Software Requirements Specification
Temi Hospital Robot Android Application

Date:	**01/07/20**
Revision Number:	**0.1**
Author:	**Brandon Lu**
Contact:	**lu0248@flinders.edu.au**

College of Science and Engineering

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Brandon Lu	01/07/20	Initial Commit (Draft)	0.1

List of Tables

No table of figures entries found.

Table of Figures

Table of Equations

No table of figures entries found.

1. Introduction

A. Purpose

This document specifies the software requirements and architecture for the Android application to be installed on a Temi Robot for use in a hospital foyer. This SRS will detail only the Android software which is to be built on out-of-scope firmware which defines and controls the robot's physical functions. This document also restricts its scope to only Stage One of the project and the subsequent Stages will be described in later revisions.

B. Intended Audience and Reading Suggestions

This document is intended to be read as a plan for developers on this project or related extension work, documentation writers, project supervisors and users who are interested in modifying the implementation.

Reading sequence…

C. Project Scope

An autonomous assistant is to be created from a modified off-the-shelf robot for use in a hospital setting. The development machine is Temi, which runs an Android OS on its robotic firmware with a basic API for robot control.

The global COVID-19 pandemic has exposed needs for the prevention of human-to-human infection on a large scale. This project will focus on two actions that will decrease the risk of infection in a public area and increase the rate at which COVID cases are detected. This may be achieved through reminding people to sanitise their hands and screening of visitors into the hospital. Currently, these goals are expensively and non-optimally achieved through human interaction and static sanitiser distribution points.

Phase One of this project will aim to increase the rate of sanitisation of visitors entering hospitals by providing an autonomous assistant equipped with a touchless sanitiser dispenser which will interact with visitors, encourage them to sanitise their hands and provide some novelty in the environment. The software required for this task is an application-layer Android implementation that will use Temi's provided SDK to control the robot and provide a touchless interaction interface. Future revisions may require interfacing with electronic components for custom functionality.

Phase Two of this project will aim to remove the cost of dedicated staff and resources by providing autonomous screening of visitors using a thermal imaging camera. This data may be used in real-time to target visitors suspected of showing fever-like symptoms of high body temperature, and act accordingly. The project will first analyse the ethical concerns about recording data in this manner and the specific engineering requirements will be defined later.

D. References

I. Excerpt from Vision Statement

Autonomous Assistant for a Hospital Lobby
Karl Sammut, Paulo Santos & Russell Brinkworth
Flinders University

In the light of the current COVID19 pandemic, hand hygiene in healthcare setting is an imperative action for the well-being of patients and health workers. The distribution of static sanitizer stations, however, is not ideal as they don't catch the attention of the users and are usually ignored by most people. One solution to this problem, that has been tried in various locations, is to place a person to remind people to sanitise their hands. Not only this is an expensive solution, but also a dangerous one, as the person in charge of this task is at risk infecting themselves and the others around. This proposal investigates the development of a mobile touchless hand sanitizer station, mounted on a mobile robot whose goal is to rover around the hospital lobby, recognise human figures and advise them to sanitise their hands while simultaneously providing a convenient way for them to do so. This shall bring a greater interest for people to use the sanitizer dispenser, while providing the opportunity for some comic relief and novelty to the hospital lobby. It could also provide an ideal platform to convey important additional information to, and/or collect useful information from, hospital visitors.

The initial development of this project has two major parts:

1. The development of mapping, localisation and the recognition of human figures. This encompasses the traditional robotic development, that is essential for the basic functioning of the mobile station.
2. The investigation of human-robot interaction strategies to better engage the robot with the human users. For this part, the development of a graphical interface (a "face") for the robot is an essential part to making the robot approachable and user friendly.

Phase two of the project will entail the addition of a thermal camera to the robot in order to check the temperature of visitors to the hospital as they are using the hand sanitiser station.

II. Temi SDK

https://github.com/robotemi/sdk/wiki

2. Overall Description

A. Product Perspective

The Android software will be newly built on top of the provided developer interface in Java and Kotlin. The developer interface interacts with the lower-level machine functions such as localisation, movement, speech recognition and synthesis, human figure recognition and joint movement; all of which are abstracted from the Android developer and has no clear direct access method. This describes Stage One of this project. It is important to note that the machine does not provide a hardware interface such as a USB port and that external hardware will need to be attached using a wireless communication medium.

