SWE40001 Software Engineering Project

Remote Guidance System

Software Requirements Specification

**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 |

**Table 2. Document Sign Off**

|  |  |  |
| --- | --- | --- |
| **Name (Position)** | **Signature** | **Date** |
| Dr. Tony Huang |  |  |
| Lyndon Prado |  |  |
| Tingcong Jimmy Li |  |  |
| Keagan Foster |  |  |
| Ayub Khan |  |  |
| Dineth Gunawardena |  |  |
| Kosala Edirisinghe |  |  |
| Krishna Adhikari |  |  |
| Liam Pan |  |  |
| Migara Gunarathne |  |  |
| Shenal Nirushka |  |  |

# Table of Contents

[**Introduction**](#_qli7055ptcma) **4**

[Purpose](#_1oridmbhgxio) 4

[Project Scope](#_nogwuw58dech) 4

[Definitions, acronyms, and abbreviations](#_90r5plbcz3c9) 5

[**Overall Description**](#_56l1jcdsy3v0) **5**

[Product Perspective](#_5iusablriw8b) 5

[Product Functions](#_c6lew3womeeb) 6

[Select role](#_dq0ylzpq73xk) 6

[Connect with user](#_yjhk436ib5sz) 6

[Send video feed](#_5ahv3h3t44n2) 7

[View video feed](#_1jpxyitcrumf) 7

[Send hand gestures](#_vlydeu5fb89x) 7

[Make sketch](#_li4xfips0gmn) 7

[Send sketch](#_jyg0xppi8cr) 7

[View sketch](#_xnfqxca4c5sq) 7

[Switch to Instructor](#_i2dmmoxdov55) 7

[Switch to Operator](#_fvx7k9n63nfm) 7

[Object Measuring](#_cda16ymb7q23) 7

[User Characteristics](#_5yiowejeyvi7) 8

[Operating Environment](#_2s8eyo1) 8

[Constraint](#_17dp8vu) 9

[Assumptions and Dependencies](#_3rdcrjn) 9

[**Specific Requirements**](#_5u9ihddyqfhc) **9**

[External Interface Requirements](#_2olfvfil7lg7) 9

[User Interfaces](#_fo378mjpft93) 10

[Hardware Interfaces](#_mek9veve7292) 12

[System Interfaces](#_tyh1nahpm5f6) 13

[Communication Interfaces](#_q0gwyv7q0vmo) 13

[Functional Requirements](#_j3i70btjmhsg) 13

[Use Cases](#_n5w5vnfvzbuk) 14

[Select Role](#_bnx9pui60wg2) 14

[Connect with user](#_aul4276l5n90) 14

[Send video feed](#_mdtl3m7niuc4) 15

[View video feed](#_9io5x2s309xl) 15

[Send hand gestures](#_wrakawciu5lb) 16

[Make sketch](#_40381slgnpev) 16

[Send sketch](#_rnvti0gj62dx) 17

[View sketch](#_af9rt4ux3eif) 17

[Switch to instructor](#_2tde4c2iscil) 18

