GestureWave

Software Requirements Specifications

V2.0

April 20, 2013

Authors: Benjamin Piatt & Eusebiu Pintilei

Team Number: 3

Mobile Application Development

CS442-02 Android Project

Instructor: Simon Hao

Spring 2013

# Revision History

GestureWave is publicly hosted at https://bitbucket.org/piatt/gesturewave/

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 11-26-12 | Commit 87e40b9 | Benjamin Piatt | small changes to fix lint warnings, added icons (baseline) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Benjamin Piatt | Student | 3-6-13 |
|  | Eusebiu Pintilei | Student | 3-6-13 |
|  | Simon Hao | Instructor |  |

**Table of Contents**

Revision History 2

Document Approval 2

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

2. General Description 5

2.1 Product Perspective 5

2.2 Product Functions 5

2.3 User Characteristics 5

2.4 General Constraints 5

2.5 Assumptions and Dependencies 5

3. Specific Requirements 6

3.1 External Interface Requirements 6

3.1.1 User Interfaces 6

3.1.2 Hardware Interfaces 7

3.1.3 Software Interfaces 7

3.1.4 Communications Interfaces 7

3.2 Functional Requirements 7

3.2.1 Proximity Sensor Support 7

3.2.2 Accelerometer Support 7

3.3 Use Cases 7

3.4 Classes / Objects 8

3.5 Non-Functional Requirements 8

3.5.1 Performance 8

3.5.2 Reliability 8

3.5.3 Availability 8

3.5.4 Security 8

3.5.5 Maintainability 8

3.5.6 Portability 8

3.6 Inverse Requirements 8

3.7 Design Constraints 9

3.8 Logical Database Requirements 9

4. Analysis Models 9

4.1 Sequence Diagrams 9

4.2 Data Flow Diagrams (DFD) 9

4.3 State-Transition Diagrams (STD) 10

5. Change Management Process 11

A. Appendices 11

# 1. Introduction

## 1.1 Purpose

## The purpose of this document is to present the details and requirements of GestureWave, our CS 442-02 android application project. In this document, we will give a general description of the project, followed by its specific requirements, followed by our analysis models of the project, followed finally by a description of our change management process and any additional information pertinent to the project.

## 1.2 Scope

## GestureWave is an android application designed to assist a smartphone user in accomplish common smartphone tasks, such as opening an application, without the need to physically interact with the screen. This is useful for a few reasons. A user may find it more convenient to use a gesture based shortcut to open an application or accomplish a similar task within the operating system without having to physically touch the screen and follow the necessary steps to accomplish the task themselves. An obvious application of this is for a user who does not want to or cannot touch the screen because they are either wearing gloves, have something on their hands, or because the phone is in their pocket and taking it out is not immediately feasible or convenient. Using GestureWave also gives power users hands-free like shortcuts to commonly used actions, such as pausing music, or scrolling through a gallery. In addition to adding a convenience to smartphone users, GestureWave is also completely customizable, allowing the user to selectively enable or disable actions as well as choose what applications they would like to bind to a gesture based shortcut.

## 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| API | Application Programming Interface |
| Proximity Sensor | Detects the presence of nearby objects without any physical contact |
| Accelerometer | Detects orientation in the x, y, and z planes |
| ICS | Ice Cream Sandwich, Google’s Android operating system, version 4.0 |
| GC | Garbage Collection: the act of an operating system removing references to objects or data that is invalid or no longer needed for use by the application. |
| XML | Extensible Markup Language: a metalanguage that allows users to define their own customized markup languages |

## 1.4 References

Gnu General Public License Version 3 can be found at http://www.gnu.org/licenses/gpl.html

# 2. General Description

GestureWave is an android application designed to assist a smartphone user in accomplish common smartphone tasks, such as opening an application, without the need to physically interact with the screen.

## 2.1 Product Perspective

This is a standalone project that has similar functionality to other applications in the Google Play Store.

## 2.2 Product Functions

The product will read user input from the proximity sensor and the accelerometer and perform predefined actions based on user input. The user can change the applications he wishes to open via the settings interface and also specify number of gestures for specific applications.

## 2.3 User Characteristics

This application is designed for any smartphone user but we would anticipate that power users would be our main target group that want to use shortcuts, like opening frequent applications and pause music. Smartphone users who are most likely to use custom launchers, roms and people who like to customize their devices will be more likely to use our application. The users of our application should know the basic use of an accelerometer and proximity sensors and also the location for the proximity sensor on their device.

