storage:**
    - All registration data (email or SSO) is validated and processed through Amazon Cognito API.
    - "New account created" confirmation is generated.

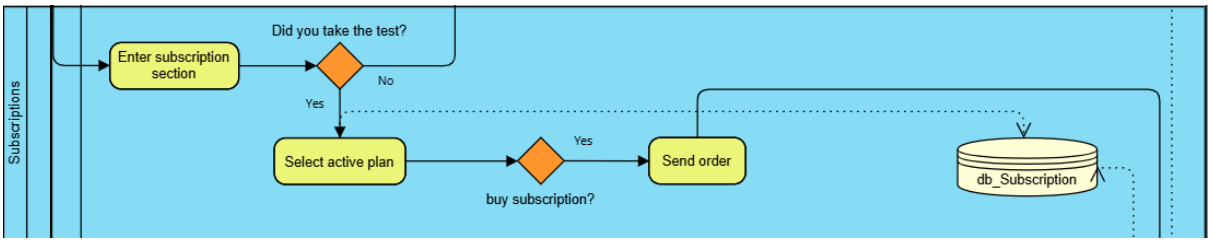
        - User credentials and profile information are securely stored in db_User database.
    - **Process completion:**
        - The user account is successfully created and ready for system access.
        - Users can now login and utilize all platform features.

3.5.1.3 Positioning test



- **Purpose**: The client will take a test to gather their information and adapt the content based on their needs.
- **Flow:**
    - **The user initiates the "Start positioning test"** to begin the profiling process.
    - **Answer the questions:**
        - The user completes a questionnaire with relevant information about their fitness goals, preferences, and current status.
    - **Generate profiles:**
        - The system processes all collected information (questionnaire responses and optional health data) to create two personalized documents:
            - **Training profile**: Customized workout plan and exercise recommendations.
            - **Nutritional profile**: Personalized diet plan and nutrition guidelines.
    - **Process completion:**
        - Both profiles are generated and delivered to the user and stored in db_Fitness_profile database.

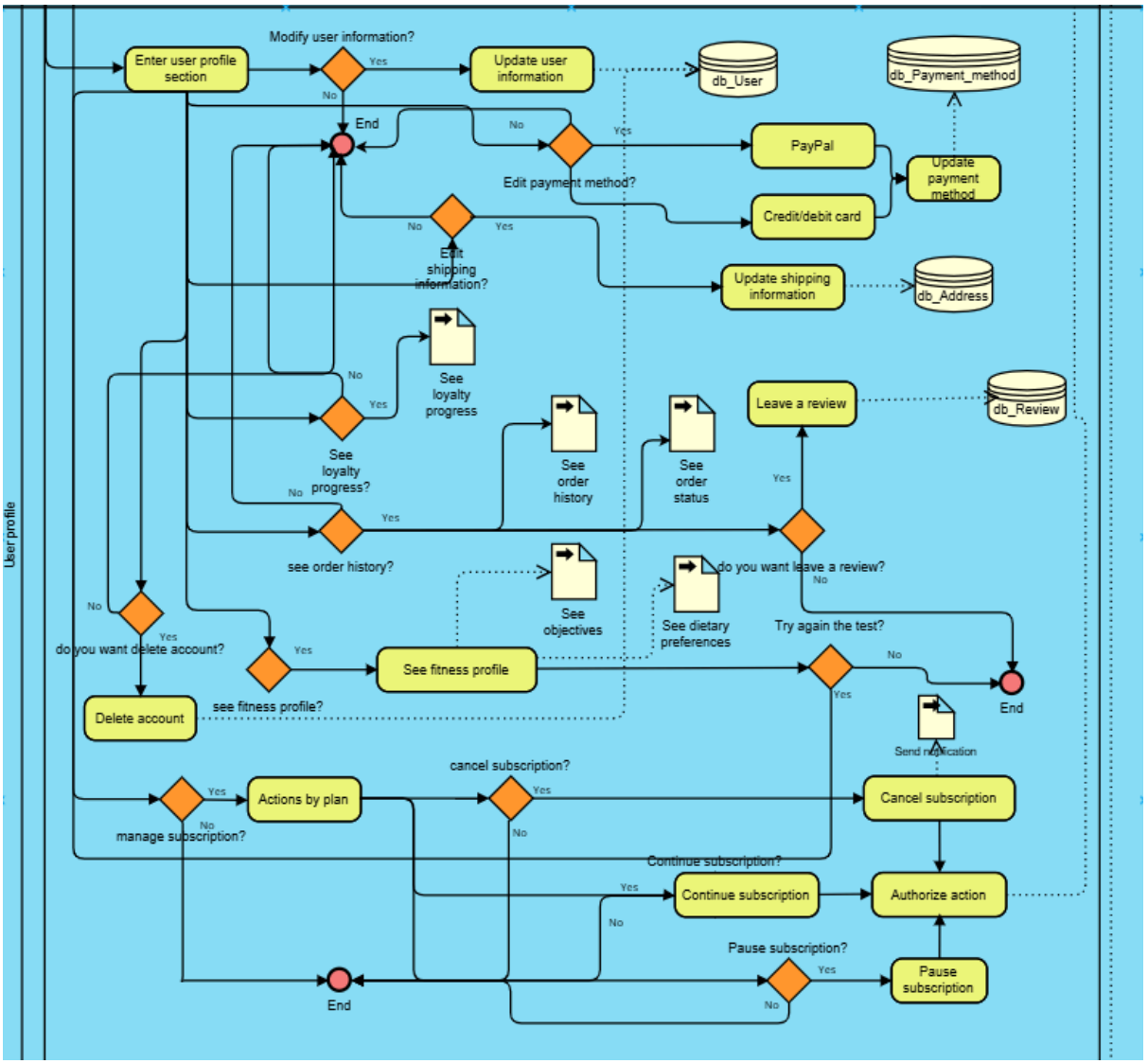
3.5.1.4 Subscription:



- **Purpose**: The client will have the option to pay for and manage a membership to get better benefits.
- **Flow:**

- **The user accesses "Enter subscription section"** to manage their subscription settings.
- **Positioning test verification:**
  - The system asks: "Did you take the test?"
  - **If No** → Navigate to "Select active plan":
    - Users browse available subscription plans and choose one that fits their needs.
  - **If Yes** → Proceed directly to subscription management options (system uses test results to recommend plans).
- **Authorization:**
  - "Authorize action" validates and confirms the user's subscription management decision.
  - The system processes the requested change and updates subscription status.
- **Process completion:**
  - The system will send an order based on a profile automatically monthly and will be saved in db_Subscription database.
  - System ends the subscription management session.