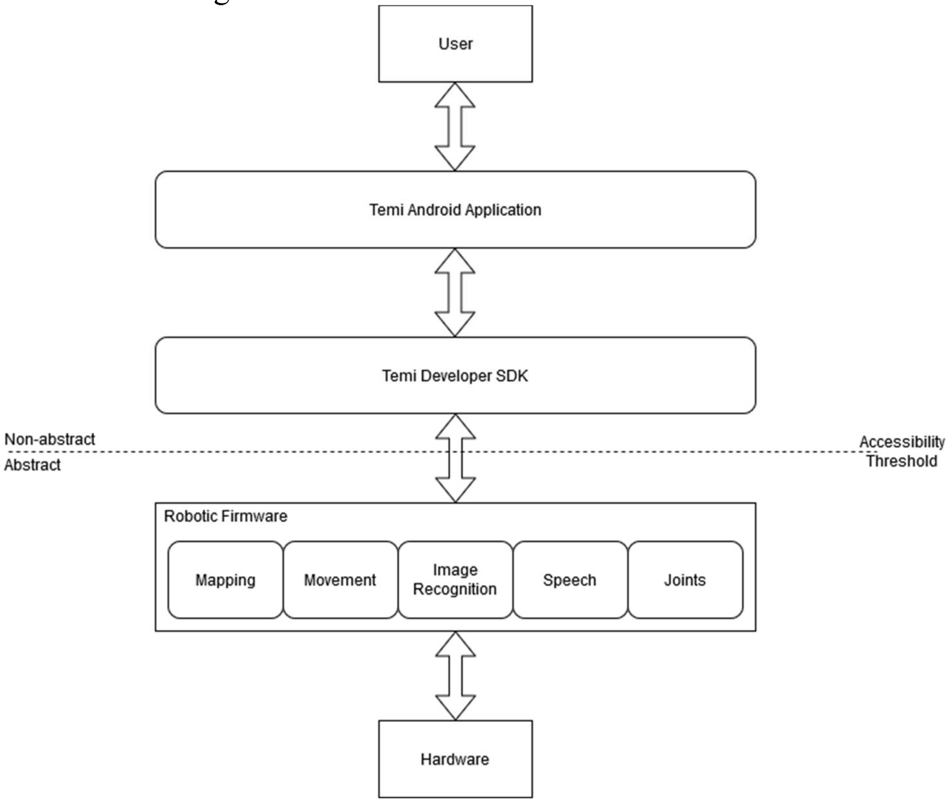


Figure 1: Phase One Software Perspective

B. Product Features

Patrol
Recognise human
Approach
Ask
Interact
Dispense
Maintain State
Provide Novelty
Do not crash into people
The details of the above functions will be described in Section 3.

C. User Classes and Characteristics

User Class 1	Visitor
Inputs	• Physical Presence • Voice (choice, confirmation)
Outputs	• Speech Synthesis • Hand Sanitiser • High-level system state feedback • Subtitles
Usage Frequency	High (one per minute)
Technical Expertise	None
Security and Privilege	None
Interactions	• Be scanned without their knowledge of the robot's presence • Provide choice using voice command whether they wish to receive sanitiser • Change the display language using voice command (this may not be possible if Temi does not support non-English voice recognition) • May damage the system intentionally or unintentionally
Characteristics	• Initially may not know of the machine's presence • Initially may not know of the machine's immediate location • Initially may not know of the machine's functions • Has an unknown infection and health status • May wear clothing that obscures their physical and facial form

User Class 1	StaffOperator extends Visitor
Inputs	• Physical Presence • Voice (choice, confirmation) • Administrative commands
Outputs	• Speech Synthesis • Hand Sanitiser • High-level system state feedback • Subtitles
Usage Frequency	Low (twice per day)
Technical Expertise	Low (user manuals)
Security and Privilege	Medium (modifications to machine and application settings)
Interactions	• Be scanned without their knowledge of the robot's presence • Provide choice using voice command whether they wish to receive sanitiser • Change the display language using voice command (this may not be possible if Temi does not support non-English voice recognition)

