Software Requirements Specification

for

Nutrifit

Version 1.0 approved

Prepared by Ilirjana Didani and Minella Bishka

Epoka University

April 05, 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document specifies the requirements for the Nutrifit software. Nutrifit is a workout and meal planning application that helps users achieve their fitness goals by providing personalized meal plans and workout routines. This document covers the initial release of Nutrifit, version 1.0.

The scope of this SRS is to describe the software requirements for the entire Nutrifit system, including all its features and functionalities. It covers both the front-end user interface and the back-end functionality required to support it. This SRS is intended for use by the development team responsible for building Nutrifit.

It should be noted that this SRS does not cover any hardware requirements or external systems that may be necessary for Nutrifit to operate. Furthermore, while the initial release of Nutrifit may focus on specific aspects of the system, such as meal planning or workout routines, this SRS aims to cover the entire system in its entirety.

1.2 Document Conventions

This SRS document follows the conventions and standards set forth by the project team. The subsequent conventions were followed when writing this document:

- Font: The entire document was written using a standard 12-point font, such as Times New Roman.
- Section Headings: All section headings were written using a standard 14-point font, such as Times New Roman, in bold and capitalized.
- Requirements: All requirements statements were written in plain text and numbered consecutively.

Furthermore, it is assumed that the priorities assigned to higher-level requirements are inherited by their detailed requirements unless otherwise stated. Moreover, every requirement statement will have its priority level assigned.

1.3 Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) document is intended for a variety of readers, including developers, project managers, marketing staff, users, testers, and documentation writers.

Each group may be interested in different aspects of the document, and as such, may require different levels of detail.

This SRS contains a comprehensive overview of the Nutrifit system, including its features and functionalities, as well as its requirements for successful implementation. The document is organized into these sections:

- Introduction: Provides an overview of the Nutrifit system and its purpose, as well as the conventions and standards used in the document.
- Overall Description: Describes the general factors that affect the product and its requirements, such as user needs.
- External Interface Requirements: Describes the system's interfaces such as user interfaces, hardware, and software interfaces.
- System Features: Describes the specific features and functionalities of the Nutrifit system.
- Other Nonfunctional Requirements: Describes additional requirements related to the system's performance, security, and other nonfunctional aspects.
- Other Requirements: Describes any additional requirements not covered in the previous sections.

For readers interested in a general understanding of the Nutrifit system, we suggest reading the Introduction and Overall Description sections first, followed by the System Features section for a detailed overview of the system's capabilities. External Interface Requirements and Other Nonfunctional Requirements may also be relevant to those interested in system integration or performance.

Developers and testers may be particularly interested in the System Features section, as it outlines the specific requirements and functionalities they will need to implement and test.

The project managers and marketing staff may find the Overall Description section particularly useful for understanding the system's goals and target audience. Finally, documentation writers may find the glossary and other appendices particularly useful for ensuring consistency and accuracy in their documentation.

1.4 Product Scope

Nutrifit is a software application designed to help users create and manage customized workout and meal plans. The primary purpose of Nutrifit is to provide users with a convenient and efficient way to track their fitness goals and progress.

The Nutrifit software offers a variety of features and functionalities, including the ability to create and customize workout and meal plans, track progress, and receive personalized recommendations based on user data. Nutrifit aims to help users achieve their fitness goals by providing a comprehensive and easy-to-use platform for tracking and managing their workouts and nutrition.

The software is designed for individuals who are looking to improve their fitness and health, including gym-goers, athletes, and fitness enthusiasts. Nutrifit can be used in a variety of settings, including at home, at the gym, or on the go, making it a flexible and accessible tool for users.

By providing users with an efficient and customizable solution for managing their fitness and nutrition, Nutrifit aims to improve user satisfaction and adherence to fitness goals, ultimately leading to improved health and well-being.

1.5 References

2. Overall Description

2.1 Product Perspective

The Nutrifit software is a new, self-contained product that is being developed to provide users with an intuitive and efficient way to manage their workout and meal plans. Nutrifit is not a member of a product family or a replacement for any existing systems.

Nutrifit is designed to be a standalone product, but it may interface with other systems or components in the future. Any interfaces between Nutrifit and other systems will be identified and documented in the External Interface Requirements section of this SRS.

A high-level diagram of the Nutrifit system and its major components is provided in the System Features section.