3.5.1.5 User profile:

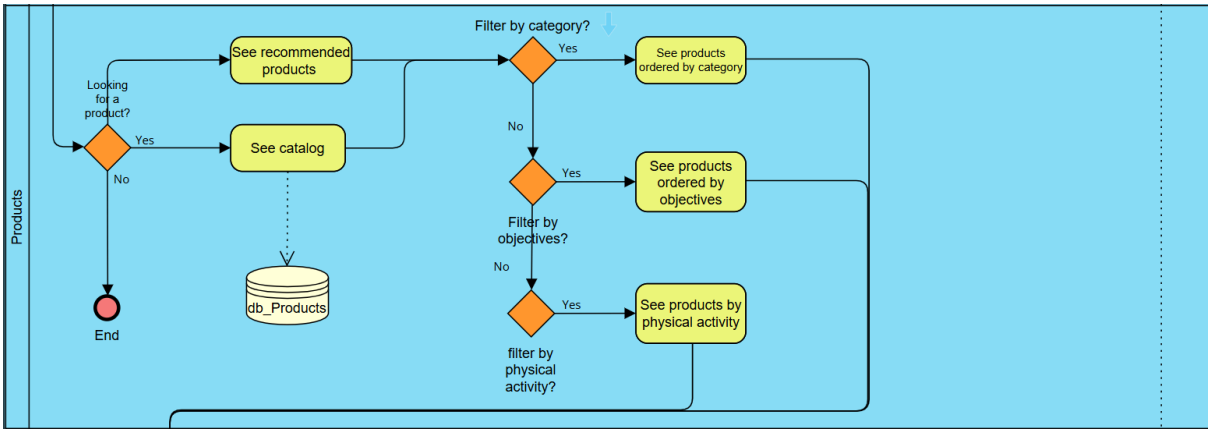


- **Purpose**: The client will have the ability to modify their personal data and payment method.
- **Flow:**
  - **The user accesses "Enter user profile section"** to manage their account settings and information.
  - **Profile information update:**
    - The system asks: "Modify user information?"
    - **If Yes** → Navigate to "Update user information":
      - The user edits personal details (name, email, phone, preferences, etc.).
      - Updated information is saved to db_User database.
    - **If No** → Proceed to other profile management options.
  - **Payment method management:**
    - The system asks: "Edit payment method?"
    - **If Yes** → Display available payment options:
      - **PayPal**: Link or update PayPal account.
      - **Credit/debit card**: Add or modify card information.

            - After selection, the system saves changes to db_Payment_method database.
        - **If No** → Continue to the next option.
    - **Shipping information management:**
        - The system asks: "Edit shipping information?"
        - **If Yes** → Navigate to "Update shipping information":
            - Users add or modify delivery addresses.
            - Updates postal information, contact details for deliveries.
            - Information is saved to db_Address database.
        - **If No** → Continue to the next option.
    - **Loyalty program tracking:**
        - The system asks: "See loyalty progress?"
        - **If Yes** → Display "See loyalty progress" report:
            - Shows points accumulated, rewards earned, tier status.
            - Provides overview of loyalty program benefits.
        - **If No** → Continue to the next option.
    - **Order history review:**
        - The system asks: "See order history?"
        - **If Yes** → Navigate to "See order history":
            - Displays complete purchase records with dates and products.
            - Links to "See order status" for tracking current deliveries and shipment information.
        - **If No** → End process.
    - **Fitness profile viewing:**
        - **If Yes** → Navigate to "See fitness profile":
            - Displays fitness assessment results and personalized recommendations.
            - Shows workout plans and nutrition suggestions based on previous test results.
        - **If No** → Continue to the next option.
    - **Product review submission:**
        - The system asks: "Do you want to leave a review?":
            - User writes and submits product reviews and ratings.
            - Review is saved to db_Review database.
        - **If No** → Continue to the next option.
    - **Positioning test retake option:**
        - The system asks: "Try again the test?"
        - **If Yes** → System loops back to allow user to retake their fitness assessment:
            - User is redirected to the fitness test module.
            - New results will update their fitness profile..
        - **If No** → Continue to the next option.
    - **Account deletion:**
        - The system asks: “Do you want to delete the account?”
        - **If Yes** → Navigate to "Delete account"t:
            - Permanently removes user profile and all associated data from the system.
            - The process ends after successful deletion.

        - **If No** → End process
    - **Subscription action selection:**
    - **Option A - Cancel subscription:**
        - The system asks: "Cancel subscription?"
        - **If Yes** → Execute "Cancel subscription" action.
            - The system sends notification to the user confirming cancellation.
            - Proceed to "Authorize action" for final confirmation.
    - **Option B - Continue subscription:**
        - The system asks: "Continue subscription?"
        - **If Yes** → Execute "Continue subscription" action.
            - Maintain current subscription plan and billing cycle.
            - Proceed to "Authorize action" for confirmation.
    - **Option C - Pause subscription:**
        - The system asks: "Pause subscription?"
        - **If Yes** → Execute "Pause subscription" action.
            - Temporarily suspend subscription without cancellation.
            - Proceed to "Authorize action" for confirmation.
        - **If No** to all options → End process.
    - **Process completion:**
        - All profile management tasks are finalized.
        - Changes are saved across all relevant databases (db_User, db_Payment_method, db_Address, db_Review).
        - User can exit the profile section or continue browsing the platform.
        - The system reaches the "End" state, concluding the user profile management session.

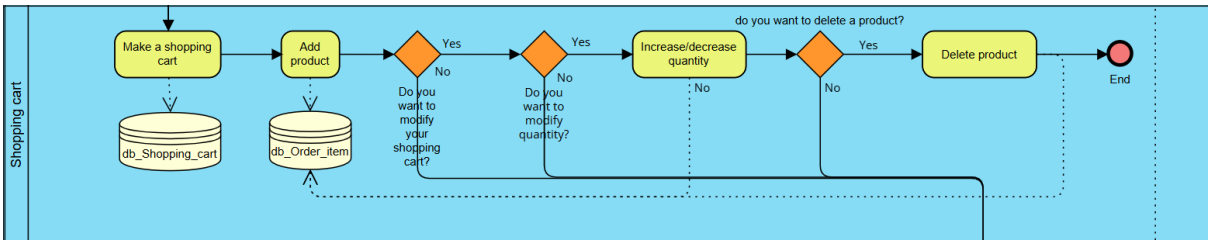
3.5.1.6 Products:



- **Purpose**: To allow users to explore and discover products through personalized recommendations, catalog browsing, and multiple filtering options based on categories, objectives, and physical activities.
- **Flow:**
    - **Product discovery entry point:**
        - The system asks: "Looking for a product?"
        - **If Yes** → Proceed to product browsing options.