## 2.4 General Constraints

* Hardware: proximity sensor and accelerometer
* API level: only supporting API 15+ due to deprecation and backwards compatability issues
* Security: properly locking and unlocking the screen without compromising user privacy

## 2.5 Assumptions and Dependencies

# In order to use GestureWave, it is assumed that the smartphone on which the application is installed has both a proximity sensor and an accelerometer. Without one or both of these embedded hardware items, functionality of the app will be severely limited, with respect to what the application is designed to do. The ability to run the application, edit settings, and start and stop the background service are not affected by these dependencies. GestureWave also has a minimum android operating system dependency. Due to compatibility issues when dealing with deprecated API functions for older versions of the android operating system, GestureWave will only be available for phones running ICS and higher.

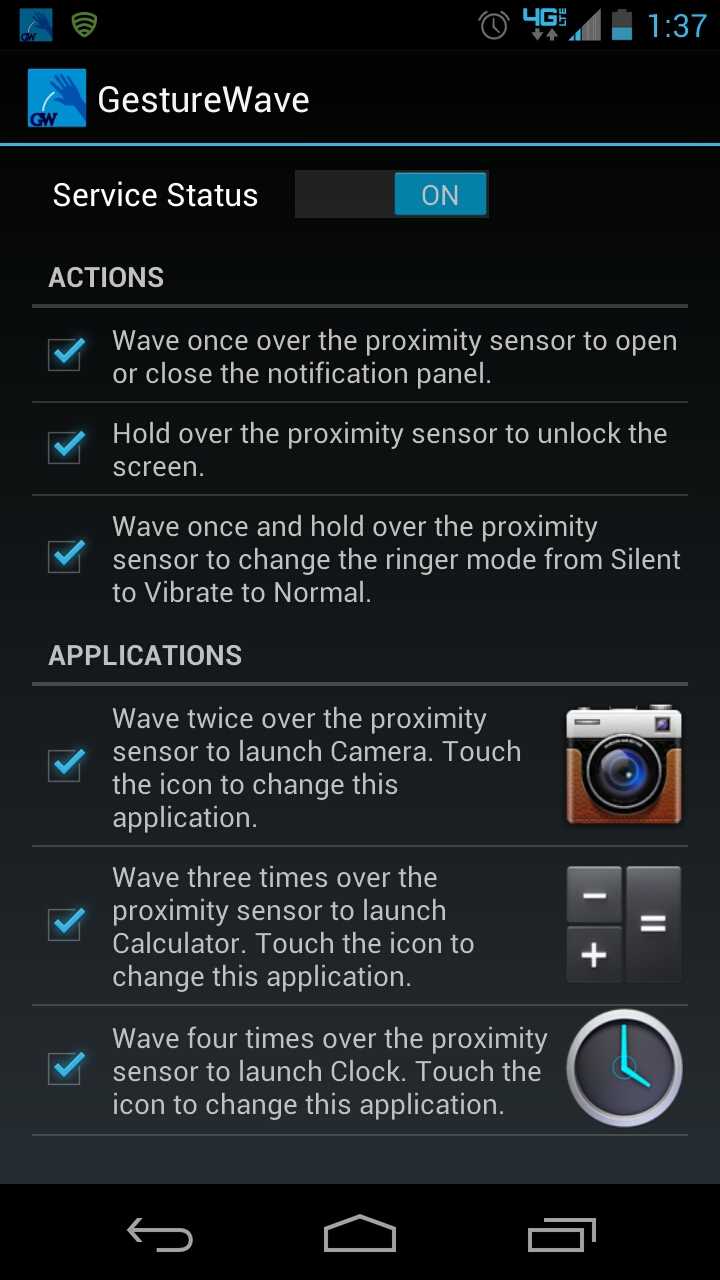
# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

We have one user settings interface which includes:

* Turning service ON/OFF
* Checkboxes to enable/disable predefined actions by the developers
* Checkboxes to enable/disable applications chosen by the user by clicking on the icons on the right
* Notification in the notification bar once the service is started



### 3.1.2 Hardware Interfaces

Using the android sensor manager interface, we connect the software to the hardware components, namely the proximity sensor and the accelerometer.

### 3.1.3 Software Interfaces

Communication between our application components is handled via android’s shared preferences framework, where any changes made in our preferences user interface are persisted to the running background service.

### 3.1.4 Communications Interfaces

Logging of user interactivity will be accomplished via network packets. User permission will be needed for the transmission of user data to the developers.

## 3.2 Functional Requirements

### 3.2.1 Proximity Sensor Support

The application supports proximity sensor input from the user, for performing some of the use cases described in Section 3.3.

### 3.2.2 Accelerometer Support

The application will support accelerometer input from the user, for performing some of the use cases described in Section 3.3, either by itself or in combination with input from the proximity sensor.

