SWE40001 Software Engineering Project

Remote Guidance System

Software Requirements Specification

Group 21

Lyndon Prado - 9740783

Tingcong Jimmy Li - 100017000

Keagan Foster - 101609822

Ayub Khan - 100667654

Dineth Gunawardena - 100862158

Kosala Edirisinghe - 101265981

Krishna Adhikari - 4953193

Liam Pan -101106174

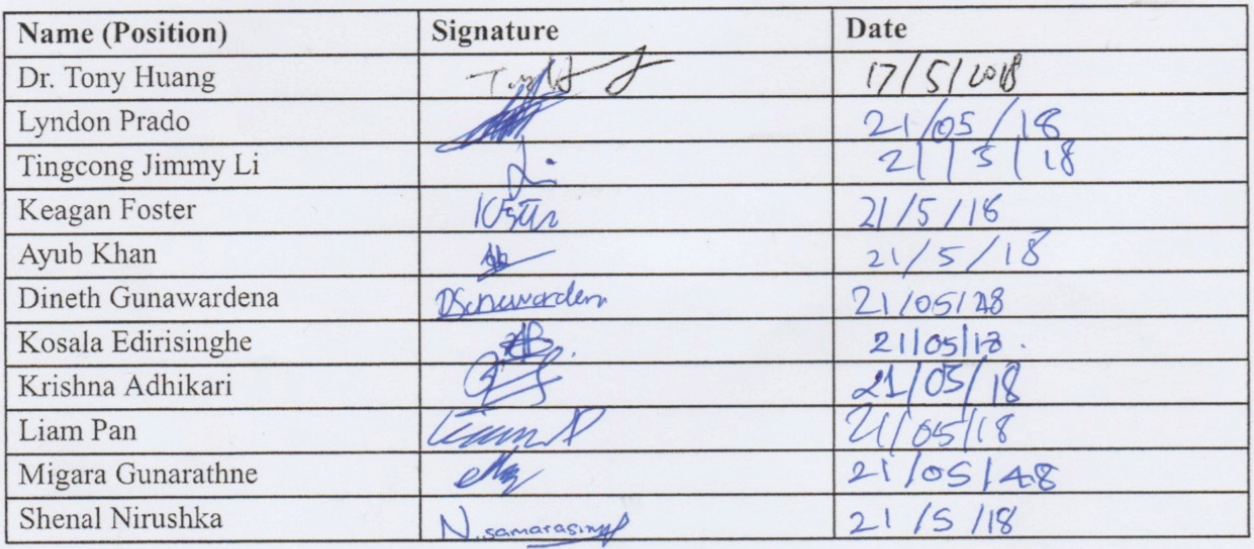
Migara Gunarathne - 101002269

Shenal Nirushka – 101054998

**Table 1. Document Change Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Authors** | **Summary of Changes** |
| 1.0 | 19/04/2018 | Tingcong Jimmy Li | Initial Draft |
| 1.2 | 08/05/2018 | Tingcong Jimmy Li | Updated based on feedback from quality assurance meeting. |
| 1.21 | 09/05/2018 | Liam Pan | Document Standards Review  Spelling and Grammar Review |
| 1.3.0 | 11/05/2018 | Tingcong Jimmy Li | Removed unnecessary product functions  Added new product functions  Update product perspective  Updated Product Block Diagram  Update use case diagram  Updated use case descriptions to reflect changes in product functions and use case diagram  Update user characteristics to reflect changes in user requirements  Update user interface to reflect changes in user requirements  Added References |
| 1.31 | 13/05/2018 | Liam Pan | Document Review |

**Table 2. Document Sign Off**



# Table of Contents

[Table of Contents 3](#_Toc514935363)

[1. Introduction 5](#_Toc514935364)

[1.1. Purpose 5](#_Toc514935365)

[1.2. Project Scope 5](#_Toc514935366)

[1.3. Definitions, acronyms, and abbreviations 5](#_Toc514935367)

[2. Overall Description 7](#_Toc514935368)

[2.1. Product Perspective 7](#_Toc514935369)

[2.2. Product Functions 8](#_Toc514935370)

[2.2.1. Send video feed 8](#_Toc514935371)

[2.2.2. View video feed 8](#_Toc514935372)

[2.2.3. Send hand gestures 8](#_Toc514935373)

[2.2.4. Make sketch 8](#_Toc514935374)

[2.2.5. Send sketch 8](#_Toc514935375)

[2.2.6. View sketch 8](#_Toc514935376)

[2.2.8. Object Measuring 8](#_Toc514935377)

[2.3. User Characteristics 9](#_Toc514935378)

[2.4. Operating Environment 10](#_Toc514935379)

[2.5. Constraint 10](#_Toc514935380)

[2.6. Assumptions and Dependencies 10](#_Toc514935381)

[3. Specific Requirements 11](#_Toc514935382)

[3.1. External Interface Requirements 11](#_Toc514935383)