        - **If No** → End process.
    - **Initial browsing options: Option A - Recommended products:**
        - Navigate to "See recommended products":
            - The system displays personalized product suggestions based on user profile, purchase history, and positioning test results.
            - Products are curated using recommendation algorithms from the admin panel.
    - **Option B - Complete catalog:**
        - Navigate to "See catalog":
            - Displays full product inventory available in db_Products database.
            - Shows all products without filtering or personalization.
    - **Product filtering system: Filter Level 1 - Category filtering:**
        - The system asks: "Filter by category?"
        - **If Yes** → Navigate to "See products ordered by category":
            - Products are organized by categories (supplements, equipment, apparel, accessories, etc.).
        - **If No** → Proceed to the next filter option.
    - **Filter Level 2 - Objectives filtering:**
        - The system asks: "Filter by objectives?"
        - **If Yes** → Navigate to "See products ordered by objectives":
            - Products are filtered by fitness goals (weight loss, muscle gain, endurance, recovery, etc.).
        - **If No** → Proceed to the next filter option.
    - **Filter Level 3 - Physical activity filtering:**
        - The system asks: "Filter by physical activity?"
        - **If Yes** → Navigate to "See products by physical activity":
            - Products are filtered by sport or activity type (running, weightlifting, yoga, cycling, etc.).
        - **If No** → Display unfiltered results.
    - **Process completion:**
        - User views filtered or unfiltered product results based on their selections.
        - Users can browse products and proceed to product details or purchase.

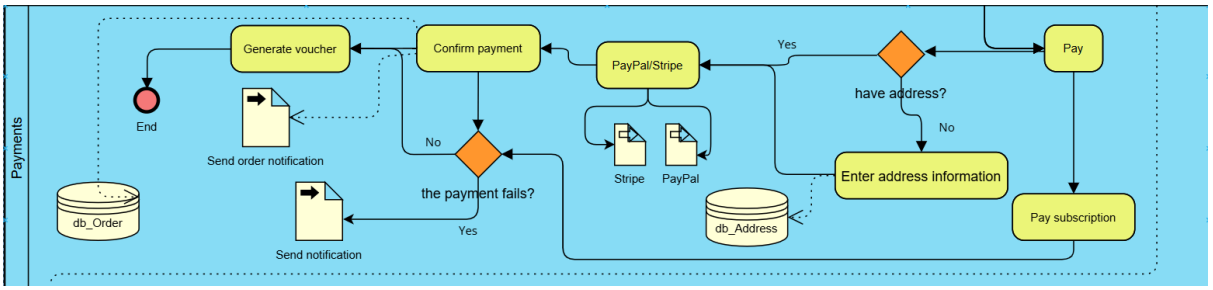
3.5.1.7 Shopping cart:



- **Purpose**: The client will be able to store products they are interested in and will be able to modify and access payment.
- **Flow:**

- **The user initiates "Make a shopping cart"** to begin building their product selection for purchase.
    - Empty cart is created and stored in db_Shopping_cart database.
- **Add products to cart:**
    - Navigate to "Add product":
        - The user selects desired products from catalog or recommendations.
        - Products are added to cart and saved in db_Order_item database.
- **Cart modification options: Option A - Modify shopping cart:**
    - The system asks: "Do you want to modify your shopping cart?"
    - **If Yes** → Proceed to quantity adjustment.
    - **If No** → Skip to deletion option.
- **Option B - Adjust product quantity:**
    - The system asks: "Do you want to modify the quantity?"
    - **If Yes** → Navigate to "Increase/decrease quantity":
        - The user adjusts the number of units for selected products.
        - Updated quantities are saved to db_Order_item database.
    - **If No** → Skip to deletion option.
- **Option C - Remove products:**
    - The system asks: "Do you want to delete a product?"
    - **If Yes** → Navigate to "Delete product":
        - The user removes unwanted items from the cart.
        - Changes are updated in db_Order_item database.
    - **If No** → End process.
- **Process completion:**
    - The shopping cart is finalized with all user modifications.
    - Users can proceed to checkout or continue shopping.
    - Cart data persists in db_Order_item database. for future sessions.

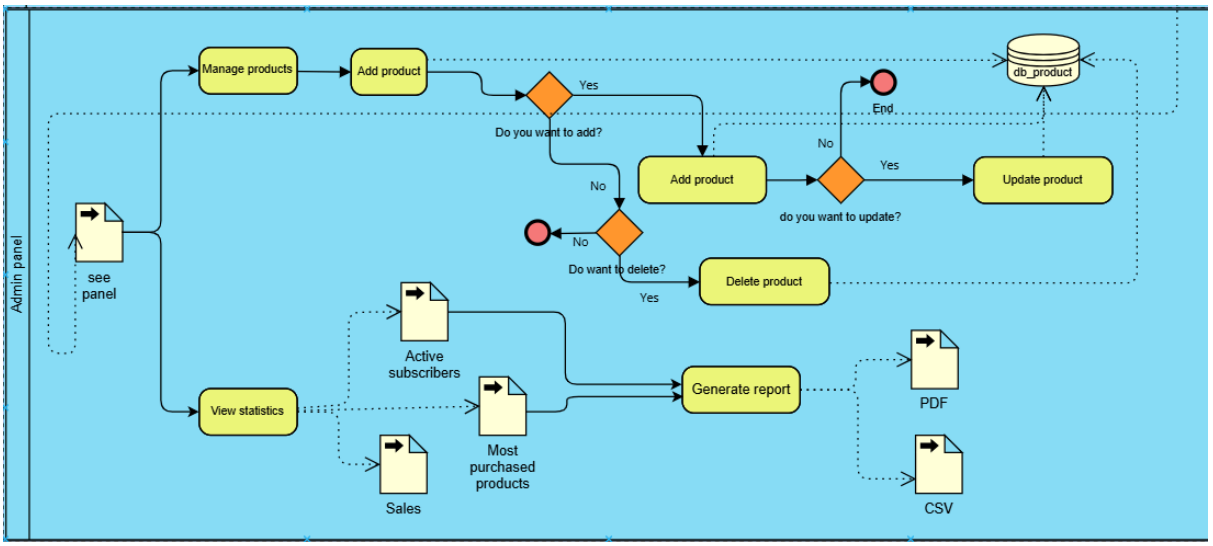
3.5.1.8 Payments:



- **Purpose**: To process secure payments for products and subscriptions through multiple payment methods, including credit/debit cards, PayPal, and Stripe, with transaction confirmation and notification system.
- **Flow:**
    - **The user initiates "Pay"** to begin the checkout and payment process.
    - **Shipping address verification:**
        - **If yes** → Proceed to payment method selection
        - **If no** → Navigate to “Enter address information”:

