Software Requirements Specification

for

Weather App

Version 1.0 approved

Prepared by:

i) Bilal Ahmed Khan (k200183)

ii) Abdullah Raza (k200184)

iii) Hameez Ahmed Siddiqui (k200242)

7th May 2023

Table of Contents

1. Introduction 1
1.1 Purpose 1
1.2 Document Conventions 1
1.3 Intended Audience and Reading Suggestions 1
1.4 Product Scope 1
1.5 References 2
2. Overall Description 2
2.1 Product Perspective 2
2.2 Product Functions 2
2.3 User Classes and Characteristics 3
2.4 Operating Environment 3
2.5 Design and Implementation Constraints 4
2.6 User Documentation 4
2.7 Assumptions and Dependencies 5
3. External Interface Requirements 6
3.1 User Interfaces 6
3.2 Hardware Interfaces 8
3.3 Communications Interfaces 8
4. System Features 8
4.1 Location-Based Weather Updates 8
4.2 Search for New Location to View Weather Updates 9
5. Other Nonfunctional Requirements 9
5.1 Performance Requirements 9
5.2 Safety Requirements 10
5.3 Usability 10
5.4 Compatibility: 10
5.5 Reliability: 10
5.6 Scalability: 10
5.7 Accessibility: 10
6. Other Requirements 10

Revision History

Name	Date	Reason For Changes	Version
Weather App	7th May 23	None	1.0.0

1. Introduction

1.1 Purpose

The purpose of the statement of a weather app that shows weather data to the user according to their location and also allows them to see weather updates of other places by entering the location is to provide users with accurate and up-to-date weather information that is relevant to their location or any other location they are interested in. The app aims to make it easy for users to access current weather conditions, temperature, humidity, wind speed, and other weather-related data by leveraging location-based services and real-time weather data.

By using the app, users can get a quick snapshot of the current weather conditions in their area, which can help them plan their day or week accordingly. They can also check the weather in other locations they are interested in, such as their travel destination or the location of their family or friends, by simply entering the location in the app. The app aims to be user-friendly, intuitive, and accessible to a broad range of users, regardless of their technical expertise.

1.2 Document Conventions

None

1.3 Intended Audience and Reading Suggestions

The intended audience for a weather app that shows weather data to the user according to their location and also allows them to see weather updates of other places by entering the location would be anyone who needs to access accurate weather information quickly and easily.

This could include individuals who are planning their day or week and need to know what the weather conditions will be like, travelers who want to check the weather conditions in their destination or along their route, outdoor enthusiasts who want to know the weather conditions for their favorite activities, and anyone who has an interest in weather-related information.

The app would be particularly useful for people who live in areas where weather conditions can change quickly, such as areas prone to severe weather events like hurricanes or tornadoes, or areas with extreme temperatures. It could also be beneficial for individuals who work in industries where weather conditions can impact their job, such as construction, agriculture, or transportation.

1.4 Product Scope

The scope of a weather app that shows weather data to the user according to their location and also allows them to see weather updates of other places by entering the location would be to provide accurate and up-to-date weather information to users in a user-friendly and intuitive way.

The app would need to have access to real-time weather data and location-based services to provide accurate information to users based on their current location. It would also need to have the ability to retrieve weather data for other locations based on user input.

In addition to providing current weather conditions, the app could include features such as hourly and daily weather forecasts, severe weather alerts, and historical weather data. The app could also offer customization options, such as the ability to set preferred units of measurement or to receive notifications for specific weather conditions.

The app's scope could also include integration with other apps or devices, such as smart home systems or navigation apps, to provide users with seamless weather-related information and functionality.

Overall, the scope of the app would be to provide users with a comprehensive and reliable source of weather-related information that is personalized to their location and preferences, and that is easily accessible and understandable.

1.5 References

None

2. Overall Description

2.1 Product Perspective

The cenral perspective under which a weather app that shows weather data to the user according to their location and also allows them to see weather updates of other places by entering the location has been developed is to provide a convenient and reliable way for users to access accurate weather information that is relevant to their needs.

The app has been developed with a user-centered perspective, with a focus on providing a seamless and intuitive user experience. The app's design and functionality have been developed with the goal of making it easy for users to access the information they need quickly and efficiently, without the need for extensive technical knowledge or expertise.