[Switch to operator](#_vab1ai9q1rg) 18

[Performance Requirements](#_ky3okik93wox) 19

# 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

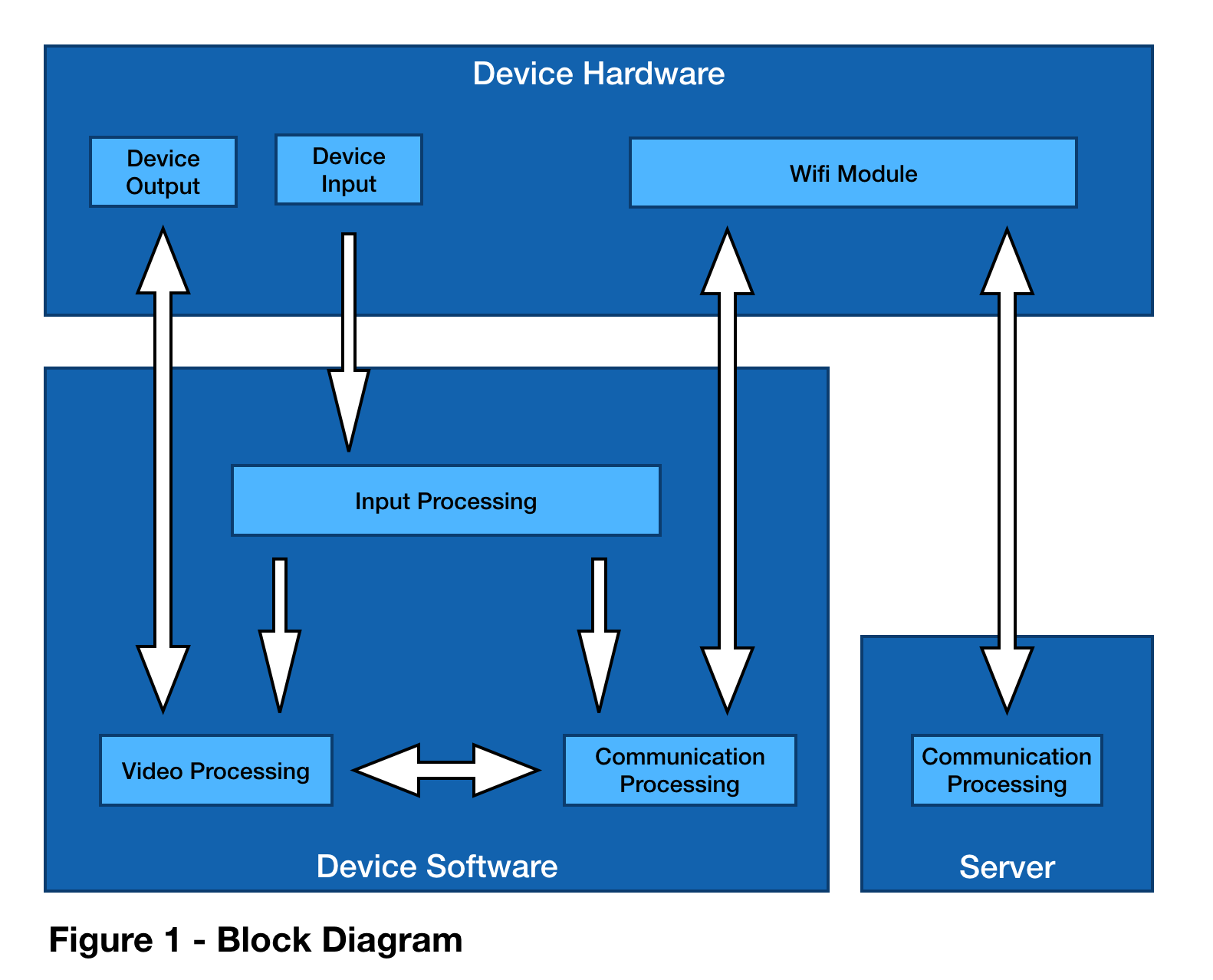
|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| RGS | Remote Guidance System |
| Smartglass | Vuzix M100 Smart Glasses |
| Instructor | User whom are giving instructions |
| Operator | User whom are receiving instructions |
| Server | The communication processor between the devices, providing 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 device’s software will also handle inputs from the device, video processing, and displaying the output onto the device hardware. The server will be the bridge between devices for both video and communications.



See Figure 1, the device software will take an input from the device hardware, which will be controlled by the user using a series of onboard sensors such as a proximity sensor, camera, gesture recognitions, and light sensors. The user will execute functions using gesture sensors on the device hardware. The camera on the device hardware will be used as the video input into the device software.

The video processed by the device software will be transmitted to and from the server through the wifi module of the hardware. The video data will output to the device’s onboard display. The server will serve as a transmission point between multiple users. The server will be limited to two concurrent connections at any given time for the purpose of testing.

## Product Functions

### Select role

This function allows users to select between two roles, “instructor” and “operator”.

### Connect with user

This function allows users to establish a connection with another user. A connection request will be presented on the screen of the second user, and once permission has been granted, the connection will be established, enabling all other functions. Users may only connect to a user with the opposite role.

### 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.

### View video feed

This function allows the user to view the video feed from the connected user.

