Qassim University جامعة القصيم

2024

CS383 Group Project

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

for AllerGuard app

version 1.0

Prepared by Group G2

Contents

1 Introduction 4-7
1.1 Purpose 4
1.2 Scope 4-5
1.3 References 5-6
1.4 Structure 6-7
2 System Overview 8-12
2.1 Product Perspective 8
2.2 Product Features 8-9
2.3 User Roles and Characteristics 9
2.4 Operating Enviromental 9-10
2.5 Assumptions and Dependencies 10
2.6 Requirements Validation. 10-12
3 Functional Requirements 13-18
3.1 Function Sign Up 13
3.2 Function Log In 13-14
3.3 Function User Profile Page 14-15
3.4 Function Scan 15-16
3.5 Function Help 16-17
3.6 Function Sign Out 17-18
4 Use Cases 19
4.1 Graphical use case 19
5 External Interface Requirements 20-22
5.1 User Interface 20
5.2. Hardware Interface 20
5.3. Software Interface 21
5.4. Communication Interface 21-22

Contents

6 Nonfunctional Requirements 23-24
6.1 Performance Requirements 23
6.2 Saftey Requirements 23
6.3 Security Requirements 23
6.4 Environmental Requirements 24
6.5 Maintenance Requirements. 24
7 Work section 25

1 Introduction

This document serves as a guide outlining the software requirements for our AllerGuard app, which focuses on detecting allergens. Its purpose is to provide a clear and comprehensive reference for everyone involved in the software's development and implementation.

1.1 Purpose

The primary aim of this Software Requirements Specification (SRS) document is to meticulously define and document both the functional and non-functional requirements for the AllerGuard app. It acts as a crucial reference point for various stakeholders, including the development team, project managers, quality assurance personnel, and other parties, ensuring a shared understanding of the software's goals, features, and constraints.

At its core, this app is designed to assist users in identifying potential allergens in products, empowering individuals with allergies to make informed choices regarding their purchases and consumption.

1.2 Scope

The scope of this SRS document is to define the boundaries and constraints of the AllerGuard app, which focuses on detecting allergens. The app provides a user-friendly interface for scanning product barcodes or searching for specific items. Users receive information about potential allergens in these products and can customize their allergen preferences.

The document covers essential requirements for allergen detection and user interface.

It serves as a reference for the development team, project managers, quality assurance personnel, and other stakeholders involved in the software's development and implementation.

The scope of this document includes:

- Definition of functional and non-functional requirements.
- Specification of primary functionalities like user login, user profile page, program exit, scanning, help, and sign-up.
- Identification of user roles and characteristics, focusing on individuals with food allergies or intolerances.
- Description of external interface requirements, including the user interface, hardware interface, software interface, and communication interface.
- Establishment of non-functional requirements like performance, safety, and security.
- Clear delineation of the boundaries and constraints of the AllerGuard app regarding allergen detection and user interface.

This scope ensures that the SRS document provides a comprehensive understanding of the software requirements, facilitating the development process and ensuring alignment among all project stakeholders.

1.3 References

To enhance the comprehension of this SRS document, it is advisable to refer to the following sources:

[1]Sommerville, Ian. Software Engineering. 9th ed., Addison-Wesley, 2011.
[2]"User Login Functional Requirements." Cal Poly, California Polytechnic State University,http://users.csc.calpoly.edu/~gfisher/classes/309/specs/FBRD/requirements/functionalRequirements/login.html