[3.2. User Interfaces 11](#_Toc514935384)

[3.3. Hardware Interfaces 13](#_Toc514935385)

[3.4. System Interfaces 13](#_Toc514935386)

[3.5. Communication Interfaces 13](#_Toc514935387)

[3.6. Functional Requirements 14](#_Toc514935388)

[3.7. Use Cases 15](#_Toc514935389)

[3.7.1. Establish Communication 15](#_Toc514935390)

[3.7.2. Send Video Feed 15](#_Toc514935391)

[3.7.3. View Video Feed 16](#_Toc514935392)

[3.7.4. Make Sketch 16](#_Toc514935393)

[3.7.5. Send Sketch 17](#_Toc514935394)

[3.7.6. Dismiss Sketch 17](#_Toc514935395)

[3.7.7. Send Hand Gestures 18](#_Toc514935396)

[3.8. Performance Requirements 19](#_Toc514935397)

[4. References 20](#_Toc514935398)

# Introduction

## Purpose

The purpose of this document is to specify the requirements for the first working prototype of a portable Remote Guidance System (RGS). This document will provide the development team with a high-level perspective of the system by illustrating its interfaces, constraints and the functional requirements. This document is intended to be a reference point for the development during the prototyping phase of the system.

## Project Scope

The Smart Glass Remote Guidance System will allow first-person interaction between instructors and operators. The system is used in combination with the Vuzix M100 Smart Glass for hands free communication between users.

Instructors can provide remote demonstrations using voice and hand gestures, which will be recorded from the first-person perspective of the instructor and live streamed onto the operator’s device. The operator simply needs to imitate the instructor’s hand gestures to complete a specific task. The operator can provide feedback to the instructor using hand sketches and object recognition, which can be used for later analysis and discussion.

The system will include a server to facilitate wireless communication between the users.

The function of object measurement for dimension and colours, is considered low priority and will be implemented in a later stage of the project.

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| RGS | Remote Guidance System |
| Smartglass | Vuzix M100 Smart Glasses |
| Instructor | User whom are giving instructions |
| Operator | User whom are receiving instructions |
| Server | The communication processor between the devices, providing communication processing and wifi hotspot for devices |
| Device | Vuzix M100 Smart Glasses |
| RTMP | Real Time Messaging Protocol, real time messaging protocol for carrying audio, video and data cross networks |
| RTP | Real-Time Transport Protocol, used by android devices to transmit real-time data from/to a external network |
| TCP | Transmission Control Protocol |
| WLAN | Wireless Local Area Network |

