Software Requirements Specification

for

CareConnect

Prepared by Group B6

Table of Contents

Table of Contents ii
Revision History ii
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
2. Overall Description 2
2.1 Product Perspective 2
2.2 Product Functions 2
2.3 User Classes and Characteristics 2
2.4 Operating Environment 2
2.5 Design and Implementation Constraints 2
2.6 Assumptions and Dependencies 3
3. External Interface Requirements 3
3.1 User Interfaces 3
3.2 Hardware Interfaces 3
3.3 Software Interfaces 3
3.4 Communications Interfaces 3
4. System Features 4
4.1 Functional Requirements 4
5. Other Nonfunctional Requirements 5
5.1 Performance Requirements 4
5.2 Security Requirements 5
5.3 Software Quality Attributes 5
Appendix A: Team members 5

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This software requirement specification provides a clear, documented model of the requirements for the CareConnect - Monitoring Isolated Individuals and Detecting Emergencies. The primary goal of the application is to utilize phone activity data for monitoring and safeguarding the well-being of users. It will achieve this by continuously analyzing the user's phone activity and promptly alerting designated contacts in instances of prolonged inactivity. This proactive approach is designed to offer reassurance to both users and their contacts, fostering a greater sense of safety and connectedness within the community.

1.2 Document Conventions

Text: Cambria, 11pt font size, single line spacing.
Lists: Bulleted lists using bullet points (•).
Tables: Numbered with table title above and centered within the table.

1.3 Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) document is intended for developers, project stakeholders, quality assurance team members, technical writers, and future maintainers involved in the development and maintenance of the mobile application. Readers should review the objectives and use cases to understand the app's purpose and functionality.

1.4 Product Scope

The software specified herein is a mobile application aimed at enhancing user well-being and safety through phone activity monitoring. The app's core objective is to analyze users' phone activity continuously, emphasizing confirmation of their presence and status. It intelligently alerts designated contacts in cases of prolonged inactivity, aligning with goals of promoting user safety and connectivity within communities.

2. Overall Description

2.1 Product Perspective

The mobile application described in this SRS is a standalone product, not associated with any existing systems. It functions independently to monitor user well-being through phone activity analysis. While it may integrate with other applications or services through APIs for certain functionalities, its core purpose remains self-contained, focusing on enhancing user safety and connectivity.

2.2 Product Functions

- Continuously monitor user's phone activity.
- Analyze phone activity to confirm user presence and status.
- Alert designated contacts in case of prolonged user inactivity.
- Enable location sharing via message.
- Allow caretaker to ping the beneficiary as needed.
- Customize time intervals between scheduled check-ins.
- Customize rest times for the beneficiary.
- Send battery low alerts to the caretaker.
- Allow caretaker to ping a member of the community to ensure beneficiary's well-being.
- Provide separate logins for caretaker and beneficiary.
- Offer a pill reminder feature.
- Implement fall detection functionality.
- Enable customization of the message sent to the caretaker.

2.3 User Classes and Characteristics

Caretaker: Caretakers actively monitor the beneficiary's well-being through the application, receiving alerts and communicating when necessary. They may vary in technical proficiency but share the common goal of ensuring the beneficiary's safety.

Beneficiary: Beneficiaries rely on the application for passive monitoring, contributing essential data for caretakers to assess their well-being. Their primary role is to benefit from the application's monitoring capabilities, fostering a sense of security knowing caretakers are informed and ready to assist.

2.4 Operating Environment

The Android application will operate within the Android mobile operating system environment, targeting Android version 14, code named "UpsideDownCake," and subsequent versions. It will be optimized to run efficiently on smartphones running Android 14, ensuring seamless performance and compatibility. The application will leverage standard hardware components found in modern smartphones, including processors, memory, storage, and network connectivity features available on Android 14-compatible devices. Compatibility with other software components and applications commonly found on Android devices will be maintained to ensure a harmonious user experience within the Android ecosystem.

2.5 Design and Implementation Constraints

- **Regulatory Compliance:** The application must adhere to relevant regulatory policies and data protection laws, ensuring user privacy and data security.
- **Hardware Limitations:** The application should be optimized to operate efficiently within the hardware constraints of mobile devices running Android 14 and above, considering factors such as memory usage and processing power.
- **Technological Compatibility:** The application must integrate seamlessly with Android 14 and above, utilizing compatible technologies, APIs, and development frameworks.
- **Security Considerations:** Robust security measures must be implemented to safeguard user data and prevent unauthorized access to sensitive information.
- **User Experience Design:** The application should follow established design conventions and user experience guidelines for Android platforms to ensure intuitive usability and accessibility.
- **Interoperability:** The application may need to interface with external services or APIs for functionalities such as location services, messaging, and notifications.
- **Maintenance and Updates:** Considerations should be made for ongoing maintenance, updates, and support, including version compatibility and bug fixes.