[3]"User Profiles." The Standard Interaction Design Process, https://thestandardinteractiondesignprocess.wordpress.com/process/understand-and-specify-context-of-use/user-profiles/
[4]"IBM i Exit Programs." IBM, https://www.ibm.com/docs/ja/i/7.3?topic=concepts-exit-programs
[5] "Use Case Diagram - Step by Step Tutorial with Example." YouTube, uploaded by Master2Teach, 23 March 2020, https://youtu.be/sQgoFjxSdxo?si=yt9gG15YNLTALuOs.
[6] "Introduction of Operating System - Set 1." GeeksforGeeks, GeeksforGeeks, www.geeksforgeeks.org/introduction-of-operating-system-set-1/
[7]"What Is HTTPS? | Cloudflare." Cloudflare, Cloudflare, www.cloudflare.com/learning/ssl/what-is-https/
[8]"Hardware Interface." Techopedia, Techopedia Inc., www.techopedia.com/definition/5968/hardware-interface
[9]“Non-Functional Requirements: Examples, Types, How to Approach.” AltexSoft, AltexSoft, 12 Feb. 2020, www.altexsoft.com/blog/non-functional-requirements/

1.4 Structure

The SRS document is organized into distinct chapters and sections, each serving a specific purpose and contributing to the overall understanding of the software requirements.

In the next chapter, the System Overview, we will describe the software system, its context, primary functionalities, roles and characteristics of the users involved, operating environment, Assumptions and Dependencies, as well as the Requirements Validation offering a comprehensive understanding of the software.

Following the System Overview, the Functional Requirements chapter will specify the precise actions, behaviors, and functionalities expected from the software system.

The Use Case chapter will outline interactions between users and the software, illustrating how the system accomplishes specific tasks and objectives.

The External Interface Requirements chapter will detail the interfaces connecting the software system with external entities, including users, other systems, hardware, and communication networks.

The Non-functional Requirements chapter will establish criteria for evaluating the software's performance, safety, security, and other essential qualities.

Finally, the Work Section will provide an overview of the work breakdown structure and the allocation of tasks within the project.

By structuring the document in this way, we ensure a logical flow of information and facilitate a comprehensive understanding of the software requirements, development process, and alignment among all project stakeholders.

In conclusion, this SRS document provides a comprehensive insight into the software requirements, encompassing the purpose, scope, and primary function of the Detection of Allergens app. It lays the foundation for the development process, ensuring a shared understanding and alignment among all project stakeholders.

2 System Overview

To save time while consuming food products in our daily life, the App "AllerGuard"is useful to help all those who have to follow a restricted diet due to food allergies or intolerance to certain types of food. People with food allergies spend a lot of time on shelves finding a product they can safely consume that shows them the ingredients.

2.1 Product Perspective

The AllerGuard app is designed to assist individuals with food allergies or intolerances in making informed choices about the products they consume. It provides a user-friendly interface that allows users to scan product barcodes or search for specific products. The app then provides information about potential allergens present in those products. It also offers alternatives to allergen-containing products, helping users identify suitable options for their dietary restrictions.

2.2 product features

The AllerGuard app offers several key features to enhance the user experience and provide valuable allergy-related information. These features include:

- Allergen Detection: The app enables users to scan product barcodes or search for specific products to detect potential allergens instead of reading the ingredients by themselves.
- Alternative Recommendations: If a product contains allergens, the app suggests alternative options that do not pose allergy risks.
- User-Friendly Interface: The app is designed to be easy to use, fast, and lightweight on devices, ensuring a seamless user experience.

- Hospital Locator: The app helps users identify the nearest hospital they can visit in case of an allergic reaction or emergency.

2.3 User roles and characteristics

The AllerGuard app caters to individuals with food allergies or intolerances, so it has two user roles:

1. Registered Users:
- Characteristics: Users with created AllerGuard accounts.
- Role: Full access to all features and functionalities.

2. Unregistered Users:
- Characteristics: Users without AllerGuard accounts.
- Role: Limited access to features and functionalities.

The AllerGuard app aims to provide a user- friendly experience for individuals with food allergies or intolerances. Registered users enjoy all features, including detailed allergen information and alternative suggestions. Unregistered users are encouraged to register to access additional features such as scanning product barcodes.

2.4 Operating Enviromental

The AllerGuard app is a mobile software designed to run on iOS and Android devices, such as smartphones and tablets. The user interface is developed using HTML and JavaScript. A stable internet connection is necessary for accessing allergen information and external communication. The app utilizes the device's camera as a barcode scanner, so the operating environment should support camera functionality. The GPS capabilities for accurate location information are required.