2.2 Product Functions

Nutrifit must provide the succeeding major functions:
- User registration and profile creation, including the user preferences and needs
- Input and tracking of meal plans
- Input and tracking of workout plans
- Personalized workout and meal plan recommendations

A high-level data flow diagram that shows the major groups of related requirements and how they relate is provided in the System Features section. This diagram will provide a visual representation of the major functions of the Nutrifit system and how they are related to one another.

2.3 User Classes and Characteristics

Nutrifit is designed for the succeeding user classes:

1. General users: These are individuals who are interested in tracking their meals and workouts to improve their overall fitness. They have a basic understanding of nutrition and exercise, but may not be experts in the field.
2. Fitness enthusiasts: These are individuals who are passionate about fitness and have a good understanding of nutrition and exercise. They may be interested in tracking their macros (protein, carbs, and fat), creating customized meal plans, and setting specific fitness goals.
3. Personal trainers: These are individuals who have a high level of expertise in nutrition and exercise and work with clients to create personalized meal and workout plans. They may use Nutrifit to manage their clients' plans and monitor their progress.

The characteristics of each user class are as follows:
- General users: They are interested in improving their fitness and want a simple and user-friendly interface to track their meals and workouts. They may not have a lot of technical expertise and may require some guidance to get started with the app.
- Fitness enthusiasts: They have a good understanding of nutrition and exercise and want to use Nutrifit to track their progress and achieve their fitness goals. They are comfortable with technology and may want to customize their meal and workout plans.
- Personal trainers: They have a high level of expertise in nutrition and exercise and want to use Nutrifit to create and manage personalized meal and workout plans for their clients. They are comfortable with technology and may require advanced features such as progress tracking and data analytics.

It is important to ensure that the Nutrifit system is designed to meet the needs of all three user classes, with a focus on providing a user-friendly interface for general users, advanced features for fitness enthusiasts, and powerful tools for personal trainers.

2.4 Operating Environment

The operating environment for Nutrifit includes the succeeding:

2.4.1 Hardware Platform

- Desktop computers or laptops running Windows 10, MacOS, or Linux
- Mobile devices running iOS 12 or later, or Android 8.0 or later

2.4.2 Operating System

- Windows 10 (64-bit) or later
- MacOS 10.13 or later
- Linux with GTK 3.18 or later

2.4.3 Software Components

- Internet connection for initial setup and cloud-based synchronization
- Database management system (e.g. MySQL)

Besides that, the software must peacefully coexist with commonly used web browsers such as Google Chrome, Mozilla Firefox, and Safari.

2.5 Design and Implementation Constraints

For the design and implementation constraints of Nutrifit software, these items should be considered:
- The software should be designed and implemented in compliance with all relevant laws and regulations related to the healthcare and fitness industry.
- The software should be able to run on common hardware platforms such as desktop computers, laptops, and mobile devices (iOS and Android).
- The software should be designed to work with the latest operating systems and versions, including Windows, macOS, and mobile operating systems.
- The software should be able to integrate with third-party applications, such as fitness trackers and meal-tracking apps.
- The software should be developed using industry-standard programming languages and frameworks.
- The software should adhere to established design conventions and programming standards to ensure maintainability and ease of modification.
- The software should be designed with security in mind and implement appropriate measures to protect user data and prevent unauthorized access.
- The software should be developed with scalability in mind to allow for future growth and expansion.
- The software should be designed to work in both online and offline environments.
- The software should be developed using agile methodology to allow for frequent updates and feature additions based on user feedback.

2.6 User Documentation

The subsequent user documentation components will be delivered with the software:
- User manual: A comprehensive guide to the software's features, functions, and user interface. The manual will be provided in both PDF and HTML formats.
- Online help: Context-sensitive help will be available within the software, assisting users as they navigate through different features and functions.

- Tutorials: A series of step-by-step tutorials will be provided to help users learn how to perform specific tasks with the software.

All user documentation will comply with the company's documentation standards and will be delivered in English.

2.7 Assumptions and Dependencies

Assumptions:

- The development team has access to the necessary hardware and software tools to develop and test the software.
- The software will be used in a standard operating environment with no unusual constraints or limitations.
- Users will have basic computer literacy and will be able to use the software with minimal training.
- The software will be developed using industry-standard programming practices and design principles.
- Required interfaces to external systems will be available and accessible.

Dependencies:

- The software is dependent on the availability and compatibility of third-party libraries and frameworks.
- The software is dependent on the timely delivery of required hardware components.
- The software is dependent on the stability and reliability of the underlying operating system and hardware platform.