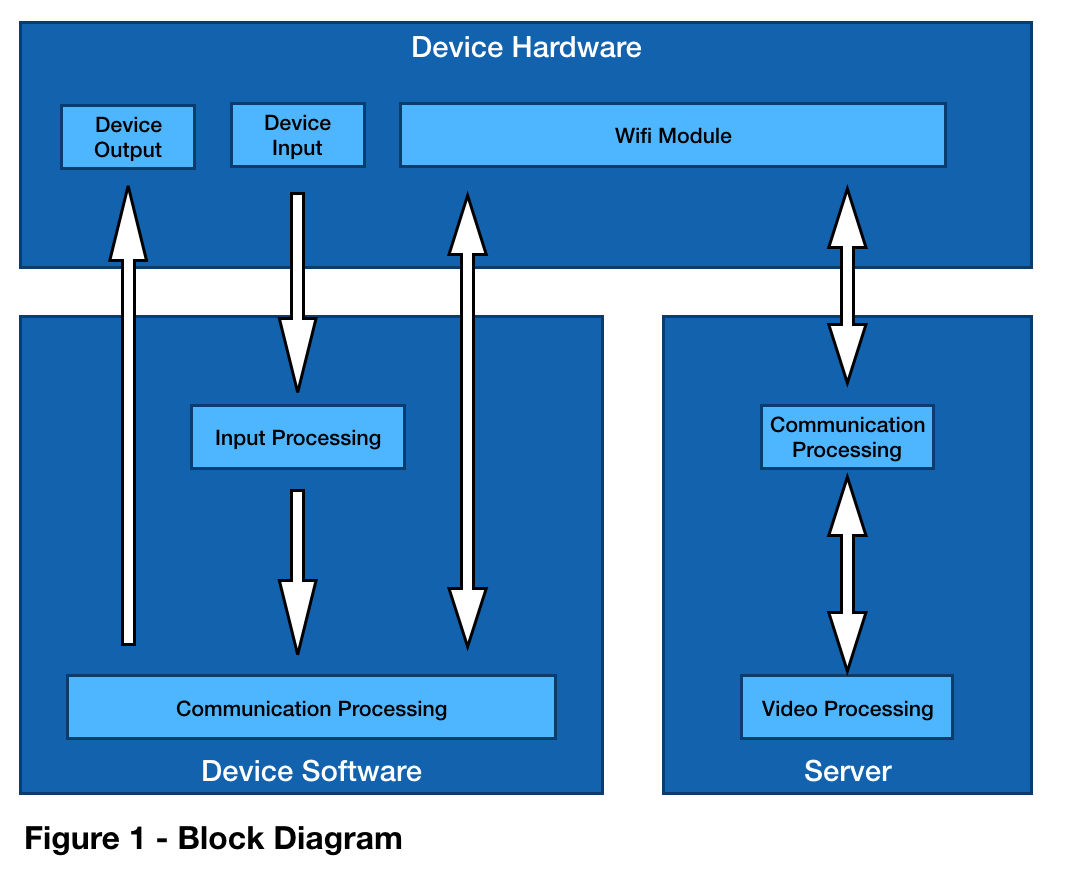
## 

# Overall Description

## Product Perspective

The system consists of the device hardware, the device software and the server. The device software will be installed onto the Vuzix M100 Smart Glass (The device hardware). The device hardware will be used as the input and the output of the device software while the device’s software will be managing the communications to and from the server. The server will be the central processing unit of the system that will manage the communication between the devices as well as video processing, see Figure 1 - Block Diagram.

User inputs(controls) and video data from device hardware captured by the device’s camera, gesture sensor and microphone, its near-eye display and speaker will be used for video and audio output.



The video data streamed into the server will be processed base on the current user input(control). Processed video data will be streamed to both users to assure synchronisation between users. The server will be limited to two concurrent connections at any given time for the purpose of development and testing.

## Product Functions

### Send video feed

This function allows the user to send a video feed to the connected user, the video feed that is currently being viewed will be stopped. This function will be automatically triggered when the users open the application.

### View video feed

This function allows the user to view the video feed from the connected user. This function will be automatically triggered when the users open the application and when the video feed is available.

### Send hand gestures

This function allows “instructor” users to send hand gestures captured by the device’s camera to the connected user. This function will replace the “Send video feed” function, causing the hand gestures video feed to be sent in place of the ordinary video feed. This function will be automatically triggered for the instructor when the application starts.

### Make sketch

This function allows “instructor” users to capture an image using the device’s camera and sketch onto the image using their finger.

### Send sketch

This function allows “instructor” users to send a sketch to the connected user. This function is only available after selecting the “Make sketch” function.

### View sketch

This function allows “operator” users to view the sketch received from the connected user. This function will be triggered when the “operator” receives sketch from the instructor.