## 3.3 Use Cases

* As a user, I can check or uncheck the following options via a settings interface, to enable or disable the following options from being available while the app is running.
* As a user, I can switch the service toggle via a settings interface, to start or stop the app’s background service.
* **UPDATE: As a user, I can link any of the actions or applications below to a custom combination of proximity gestures and accelerometer positioning via the settings interface. Instead of preset gestures using only the proximity sensor, the user will have the option to create a custom combination via pickers to bind to each predefined action or application that they want to enable.** **Here are the predefined action which we have implemented:**
  + **unlock the screen**
  + **expand or contract the notification panel**
  + **change the ringer mode from silent to vibrate, vibrate to normal, or normal to silent**
  + **launch an app of my choosing**
* As a user, I can choose an app from a list of installed apps via a settings interface, to determine which of the above gestures launches that app.

## 3.4 Classes / Objects

* Settings Activity: The settings activity class creates a preference fragment that hosts the user interface
  + Toggle preference: this turns the service ON/OFF
  + Action preference: this enables and disables predefined actions
  + Application preference: this enables and disables actions bound to custom applications
    - List adapter: displays a list of installed applications from which the user can choose
  + **UPDATE: Custom gesture picker: displays options from which the user can choose a combination of proximity sensor input and accelerometer positioning to bind to the action or application.**
* Sensor Service: This is a background service which reads and processes sensor input and executes the actions specified by the settings interface

## 3.5 Non-Functional Requirements

### 3.5.1 Performance

Memory usage while the service is running in background mode should not increase with time. GC should only need to occur after the user chooses a new application from the list of installed apps for binding to a custom action.

### 3.5.2 Reliability

Proximity sensor input is disabled while in a phone call.

### 3.5.3 Availability

Proximity sensor input is only available when the service is turned on and the phone screen is lit, except for when the user is in an active phone call.

### 3.5.4 Security

For unlocking the screen, the feature only works when the type of unlock is non-secure. For future features we may implement secure unlocking of screen with user specified proximity sensor input. In addition, we intent to use the latest API level to avoid using deprecated calls or code that is not fully backwards compatible.

### 3.5.5 Maintainability

Our XML classes have all string requirements which make it easy to maintain the project and changing the attributes only in one place to stay consistent i.e. no hard coded values.

### 3.5.6 Portability

Our application is for use on Google’s Android operating system, and we currently support API level 15 and higher.

## 3.6 Inverse Requirements

The user should not be able to expose any functionality that is not inherently intended or enabled by the user or by the developer while using the application.

## 3.7 Design Constraints

This application is licensed under GPL v3, and as such, the source code is publicly accessible, but cannot be used for monetary gain by any other party who adopts our source code and must continue to be licensed under GPL v3.

## 3.8 Logical Database Requirements

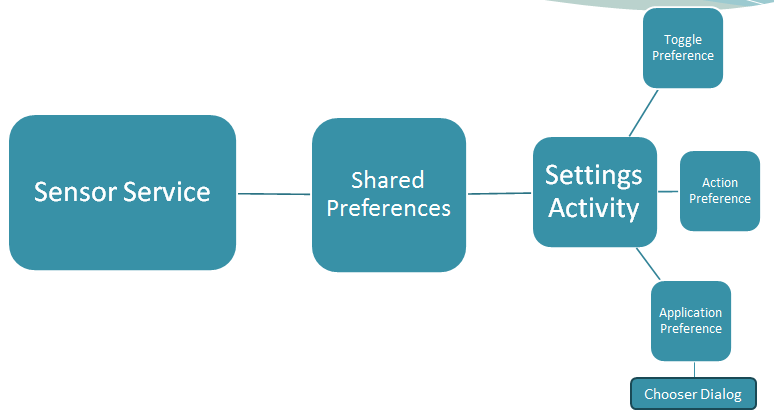
Instead of using an internal database for storing preferences, this application uses the exisiting shared preferences framework for data persistence, provided to us by the android API.

# 4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.

## 4.1 Sequence Diagrams

## 4.2 Data Flow Diagrams (DFD)



## 4.3 State-Transition Diagrams (STD)

# https://lh5.googleusercontent.com/8FPwrUDASk3SGHcMvdJe5-a7uvNkSaGbbOl6y8iXzcXV8Ne7DjafdR3IhgLNMz_f7JHMJBTtEns9WyA6V7Ukn1Ichv48iV2svWnIU7TcU7of3B2WvBvspb1W5. Change Management Process

The developers will add any additional requirements and use cases as the project changes. Since the developers on this project are also the designers, testers, and maintainers, any changes will be internally approved and submitted.

**A. Appendices**