The app has also been developed with a data-driven perspective, with a focus on using real-time weather data and location-based services to provide users with accurate and up-to-date information. The app's features and functionality have been developed based on user feedback and input, as well as insights from weather-related data and trends.

2.2 Product Functions

Here are some of the product functions of a weather app that shows weather data to the user according to their location and also allows them to see weather updates of other places by entering the location:

1. **Current weather conditions:**
   The app would display the current weather conditions based on the user's location, including temperature, humidity, wind speed, and precipitation.

2. **Customization options:**
   The app would allow users to customize their experience by setting the preferred location and see the weather of a particular area.
3. **Location-based services:**
   The app would use location-based services to automatically update the weather information for the user's current location.

2.3 User Classes and Characteristics

The user classes for a weather app that shows weather data to the user according to their location and also allows them to see weather updates of other places by entering the location can be defined as follows:

1. **Casual users:**
   These are users who are interested in checking the weather conditions occasionally. They might use the app to get an idea of the weather forecast for their location for the next few hours or days. They may not require detailed weather information and may use the app for general purposes.
2. **Travelers:**
   These are users who are interested in getting weather information for multiple locations, including their travel destinations. They may use the app to plan their travel based on weather conditions or to be prepared for any weather changes during their travel.
3. **Outdoor enthusiasts:**
   These are users who participate in outdoor activities such as hiking, camping, fishing, or skiing. They may use the app to plan their activities based on weather conditions and to stay safe during their outdoor adventures.
4. **Professionals:**
   These are users who require accurate weather information for their work, such as construction workers, farmers, or pilots. They may use the app to plan their work based on weather conditions and to ensure their safety while working in different weather conditions.

Characteristics related to this app for these user classes are:

1. **User-friendly interface:**
   The app should have a simple and easy-to-use interface to allow casual users to check the weather information quickly and easily.
2. **Accurate and detailed information:**
   The app should provide accurate and detailed weather information for travelers and outdoor enthusiasts, who require more specific weather data for their planning.
3. **Customizable settings:**
   The app should have customizable settings to allow users to enable or disable notifications, and choose the locations they want to monitor.
4. **Real-time updates:**
   The app should provide real-time weather updates for professionals who require accurate and up-to-date weather information for their work.

2.4 Operating Environment

The app has been developed used the Flutter framework that uses dart language.
It can on both:

i. Andriod
ii. iOS

platforms natively.

2.5 Design and Implementation Constraints

Following design and implementation constraints were encountered in the creation of this app

1. **Data accuracy:**
   One of the most critical constraints for this app is the accuracy of the weather data. The app needs to rely on accurate and reliable sources of weather data to provide precise and up-to-date information to its users.
2. **Location-based services:**
   The app needs to use location-based services to identify the user's location accurately. This requires the app to have access to the user's device location, which may be subject to various limitations and restrictions.
3. **Network connectivity:**
   The app needs to have access to a reliable network connection to fetch and update weather information from its sources. This may be challenging in areas with poor network coverage or limited bandwidth.
4. **User interface:**
   The app needs to have an intuitive and user-friendly interface that allows users to access and interact with the weather information easily. This requires careful consideration of the app's layout, typography, color scheme, and other design elements.
5. **Compatibility:**
   The app needs to be compatible with a wide range of devices, operating systems, and screen sizes to ensure maximum reach and usability.
6. **Security:**
   The app needs to ensure the security and privacy of its users' data, including their device location and other sensitive information.
7. **Performance:**
   The app needs to be designed and implemented to provide fast and responsive performance, even under heavy load or poor network conditions.
8. **Compliance:**
   The app needs to comply with various legal and regulatory requirements, such as data protection laws and regulations.
9. **Cost:**
   The app needs to be developed within a reasonable budget, considering the costs of development, hosting, and maintenance.

2.6 User Documentation

A brief guide on how to use the app to get weather information for your location and other places:

1. Download and install the app from your device's app store.

2. Upon launching the app, you'll be asked to allow the app to access your device's location. You can choose to allow or deny the request.

3. Once you've granted the app permission to access your location, it will automatically detect your current location and display the weather conditions for that location on the home screen.