Compatibility efforts are made to ensure accessibility and usability across a wide range of devices and operating environments.
The AllerGuard app provides users with a seamless experience, enabling efficient detection and management of potential allergens in products.

2.5 Assumptions and Dependencies

The successful implementation and functioning of the AllerGuard app depend on certain assumptions and dependencies. These include:

- Availability of accurate and up-to-date product information.
- Users of the app have awareness of their allergies, including specific allergens, severity, and past reactions.
- Users actively engage with the app, inputting and updating their allergy information.
- The app must be synchronized with the user's location, date, and time for accurate information, with user permission.
- User-provided data accuracy and timeliness affect the effectiveness of the app.
- User feedback can impact the app's effectiveness.

2.6 Requirements Validation

The primary goal of the AllerGuard app is to identify and detect allergens in food products, empowering individuals with allergies to make informed choices. To ensure that the app is effective and feasible, it is important to consider the following validation criteria and activities:

Validation Criteria:

1. Validity:

- Align the requirements with the real-world need for allergen detection in food products.
- Verify that the proposed features effectively address the core problem of allergen detection.

2. Consistency:
   - Ensure that the requirements are consistent with each other, avoiding conflicting statements.
   - Verify that all requirements work together harmoniously.
3. Completeness:
   - Validate that the requirements cover all essential aspects of allergen detection.
   - Confirm that no critical functionalities are overlooked in the requirements.
4. Realism:
   - Assess the realism of the requirements.
   - Evaluate whether the app can realistically achieve the specified goals, considering technical and resource limitations.
5. Verifiability:
   - Confirm that each requirement is verifiable and can be tested or validated.

Validation Activities:

1. Interviewing Users:
   - Engage with individuals with allergies.
   - Gather feedback to ensure that the requirements align with user needs and expectations.

2. Requirements Reviews:
   - During the requirement review process, we thoroughly analyze existing documentation and research materials related to allergen detection.

This ensures that specific requirements and restrictions, such as allergen labeling regulations, are considered and incorporated into the development of the AllerGuard software, and will address any issues.

By adhering to these validation criteria and engaging in the activities, the AllerGuard app can ensure that its requirements are well-founded, practical, and capable of effectively addressing the critical need for allergen detection in food products.

3 Functional Requirements

3.1 Function Sign Up (FR001)

- Description: This function allows users to create an account and provide personal information, including allergy assessment questions, name, phone number, region, email, password, and relative number.
- Inputs: Allergy assessment responses, name, phone number, region, email, password, relative number.
- Source: User input.
- Outputs: User account creation confirmation.
- Destination: User account database.
- Action:
  1. User provides allergy assessment responses.
  2. User provides their name, phone number, region, email, password, and relative number.
  3. The app validates the provided information.
  4. The app creates a user account using the provided information.
  5. The app stores the user account information in the user account database.
  6. The app confirms the successful creation of the user account.
- Pre-condition: The user has launched the app and initiated the sign-up process.
- Post-condition: The user has successfully created an account with the provided information.
- Side effects: None.
- Priority: High.

3.2 Function Log In (FR002)

- Description: this function log in using the password, In order to use the program and interact with other users, you must log in using the password to verify the user. If you forget the password, the program offers you to re-enter a new password.
- Inputs: entered user password.
- Source: user input , user-password data.
- Outputs: logging the user in the app Successfully.
- Destination: User interface,user account database.
- Action:
  1. The program displays an icon for the user to enter the password.
  2. The user enters his own password, preferably distinct so that it is not similar to another password.
  3. The program verifies the password inputs and then the user enters the private page. With it and its data.
- Pre-condition: The user has launched the app and have valid password.
- Post-condition: The user logging in the app.
- Side effects: None.
- Priority: High.