            - User inputs complete shipping details, such as: address, state, city and zip code.
            - Address is validated and stored for delivery in db_Address database.
    - **Subscription payment (if applicable):**
        - Navigate to "Pay subscription":
            - Processes recurring payment for active subscription plans.
            - Set up an automatic billing cycle.
    - **Payment confirmation:**
        - Navigate to “Confirm payment”:
            - The system validates transaction details and processes payment through a selected method.
            - Payment gateway authorized and captures funds.
    - **Payment confirmation:**
        - Navigate to "Confirm payment":
            - The system validates transactions and processes payment through a selected method.
    - **Payment verification:**
        - The system checks: "The payment fails?"
        - **If Yes** → "Send notification":
            - The user receives an error notification explaining payment failure.
            - The system prompts users to retry with different payment methods or correct information.
        - **If No** → Payment is successful, proceed to order completion.
    - **Order finalization:**
        - Navigate to "Generate voucher":
            - The system creates a purchase receipt/invoice.
            - Transaction details are recorded.
        - "Send order notification":
            - The user receives a confirmation email with order details, tracking information, and voucher.
    - **Process completion:**
        - Payment is successfully processed and recorded.
        - Order is confirmed and ready for fulfillment
        - Order is stored in db_Order database.

3.5.1.9 Administrator panel:



- **Purpose**: To manage the complete lifecycle of products in the system, including creation, modification, deletion, positioning, recommendation logic, and performance analytics.
- **Flow:**
    - **The administrator accesses the admin panel**, which serves as the central hub for all product management operations.

    Manage products section:

    - The system asks: "Do you want to add?"
    - **If Yes** → Navigate to "Add product" form, input product details, and save to **db_product** database.
    - **If No** → The system asks: "Do you want to update?"
        - **If Yes** → Navigate to "Update product" form, modify existing product information, and save changes to db_product database.
        - **If No** → The system asks: "Do you want to delete?"
            - **If Yes** → Execute "Delete product" action and remove from **db_product** database.
            - **If No** → End process.

    View statistics:

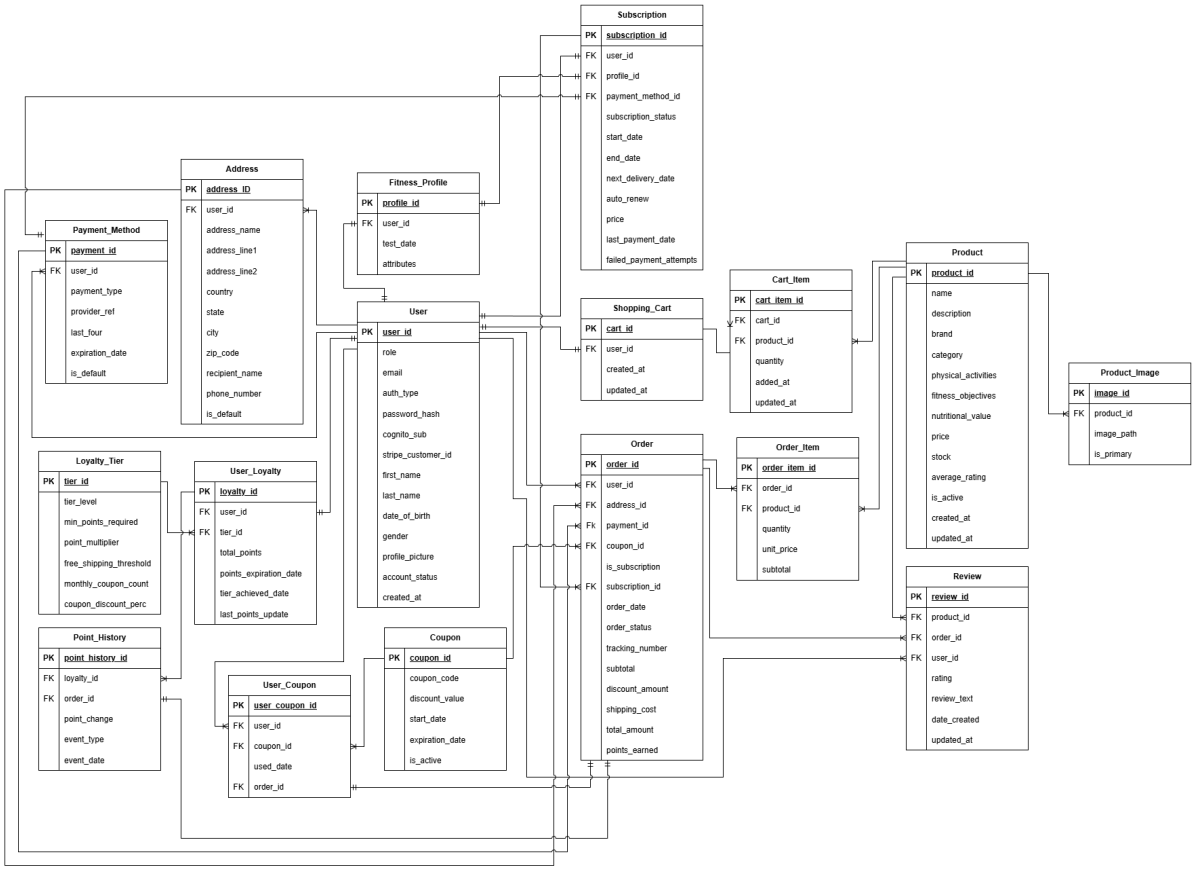
    - Administrator accesses reporting dashboard displaying:
        - **Active subscribers** metrics
        - **Most purchased products** analysis
        - **Sales** performance data
    - The system can automatically reports to compile a comprehensive analytics document in PDF or CSV format, allowing administrators to export and share data insights easily.

    Process completion:

- All operations conclude with an "End" indicator, confirming successful task completion.

3.6 Database diagrams

3.6.1 Entity Relationship Diagram



User

Attributes

Attribute	**Data type**	**Key**	**Description**
user_id	int	Primary Key	Unique identifier for each user
role	enum		Defines user role/privileges (admin, user)
email	varchar	Unique	Email address related to the account (NULL for external auth)
auth_type	enum		Specifies the authentication type used for the account (email, google, facebook)

password_hash	varchar		Hashed password (NULL for external auth)
cognito_sub	varchar		Cognito sub (ID)
stripe_customer_id	varchar		Stripe account identification (NULL if not used stripe)
first_name	varchar		User's name (for profile)
last_name	varchar		User's last name (for profile)
date_of_birth	date		User's DOB
gender	enum		User's gender (M, F, Prefer not say)
profile_picture	varchar		URL to image
account_status	bool		Defines if the account is active or inactive
created_at	datetime		When the account was created