4. To view weather information for other locations, tap on the search bar at the top of the screen and enter the name of the location you want to view. You can also search for a location by entering its zip code or latitude and longitude coordinates.

5. Once you've entered the location, the app will display the current weather conditions and a forecast for the next few days.

6. You can also customize the app's settings to enable or disable notifications, and choose the locations you want to monitor.

2.7 Assumptions and Dependencies

Assumptions:

Following assumptions were made about the users of this app throughout the development process:

1. Users of this app are likely to be interested in weather conditions and updates, either for personal or professional reasons.
2. Users may have varying levels of knowledge and experience with weather-related terminology and concepts.
3. Users may come from different backgrounds and age groups, but are likely to have access to a smartphone or other mobile devices.
4. Users may use this app for planning their daily activities or outdoor events, and may require accurate and up-to-date weather information.
5. Users may have varying levels of technical proficiency, ranging from casual to advanced users.
6. Users may have different preferences regarding the user interface and design of the app.
7. Users may be located in different parts of the world and have different weather-related needs and concerns.

These assumptions helped us guide the design and development of the app to ensure that it meets the needs and expectations of its users

Dependencies:

The weather app has the following dependencies:

```
dependencies:
    flutter:
     sdk: flutter

  cupertino_icons: ^0.1.2
  geolocator: ^7.0.0
```

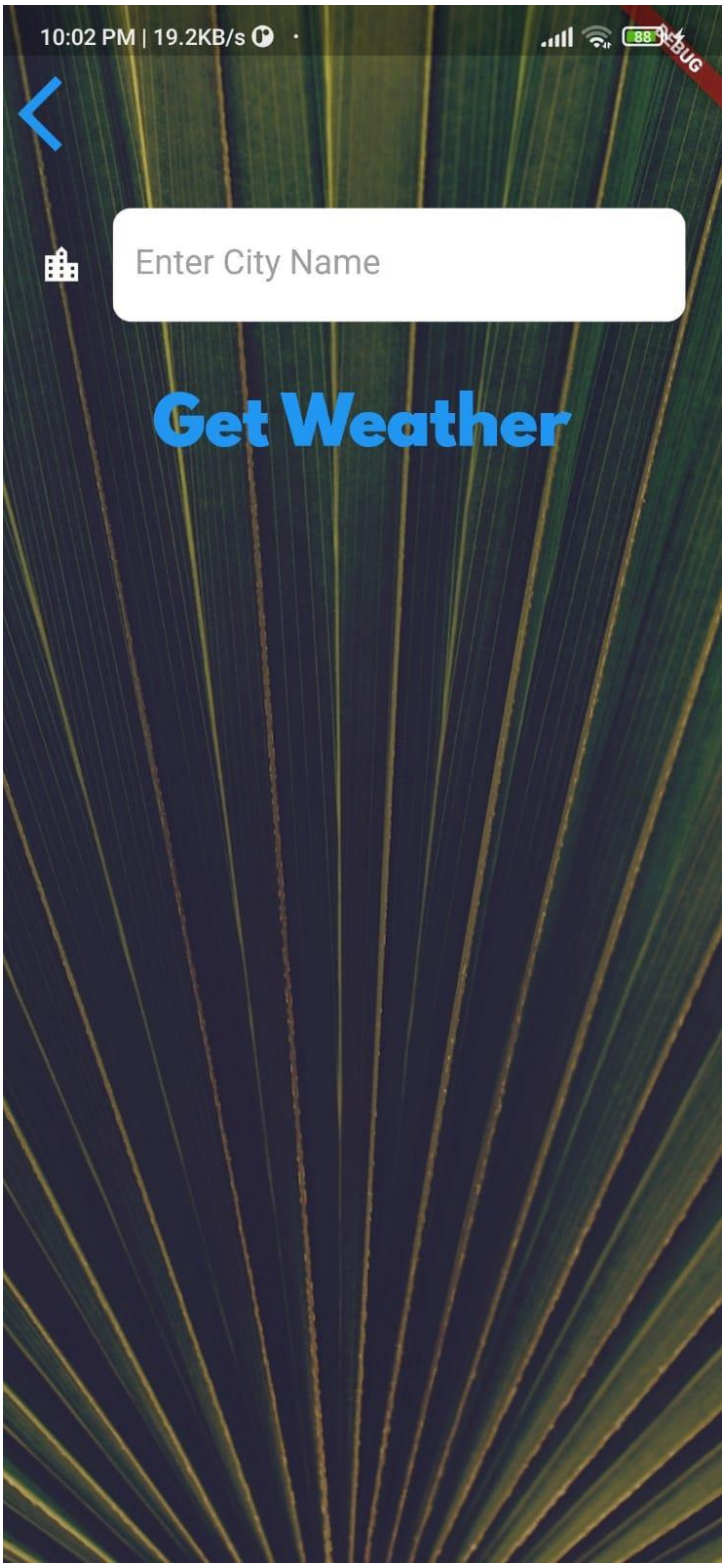
```
  http: ^0.13.4 permission_handler:
  ^8.2.5
  flutter_spinkit: ^5.1.0

dev_dependencies:
    flutter_test:
    sdk: flutter
```