3. External Interface Requirements

3.1 User Interfaces

1. Dashboard: The dashboard module is designed to deliver a succinct yet informative overview of the user's fitness progress and goals. It showcases a comprehensive range of fitness-related data such as the number of steps taken, calories burned, and workout duration. Moreover, it enables users to establish fitness objectives and trace their progress over time, making it an invaluable tool for both beginners and experienced fitness enthusiasts.

2. Exercise tracker: The workout tracker module is an essential component of the app, intended to allow users to log and monitor their workout progress with accuracy and ease. Users can input a diverse range of data such as the type of exercise they performed, the duration, and the intensity level, which are then aggregated to provide detailed progress reports. Furthermore, the app provides suggestions for new workouts based on the user's preferences, ensuring that users stay motivated and challenged.

3. Exercise Library: The exercise library module aims to provide users with a comprehensive selection of exercises tailored to their fitness requirements. It includes instructional videos, images, and detailed descriptions of how to perform each exercise effectively. Using filters for various criteria such as body parts, equipment, and exercise type, users can easily customize their workout plans and achieve their fitness goals.

4. Personalized recommendations: The personalized recommendations module is a powerful feature that harnesses user data such as workout history, goals, and preferences to provide customized suggestions for workouts, exercises, and nutrition. The recommendations are based on several factors such as the user's fitness level, available equipment, and other pertinent considerations. By providing tailored suggestions, users can maximize their workout efficacy and achieve optimal fitness results.

3.2 Hardware Interfaces

Some of the supported devices are shown below:

1. Bluetooth-enabled equipment: Our fitness application possesses the capability to connect with Bluetooth-enabled equipment such as treadmills, stationary bikes, or weightlifting machines. This advanced feature enables the app to track the user's workout data in real-time, encompassing metrics such as the speed and distance of a run or the weight and reps of a weightlifting set.

2. Audio devices: Nutrifit leverages audio devices such as headphones or speakers, to provide optimal guidance and feedback during workouts.

3. Camera and sensors: Nutrifit will exploit the camera and sensors present on a user's device to track their movements and deliver feedback on their form.

4. Voice assistants: Our application will integrate with voice assistants such as Amazon Alexa or Google Assistant, to offer hands-free control and feedback during workouts. This innovative feature empowers the user to conveniently ask the voice assistant to initiate a workout, monitor their progress, or provide comprehensive guidance on how to execute a specific exercise with precision and accuracy.

3.3 Software Interfaces

Nutrifit will interface with the subsequent software components:
- **Database**: Nutrifit will use a MySQL database (version 8.0) to store user information, workout templates, and meal plans.
- **Operating System**: Nutrifit will be developed using mostly Java programming language and will be compatible with any operating system that supports Java.
- **Libraries**: Nutrifit will use the Spring Framework (version 6.0) for dependency injection and inversion of control.
- **Integrated Commercial Components**: Nutrifit will integrate with the Stripe payment gateway (version 2.8.13) to handle subscription payments.

3.3.1 Data Items and Messages
Nutrifit will receive these data items and messages:
- User profile information, including name, email address, and password
- User preferences for workout templates and meal plans
- The workout and meal plan templates, which will be stored in the database
- Payment information, including credit card details and subscription status

3.3.2 Services and Communications
Nutrifit will require the subsequent services:
- Database connectivity to store and retrieve user information, workout templates, and meal plans
- Payment gateway integration to handle subscription payments

Communication between Nutrifit and the database will be through JDBC (Java Database Connectivity). Communication between Nutrifit and the payment gateway will be through a RESTful API provided by Stripe.

3.3.3 Data Sharing
Data will be shared between the Nutrifit application and the MySQL database, with the database serving as the central repository for all user information, workout templates, and meal plans. Nutrifit will use JDBC to connect to the database and retrieve the required data.

3.3.4 Implementation Constraints
Nutrifit will require access to a MySQL database for proper functioning. The database schema and access credentials will be specified in a separate document. Nutrifit will also require internet connectivity for payment gateway integration and user authentication.

3.4 Communications Interfaces

Nutrifit requires several communications functions to enable users to interact with the application and to provide access to external systems. This section describes the requirements associated with these functions, including email, web browser, network server communications protocols, electronic forms, and other pertinent message formatting. It also identifies communication standards, security requirements, data transfer rates, and synchronization mechanisms.

3.4.1 Email

Nutrifit will require email functionality to enable users to reset their passwords, receive notifications, and communicate with support personnel. Email messages will conform to standard message formatting conventions, such as the Multipurpose Internet Mail Extensions (MIME) standard.