	• Enter a secret menu to change machine settings and perform other administrative features • Interact with the hardware to refill sanitiser • Interact with hardware to turn the machine on and off • Store the machine when not in use • Set up the machine from storage
Characteristics	• Initially may not know of the machine's immediate location • Has an unknown infection status (asymptomatic COVID-19) • May wear clothing that obscures their physical and facial form • Will have a copy of the user documentation

D. Operating Environment

The software application must be written in Java and Kotlin for Android 6 API Level 23 and must not interfere with the other applications installed in the factory version of Temi's Launcher. The Android OS interfaces through an unknown tablet-style touch display.

E. User Documentation

The deliverable user documentation will consist of a project logbook, scrum backlog, a physical user manual, an online user manual and developer documentation.

F. Assumptions and Dependencies

This software is only dependent on the correctness of the robotic firmware and launcher provided by the robot manufacturer.

```
testImplementation 'org.robolectric:robolectric:4.3'
testImplementation 'junit:junit:4.12'
```

3. Specific System Requirements

A. Visitor

A Visitor is a user that enters the physical environment and may not know of the machine's existence. The visitor may interact with the machine when prompted.

FRQ1. The system shall approach visitors. When in patrol mode, the system shall continuously look for human forms through the RGB camera. After applying logic [TBD] the system shall decide to stop patrolling and follow a recognised human form.

FRQ1.1. The system shall autonomously patrol an area.

FRQ 1.1.1. The system shall move to a pre-defined set point.

FRQ 1.1.2. If a location is not reached in an allocated time, the system shall call for assistance.

FRQ 1.1.3. When at the set point, the system shall speak some dialogue and wait for input for a set amount of time.

FRQ 1.1.4. The system shall monitor its battery life and act accordingly.

FRQ1.2. The system shall search for human forms when in the patrol state.

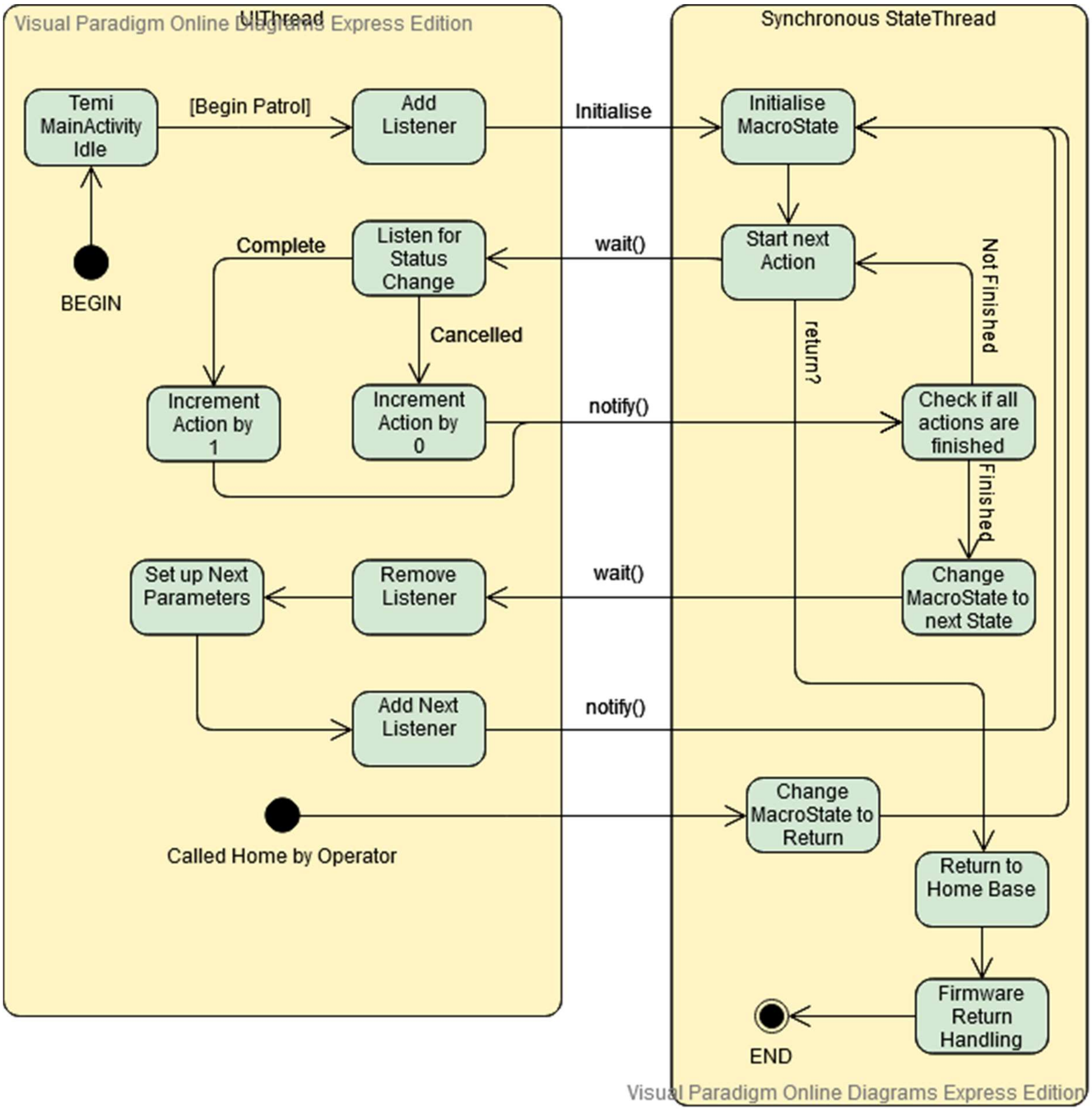
FRQ 1.2.1. When in patrol state, the system shall use the RGB camera to detect humans.

