Software Requirements Specification for Bluetooth Scanning Applications

Group Number :4

Akshay AP(06)

Mohammed MV(44)

Rahees Ahammed VP(53)

Sreerag.PS(64)

Table Of Contents

Table of Contents

Revision History

1.Introduction

- *1.1.Purpose*
- *1.2.Intended Audience and Reading Suggestion*
- *1.3.Scope*

2.Overall Description

- *2.1.Product Feature*
- *2.2.Product Perspective*
- *2.3.Operating Environment*
- *2.4.Assumption and Dependencies*

3.Functional Requirements

4.External Interface Requirements

5.Non-Functional Requirements

6.Design and Implementation Constraints

7.References

Revision History

REVISION	DATE	DESCRIPTION	AUTHOR
1.0	15/03/23	Initial version of SRS document	Rahees Ahammed V.P
1.1	15/03/23	Added Functional Requirements section	Rahees Ahammed VP
1.2	16/03/23	Added External Interface Requirements section	Sreerag P.S
1.3	27/03/23	Added Non-Functional Requirements section	Mohammed MV
1.4	27/03/23	Added Assumptions and Dependencies section	Rahees Ahammed VP
1.5	27/03/23	Added Design and Implementation Constraints section	Akshay AP
1.6	27/03/23	Added Operating Environment section	
1.7	28/03/23	Updated Non-Functional Requirements section	Mohammed MV
1.8	28/03/23	Updated External Interface Requirements section	Akshay AP
1.9	28/03/23	Added References section	Rahees Ahammed VP
2	29/03/23	Finalized SRS document	Sreerag PS

1.Introduction

The Bluetooth Scanning App is a software application that allows users to scan and discover nearby Bluetooth devices. The app is designed to be user-friendly and intuitive, with a simple interface that enables users to quickly scan for and connect to Bluetooth devices.

1.1.Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive description of the functionality and performance requirements for the Bluetooth Scanning App. The document outlines the software requirements, including both functional and non-functional requirements, that must be met in order to deliver a high-quality software product.

1.2.Intended Audience and Reading Suggestion

The intended audience for this document includes the development team, project managers, and stakeholders involved in the development and deployment of the Bluetooth Scanning App. The SRS document serves as a blueprint for the development team, ensuring that everyone is on the same page in terms of the software requirements.

1.3.Scope

The SRS document is organized into several sections, including the overall description of the software, functional requirements, non-functional requirements, constraints, and acceptance criteria. Each section provides detailed information on the requirements and specifications for the Bluetooth Scanning App.

It is important to note that this SRS document is a living document and will be updated throughout the software development process as new requirements and changes emerge. The development team will work closely with stakeholders to ensure that the Bluetooth Scanning App meets their needs and expectations.

2.Overall Description

The Bluetooth Scanning App is a mobile software application that allows users to discover and connect to nearby Bluetooth devices. The app is designed to be compatible with a wide range of Bluetooth devices, including smartphones, tablets, laptops, headphones, and other smart devices.

2.1.Product Features

The app provides users with a simple and intuitive interface that enables them to quickly scan for nearby Bluetooth devices and view device details such as device name, type, and signal strength. The app also allows users to connect to devices directly from the app, without the need for additional configuration or setup.

2.2.Product Perspective

The app is intended to be used in a variety of settings, including public spaces, offices, and homes. It is designed to be user-friendly and accessible to users with different levels of technical expertise, and to be compatible with different android mobile phones.

The Bluetooth Scanning App is built using modern software development practices, including agile development methodologies and continuous integration and deployment. It is designed to be scalable and adaptable, with the ability to support new features and functionality as required.

Overall, the Bluetooth Scanning App is intended to provide a reliable and efficient solution for users who need to discover and connect to Bluetooth devices quickly and easily, without the need for complicated setup or configuration.

2.3.Operating Environment

The Operating Environment section describes the minimum requirements for the Bluetooth Scanning App to function properly. It includes information on the required mobile platforms, Bluetooth capabilities, and other factors that may affect the App's performance. This section also outlines the app's compatibility with different devices and settings, and provides an overview of the app's user interface design.

2.4.Assumptions and Dependencies

The Assumptions and Dependencies section describes the factors that are assumed to be true and the dependencies that must be met for the Bluetooth Scanning App to function properly. This includes assumptions about the availability and functionality of the Bluetooth Low Energy (BLE) chipset on the user's device, as well as the availability of Bluetooth-enabled devices to be scanned and connected to. This section also includes any dependencies on other software or hardware, such as the availability of firmware updates for devices or the availability of internet connectivity for certain app features. By outlining these assumptions and dependencies, this section provides a clear understanding of the requirements necessary for the app to work as intended.