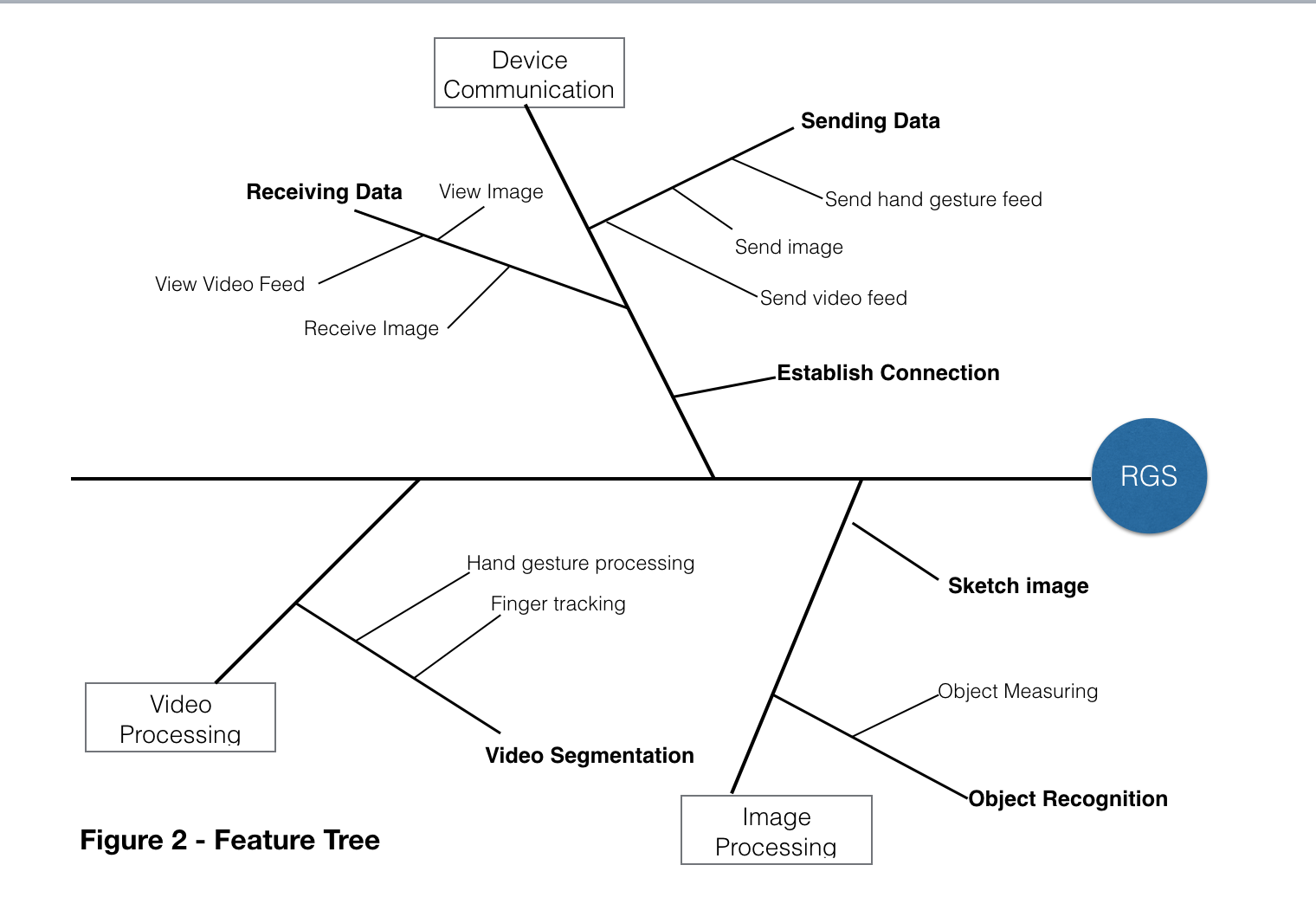
* + 1. Dismiss Sketch

This function allows the “operator” users to dismiss a sketch after viewing.

### Object Measuring

This function allows the “operator” user to measure objects that are captured by the camera. The object dimensions and colour(s) will be shown to both users.

The following feature-tree diagram (Figure 2) shows the dependencies between major features and its sub-features that are required to implement the above functions.



## User Characteristics

The system will support two types of users, instructors and operators. Functions are limited to each user type.

Instructor can send hand gestures, view video streams from the operator. The instructor can also capture images from the stream and make sketches using his/her finger(s). Video streamed will be displayed on his/her near-eye display as well as sketches when available. The operator will be able to dismiss a sketch by waving his/her hand.

Both users will be able to broadcast video streams to the other user, however only one user is allowed to broadcast video at any given time, and any existing broadcasting stream will be canceled when a new broadcast stream is initiated.

Considering the possibility that the operator may lack the training and knowledge of using the system. Furthermore, the operator is expected to be operating the system while performing other tasks, where user interactions between the operator and the system are reduced to a minimum. On the other hand, the instructor is most likely to be situated in a safe environment where active interactions with the system will not interfere with his/her safety. For these reasons, operator functions are designed to be passive and minimal while communications and interactions will most be initiated and conducted by the instructor.

The operator will not be able and are not required to initiate any functions from his/her device. Operator’s device will be automatically connected to the instructor as soon as a connection is available. The only action the operator is required to perform is to turn on the device.

The user interface should be easy to use without training. Interface control for the operator is reduced to a minimum, and the controls from the instructor will be handle using device buttons. The two devices will be configured to instructor and operator mode respectively.

## Operating Environment

RGS operates with the following environments

* Operating System: Android
* Programming Languages: Java, C++, Python, JavaScript
* Database: SQL/NoSQL
* Network: WLAN

## Constraint

The onboard computer of the smart glass currently operates with Android 4.0.4 Icecream Sandwich. Considering the hardware was released in 2009, the performance of the system may be impacted once it is updated to a higher version of Android. The development of the RGS will be using components and open source libraries that are compatible and available to Android 4.0.

A feature requested by the client is to measure object dimensions and colour. The smart glass provided by the client is equipped with a single built-in camera, but while there are different methods of measuring object dimensions and distance using a single camera, the overall accuracy and reliability of these methods are greatly affected by external factors such as lighting conditions and movements. Therefore, a single camera is not sufficient to achieve this goal. A external tool such as a ruler and color palettes would be required to serve as a point of reference for measuring.

The client has requested that the system should able to operate without internet connections. A local Wi-Fi hotspot will be used to facilitate communication between the two devices. The hotspot will also provide necessary messaging protocols such as RTMP and TCP to support video communication between the devices as well as to serve as the video processing unit for the system. Neural network models are necessary for the object recognition feature of the system, network models will be trained on an external device and then transferred onto the devices to eliminate the need of internet connection. The accuracy of this feature is limited by the quality of the initial trained mode and its training data, the “local” trained models will not be trained with new data due to limited computational power.

## Assumptions and Dependencies

The performance of the system is highly dependent on device hardware. The device was released in 2009 and operates on Android 4.0.4. The choice of libraries is dependent on their compatibilities with the operating system. We are assuming that the system performance will decrease when upgraded to higher version of Android.