FRQ 1.2.2. The system shall decide to approach a human and then follow them.

FRQ 1.2.3. While performing FRQ 1.2.2, the system shall communicate with the user using speech synthesis and through the display.

FRQ1.3. The system shall not travel out of its operating range.

FRQ 1.3.1. The system shall provide some mechanism for staying within its operating range.
Note: this may be difficult due to the lack of access available to localisation data.



FRQ2. The system shall interact with visitors.

FRQ2.1. The system shall greet a visitor using combination of speech synthesis, music, videos, images and text.

FRQ 2.1.1. The system shall provide a brief explanation of its directive. Then ask for permission to deliver sanitiser.

FRQ 2.1.2. The system shall listen for and act upon responses from a visitor.

FRQ2.2. The system shall dispense hand sanitiser to a visitor upon receiving permission.

FRQ2.3. The system shall include an interface for changing the written language.

FRQ2.4. The system may include an interface for changing the spoken language.

FRQ3. The system shall avoid visitors and other obstacles when patrolling.

B. Staff Operator extends Visitor

A Staff Operator (or just "Operator") is a subclass of Visitor and has a higher privilege level. Operators will have read the user documentation and will be somewhat familiar with the machine's functions and interfaces. The set of Operator use cases completely includes the set of Visitor use cases.

FRQ1. The system shall store and delete locations from memory.

FRQ1.1. The system shall be able to follow an operator to a location.

FRQ1.2. The system shall be able to store its current location in memory.

FRQ1.3. The system shall be able to display all saved locations.

FRQ1.4. The system shall be able to delete locations.

FRQ2. The system shall return to its docking station or "home base" at an operator's request.

FRQ2.1. The system shall respond to a pre-defined voice command and stop all actions and then return to the home base.

FRQ2.2. If a location is not reached in an allocated time, the system shall call for assistance.

FRQ3. The system shall provide a secure mechanism for an operator to access privileged functions.

<Don't really say "User Class 1." State the class name and describe it in just a few words based on 2.3.>

3.1.1 Functional Requirement 1

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<The format required for each functional requirement is the sample use-case format as given in lectures, topic manual page 45. Each functional requirement should have a data flow diagram supplement where possible to outline the data flows associated with the requirement (most likely a level 2 DFD). {Some minor functions will not need an associated data flow diagram as they are sufficiently self-explanatory without}>

4. External Interface Requirements

A. User Interfaces

Due to the nature of COVID and other infectious diseases, the physical interaction between users and the machine must be minimised. Therefore, the user interface must encourage audio interaction over touching the screen.