2.6 Assumptions and Dependencies

Assumptions:

1. **Stable Android 14 Environment:** The development assumes a stable and consistent Android 14 operating environment, including reliable APIs and frameworks for application development.
2. **Availability of Necessary Permissions:** The application assumes access to necessary permissions on users' devices, such as location, messaging, and phone activity monitoring permissions.
3. **Data Connectivity:** The application assumes consistent data connectivity to enable real-time monitoring and communication features.
4. **Compliance with Regulatory Standards:** Assumption is made that the application development and deployment will comply with relevant regulatory standards and data protection laws.

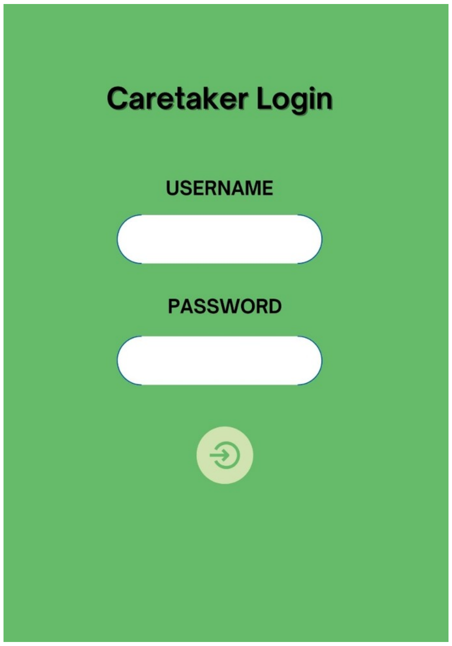
Dependencies:

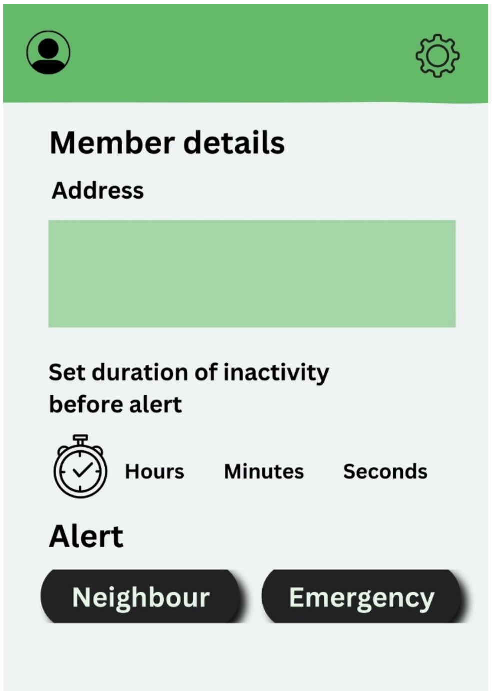
1. **Third-Party Libraries and APIs:** The project may depend on third-party libraries and APIs for functionalities such as location services, messaging, and notifications.
2. **Integration with External Services:** Dependencies exist on external services for features like location sharing and message delivery.
3. **Hardware Compatibility:** The application's functionality is dependent on the compatibility of the device hardware with Android 14 and its capabilities.
4. **Timely Updates and Support:** Dependence on timely updates and support from third-party service providers for compatibility with evolving Android environments and security patches.