3.4.2 Web Browser

Nutrifit will provide a web-based interface that users can access using standard web browsers, such as Google Chrome, Mozilla Firefox, and Safari. The web interface will conform to standard web protocols, such as Hypertext Transfer Protocol (HTTP) and Hypertext Markup Language (HTML).

3.4.3 Network Server Communications Protocols

Nutrifit will communicate with external systems using standard network protocols, such as HTTP, File Transfer Protocol (FTP), and Secure Shell (SSH). Nutrifit will utilize secure protocols, such as HTTPS and SFTP, where appropriate. Nutrifit will also implement appropriate data transfer rates and synchronization mechanisms to ensure timely and accurate data transfer.

3.4.4 Electronic Forms

Nutrifit will require users to complete electronic forms to input data into the application. These forms will conform to standard formatting conventions, such as the Extensible Markup Language (XML) or JavaScript Object Notation (JSON) formats. Nutrifit will also implement appropriate data validation and error-checking mechanisms to ensure accurate and complete data input.

3.4.5 Security and Encryption

Nutrifit will implement appropriate security measures to protect user data and prevent unauthorized access. These measures will include data encryption, user authentication, access control, and secure communication protocols, such as HTTPS and SFTP. Nutrifit will

also comply with applicable data protection regulations, such as the General Data Protection Regulation (GDPR).

4. System Features

The system features section describes the major services provided by the Nutrifit application. These features are organized by functional hierarchy to ensure a logical presentation of the software capabilities that must be present to carry out the services provided by each feature.

4.1 Meal Planning Feature

The meal planning feature allows users to plan their meals for the week, providing them with a recommended daily calorie intake based on their fitness goals and food preferences.

4.1.1 Description and Priority
This feature is of high priority as it is one of the main functions of the Nutrifit application, catering to the needs of users who wish to follow a structured meal plan.

4.1.2 Stimulus/Response Sequences
- The user selects the meal planning feature from the main menu
- The user inputs their fitness goals and food preferences
- Nutrifit provides the user with a recommended daily calorie intake based on their inputs
- The user plans their meals for the week by selecting from a list of suggested meal options
- Nutrifit provides the user with a shopping list of ingredients required for the week's meals

4.1.3 Functional Requirements
- REQ-1: Nutrifit must be able to calculate a user's recommended daily calorie intake based on their fitness goals and food preferences
- REQ-2: Nutrifit must provide a list of suggested meal options for each day of the week based on the user's recommended daily calorie intake and food preferences
- REQ-3: Nutrifit must allow the user to select a meal option for each day of the week
- REQ-4: Nutrifit must provide a shopping list of ingredients required for the week's meals
- REQ-5: Nutrifit must allow the user to customize their meal plan by adding or removing meals and adjusting the portion sizes

- REQ-6: Nutrifit must provide the user with the option to save and access their meal plan for future reference

4.2 Exercise Tracking Feature

4.2.1 Description and Priority

This feature will allow users to track their exercise progress over time, including sets, reps, and weights lifted. It is of medium priority as it is a useful feature but not as critical as meal planning.

4.2.2 Stimulus/Response Sequences

1. The user selects the "Track Exercise" option from the main menu.
2. User selects the type of exercise they want to track, such as bench press or squats.
3. The user inputs the number of sets, reps, and weights lifted for each exercise.
4. The system records the data and displays a summary of the user's exercise progress.
   4.2.3 Functional Requirements:

- The system must allow the user to select the type of exercise they want to track.
- The user must be able to input the number of sets, reps, and weights lifted for each exercise.
- The system must record and store the user's exercise data.
- The system must display a summary of the user's exercise progress.

4.1.3 Functional Requirements

- REQ-1: The system must allow the user to select the type of exercise they want to track.
- REQ-2: The user must be able to input the number of sets, reps, and weights lifted for each exercise.
- REQ-3: The system must record and store the user's exercise data.
- REQ-4: The system must display a summary of the user's exercise progress.

4.3 Exercise Planning Feature

4.3.1 Description and Priority

The Workout Planning feature allows users to create and customize workout plans based on their fitness goals, preferences, and available equipment. This feature is of high priority as it is a core functionality of the Nutrifit application.

4.3.2 Stimulus/Response Sequences

To use this feature, the user will first select their fitness goals, workout preferences, and available equipment. Nutrifit will then provide a list of recommended workouts based on these preferences, which the user can customize according to their fitness level and schedule.