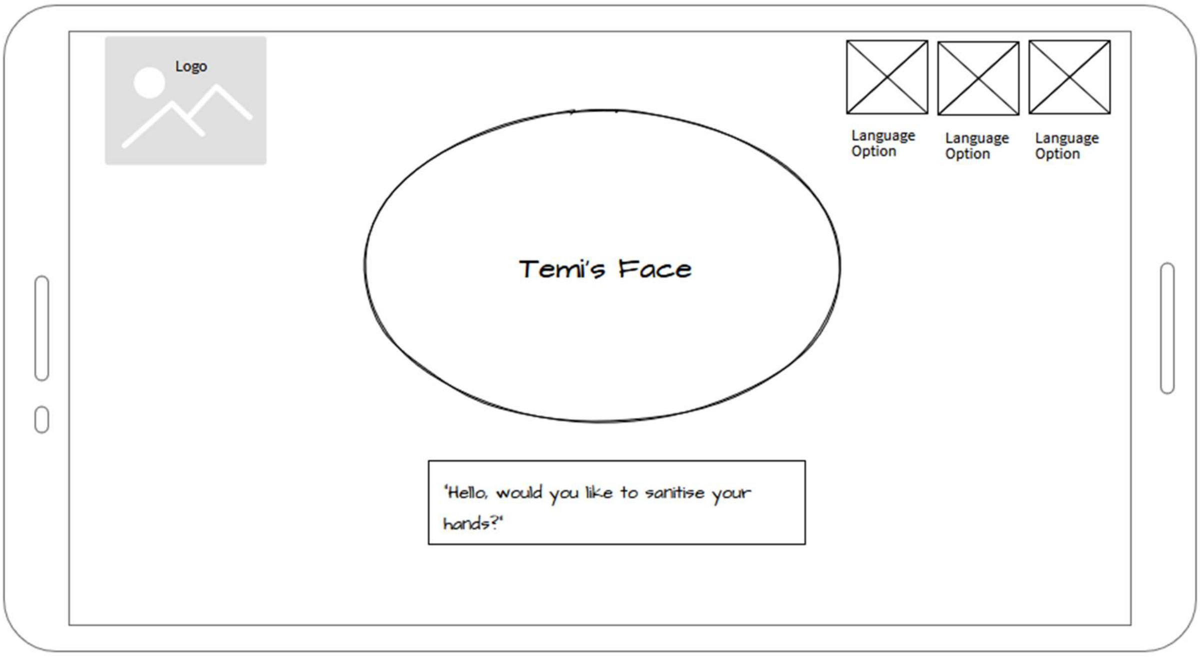
1. Temi Launcher Interface
   The Temi Launcher is the pre-loaded interface used to interact with the machine. Developed functions should aim to minimise the use of this provided interface to keep

activity within the software system (Android application), rather than the launcher application.