### 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.

### Make sketch

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

### Send sketch

This function allows “operator” 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 “instructor” users to view the sketch received from the connected user.

### Switch to Instructor

This function allows the user to switch from the “operator” role to the “instructor” role. Permission from the connected user is required for the switch to occur.

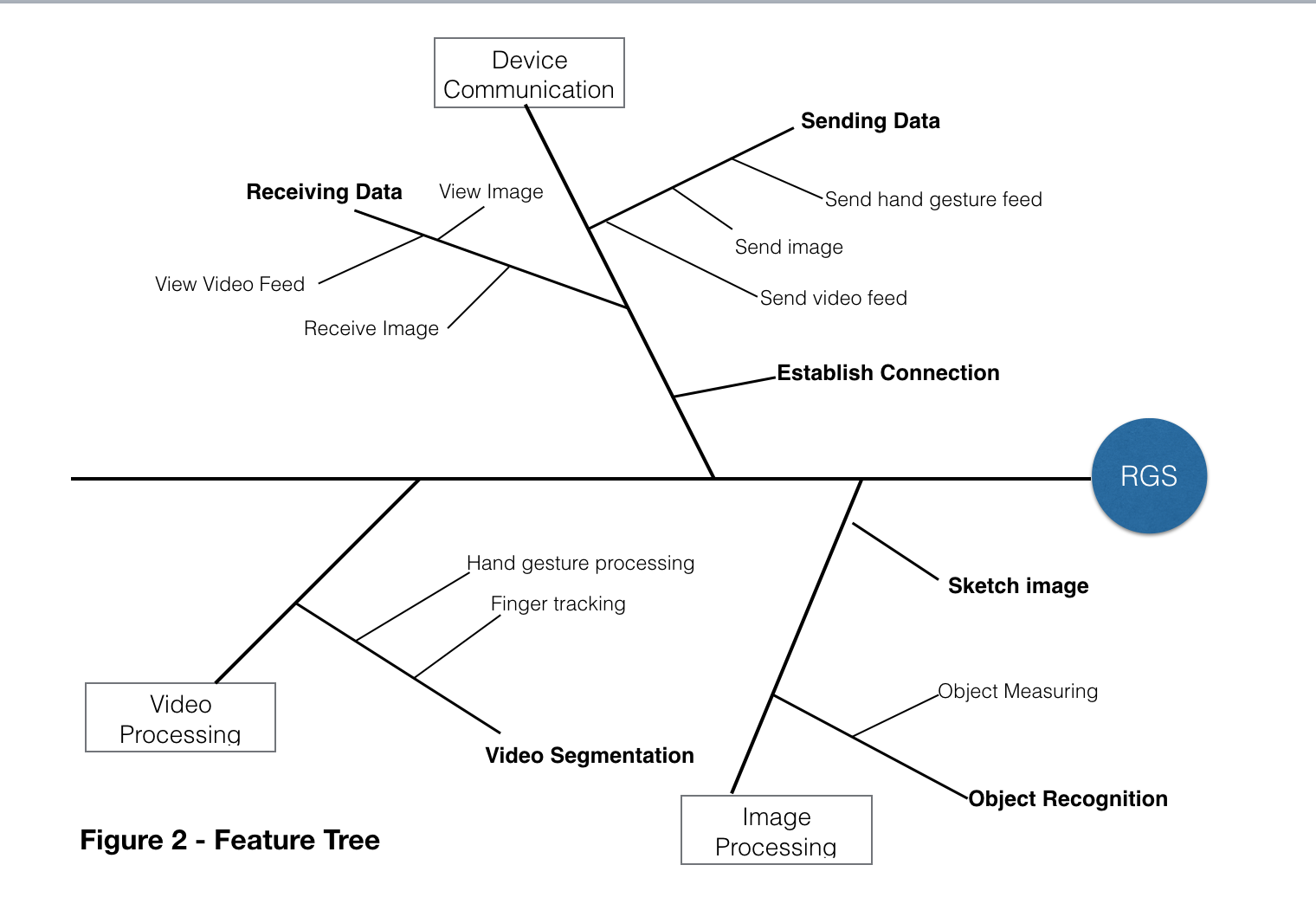
### Switch to Operator

This function allows the user to switch from the “instructor” role to the “operator” role. Permission from the connected user is required for the switch to occur.

