

# Requirements Documentation

## ABSTRACT

This document outlines the functional and non-functional requirements for the mobile location verification application designed to track service personnel during service visits to customer homes. The system ensures that service personnel arrive at the correct location and allows both the homeowner and supervisor to track the personnel in real-time. The document includes use case models to represent functional requirements, a rationale for each use case, and a list of non-functional requirements that the system must meet.

## TABLE OF CONTENTS

LIST OF FIGURES	2
INTRODUCTION	2
Purpose and Scope of the document	2
USE CASE MODEL FOR FUNCTIONAL REQUIREMENTS	3
Figure 1 - Use Case Diagram 1: Customer Verifies Service Personnel's Identity	3
Figure 2 - Use Case Diagram 2: Supervisor Tracks Service Personnel in Real-Time	5
Figure 3 - Use Case Diagram 3: Supervisor Generates Service Report from the System	6
Figure 4 - Use Case Diagram 4: Service Personnel downloads Software onto Customer Phone	7
RATIONALE FOR YOUR USE CASE MODEL	8
NON-FUNCTIONAL REQUIREMENTS	8
EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT	9
ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS	9
ADDITIONAL REFERENCES	10

## LIST OF FIGURES

Figure 1 - Use Case Diagram 1: Customer Verifies Service Personnel's Identity  
Figure 2 - Use Case Diagram 2: Supervisor Tracks Service Personnel in Real-Time  
Figure 3 - Use Case Diagram 3: Supervisor Generates Service Report from the System  
Figure 4 - Use Case Diagram 4: Service Personnel downloads Software onto Customer Phone

## INTRODUCTION

This document serves as a comprehensive description of the requirements for the mobile location verification application. The purpose of the application is to track the real-time location and identity of service personnel providing services at customer homes. It provides visibility for both homeowners and supervisors to ensure that the service personnel is verified and reaches the correct destination.

### *Purpose and Scope of the document*

The purpose of this document is to capture and present the functional and non-functional requirements of the system in a clear, structured manner. The scope includes identifying the key actors interacting with the system, detailing the system's functionality through use case models, and outlining the requirements that the system must fulfill. The document will serve as a foundation for the system's design, development, and testing.

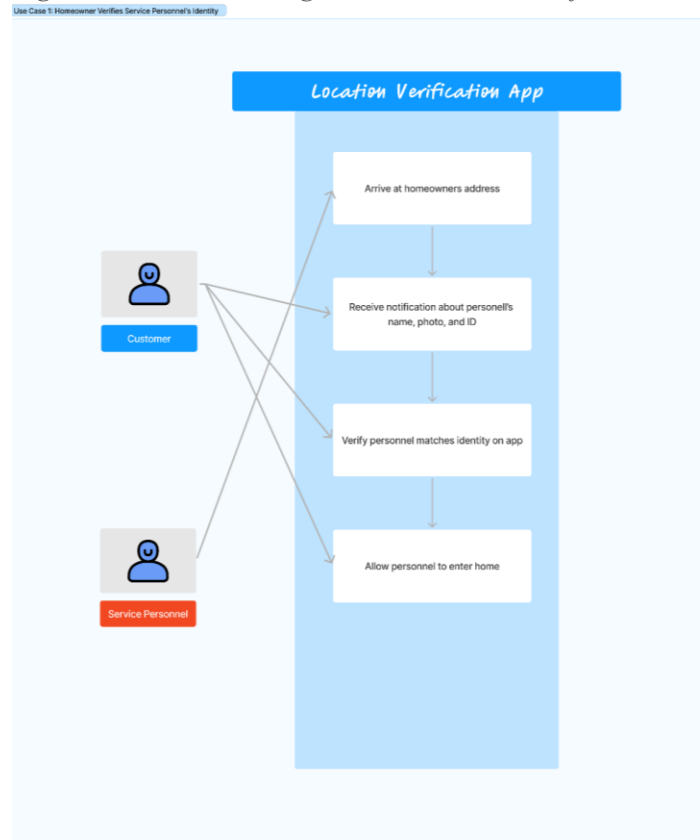
The document is structured as follows:

- **Use Case Models for Functional Requirements:** This section contains graphical and textual descriptions of the use cases that represent the system's core functionality.

- **Rationale for Use Case Models:** Each use case is explained in detail, including the reason for its inclusion and its role in meeting the overall system objectives.
- **Non-Functional Requirements:** These are the requirements that define the system's quality attributes, such as performance, security, and usability.