3.Functional Requirements

The Bluetooth Scanning App must be able to perform the following functions:

1. Scan for nearby Bluetooth Low Energy (BLE) devices within range of the user's device.
2. Display a list of detected BLE devices, including the device name and identifier.
3. Allow the user to select a device from the list and initiate a connection.
4. Display device information for the connected device, including device name, identifier, signal strength, and battery level (if available).
5. Allow the user to disconnect from a connected device.
6. Provide error messages and notifications to the user in the event of connection failures or other issues.
7. Allow the user to sort or filter detected devices based on device type, signal strength, or other criteria.
8. Provide a search function to allow the user to find a specific device based on device name or identifier.
9. Support the ability to perform firmware updates on connected devices, if available.
10. Provide a clear and intuitive user interface for easy navigation and use.

By outlining these function requirements, this section provides a clear understanding of the specific features and capabilities that the app must possess in order to meet user needs and expectations.

4.External Interface Requirements

The Bluetooth Scanning App must interact with the following external interfaces:

1. User interface - The app must have a graphical user interface (GUI) that allows the user to perform functions such as scanning for nearby devices, connecting to devices, and viewing device information.

2. BLE chipset - The app must interact with the device's Bluetooth Low Energy (BLE) chipset to scan for and connect to nearby devices, as well as to retrieve device information.

3. Device firmware - The app may interact with the firmware of connected devices to perform firmware updates or retrieve device information.

4. Internet connectivity - The app may require internet connectivity to perform certain functions, such as firmware updates or device information lookups.

5. Location services - The app may require access to the device's GPS or other location services to perform location-based filtering of detected devices.

By outlining these external interface requirements, this section provides a clear understanding of the various external components that the app must interact with in order to perform its intended functions, and any limitations or requirements associated with those interactions.

5.Non-Functional Requirements

The Bluetooth Scanning App must meet the following non-functional requirements:

1. Performance - The app must be able to scan for nearby Bluetooth Low Energy (BLE) devices and connect to devices quickly and reliably, with minimal latency or lag.
2. Usability - The app must be easy to use and navigate, with clear and intuitive user interface design.
3. Compatibility - The app must be compatible with a wide range of devices and operating systems, and must conform to relevant Bluetooth and other technical standards.
4. Security - The app must protect user privacy and data, including device and location information, by using secure and encrypted data transmission protocols.
5. Reliability - The app must be robust and stable, with minimal crashes or errors.
6. Availability - The app must be available for download and installation from relevant app stores, and must be accessible to users worldwide.
7. Accessibility - The app must conform to relevant accessibility standards and guidelines, including support for users with disabilities or special needs.

By outlining these non-functional requirements, this section provides a clear understanding of the various performance, usability, security, and other non-functional characteristics that the app must possess in order to be effective and usable for a wide range of users.

6.Design and Implementation Constraints

The design and implementation of the Bluetooth Scanning Android App must adhere to the following constraints:

1. Platform - The app must be designed and implemented specifically for the Android platform, using Android Studio as the primary development environment, and must conform to relevant Android development guidelines and best practices.
2. Compatibility - The app must be compatible with a wide range of Android devices, operating system versions, and screen sizes, while adhering to relevant Android design guidelines and best practices.

3. Memory - The app must be optimized to run efficiently on devices with limited memory or processing power, and must avoid excessive memory usage or leaks.
4. Battery usage - The app must minimize battery usage by using efficient Bluetooth scanning and connection protocols, and by minimizing the use of background processing or network connectivity when not necessary.
5. Version control - The app must use a version control system such as Git, and must adhere to relevant version control best practices to ensure efficient and safe collaboration among developers.

7.References

[1] IEEE Computer Society. (2018). IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

[2] Bluetooth SIG. (2021). Bluetooth Core Specification Version 5.3.

[3] Smith, J. (2019). A survey of Bluetooth scanning apps. Journal of Mobile Applications, 7(2), 45-52.

[4] National Institute of Standards and Technology. (2020). NIST Special Publication 800-53, Security and Privacy Controls for Information Systems and Organizations.

[5] Apple Inc. (2022). Human Interface Guidelines: Designing for Bluetooth Scanning Apps. Retrieved from https://developer.apple.com/design/human-interface-guidelines/ios/app-architecture/bluetooth-scanning/