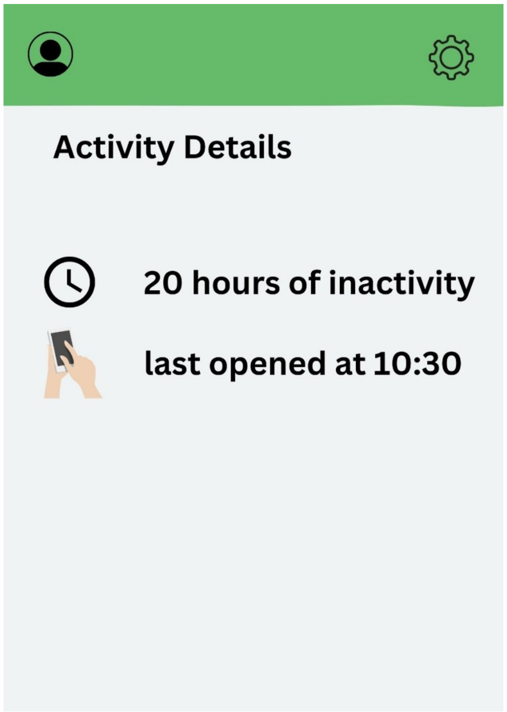
3. External Interface Requirements

3.1 User Interfaces

- **Caretaker Interface:**
- The caretaker interface will feature a dashboard providing an overview of the beneficiary's status and recent activity.
- It will include options to customize monitoring settings, such as time intervals for check-ins and rest times.
- Caretakers will be able to ping the beneficiary, and receive notifications for battery status and alerts.

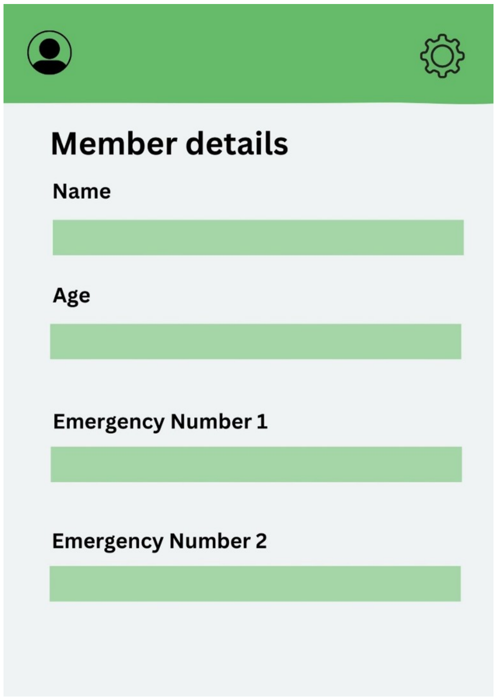


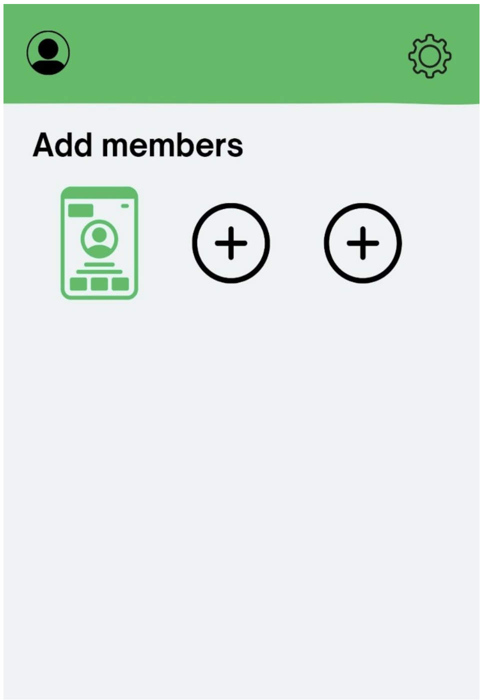

Member details


Activity Details

20 hours of inactivity

last opened at 10:30

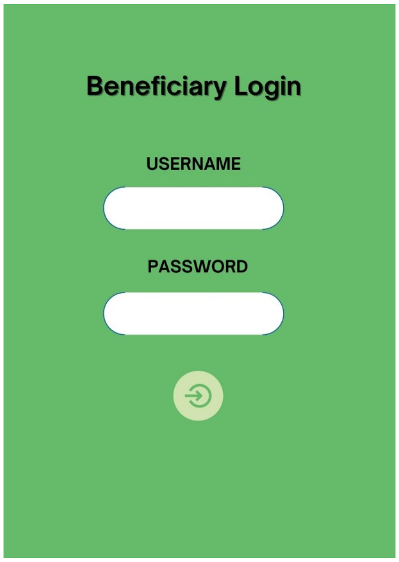

Member details

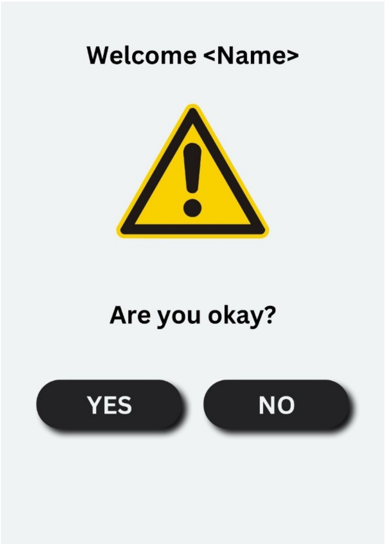

Add members

The images attached are representations. May not be final

- **Beneficiary Interface:**

- The beneficiary interface will offer a simplified view, focusing on providing reassurance and ease of use.
- It will display basic status indicators, such as the current monitoring status.
- Beneficiaries may have access to features like pill reminders and fall detection alerts, depending on their settings.