Relationships

Entity	**Relationship type**	**Description**
Fitness_Profile	1 to 1	One user can only have one fitness profile
Address	1 to many	One user can save multiple shipping addresses
Payment_Method	1 to many	One user can save multiple payment methods
Shopping_Cart	1 to 1	One user can only have one shopping cart
Order	1 to many	One user can have multiple orders
Review	1 to many	One user can make multiple reviews
Subscription	1 to 1	One user can have one subscription
User_Loyalty	1 to 1	One user has one loyalty record
User_Coupon	1 to many	One user can have/apply multiple coupons

Fitness_profile

Attributes

Attribute	Data type	Key	Description
profile_id	int	Primary Key	Unique identifier for each profile
user_id	int	Foreign Key	Associated user
test_date	date		Date test was last taken
attributes	JSON		Weight of the test results

Relationships

Entity	Relationship type	Description
User	1 to 1	Each profile is associated with a unique user
Subscription	1 to 1	A fitness profile is used to create a subscription

Address

Attributes

Attribute	Data type	Key	Description
address_id	int	Primary Key	Unique identifier for each address
user_id	int	Foreign Key	Associated user
address_name	varchar		Label/Identifier (NULL if not given)
address_line1	varchar		First line of address
address_line2	varchar		Second line of address (NULL if not given)
country	varchar		Country of address
state	varchar		State of address
city	varchar		City of address
zip_code	varchar		Zip code of address
recipient_name	varchar		Name for the recipient of the package
phone_number	varchar		Phone number for

			contact
is_default	bool		Defines if it is used by default at checkout

Relationships

Entity	**Relationship type**	**Description**
User	Many to 1	Many addresses can be saved by one user
Order	1 to many	One address can be used for multiple orders

Payment_Method

Attributes

Attribute	**Data type**	**Key**	**Description**
payment_id	int	Primary Key	Unique identifier for each payment method
user_id	int	Foreign Key	Associated user
payment_type	enum		Paypal, credit card, debit card
provider_ref	varchar		Token or reference from Stripe or PayPal
last_four	varchar		For identification and display
expiration_date	varchar		For display
is_default	bool		Defines if it is used by default at checkout

Relationships

Entity	**Relationship type**	**Description**
User	Many to 1	Many payment methods can be added by a user
Order	1 to many	One payment method can be used in multiple orders
Subscription	1 to 1	One payment method can be associated to

		one subscription

Shopping_Cart

Attributes

Attribute	**Data type**	**Key**	**Description**
cart_id	int	Primary Key	Unique identifier for the shopping cart
user_id	int	Foreign Key	Associated user
created_at	datetime		Date in which cart was created
updated_at	datetime		Date in which the cart was last updated

Relationships

Entity	**Relationship type**	**Description**
User	1 to 1	One cart is associated with one unique user
Cart_item	1 to many	One cart can hold multiple items/products.

Cart_Item

Attributes

Attribute	**Data type**	**Key**	**Description**
cart_item_id	int	Primary Key	Unique identifier for the cart-item relationship
cart_id	int	Foreign Key	Associated cart
product_id	int	Foreign Key	Associated product
quantity	int		Quantity of each product in cart
added_at	datetime		Date in which items were added to cart
updated_at	datetime		Date in which

			relationship was updated

Relationships

Entity	**Relationship type**	**Description**
Shopping_Cart	Many to 1	Many cart-item relationships can be related to one cart
Product	Many to 1	Many cart-item relationships can include one product

Order

Attributes

Attribute	**Data type**	**Key**	**Description**
order_id	int	Primary Key	Unique identifier for an order
user_id	int	Foreign Key	Associated user
address_id	int	Foreign Key	Associated shipping address
payment_id	int	Foreign Key	Associated payment method
coupon_id	int	Foreign Key	Used coupon (Nullable)
is_subscription	bool		Identifies if the order is related to subscription
subscription_id	int	Foreign Key	Associated subscription (Nullable)
order_date	date		Defines the date in which the order was placed
order_status	enum		Defines the status in which the order is (pending, paid, shipped, delivered, cancelled)

tracking_number	varchar		Tracking number related to the order shipment
subtotal	decimal		Sum of products totals
discount_amount	decimal		Calculated value of amount to discount if coupon is used
shipping_cost	decimal		Cost of shipping
total_amount	decimal		Sum of subtotal plus shipping (final cost)
points_earned	int		Amount of loyalty points earned with the purchase

Relationships

Entity	**Relationship type**	**Description**
User	Many to 1	Multiple orders can be done by one user
Address	Many to 1	Multiple orders can be shipped to the same address
Payment_Method	Many to 1	Multiple orders can be paid with the same payment method
Order_Item	1 to many	One order can have multiple items/products
Review	1 to many	One order can span many reviews (for different products)
Subscription	Many to 1	Multiple orders can be from one subscription
Coupon	Many to 1	Many orders can use the same coupon
User_Coupon	1 to 1	One order can generate one coupon usage record
Point_History	1 to 1	One order can update a user's point history once.

Order_Item

Attributes

Attribute	Data type	Key	Description
order_item_id	int	Primary Key	Unique identifier for the order-item relationship.
order_id	int	Foreign Key	Associated order
product_id	int	Foreign Key	Associated product
quantity	int		Quantity of each product in the order
unit_price	decimal		Cost of singular product
subtotal	decimal		Product of the individual product cost times the quantity

Relationships

Entity	Relationship type	Description
Order	Many to 1	Many order-item relationships can be associated with one order
Product	Many to 1	Many order-item relationships can include one product

Product

Attributes

Attribute	Data	Key	Description
product_id	int	Primary Key	Unique identifier for a product
name	varchar		Name of the product
description	text		Description of the

			product
brand	varchar		Brand of the product
category	varchar		Category associated
physical_activities	json		List of physical activities that are related to the product
fitness_objectives	json		List of fitness objectives that are related to the product
nutritional_value	text		Nutritional and dietary information
price	decimal		Unitary price for the product
stock	int		Amount of product in stock
average_rating	decimal		Average rating of product based on reviews
is_active	bool		Defines if user is active or inactive
created_at	datetime		Date in which product was created
updated_at	datetime		Date in which product was last updated

Relationships

Entity	**Relationship type**	**Description**
Cart_Item	1 to many	One product can take part in many

		cart-item relationships
Order_Item	1 to many	One product can take part in many order-item relationships
Review	1 to many	One product can have multiple reviews
Product_Image	1 to many	One product can have multiple images

Product_Image

Attributes

Attribute	**Data type**	**Key**	**Description**
image_id	int	Primary Key	Unique identifier for an image
product_id	int	Foreign Key	Associated product
image_path	varchar		URL to image
is_primary	bool		Defines if it's the main image for the product

Relationships

Entity	**Relationship type**	**Description**
Product	Many to 1	Many images can be associated to one product