4.3.3 Functional Requirements

- REQ-1: Nutrifit shall allow users to select their fitness goals, workout preferences, and available equipment.
- REQ-2: Nutrifit shall provide a list of recommended workouts based on the user's fitness goals, workout preferences, and available equipment.
- REQ-3: Nutrifit shall allow users to customize the recommended workouts according to their fitness level and schedule.
- REQ-4: Nutrifit shall allow users to save and modify their workout plans.

4.4 Recipe Library Feature

4.4.1 Description and Priority

The recipe library feature provides users with a comprehensive library of healthy recipes for meals and snacks. This feature is of high priority as it is one of the core functionalities of the Nutrifit application and is essential for providing users with a seamless experience when creating meal plans and tracking their food intake.

4.4.2 Stimulus/Response Sequences

The user accesses the recipe library through the Nutrifit application interface. The system responds by displaying a list of recipes. The user selects a recipe of interest, and the system responds by displaying the recipe details, including ingredients, preparation instructions, nutritional information, and serving size.

4.4.3 Functional Requirements

- REQ-1: The system shall allow users to search for recipes by keyword, meal type, cuisine, ingredients, and nutritional information.
- REQ-2: The system shall allow users to filter search results based on dietary preferences, such as vegetarian, vegan, gluten-free, dairy-free, and low-carb.

- REQ-3: The system shall allow users to save their favorite recipes to their recipe book.
- REQ-4: The system shall provide users with nutritional information for each recipe, including calories, macronutrient content (protein, carbohydrates, and fat), and micronutrient content (vitamins and minerals).
- REQ-5: The system shall allow users to adjust the serving size of a recipe and automatically recalculate the ingredient quantities and nutritional information.
- REQ-6: The system shall display clear and concise preparation instructions for each recipe, including cooking time, temperature, and special techniques.
- REQ-7: The system shall allow users to rate and leave feedback on recipes, which will be visible to other users.
- REQ-8: The system shall provide users with the option to add ingredients to their shopping list, making it easy to purchase the necessary ingredients for a recipe.
- REQ-9: The system shall ensure that all recipes meet certain quality standards, such as being tested for taste, accuracy, and nutritional value.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. Speed: The swiftness and responsiveness of the fitness application are of paramount importance, as users anticipate immediate outcomes when interacting with the software.
2. Scalability: The software's scalability is a crucial aspect that should be taken into account during the developmental process. It should be capable of handling a multitude of users without any performance issues. As the number of users increases, the software must scale up effortlessly.
3. Reliability: The reliability of the fitness application is essential, and it must be available to users at all times. The software should be accessible 24/7 without any maintenance or downtime, ensuring a seamless user experience.
4. Security: The security of the fitness application is of utmost importance, and all user data and information must be protected from unauthorized access or breaches.
5. Compatibility: The fitness application must be compatible with different operating systems and devices, such as iOS, Android, and web browsers to ensure maximum accessibility to users.
6. Usability: The usability of the fitness application should be optimized to ensure ease of use and seamless navigation. The user interface must be simple and intuitive, allowing users to promptly locate the information they require.

7. Customization: The fitness application should allow users to personalize their experience by setting their own fitness goals, tracking their progress, and adjusting their routines accordingly, promoting adherence and engagement.

5.2 Safety Requirements

1. Security: Nutrifit must be designed with privacy and security in mind. User data should be encrypted and protected from unauthorized access, both during transit and at rest. The application should also be compliant with all relevant data privacy regulations and laws, such as GDPR.
2. Accuracy and Precision: The application should provide accurate and precise information to the user, including real-time feedback on their heart rate, calorie burn, and other fitness metrics. Inaccurate information can not only lead to injuries but also cause user distrust and dissatisfaction.
3. Health and Medical Considerations: The application should consider the user's health and medical history when providing workout recommendations and progress tracking. The app should be able to identify potential risks or limitations that may affect the user's ability to perform certain exercises safely.
4. Safety Guidelines: The application should provide clear and concise safety guidelines and warnings for each exercise, including any potential hazards or risks associated with the exercise. The app should also remind the user to consult with a medical professional before beginning any new exercise routine.
5. Emergency Preparedness: The application should have a clear and easy-to-use emergency response system in place. This includes providing emergency contacts, medical history, and medication information. The app should also have a feature to alert emergency services if necessary, such as if a user experiences a medical emergency or injury during a workout.