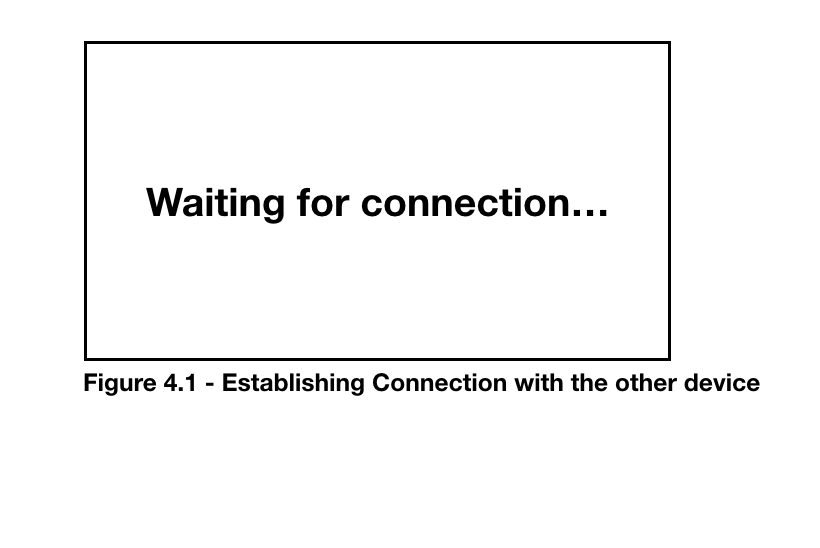
# Specific Requirements

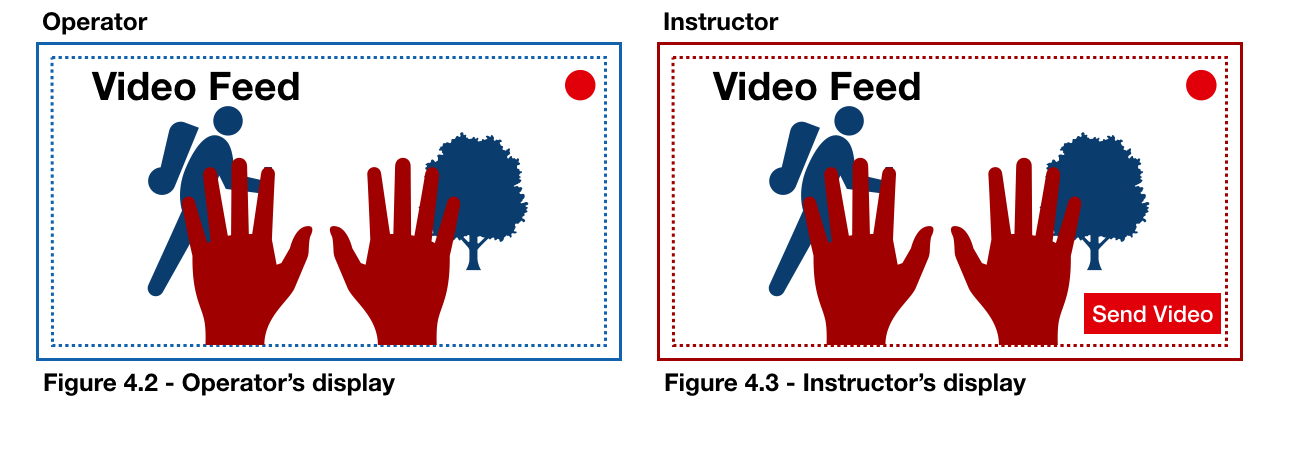
## External Interface Requirements

Both devices must be connected to the server through WIFI to enable cross device communication. Both devices will be connected to the same WLAN, and communication will be achieved using a combination of Nginx, Socket IO and Node.js on the server.