Review

Attributes

Attribute	**Data type**	**Key**	**Description**
review_id	int	Primary Key	Unique identifier for a review
product_id	int	Foreign Key	Product associated

order_id	int	Foreign Key	Order associated
user_id	int	Foreign Key	User associated
rating	decimal		Value (1-5) given by the user
review_text	text		Written opinion (NULL if not given)
date_created	datetime		Date the review was written/ edited on
updated_at	datetime		Date when review was last updated

Relationships

Entity	**Relationship type**	**Description**
Product	Many to 1	Many reviews can exist for the same product
Order	Many to 1	Multiple reviews can be created from one order
User	Many to 1	Multiple reviews can be done by one user

Subscription

Attributes

Attribute	**Data type**	**Key**	**Description**
subscription_id	int	Primary Key	Unique identifier for a subscription
user_id	int	Foreign Key	Associated user
profile_id	int	Foreign Key	Associated fitness profile
payment_method_id	int	Foreign Key	Associated payment

			method
subscription_status	enum		Defines the status of the subscription (active, suspended, cancelled)
start_date	date		Date in which the subscription was started
end_date	date		Date in which the subscription ends
next_delivery_date	date		Date for next delivery (calculated)
auto_renew	bool		Defines if the subscription should auto renew after end date
price	decimal		Price of a subscription
last_payment_date	datetime		Date of last payment
failed_payment_attempts	int		Amount of failed attempts to get payment

Relationships

Entity	**Relationship type**	**Description**
User	1 to 1	One subscription is associated to one user
Fitness_Profile	1 to 1	One subscription is based on one fitness profile
Order	1 to many	One subscription can generate multiple orders
Payment_Method	1 to 1	One subscription can be paid for with one payment method

Coupon

Attributes

Attribute	**Data type**	**Key**	**Description**
coupon_id	int	Primary Key	Unique identifier for coupons
coupon_code	varchar		Code for coupon usage at checkout
discount_value	decimal		Percentage to be discounted
start_date	date		Date in which the coupon becomes active
expiration_date	date		Date in which the coupon becomes inactive
is_active	bool		Defines if the coupon is active

Relationships

Entity	**Relationship type**	**Description**
User_Coupon	1 to many	One coupon can appear in different user's histories
Order	1 to many	One coupon can be used in different orders

User_Coupon

Attributes

Attribute	**Data type**	**Key**	**Description**
user_coupon_id	int	Primary Key	Unique identifier for the user-coupon relationship

user_id	int		Associated user
coupon_id	int		Associated coupon
used_date	date		Date in which coupon was used
order_id	int		Order in which coupon was used

Relationships

Entity	**Relationship type**	**Description**
User	Many to 1	Many user-coupon relationships can be associated to one user
Coupon	Many to 1	Many user-coupon relationships can stem from one coupon
Order	1 to 1	One user-coupon entry can originate from one order

User_Loyalty

Attributes

Attribute	**Data type**	**Key**	**Description**
loyalty_id	int	Primary Key	Unique identifier for the user's loyalty program relationship
user_id	int	Foreign Key	Associated user
tier_id	int	Foreign Key	Associated tier
total_points	int		Total points that the user has
points_expiration_date	datetime		Date in which points expire

tier_achieved _date	date		Date in which tier was changed
last_points_update	date		Last date in which points where updated

Relationships

Entity	**Relationship type**	**Description**
User	1 to 1	One user-loyalty relationship belongs to one user
Point_History	1 to many	One user-loyalty relationship can update point history multiple times
Loyalty_Tier	Many to 1	Multiple user-loyalty relationships can reference the same tier

Point_History

Attributes

Attribute	**Data type**	**Key**	**Description**
point_history_id	int	Primary Key	Unique identifier for the point history record
loyalty_id	int	Foreign Key	User's loyalty profile associated
order_id	int	Foreign Key	Order associated with the change (nullable if event is expiration)
point_change	int		Amount of points changed (positive or negative)
event_type	enum		Type of event (earned, expired)

event_date	date		Date of event

Relationships

Entity	Relationship type	Description
User_Loyalty	Many to 1	Many point movements can be associated to one user's loyalty profile
Order	1 to 1	One point movement can be related to one order

Loyalty_Tier

Attributes

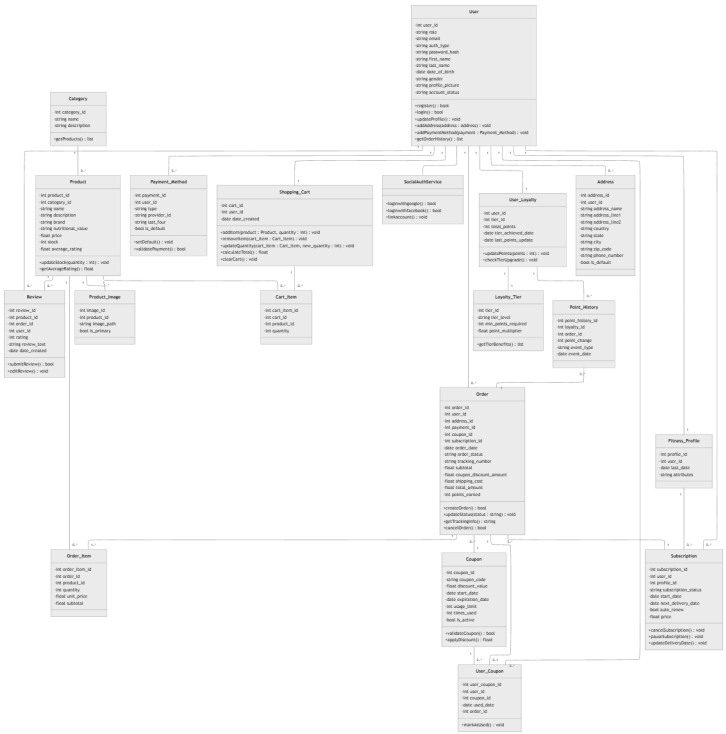
Attribute	Data type	Key	Description
tier_id	int	Primary Key	Unique identifier for each tier
min_point_required	int		Minimal amount of point required to reach tier
point_multiplier	decimal		Multiplier for points earned associated to the tier
free_shipping_threshold	decimal		Amount needed for order to have free shipping depending on tier
monthly_coupon_count	int		How many coupons are given to user
coupon_discount_perc	int		Percentage of discount in coupons

Relationships

Entity	Relationship type	Description

User_Loyalty	1 to many	One tier can be in multiple loyalty profiles

3.7 Class diagram



3.7.1 Class Diagram Description

User

Attributes:

-int user_id

-string role

-string email

-string auth_type

-string password_hash

-string first_name

-string last_name

-date date_of_birth

-string gender

-string profile_picture

-string account_status

Methods:

+register() bool

+login() bool

+updateProfile() void

+addAddress(address : Address) void

+addPaymentMethod(payment : Payment_Method) void

+getOrderHistory() list

Methods:

+addAddress(address : Address) void

+addPaymentMethod(payment : Payment_Method) void

+getOrderHistory() list

+resetPassword() void

The User class has a connection with Fitness_Profile, 1 to 1.