3. External Interface Requirements

3.1 User Interfaces

1. Weather Screen

2. Search Screen



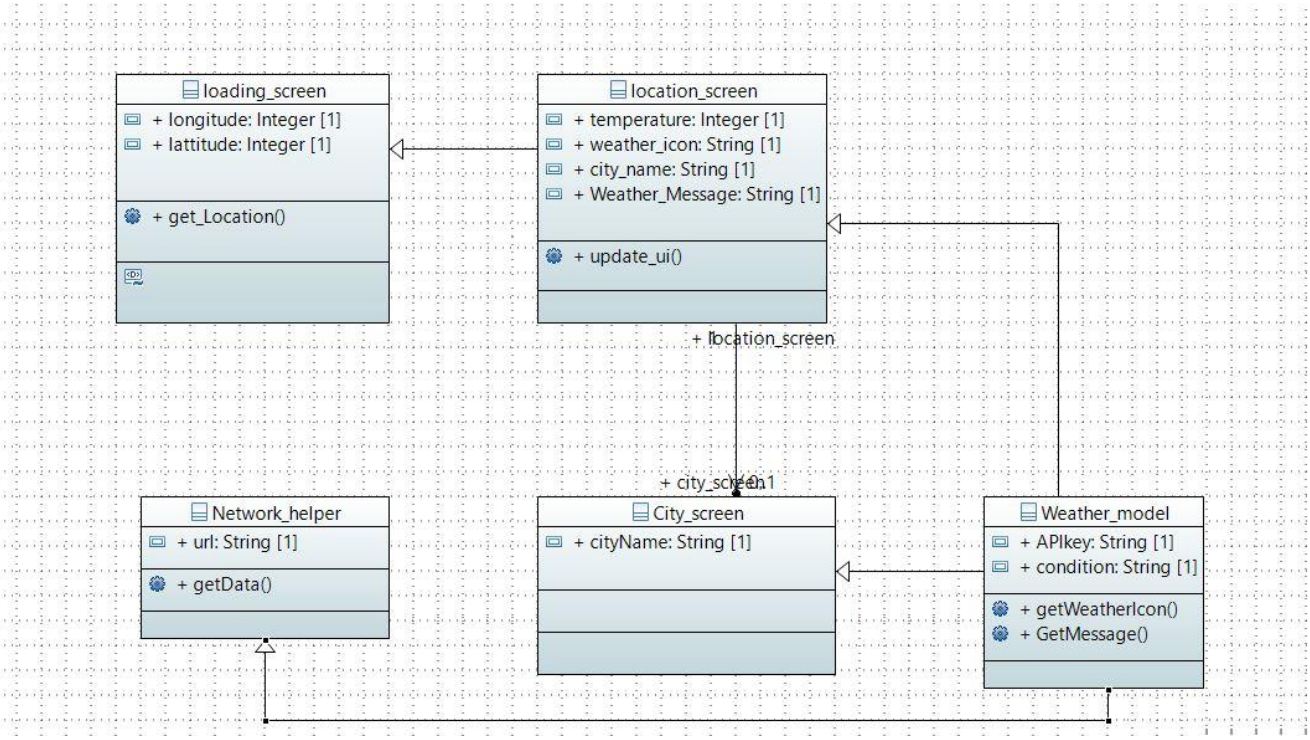
3.2 Hardware Interfaces

None

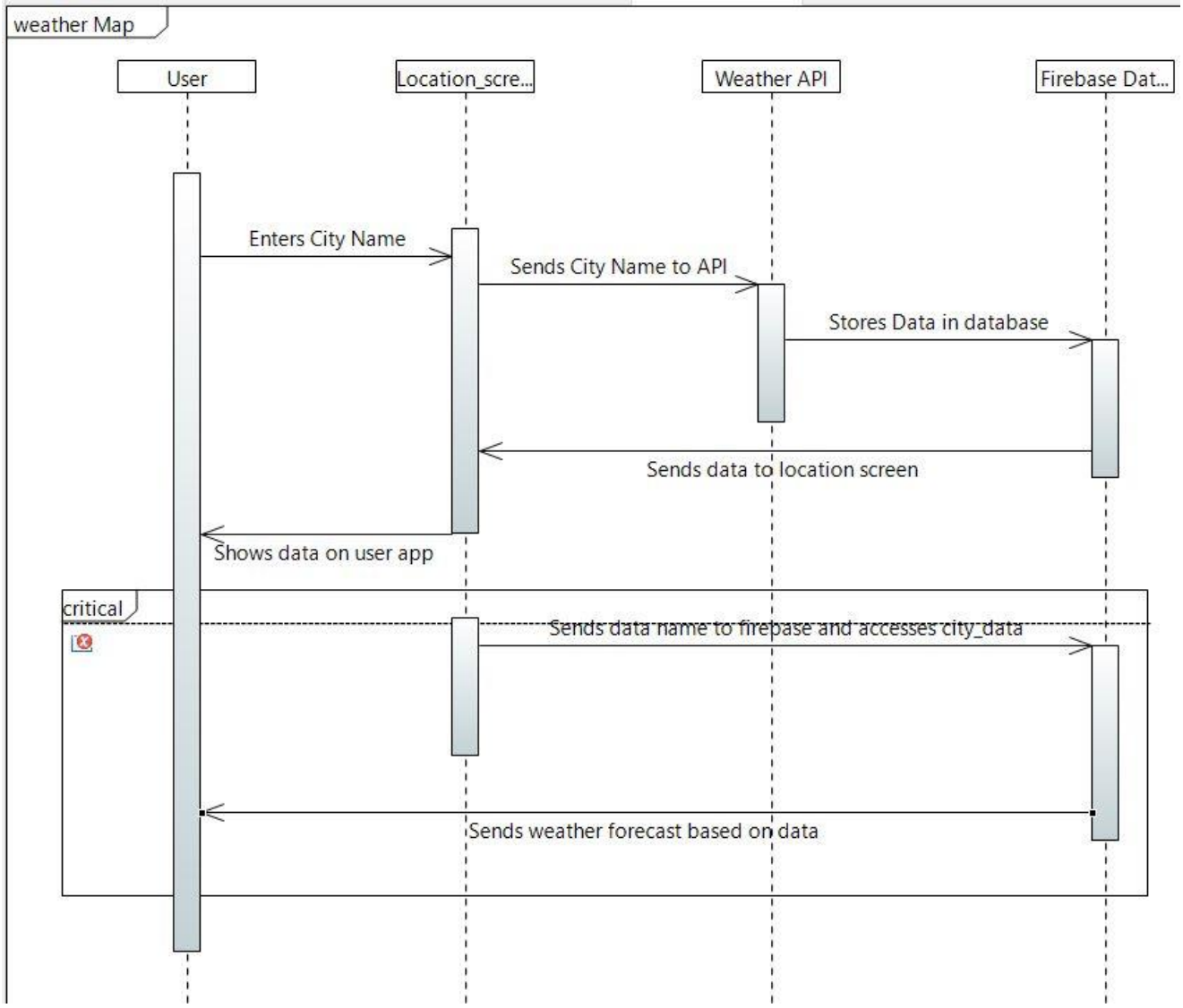
3.3 Communications Interfaces

The weather app uses a range of communication interfaces to interact with external services and provide weather information to its users. For instance, when the user opens the app, it sends an HTTP request to the server API to fetch the weather data for their current location. The app also utilizes the device's built-in location services to obtain the user's latitude and longitude coordinates, which are then passed as parameters in the API call. The server then responds with a JSON object that contains various weather-related parameters, such as temperature, humidity, wind speed, and more. The app then processes this data and displays it on the screen in a user-friendly format.
In summary, the weather app uses a combination of networking protocols, APIs, and server-side technologies to provide accurate and up-to-date weather information to its users while leveraging the latest technologies to enhance the overall user experience.