### 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

There are two types of users, instructors and operators, users may select a user type when using the system. Usable functions are limited depending on the selected type. Instructor users are able to send hand gestures, view video streams from the operator, and view hand sketches from the operator. Operator users can make and send hand sketches.

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.

The user interface should be easy to use without training. Control of the interface should be done without physical interactions.

## 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 smartglass currently operates with Android 4.0 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 smartglass 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 wifi 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. 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 model, the “local” trained models will not be trained with new data due to limited computational power.

## Assumptions and Dependencies

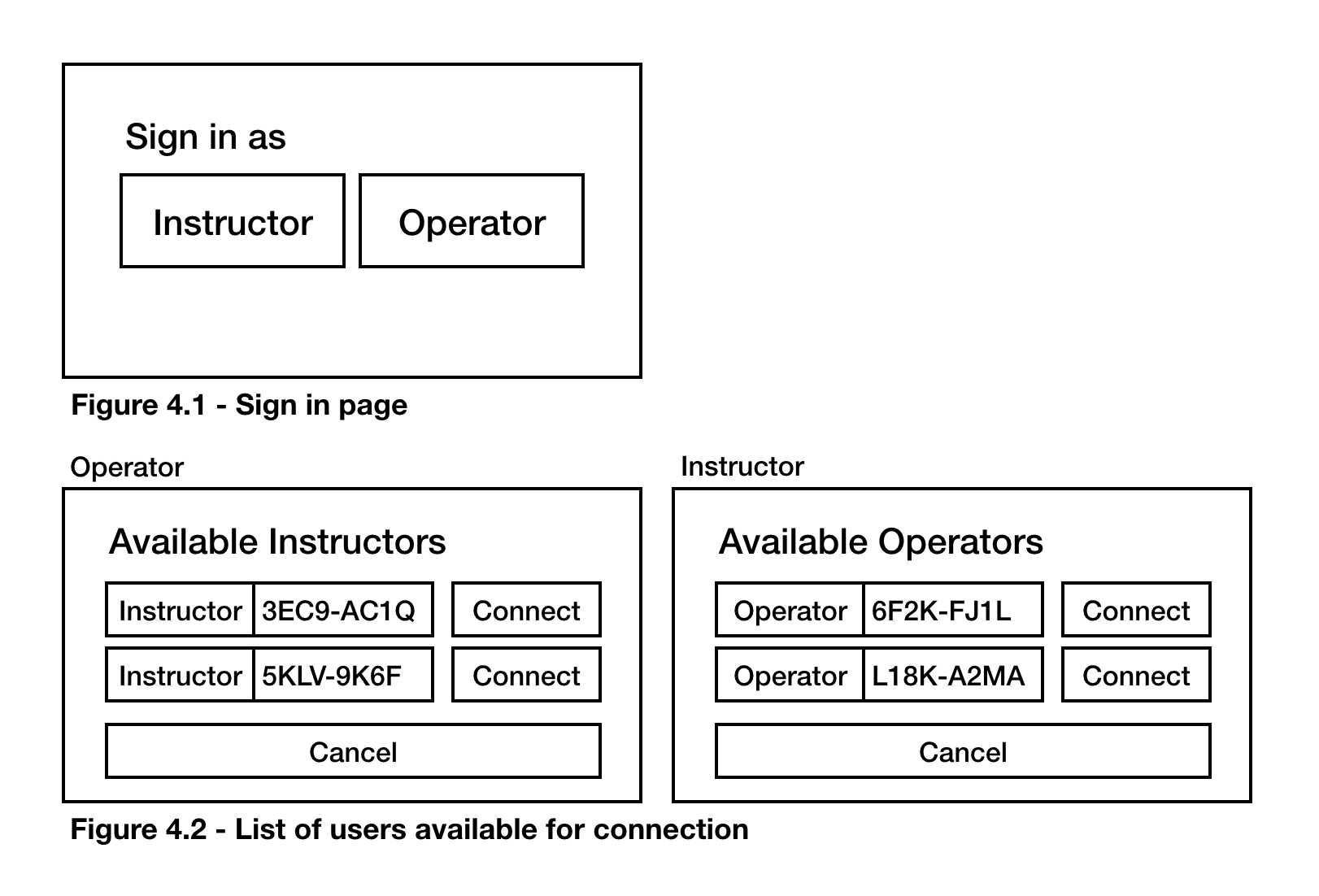
The performance of the system is highly dependent on device hardware. The device was released in 2009 and operates on Android 4.0. The choice of libraries are dependent on their compatibilities with the operating system. We are assuming that the system performance will decrease when upgraded to higher version of Android.