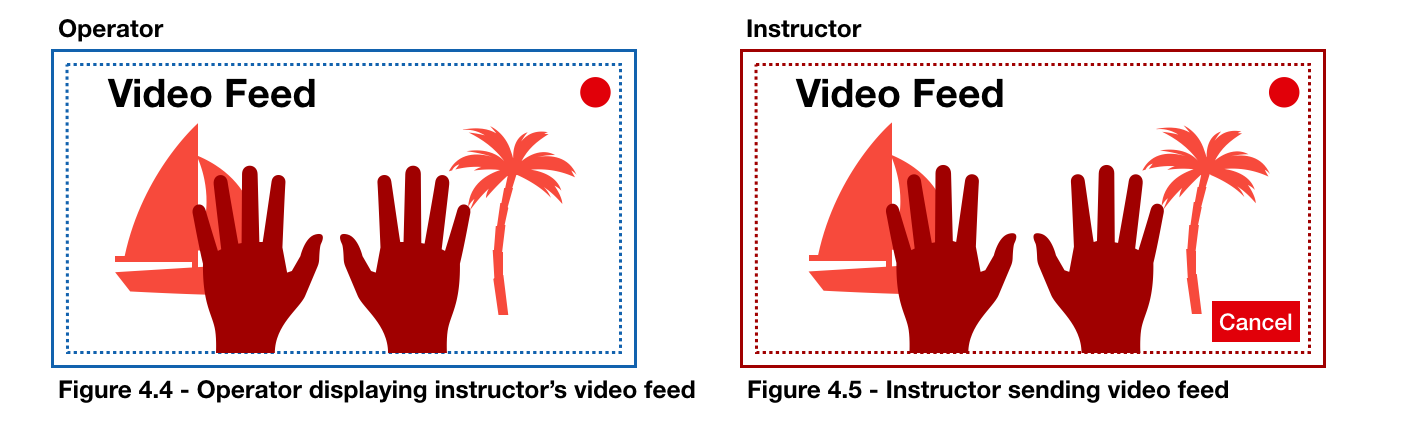
## User Interfaces

The two devices will be configured to operate as Instructor and Operator respectively. When the users have turned on the device and the application, the application will attempt to establish a connection with the other device, see Figure 4.1.

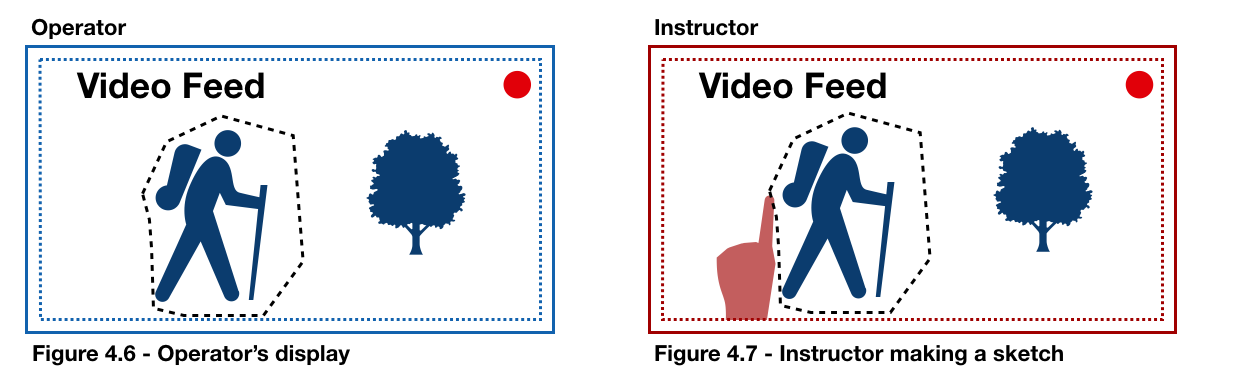


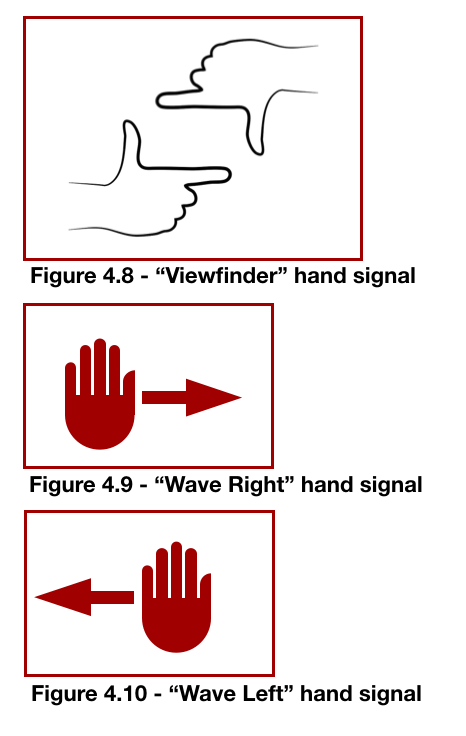
****

Once a connection has been established between the two devices. Operator’s device will begin capturing video data and broadcasted on the both devices, see Figure 4.2 and Figure 4.3. The video feed will be a combination of the operator’s captured video (blue) and the instructor’s hands (red), both users should be able to see the same video feed. A red dot will be shown on both user’s display to indicate communication is currently in progress.



By click “send video” (see Figure 4.3), the instructor will be able to send his/her video feed (red) to the operator. The video feed will be displayed on both user’s devices, see Figure 4.4 and Figure 4.5.





Instructor can capture a screenshot of the current video feed by performing a hand signal shown in Figure 4.8. The captured image will be displayed on the instructor’s device. The instructor is then able to sketch on the image with his/her finger, see Figure 4.7. The instructor can send the sketch to the operator by performing the hand signal “wave right” shown in Figure 4.9 or discard the sketch by performing the “wave left” hand signal, shown in Figure 4.10.

The operator’s display will be shown the sketch from the instructor. Once the operator has viewed the sketch, the sketch can be dismissed by performing the “wave left” or “wave right” hand signal.

Once the operator has dismissed the sketch, the display will return to the video feed. The instructor will also be returned to the video feed once the sketch has been sent or dismissed.