3.3 Function User Profile Page (FR003)

- Description: modify user personal information

The program must allow the user to modify his personal information so that the program can give the user accurate results every time.

- Inputs: enter all user personal information.
- Source: User input, user's contact information.
- Outputs: the program can give the user accurate results.
- Destination: User interface, User account database.
- Action:

1. On the user's profile page, it asks you about your complete personal information.
2. This function allows you to change and update if you need.
3. It is stored in the database.

- Pre-condition: The user has launched the app have valid personal information.
- Post-condition: create a user profile page.
- Side effects: None.
- Priority: High.

3.4 Function Scan (FR004)

- Description: This function allows the user to access the camera functionality within the app and scan/capture the barcode or serial number of a product. It compares the scanned product's information with the user's entered allergen information and provides appropriate feedback and suggestions.
- Inputs: Scanned product information, user's entered allergen information.
- Source: User input, scanned product data.
- Outputs: Message indicating whether the product is safe to consume, message informing about allergens present in the product, alternative product suggestions.
- Destination: User interface.
- Action:
  1. User accesses the camera functionality within the app.
  2. User scans and captures the barcode or serial number of a product.
  3. The app compares the scanned product's information with the user's entered allergen information.
  4. If the product does not contain any allergens specified by the user, the app displays a message indicating that it is safe to consume.

5. If the product contains any allergens specified by the user, the app displays a message informing the user about the allergens present in the product.
6. The app provides alternative product suggestions that do not contain the user's specified allergens.

- Pre-condition: The user has launched the app and granted necessary permissions.
- Post-condition: The user receives feedback about the scanned product's allergen information and alternative product suggestions, if applicable.
- Side effects: None.
- Priority: High.

3.5 Function Help (FR005)

- Description: This function provides assistance and emergency support to the user. It includes features related to accessing location information, finding nearby hospitals, viewing contact details of hospitals, accessing contact numbers, and making emergency calls.
- Inputs: User's location information, selected hospital, user's contact numbers.
- Source: User input, hospital database or API, user's contact information.
- Outputs: List of nearby hospitals, hospital contact details, emergency call feature.
- Destination: User interface, phone call functionality.
- Action:

1. User grants permission for the app to access their location information.
2. The app has access to a database or API that contains information about nearby hospitals.
3. The app provides a feature to search for and display the nearest hospitals based on the user's current location.
4. The user is able to select a hospital from the displayed list.
5. The app retrieves and displays the contact details of the selected hospital.
6. The user grants permission for the app to access their contact numbers.
7. The app provides a feature to call the nearest relative/contact in case of an emergency.

- Pre-condition: The user has launched the app and granted necessary permissions.
- Post-condition: The user has access to nearby hospital information and can make emergency calls if needed.
- Side effects: None.
- Priority: Medium.