The User class has a connection with Address, 1 to N.

The User class has a connection with Payment_Method, 1 to N.

The User class has a connection with Shopping_Cart, 1 to 1.

The User class has a connection with Order, 1 to N.

The User class has a connection with Review, 1 to N.

The User class has a connection with Subscription, 1 to N.

The User class has a connection with User_Loyalty, 1 to 1

The User class has a connection with User_Coupon, 1 to N.

The User class uses the SocialAuthService.

The User class has a connection with Notification, 1 to N.

Fitness_Profile

Attributes:

-int profile_id

-int user_id

-date test_date

-string attributes

The Fitness_Profile class has a connection with **Subscription**, 1 to N.

Address

Attributes:

-int address_id

-int user_id

-string address_name

-string address_line1

-string address_line2

-string country

-string state

-string city

-string zip_code

-string phone_number

-bool is_default

Payment_Method

Attributes:

-int payment_id

-int user_id

-string type

-string provider_id

-string last_four

-date expiration_date

-bool is_default

Methods:

+setDefault() void

+validatePayment() bool

Shopping_Cart

Attributes:

-int cart_id

-int user_id

-date

-date_created

Methods:

+addItem(product : Product, quantity : int) void

+removeItem(cart_item : Cart_Item) void

+updateQuantity(cart_item : Cart_Item, new_quantity : int) void

+calculateTotal() float

+clearCart() void

The Shopping_Cart class has a connection with Cart_Item, 1 to N.

Cart_Item

Attributes:

-int cart_item_id

-int cart_id

-int product_id

-int quantity

Order

Attributes:

-int order_id

-int user_id

-int address_id

-int payment_id

-int coupon_id

-int subscription_id

-date order_date

-string order_status

-string tracking_number

-float subtotal

-float coupon_discount_amount

-float shipping_cost

-float total_amount

-int points_earned

Methods:

+createOrder() bool

+updateStatus(status : string) void

+getTrackingInfo() string

+cancelOrder() bool

The Order class has a connection with Order_Item, 1 to N.

The Order class has a connection with Coupon, N to 1.

The Order class has a connection with Subscripti**on**, N to 1.

The Order class has a connection with User_Coupon, 1 to 1

The Order class has a connection with Point_History, 1 to N

Order_Item

Attributes:

-int order_item_id

-int order_id

-int product_id

-int quantity

-float unit_price

-float subtotal

Product

Attributes:

-int product_id

-int category_id

-string name

-string description

-string brand

-string nutritional_value

-float price

-int stock

-float average_rating

Methods:

+updateStock(quantity : int) void

+getAverageRating() float

The Product class has a connection with Product_Image, 1 to N.

The Product class has a connection with Category, N to 1.

The Product class has a connection with Cart_Item and Order_Item, 1 to N.

Review

Attributes:

-int review_id

-int product_id

-int order_id

-int user_id

-int rating

-string review_text

-date date_created

Methods:

+submitReview() bool

+editReview() void

Product_Image

Attributes:

-int image_id

-int product_id

-string image_path

-bool is_primary

Category

Attributes:

-int category_id

-string name

-string description

Methods:

+getProducts() list

Subscription

Attributes:

-int subscription_id

-int user_id

-int profile_id

-string subscription_status

-date start_date

-date next_delivery_date

-bool auto_renew

-float price

Methods:

+cancelSubscription() void

+pauseSubscription() void

+updateDeliveryDate() void

The Subscription class has a connection with **Order**, 1 to N.

Loyalty_Tier

Attributes:

-int tier_id

-string tier_level

-int min_points_required

-float point_multiplier

Methods:

+getTierBenefits() list

The Loyalty_Tier class has a connection with **User_Loyalty**, 1 to N.

User_Loyalty

Attributes:

-int user_id

-int tier_id

-int total_points

-date tier_achieved_date

-date last_points_update **Methods:**

+updatePoints(points : int) void

+checkTierUpgrade() void

The User_Loyalty class has a connection with **Point_History**, 1 to N.

Point_History

Attributes:

-int point_history_id

-int loyalty_id (FK to User_Loyalty)

-int order_id

-int point_change

-string event_type

-date event_date

Coupon

Attributes:

-int coupon_id

-string coupon_code

-float discount_value

-date start_date

-date expiration_date

-int usage_limit

-int times_used

-bool is_active

Methods:

+validateCoupon() bool

+applyDiscount() float

The Coupon class has a connection with **User_Coupon**, 1 to N.

The Coupon class has a connection with **Order**, 1 to N.

User_Coupon

Attributes:

-int user_coupon_id

-int user_id

-int coupon_id

-date used_date

-int order_id

Methods:

+markAsUsed() void

Api

Attributes:

+loginwithgoogle() bool

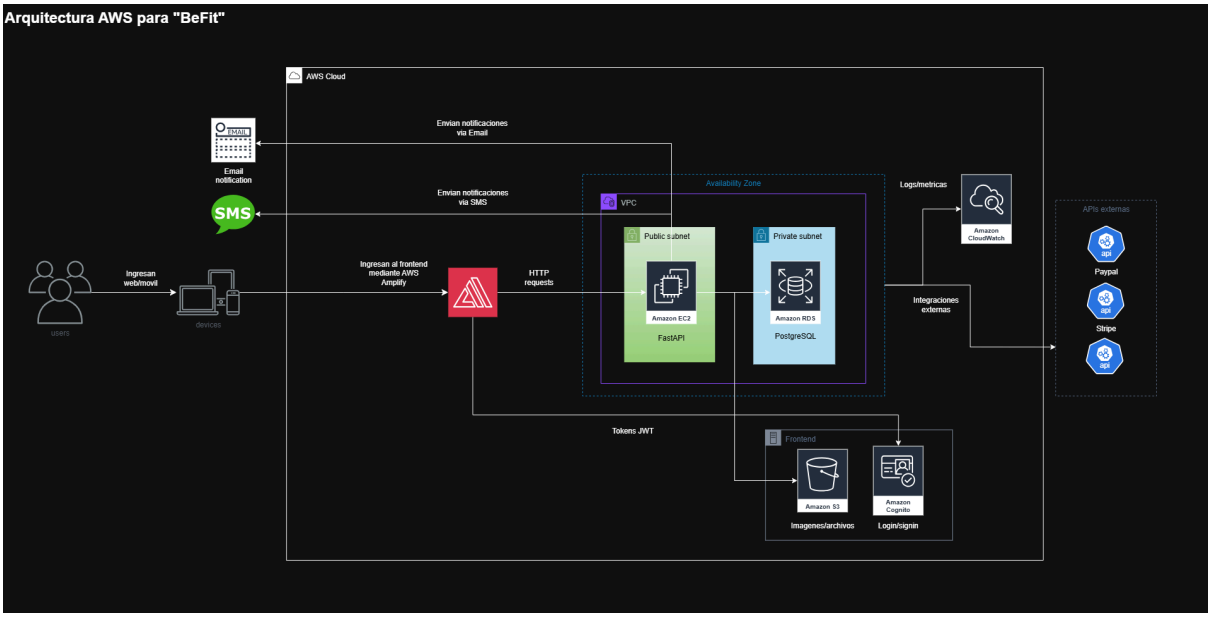
+loginwithfacebook() bool

+linkaccount() void

3.8 Design Constraints

https://www.figma.com/design/WaS7B5Rg99ll3yqHYbcc5B/BeFit-Prototype?node-id=0-1&t=P5MxxgFtxIxBqSM0-1