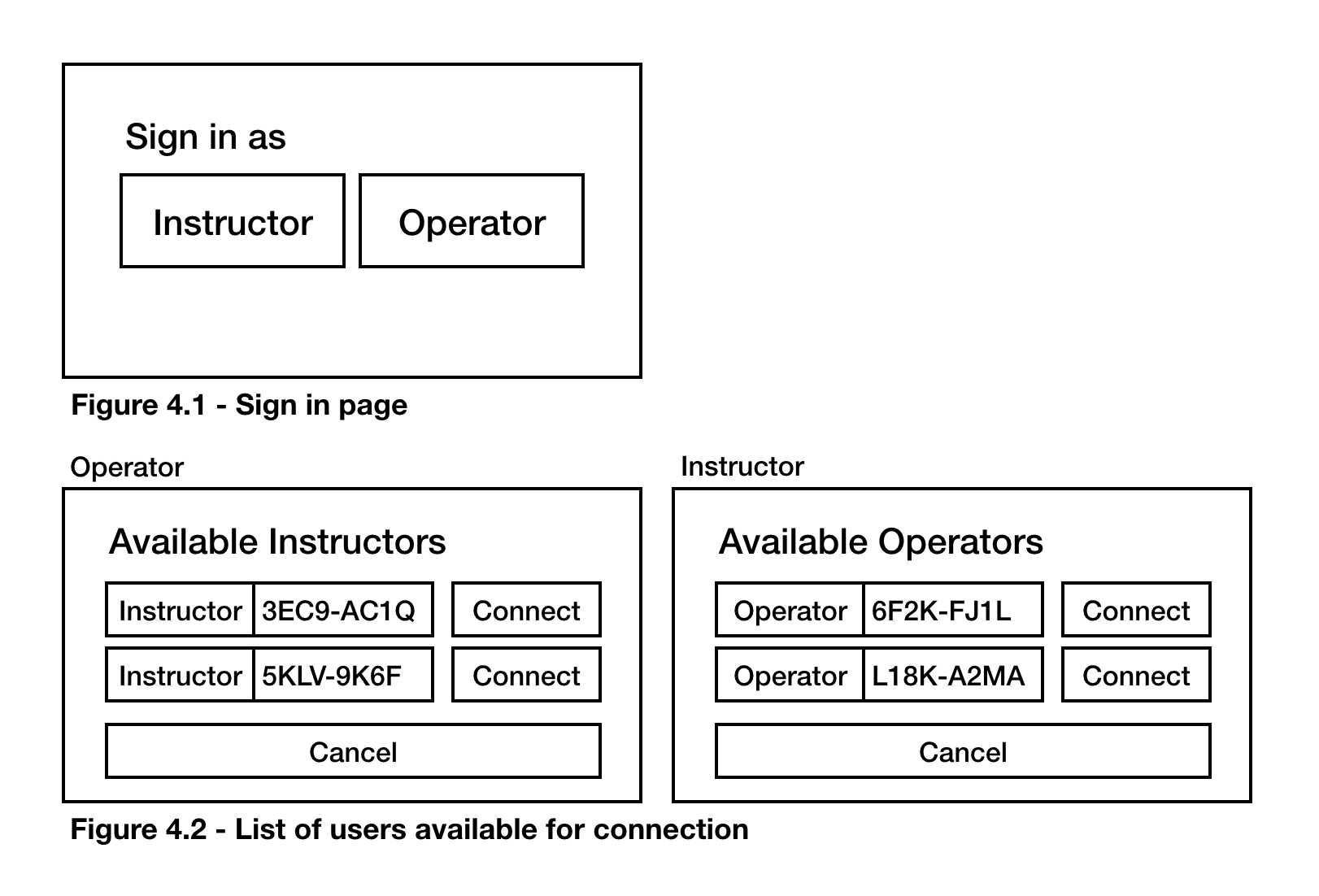
# Specific Requirements

## External Interface