3.6 Function Sign Out (FR006)

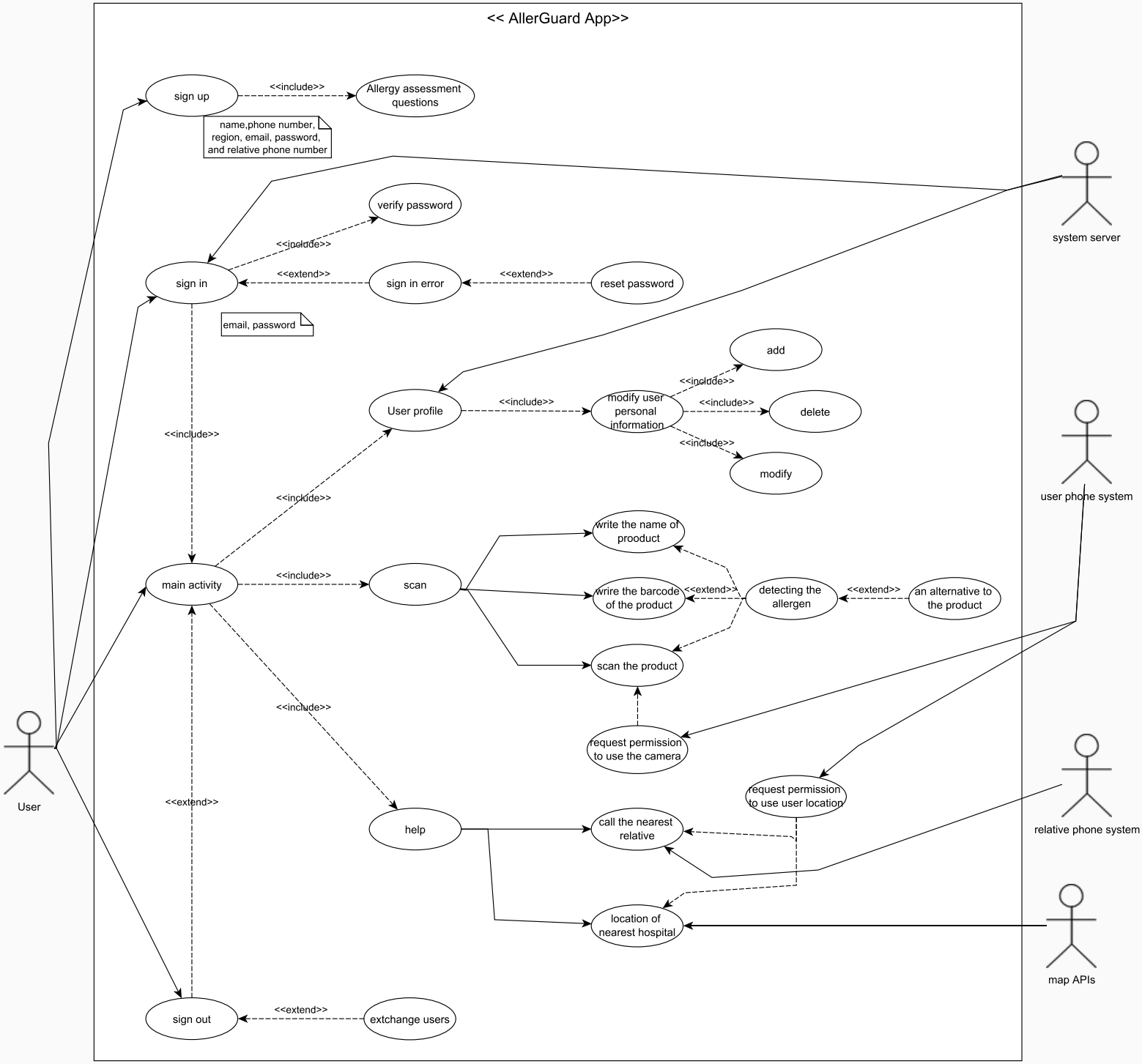
- Description: Allows the user to sign out of the app. By signing out, the user can switch between multiple accounts.
- Inputs: Press the sign out option.
- Source: User input.
- Outputs: The user is signed out of the program, enabling them to use multiple accounts.
- Destination: User interface.
- Action:

1. When the user finishes using the program and wishes to sign out, they click on the sign out option.
2. The user's session is terminated, and they are logged out of the system and the program.

- Pre-condition: The user has launched the app and completed the sign-out process.
- Post-condition: The user has successfully logged out of their account.
- Side effects: None.
- Priority: Medium.

4 Use Cases

4.1 Graphical use case



5 External Interface Requirements

5.1 User interface

- The software provides a seamless and user-friendly experience. This means that the interface is designed in a way that facilitates easy interaction and usability for the users.
- Users can easily input or search for product barcodes. This can be done through custom input fields or a search function.
- The interface quickly displays relevant details about the products. This includes information such as ingredients, nutritional value, and allergy warnings. The information is presented clearly and in an easily understandable manner.
- The user interface is designed with a focus on clarity, simplicity, and ease of use. This makes the interface visually appealing and features user-friendly controls and a logical design.