## USE CASE MODEL FOR FUNCTIONAL REQUIREMENTS

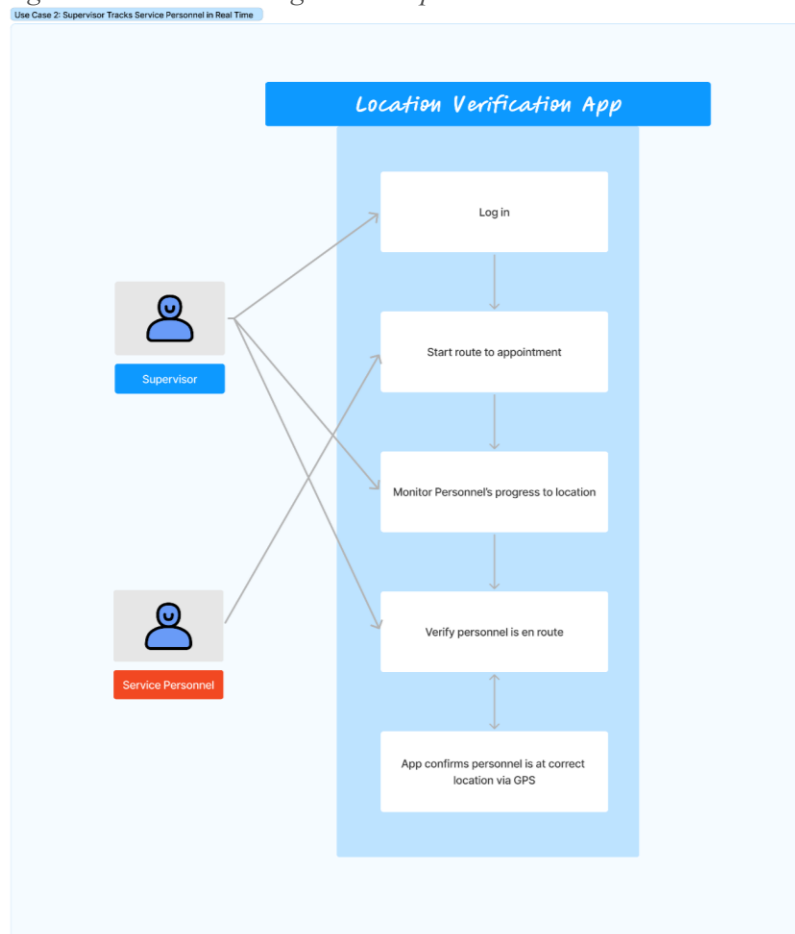
Figure 1 - Use Case Diagram 1: Customer Verifies Service Personnel's Identity



- **Use Case Name:** UC1 - Customer Verifies Service Personnel's Identity
- **Actors:** Homeowner, Service Personnel
- **Entry Conditions:**
  - Service personnel installed app
  - Customer has access to app
  - Service appointment is scheduled and recorded into system
- **Normal Flow of Events:**
  - Service personnel arrives to customer's house for appointment
  - Mobile app verifies service personnel's identity using unique ID or biometric Data (fingerprint or face recognition)
  - App sends notification to customer, notifying them about personnel's name, photo, and ID
  - Customer opens app to verify person at door matches verified identity
  - App confirms service personnel is at correct location via GPS
  - Customer allows personnel to enter once personnel is verified

- **Exit Conditions:**
  - Customer successfully verifies service personnel's identity and allows access
  - Homeowner denies access if identity verification fails
- **Exceptions (Alternate Flow of Events):**
  - Homeowner can't access mobile app
    - System sends SMS/call notification to customer with personnel's details
    - Customer can manually verify identity via other methods (call verification with service provider)
- **Special Requirements:**
  - Secure communication for identity verification
  - App should support multiple authentication methods
  - App must have fallback mechanisms