3.4 Class Diagram



3.5 Sequence Diagram



4. System Features

The weather app is designed to provide users with real-time weather information and updates for their current location and other selected locations. The app offers a range of features and functionalities that allow users to stay informed about the weather conditions in their area and plan their day accordingly. Some of the key system features of the app include location-based weather updates, customizable preferences, interactive weather maps, and push notifications for severe weather alerts. With these features, users can easily access the latest weather information and make informed decisions based on their weather-related needs and preferences. In the following sections, we will delve deeper into these system features and explore how they enhance the overall functionality and usability of the app.

4.1 Location-Based Weather Updates

4.1.1 Description and Priority

This feature enables the app to provide weather updates based on the user's current location. It is a critical system feature that prioritizes accuracy and reliability in delivering real-time weather information to users.

4.1.2 Stimulus/Response Sequences

When the user opens the app, the app sends an HTTP request to the server API to fetch the weather data for their current location. The app also utilizes the device's built-in location services to obtain the user's latitude and longitude coordinates, which are then passed as parameters in the API call. The server then responds with a JSON object that contains various weather-related parameters, such as temperature, humidity, wind speed, and more. The app then processes this data and displays it on the screen in a user-friendly format.

4.1.3 Functional Requirements

- Integration with device location services to obtain user's current location
- API call to external weather data provider
- Processing of weather data to display on screen
- User-friendly interface for displaying weather information

4.2 Search for New Location to View Weather Updates

4.2.1 Description and Priority

This feature allows users to search for weather information for a new location other than their current location. It is a critical system feature that prioritizes flexibility and user choice in accessing weather data.

4.2.2 Stimulus/Response Sequences

When the user selects the option to search for a new location, the app prompts them to enter the name or address of the desired location. The app then sends an HTTP request to the server API to fetch the weather data for the specified location. The server then responds with a JSON object that contains various weather-related parameters, such as temperature, humidity, wind speed, and more. The app then processes this data and displays it on the screen in a user-friendly format.

4.2.3 Functional Requirements

- Search function that allows users to enter the name or address of a desired location
- Integration with external location-based services to retrieve latitude and longitude coordinates for the specified location
- API call to external weather data provider
- Processing of weather data to display on screen
- User-friendly interface for displaying weather information for the selected location

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The app should be responsive and provide fast loading times to ensure a smooth user experience. This includes quick access to weather data, smooth transitions between screens, and minimal app crashes or errors.

5.2 Safety Requirements

The app should ensure the security of user data and maintain the confidentiality of sensitive information such as location data. This includes implementing secure user authentication methods and encrypting user data in transit and at rest.

5.3 Usability

The app should be designed with user-friendly interfaces, clear and concise labeling of functions and buttons, and an easy-to-navigate layout. This includes providing appropriate font sizes, color schemes, and visual cues to aid users in navigating the app.

5.4 Compatibility:

The app should be compatible with a variety of mobile devices and operating systems, ensuring that users can access the app on their preferred device.

5.5 Reliability:

The app should be designed to minimize the risk of errors, system crashes, or data loss. This includes implementing regular backups of user data and system configurations, and providing error messages and notifications to users when needed.

5.6 Scalability:

The app should be designed to handle large volumes of user requests and data, ensuring that it can scale to accommodate a growing user base and increased usage over time.

5.7 Accessibility:

The app should be designed with accessibility in mind, ensuring that it can be used by users with disabilities or impairments. This includes providing support for assistive technologies such as screen readers and voice assistants, and adhering to accessibility standards such as WCAG 2.1.

6. Other Requirements

None