1.1. Add Location (pre-loaded)
1.2. Delete Location (pre-loaded)
1.3. Display Location (pre-loaded)

2. Application Touch Screen Face `Activity MainActivity`

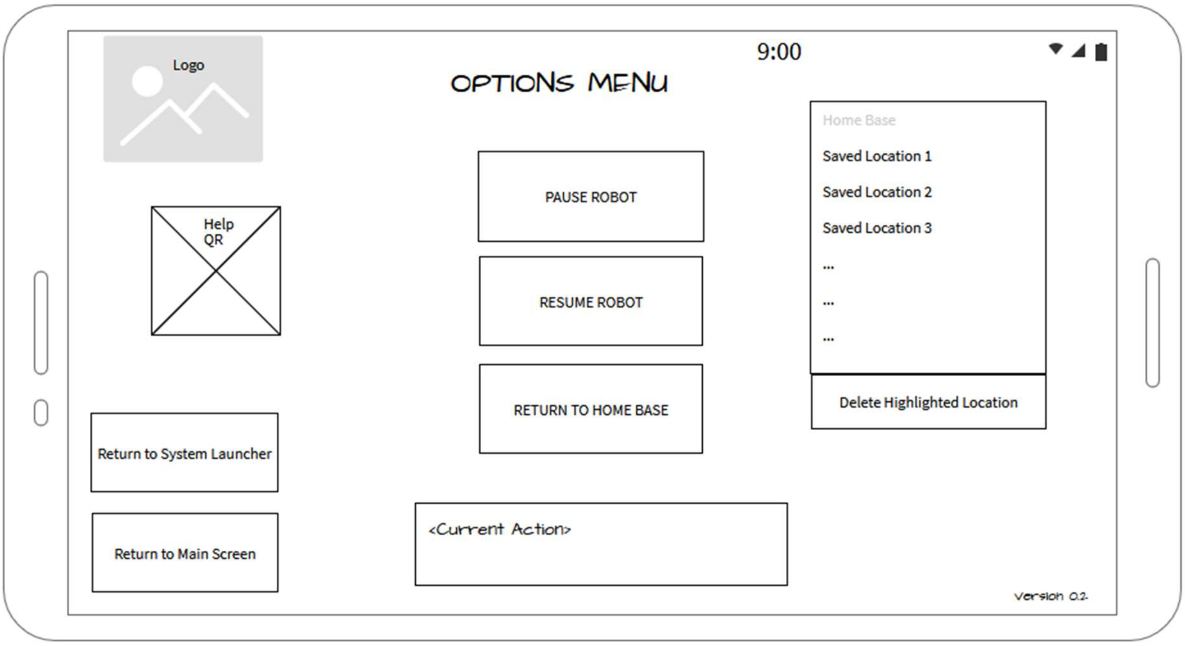
2.1. Temi will be humanised through a face on the main screen. This may be animated.
2.2. The face shall provide a text component that displays Temi's "thoughts" or current TTS requests.
2.3. The application interface may provide a touch method of changing the interface display language.
2.4. The main interface may show the battery state of charge.
2.5. The main interface may show the current level of remaining sanitiser.
2.6. The main screen shall include an enabled invisible UI element that directs an Operator to Interface 3 to perform the actions required to fulfil Requirement FRQ3. The location of the hidden button shall be described in the physical user manual.



3. Operator Menu `Activity OperatorMenu`

3.1. A menu shall be provided accessible through an element described by Interface Requirement 2.6 that provides access to the following privileged functions.
3.1.1. Exit to Temi Launcher Menu
3.1.2. Volume Control prompts
3.1.3. Pause State Machine
3.1.4. Resume State Machine
3.1.5. Return to Home Base
3.1.6. Display System Diagnostics
3.1.7. View Version number and developer information
3.1.8. Access online documentation through QR code
3.2. A visible element shall be provided to return to the `MainActivity`.
3.3. The interface shall return to the `MainActivity` after a short amount of idle time.

3.4. Access to the `OperatorMenu` may be password protected.



B. Software Interfaces

None.

C. Hardware Interfaces

3.4.1. The system may have an interface with the dispenser to control the timing and manage the amount of remaining liquid.

D. Communications Interfaces

3.4.2. The system shall have minimal dependencies on external communications such as wireless LAN.

3.4.3. If connections are mandatory for function, the system shall handoff to different access points without user intervention.

5. Other Non-functional Requirements

A. Performance Requirements

3.4.4. The system shall not exceed [TBD] metres of Line-of-Sight displacement from the home base.

3.4.5. The system shall not exceed [TBD] metres of displacement from the home base in the presence of walls and obstacles.

B. Safety Requirements

1. The system shall always make users aware of its presence for safety reasons.
    1. The system shall announce its actions through audio and visual means as it performs them.
    2. The system shall broadcast its location through audio while idle.

C. Security Requirements

1. Privileged Actions shall be inaccessible to Visitors but not Operators.

D. Software Quality Attributes

Usability

6. Other Requirements

This application shall be installed through the Android Debug Bridge utility through Android Studio. To make a more extensible package, the application may also be made available and installable through version control (eg. Github) or the Temi App Store.

7. Conclusion

Appendix A – Glossary

Appendix B – Issues List

Appendix C – Survey and Interview Questions

Appendix D – How-to Notes