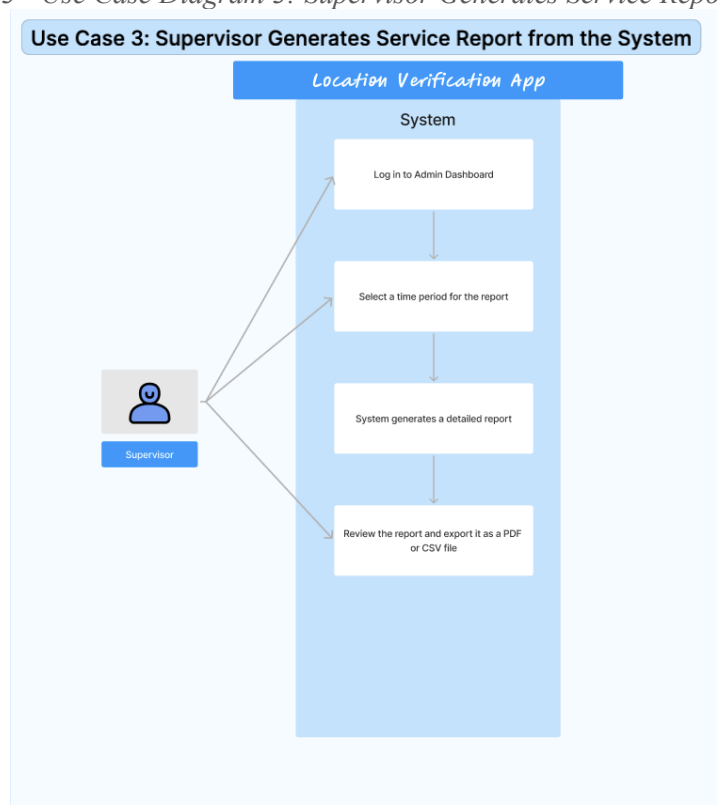
Figure 2 - Use Case Diagram 2: Supervisor Tracks Service Personnel in Real-Time



- **Use Case Name:** UC2 - Supervisor Tracks Service Personnel in Real-Time
- **Actors:** Homeowner, Service Personnel
- **Entry Conditions:**
  - Service personnel is logged into mobile app
  - Service personnel enables GPS
- **Normal Flow of Events:**
  - Service personnel start route to appointment
  - Location verification app tracks personnel in real time

- Supervisor views real-time location of personnel using admin dashboard
- Supervisor monitors personnel's progress towards the service location
- Supervisor is notified if there are deviations from expected route
- **Exit Conditions:**
  - Supervisor can verify personnel are en route and on time to appointment and track personnel until job is complete
- **Exceptions (Alternate Flow of Events):**
  - Personnel's GPS is off or unavailable
    - App sends reminder to enable GPS
    - If GPS unavailable for extended period, an alert is sent to the supervisor to directly contact the personnel
- **Special Requirements:**
  - Continuous GPS tracking
  - App must alert supervisor if there are any tracking interruptions
  - Ability for service provider to override automatic alerts (if deviations are expected)

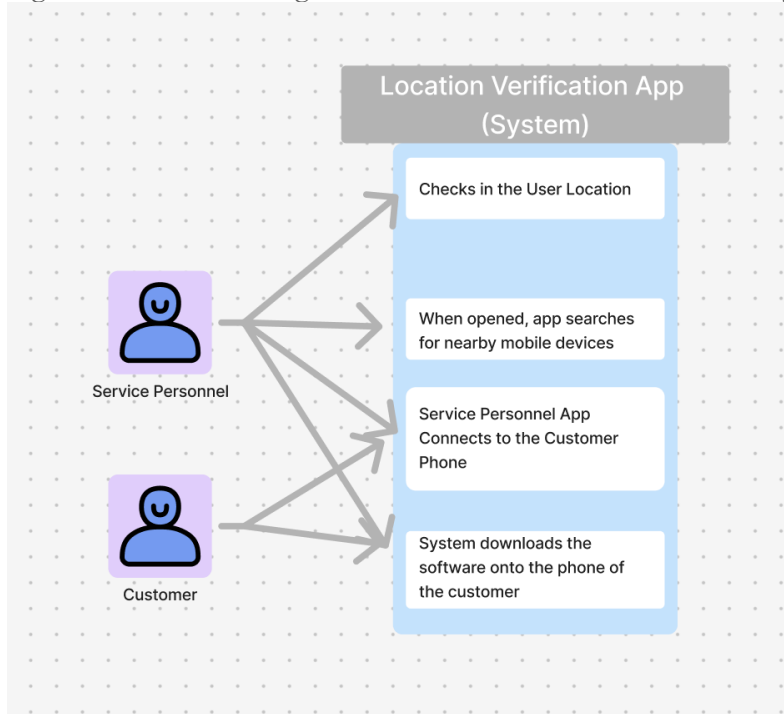
Figure 3 - Use Case Diagram 3: Supervisor Generates Service Report from the System



- **Use Case Name:** UC3 - Supervisor Generates Service Report from the System
- **Actor:** Supervisor
- **Entry Conditions:**
  - The supervisor is logged into the app's admin dashboard.
  - The system has recorded service personnel's location history and performance data during a selected time period.
  - The supervisor has access to reporting features in the app.
- **Normal Flow of Events:**
  - Supervisor logs into the admin dashboard of the app.