5.2 Hardware Interface

- The software is compatible with a variety of computers, mobile devices, and servers. This ensures that the software is accessible from a wide range of devices.
- The use of the software requires an internet connection. This is necessary as the software relies on online resources such as a product database or external APIs to provide accurate and up-to-date information.
- The software also relies on GPS capabilities to determine the user's location accurately. This is particularly important to identify the nearest hospital in case of emergencies related to allergies. Therefore, the hardware interface should include support for GPS capabilities to fulfill this requirement.

5.3 Software Interface

Operating Systems:

- The software is designed to support all major operating systems, including iOS and Android. This ensures that users can access the software regardless of the device they are using.

Database:

- The software utilizes the SQLite database management system. SQLite is a lightweight and widely used database that is efficient in storing and retrieving data. It seamlessly integrates with Flutter, a popular cross-platform mobile app development framework. This ensures the use of a robust and reliable database in the software.

5.4 Communication Interface

- The software integrates with a chatbot using HTML and JavaScript models to enable users to navigate and make inquiries within the app. Users can interact with the chatbot to seek assistance or get answers to specific questions.
- Users can send text messages to the support team through embedded HTML and JavaScript forms in the app. They can submit their inquiries or share their issues with the support team and receive appropriate assistance.
- Data transmission is secured using HTTPS, a secure protocol for data transmission over the internet.
- Email support is provided to users, allowing them to communicate with the customer service team by sending emails directly from the app using pre-packaged HTML forms. The communication is secured using HTTPS.

- Important updates are sent to users using JavaScript and the Web Push Notifications API. This ensures that users receive timely updates and notifications.
- Health checks and server authentication procedures are implemented to ensure data privacy and protection. These procedures help validate and sanitize user inputs, verify user identities, and prevent security vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). By implementing these measures, the software is protected from unauthorized access and ensures data integrity and safety.

6 Nonfunctional Requirements

6.1 Performance Requirements

Launch Time and Speed

- Description: The app's initial launch should take no longer than 3 seconds. Interactions within the app should be completed in less than 300 milliseconds.
- Rationale: Users expect rapid responses from the app, and a speedy experience is crucial for user satisfaction. Users often become frustrated when the app is slow to respond.

6.2 Safety Requirements

Accuracy

- Description: The scanning process should accurately identify the products and provide precise allergen information.
- Rationale: Accurate product identification and allergen information are crucial for user safety. Inaccurate data can pose health risks, especially for individuals with severe allergies.

6.3 Security Requirements

Data Authorization

- Description: The app should provide secure and reliable storage of user data, including allergen information, contact numbers, region, email, and passwords. Access to sensitive data should be strictly restricted to authorized users only, preventing unauthorized access and ensuring the confidentiality and security of user information.
- Rationale: Protecting user data is essential for building trust with users. Ensuring the information remains confidential and secure by limiting access to authorized users is a fundamental aspect of maintaining user confidence and trust in the app's security.

6.4 Environmental Requirements

Energy Efficiency

- Description: The software should be designed to minimize energy consumption. It should not consume more than 1% of the device battery per hour.
- Rationale: High consumption of energy can quickly deplete the device battery, which is undesirable for users. Minimizing energy consumption is environmentally responsible.

6.5 Maintenance Requirements

Update and maintain

- Description: The system's architecture should be designed to facilitate updates and maintenance activities. Bug fixes and enhancements should be done without interrupting or affecting the system's availability to users.
- Rationale: Ensuring ease of maintenance and updates is critical for minimizing disruption to users and keeping the system up to date with evolving requirements and technologies. Which Make the app more reliable and robust.

7 Work Section

Section	Student name
Introduction & Report design	Aliaa AlRishan
System Overview	Amani AlHarbi & Aliaa AlRishan
Functional Requirements	Rafeef AlSuhaibani & Sumayh AlHarbi
Use Cases	Raghad AlYaeesh
External Interface Requirements	Lama AlShmali
Non-functional Requirements	Raghad AlSuhaibani