## Hardware Interfaces

The Vuzix M100 Smart Glass is equipped with a series of sensors and a camera. The system will use its camera for video capturing. Video streams will be displayed on its near-eye display. Wireless communication between the Smart Glass and the server will be transmitted using its built-in Wi-Fi module. The server will be hosted locally on a laptop capable of sending and receiving wireless traffic from multiple devices as well as video processing.

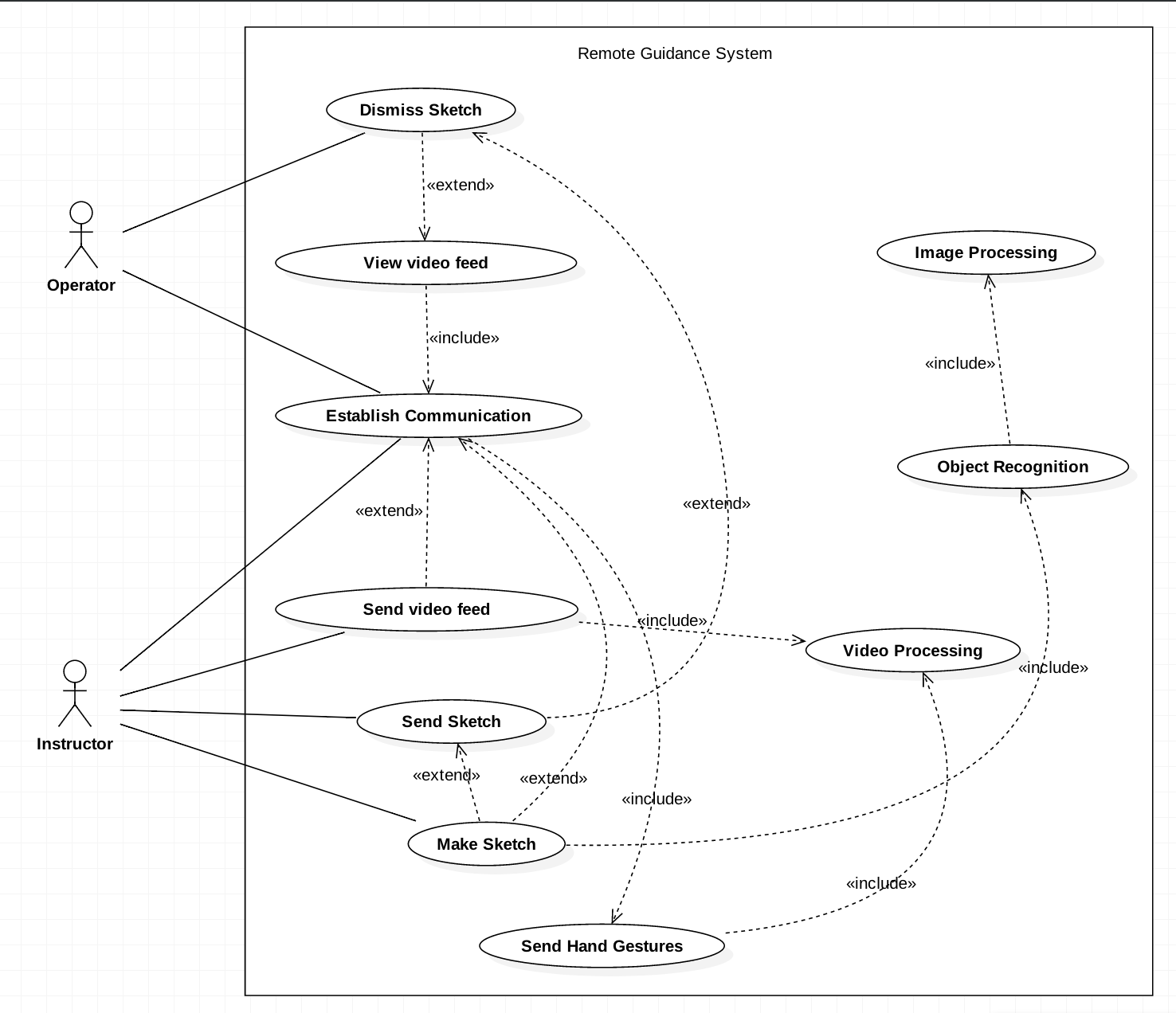
## System Interfaces

Input and video data captured by the device’s software and hardware will be transmitted to the server. Video, audio and input (user commands) data will be processed on the server based on user commands. Video processing methods include object recognition, hand tracking, video segmentation. The processed data will be transformed into video data and broadcasted(stream) to the users. Streaming on the devices will be utilising HLS (Http Live Streaming), which will be coupled with Real Time Protocols on the server.

## Communication Interfaces

The devices will be communicating through a WLAN network hosted by the server. Video communication will be supported by HLS and RTP. Video communications will be routed to the server, which will be broadcasted to the other device.