- Supervisor navigates to the reporting section and selects a time period for which the report is required (daily, weekly, or monthly).
- The System compiles data related to service personnel's location history, service completion times for the selected period.
- Supervisor reviews the generated report in the app, and exports the report as a PDF or CSV file if needed.
- **Exit Conditions:**
  - The supervisor successfully reviews the report and optionally exports it.
  - The report is generated based on the selected criteria and presented to the supervisor.
- **Exceptions (Alternate Flow of Events):**
  - If the System does not have sufficient data for the selected time period, it notifies the Supervisor that no report can be generated for the chosen period.
  - If there is a system error or issue generating the report, the System displays an error message and allows the Supervisor to retry the process or select a different time period.
- **Special Requirements:**
  - The system should have access to accurate, up-to-date service personnel data.
  - The application must be able to export reports in both PDF and CSV formats.
  - The system should handle cases where data is incomplete or missing and notify the supervisor accordingly.

*Figure 4 - Use Case Diagram 4: Service Personnel downloads Software onto Customer Phone*



- **Use Case 4:** Service Personnel downloads Software onto Customer Phone
- **Actors:** Service Personnel, Customer
- **Entry Conditions:**
  - The service personnel is within the range of the customer's phone

- The service personnel must have the app on their phones
- The service personnel must have internet or data based connection
- **Normal Flow of Events:**
  - The Service Personnel arrives to the location of the user
  - The System notes that the system personnel has arrived
  - The Service Personnel opens up the application
  - The System searches for nearby mobile devices
  - The Service Personnel connects to the customer's phone
  - the System downloads the software onto the phone of the user

**Exit Conditions:**

The system completes its download of the software  
the location of the user is confirmed to be tracked

**Special Requirements:**

- Location is being continuously traced
- The customer device must have space to store the software

## RATIONALE FOR YOUR USE CASE MODEL

The use case models are designed to address important security and efficiency needs in service-based businesses. The identity verification use case helps homeowners feel safe by confirming the identity of the service personnel through biometric methods and notifications. This improves security and builds trust in the process.

The real-time tracking use case helps service providers monitor their workers. Supervisors can see where service personnel are in real-time, ensuring they stay on schedule and helping if there are any issues. This improves transparency and reduces delays.

In addition, generating service reports helps supervisors track performance, increase accountability, and export data in various formats for easier analysis and decision-making. The system's error handling ensures reliability, even when data is incomplete.

Lastly, the service personnel downloads the app on the customer's phone so that their location can be tracked. This allows the service personnel's phone to connect to the customer, and this allows for the identity and location to be confirmed.

These use cases enhance security, transparency, and operational efficiency while ensuring reliable performance tracking and communication in service-based businesses.