5.3 Security Requirements

1. Secure Authentication: The application should require users to authenticate themselves securely, preferably through multi-factor authentication. This will help prevent unauthorized access to user accounts.
2. Data Encryption: All user data and information should be encrypted, both in transit and at rest, to prevent unauthorized access and protect user privacy.
3. Secure Data Storage: User data and information should be stored securely in a database or server, with appropriate security controls in place to prevent data breaches and ensure data integrity.

4. Regular Updates and Patches: The application should be updated regularly to fix any security vulnerabilities or bugs that could be exploited by hackers.

5. Secure APIs: If the application uses APIs to integrate with third-party services or devices, the APIs should be secure and protected from unauthorized access or attacks.

6. Compliance with Privacy Regulations: The application should comply with all relevant privacy regulations, such as the General Data Protection Regulation (GDPR) in Europe or the California Consumer Privacy Act (CCPA) in the United States.

7. Penetration Testing: The application should be tested regularly through penetration testing, to identify any security vulnerabilities or weaknesses that could be exploited by hackers.

5.4 Software Quality Attributes

1. Functionality: The fitness application should function as intended, with all features and functionalities working correctly and as expected. The application should allow users to set goals, track progress, and provide feedback on workouts.

2. Usability: The application should be user-friendly and easy to use, with a clear and intuitive user interface. The application should be easy to navigate and use, with minimal user input required.

3. Reliability: The fitness application should be reliable, with minimal downtime or technical issues. Users should be able to rely on the application to provide accurate and up-to-date information on their workouts and progress.

4. Performance: The application should be fast and responsive, with minimal lag time or delays. The application should be able to handle a large number of users and provide fast results.

5. Security: The fitness application should be secure, with all user data and information protected from unauthorized access or breaches.

6. Maintainability: The application should be maintainable, with easy-to-understand code and a straightforward development process. The application should be easy to update and maintain as new features and functionalities are added.

7. Scalability: The application should be scalable, with the ability to handle a large number of users and provide fast results. The application should be able to scale up as the number of users increases, without compromising performance or functionality.

5.5 Business Rules

1. Membership: The fitness application should have clear rules around membership, including how users sign up, what information they need to provide, and what fees they may be required to pay.

2. Billing: The application should have clear billing rules, including how users are charged for using the app, what payment methods are accepted, and what happens if a user cancels their membership.

3. Exercise Restrictions: The application should have clear rules around workout restrictions, particularly for users with pre-existing medical conditions or injuries. The application should provide users with guidance on which exercises are safe and appropriate for their fitness level and medical history.

4. Use of Personal Information: The application should have clear rules around the use of personal information, including how user data is collected, stored, and used. The application should comply with all relevant privacy regulations and should be transparent about how user data is used.

5. Advertising and Sponsorship: The application should have clear rules around advertising and sponsorship, particularly if the application features sponsored content or advertising. The application should be transparent about sponsored content and should comply with all relevant advertising regulations.

6. User Conduct: The application should have clear rules around user conduct, including what behavior is acceptable and what is not. The application should have a code of conduct that outlines the expected behavior of users and what consequences may result from inappropriate behavior.

6. Other Requirements

Appendix A: Glossary

API: Application Programming Interface

Back end: The part of an application or system that handles data storage, processing, and management.

Data breach: A security incident in which sensitive or confidential data is accessed, disclosed, or stolen by an unauthorized individual.

Data encryption: The process of converting data into a code to prevent unauthorized access.

Front end: The part of an application or system that handles user interaction and presentation.

GDPR (General Data Protection Regulation): A regulation that governs data protection and privacy for all individuals within the European Union (EU) and the European Economic Area (EEA).

GUI: Graphical User Interface

HTTP: Hypertext Transfer Protocol

HTTPS: Hypertext Transfer Protocol Secure

JDBC (Java Database Connectivity): An API that allows Java applications to access databases, such as MySQL.

JSON: JavaScript Object Notation

Multipurpose Internet Mail Extensions (MIME): A standard for formatting email messages to allow for the exchange of different kinds of data, such as text, images, and audio.
RESTful API: A web service architecture that uses HTTP requests to access and manipulate data.

Spring Framework: A popular framework for building Java-based web applications.

SRS: Software Requirements Specification Document

Stripe payment gateway: An online payment processing platform that enables secure and reliable online transactions

UI: **User Interface**

UX: **User Experience**

Version control: **The management of changes to software code, including tracking, versioning, and collaboration.**

XML: **Extensible Markup Language**

Appendix B: Analysis Models

Appendix C: To Be Determined List