3.9 AWS architecture



3.10 Software system attributes

3.10.1 Scalability

The system is designed to handle growth in user base, data volume, and transaction frequency. It also employs a model that allows the system to add and collaborate with other services, such as PayPal and Stripe.

3.10.2 Availability

The system must be up for use 24 hours a day, 7 days a week, 365 a year. This is in order to ensure that users can access the system at any moment, at any time, as it is important to the business model. System maintenance and problem fixing must be done while the system is up with utmost briefness to avoid disturbing user experience.

3.10.3 Functionality

The system provides core functionalities such as user authentication, personalized product recommendations, product browsing, and flexible purchasing through subscriptions or single orders. It includes secure payments, notifications, an admin panel for management, and a loyalty program to enhance user engagement. All features are accessible to authenticated users and follow defined business rules.

3.10.4 Security

The system ensures security through HTTPS, password hashing. Only registered users can access sensitive features. Payments are securely processed via PayPal and Stripe to prevent data breaches.

3.10.5 Usability

The system is designed in a simple and user-friendly way to provide a seamless experience. It has optimized responsiveness and complies with accessibility guidelines so that anyone can use the system.

3.10.6 Reliability

The system is designed to support a high number of users accessing it concurrently without performance degradation.

3.11 Organizing the specific requirements

To prioritize the requirements of the system for BeFIt, the MoSCOW method was used, which classifies requirements into four categories:

- **Must Have:** Core functionalities that are critical for the system´s operation.
- **Should Have:** Important functionalities that enhance system performance or user experience.
- **Could Have:** Non-essential features that provide additional value or convenience.
- **Won´t Have:** Requirements that have been explicitly excluded from the current scope of development

Must Have	**Should Have**
● Sign Up ● Log In ● Positioning Test ● Payment Method ● Shopping Cart ● Order Processing ● Subscription ● Automatic Orders ● Admin Panel	● User Profile ● Catalogue / Home Page ● Product Listing ● Product Review ● Loyalty Program ● Push Notification System
Could Have	**Won´t Have**
● Machine Learning Model ● Administrative Reports (Downloadable Formats)	● Integration with physical Stores ● Medical Reports ● Integration with Health API (Google

	Fit)

4 Supporting information

4.1 Table of contents and index

4.2 Appendixes

4.2.1 Daily meetings

The team meets via Microsoft Teams to discuss scheduled activities, as well as to review progress and provide feedback on those activities. These meetings are documented using video recordings and a supporting document, which includes information such as:

- The attendees and absentees
- The date and time of each meeting
- Any attached materials, such as the call transcription and other documents
- The purpose of the meeting
- A summary of the topics discussed
- Action plans to address identified issues

The documentation of the meeting minutes was organized into two key stages. Phase One was dedicated to conceptualization, recording the presentation of ideas by the team, the analyses performed, and the evaluation of the proposals, until the project with the best qualities was selected.

Phase Two focuses on the already chosen project, documenting the division of tasks, the creation of the prototype, and the management of inquiries and resolution of doubts, among other development aspects.

Phase 1: Documento-Minutas.docx

Phase 2: Minutas-Etapa2.docx

4.2.2 Activities schedule

The project management was done with ClickUp, where the team divided the activities regarding the elaboration of this document. The division of responsibilities and dates was done in the following matter:

BeFit Task Management		Sprint 1							
		20/10/2025	21/10/2025	22/10/2025	23/10/2025	25/10/2025	24/10/2025	26/10/2025	27/10/2025
Task	**Responsable**	**M**	**T**	**W**	**T**	**S**	**F**	**S**	**M**
Levantamiento de requerimientos	Todos								
Imagen corporativa	Todos								
SRS									
Sección 1									
1.1 Introducción y proposito	Gabriel V. & Lizbeth B.								
1.2 Alcance	Luis F.								
1.3 Definiciones	Todos								
1.4 Referencias	Todos								
1.5 Overview	Lizbeth B.								
Sección 2									
2.1 Perspectiva de producto	Luis F.								
2.2 Funciones del producto	Gabriel V. & Luis F. & Diego J.								
2.3 Caracteristicas de usuario	Gabriel V. & Lizbeth B.								
2.4 Restricciones	Diego J.								
2.5 Dependencias	Lizbeth B.								
2.6 Distribucion de Requerimientos	Lizbeth B.								
Sección 3									
3.1 Interfaces externas	Ricardo R.								
3.2 Funciones	Lizbeth B. & Ricardo R.								
3.3 Requerimientos funcionales	Diego J.								
3.4 Reglas de negocio	Luis F.								
3.5 Diagrama de procesos	Luis F. & Gabriel V. & Diego J.								
3.6 Diagrama ER	Lizbeth B.								
3.7 Diagrama de clases	Ricardo R.								
3.8 Restricciones de diseño	Ricardo R.								
3.9 Atributos de software	Gabriel V.								
3.10 Organización de requerimientos	Diego J.								
Sección 4									
4.1 Indice y tabla de contenidos	Luis F.								
4.2 Minutas	Luis F.								
4.3 Cronograma de actividades	Lizbeth B.								
4.4 Matrices de riesgo	Todos								

Gantt diagram: Diagrama de Gantt & Matrices de riesgo

4.3 Risk matrix

During the development of this SRS document, risk matrices were created for each completed sprint. Each matrix includes the probability of occurrence, the impact it will have, the justification for the impact and its probability, and its mitigation. These matrices are aligned with the project´s activity schedule.

Risk matrix: Diagrama de Gantt & Matrices de riesgo

4.4 Manuals

During the development of this project, the integration of APIs and the use of cloud infrastructure is a crucial point. Considering that not everyone has the same experience in these processes, two types of manuals were created to help team members learn more about these processes, as well as the tools provided by AWS.

AWS guide: Manuales AWS

API integration: Integracion APIs