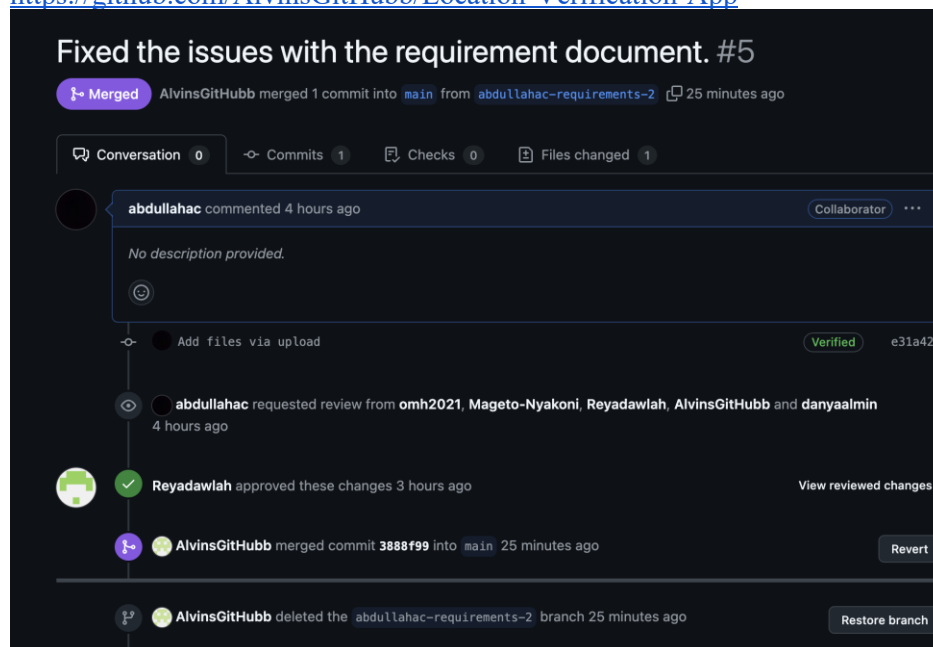
## NON-FUNCTIONAL REQUIREMENTS

- The app should be able to process location verification requests within 2 seconds to ensure timely updates
- The app should have an intuitive user interface that allows users to easily navigate and access features
- The app must be compatible with a range of devices and operating systems specifically targeting the latest two versions of iOS and Android
- The app should have a 99.9% uptime ensuring that location verification services are available at all times
- The app's codebase should be modular and well-documented to facilitate easy updates and maintenance
- The app must implement end-to-end encryption for all location data transmission to protect user privacy and prevent unauthorized access
- The app must comply with relevant legal and regulatory standards to ensure that user data is

- handled in accordance with privacy laws
- The app should adhere to accessibility standards to enable easy navigation for users with disabilities
- The app must provide consistent performance across different screen sizes, resolutions, and form factors.
- The app should have the capability to integrate with third-party service provider company apps
- The app must enforce additional layers of security such as multi-factor authentication for logging into the app or accessing sensitive data.
- The app should implement session timeouts and automatic logouts after periods of inactivity to improve security.
- The app should allow users to customize alert types (e.g., SMS, push notifications) for location updates or changes in service status.
- The app should ensure users can easily export their location history or account data in a structured format if required.
- The app should support real-time updates for continuous or frequent location tracking without significant delays.
- The app should be optimized to minimize battery consumption during location tracking and data transmission.
- The app should be able to handle an increasing number of users and location verification requests without compromising performance.

## EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT

- We will be using GitHub to keep a track of who is working on what part. It is a configuration management tool that we will use. Also, it will ask for approval from 2 group members before checking in a change to the document.
- <https://github.com/AlvinsGitHubb/Location-Verification-App>



## ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS

- Requirements Documentation



- IEEE Std 830-1998: Software Requirements
- IEEE Std 29148: Requirements Engineering
- ISO/IEC/IEEE Std 29148-2018: Systems and Software Engineering
  - Life Cycle Processes
  - Requirements Engineering

## ADDITIONAL REFERENCES

Belligoni, Sara, et al. "Privacy and Security Concerns with Passively Collected Location Data for Digital Contact Tracing among U.S. College Students." *PloS One*, U.S. National Library of Medicine, 22 Nov. 2023, [www.ncbi.nlm.nih.gov/pmc/articles/PMC10664924/](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC10664924/).

Priego, Nuria Rodríguez, et al. "Sharing but Caring: Location Based Mobile Applications (LBMA) and Privacy Protection Motivation." *Journal of Business Research*, Elsevier, 23 Nov. 2021, [www.sciencedirect.com/science/article/pii/S0148296321008298](http://www.sciencedirect.com/science/article/pii/S0148296321008298).

Wisniewski, Pamela, et al. "Predicting Smartphone Location-Sharing Decisions through Self-Reflection on Past Privacy Behavior." *OUP Academic*, Oxford University Press, 22 Sept. 2020, [academic.oup.com/cybersecurity/article/6/1/tyaa014/5910009](http://academic.oup.com/cybersecurity/article/6/1/tyaa014/5910009).

Yan, Shihao, and Robert Malaney. "Location Verification Systems in Emerging Wireless Networks." *arXiv.Org*, 12 July 2013, [arxiv.org/abs/1307.3348](http://arxiv.org/abs/1307.3348).

- Requirements Documentation
  - Lamsweerde, A.V., 2009. Requirements Engineering: From System Goals to UML Models to Software Specifications. John Wiley