## Functional Requirements



## Use Cases

### Establish Communication

|  |  |
| --- | --- |
| **Use case name** | Establish Communication |
| **Actor(s)** | Instructor, Operator |
| **Precondition(s)** | None |
| **Trigger** | User turns on the device, the system establish connection with the other device |
| **Procedure** | 1. User turns on the device 2. System attempts to establish a connection with the other device |
| **Extensions** | 2a. System displays a “loading…” message while waiting for a connection to be established  2b. System successfully established a connection with the other device |
| **Post Condition(s)** | Users can communicate  Connection between devices established |
| **Exception(s)** | Device cannot be powered off  Device is out of battery  Network not available  Application crashed |

### Send Video Feed

|  |  |
| --- | --- |
| **Use case name** | Send video feed |
| **Actor(s)** | Instructor, Operator |
| **Precondition(s)** | Device is switched on  Connection between devices established |
| **Trigger** | Instructor clicks “send video” |
| **Procedure** | 1. System captures video feed from the instructor’s device 2. System streams video feed to both users |
| **Extensions** | This function is automatically triggered on the operator’s device once a connection has been established between the devices. In which case the operator’s video feed will be streamed to both users. |
| **Post Condition(s)** | Both users are shown the instructor’s video feed captured by the instructor’s device |
| **Exception(s)** | Device is powered off  Application crashed  User has selected a role |

### View Video Feed

|  |  |
| --- | --- |
| **Use case name** | View video feed |
| **Actor(s)** | Instructor, Operator |
| **Precondition(s)** | Connection between devices established |
| **Trigger** | Triggered when the connection between instructor and operator has been established |
| **Procedure** | 1. System successfully establish connection between the two users 2. System captures and stream video data from operator’s device to server 3. System broadcast video to both users |
| **Post Condition(s)** | Both users can see video captured by the operator on their device’s display |
| **Exception(s)** | Device is powered off  Network crashed  Application crashed  User disconnected |

### Make Sketch

|  |  |
| --- | --- |
| **Use case name** | Make sketch |
| **Actor(s)** | Instructor |
| **Precondition(s)** | Connection between devices established  Video stream is active |
| **Trigger** | Instructor performs a “screenshot” hand signal in front of his/her device camera |
| **Procedure** | 1. System captures an image from the video stream 2. System displays the image on the instructor’s display 3. Instructor draws on the image using his/her finger 4. System captures the paths drawn by the instructor and overlays the path on top of the image 5. Instructor is shown the image with draw path |
| **Post Condition(s)** | Instructor is shown the sketch with paths they have drawn  Instructor can send the sketch  Instructor can discard the sketch |
| **Exception(s)** | Device is powered off  Network crashed  Connection failure  Application crashed |

### 

### Send Sketch

|  |  |
| --- | --- |
| **Use case name** | Send Sketch |
| **Actor(s)** | Instructor |
| **Precondition(s)** | Device is switched on  Instructor has created a sketch in “make sketch” function |
| **Trigger** | Instructor performs a “wave forward” hand signal to send sketch |
| **Procedure** | 1. System sends instructor’s sketch to the operator 2. System replaces the video stream on the operator’s display with the sketch |
| **Post Condition(s)** | Instructor’s display resumes to video stream capture by operator’s device  The sketch is shown on the operator’s display  Operator can choose to dismiss the image |
| **Exception(s)** | Device is powered off  Application crashed  Network crashed |

### Dismiss Sketch

|  |  |
| --- | --- |
| **Use case name** | Dismiss Sketch |
| **Actor(s)** | Operator |
| **Precondition(s)** | Device is switched on  Network is available  Operator has received the sketch |
| **Trigger** | Operator performs a “wave away” hand signal to dismiss the sketch |
| **Procedure** | 1. System removes the sketch from the operator’s display |
| **Post Condition(s)** | Operator’s display resumes to video stream capture by his/her device |
| **Exception(s)** | Device is powered off  Application crashed  Network crashed |

### Send Hand Gestures

|  |  |
| --- | --- |
| **Use case name** | Send hand gestures |
| **Actor(s)** | Instructor |
| **Precondition(s)** | User is connected with another user  Device is switched on  Network is available  Connection is established |
| **Trigger** | Once connection has been established between the two devices, this function will be triggered automatically |
| **Procedure** | 1. System captures video data from instructor’s device 2. System extracts instructor’s hands from the video 3. System overlays instructor’s hands on top of operator’s video stream 4. System streams the video (operator + instructor’s hand) to both users |
| **Post Condition(s)** | Both users can see the video of on their devices |
| **Exception(s)** | Device is powered off  User(s) disconnected  Application crashed  Network crashed |

## Performance Requirements

* The network delay between the clients and the server should be less than 100ms
* The time delay of hand gestures shown on the operator’s display and the current(real-time) hand gestures of the instructor should be less than 500ms
* The system should be able to process minimum 10 frames per second from the camera when extracting hand gestures
* The server should be able to support 2 users at any given time

# References

* "IEEE Recommended Practice for Software Requirements Specifications," *IEEE Std 830-1998*, pp. 1-40, 1998.