Welcome <Name>

Are you okay?

3.2 Hardware Interfaces

The software interacts with Android devices running Android 14 and above, supporting various smartphones. Built-in sensors such as accelerometer and gyroscope may be utilized for fall detection. The software's interaction with hardware components enables monitoring, alerting, and ensuring user safety and well-being.

3.3 Software Interfaces

1. **Android Operating System (Version 14):**
    - The application runs on the Android OS, utilizing its APIs for accessing device features like sensors, network connectivity, and user interface elements.
2. **Google Play Services (Latest Version):**
    - Utilized for functionalities such as location services.

3. **Third-Party Libraries and APIs:**
    - Integration with third-party libraries and APIs.

4. **Custom Backend Services:**
    - Developed to handle user authentication, data storage, and business logic processing.

3.4 Communications Interfaces

1. **Network Communication Protocols:**
    - Used for communication with external servers, APIs.
2. **Messaging Protocol:**
    - Utilized for communication between the application and external services, including message formatting, sending, and receiving.

4. System Features

4.1. Functional Requirements

1. **Continuous Monitoring:**
    - The application shall continuously monitor user's phone activity
2. **Alerting for Prolonged Inactivity:**
    - The application shall alert designated contacts in case of prolonged user inactivity.
3. **Caretaker Interaction:**
    - The application shall allow caretaker to ping the beneficiary as needed.
4. **Customizable Check-in Intervals:**
    - The application shall customize time intervals between scheduled check-ins.
5. **Customizable Rest Times:**
    - The application shall customize rest times for the beneficiary.
6. **Battery Low Alerts:**
    - The application shall send battery low alerts to the caretaker.
7. **Community Check-ins:**
    - The application shall allow caretaker to ping a member of the community to ensure beneficiary's well-being.
8. **Separate Logins:**
    - The application shall provide separate logins for caretaker and beneficiary.
9. **Pill Reminder Feature:**

- The application shall offer a pill reminder feature.

10. **Fall Detection Functionality:**
    - The application shall implement fall detection functionality.
11. **Customization of Alert Messages:**
    - The application shall enable customization of the message sent to the caretaker.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1) **Interface Responsiveness:**

   - The application shall provide a user-friendly and intuitive interface for both users and contacts, responding to user interactions within 1 second to ensure smooth and efficient navigation.

2) **Efficient Operation:**

   - The application shall operate efficiently with minimal battery and resource consumption on the monitored user's device, aiming to consume no more than 10% of the device's battery capacity per hour and utilize no more than 100 MB of RAM during regular operation.

5.2 Security Requirements

1. **User Authentication:**
   - The application shall implement user identity authentication mechanisms, such as password protection or biometric authentication, to ensure secure access to user accounts and prevent unauthorized access.

5.3 Software Quality Attributes

1) **Usability:** The application should prioritize usability, ensuring a straightforward and intuitive user experience for both caretakers and beneficiaries. A usability score of at least 90% should be achieved in user testing surveys, with feedback indicating ease of navigation and task completion.

2) **Reliability:** The application must be reliable, with a mean time between failures (MTBF) of at least 30 days. It should operate without unexpected crashes or errors, ensuring consistent performance and minimal downtime.

3) **Maintainability:** The application codebase should be maintainable, allowing for easy updates, bug fixes, and enhancements. Code changes should be implemented efficiently, with an average resolution time for reported issues not exceeding 48 hours.

4) **Robustness:** The application should be robust, capable of handling unexpected inputs, errors, and adverse conditions without crashing or data loss. It should gracefully recover from errors and maintain data integrity under all circumstances.

Appendix A: Team members

1	Sai Anand K	-	20221074
2	Shiva Surendran	-	20221076
3	Sidharth Thejas	-	20221079
4	S R Angleo Antony	-	20221081