Requirements

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

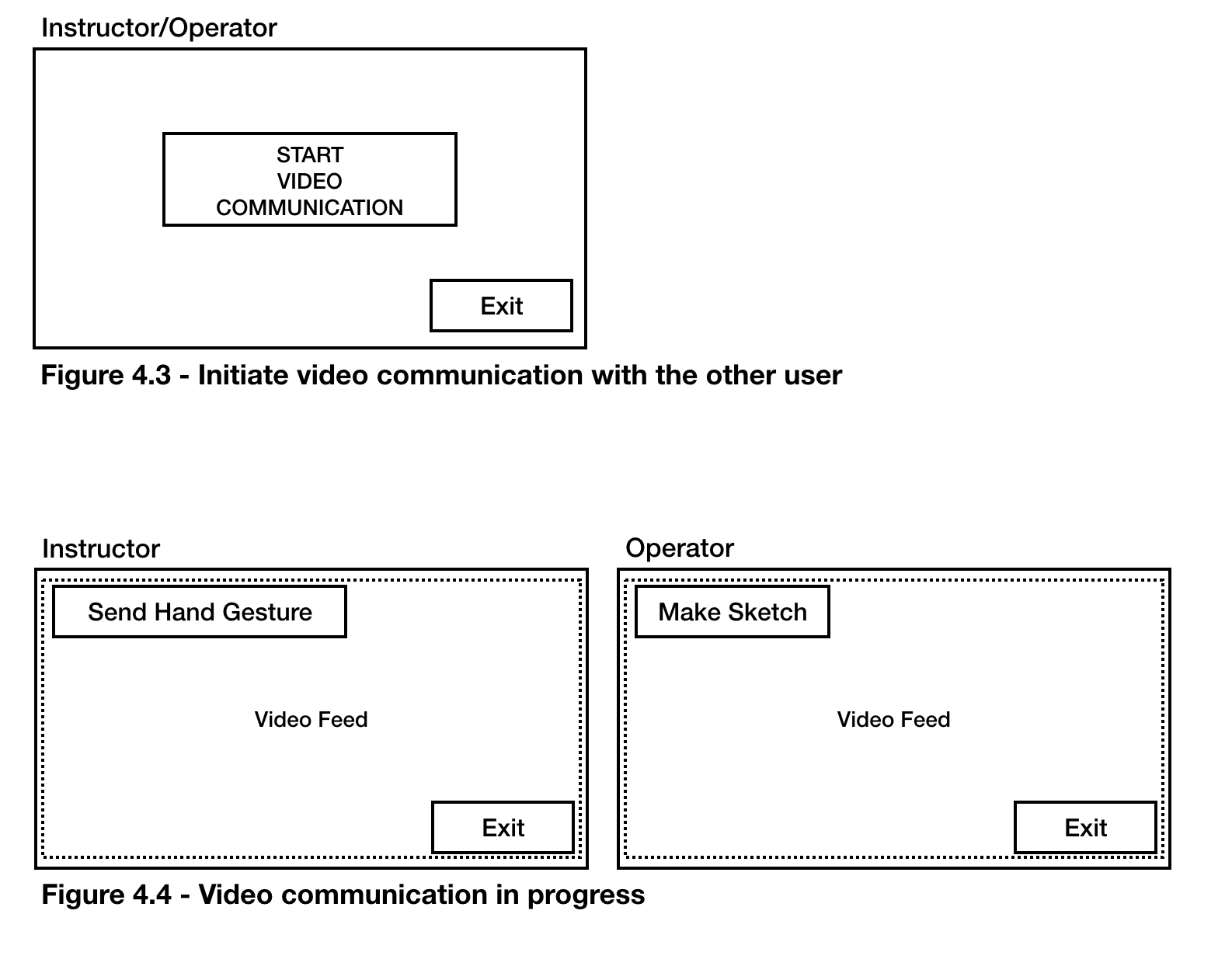
## User Interfaces

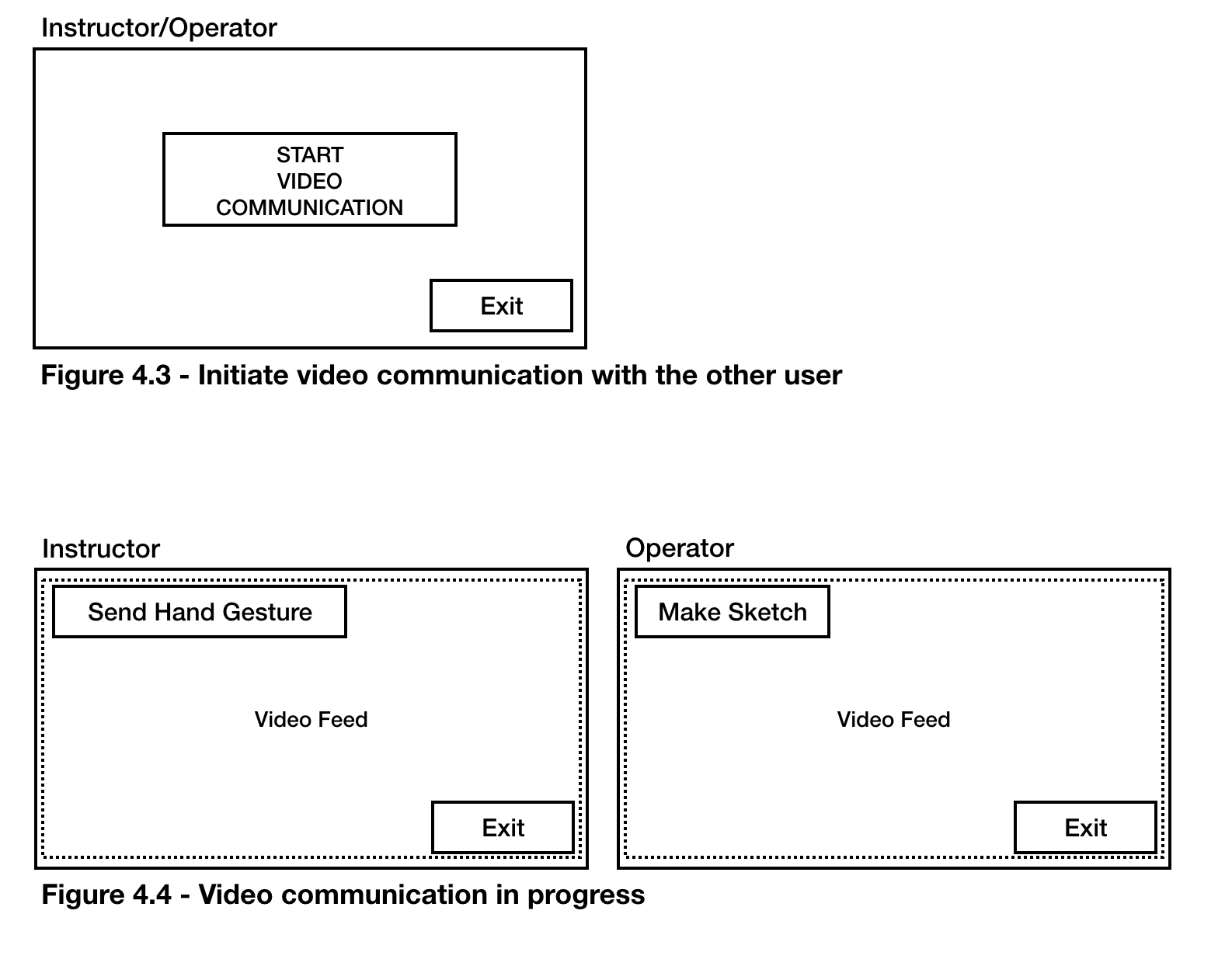




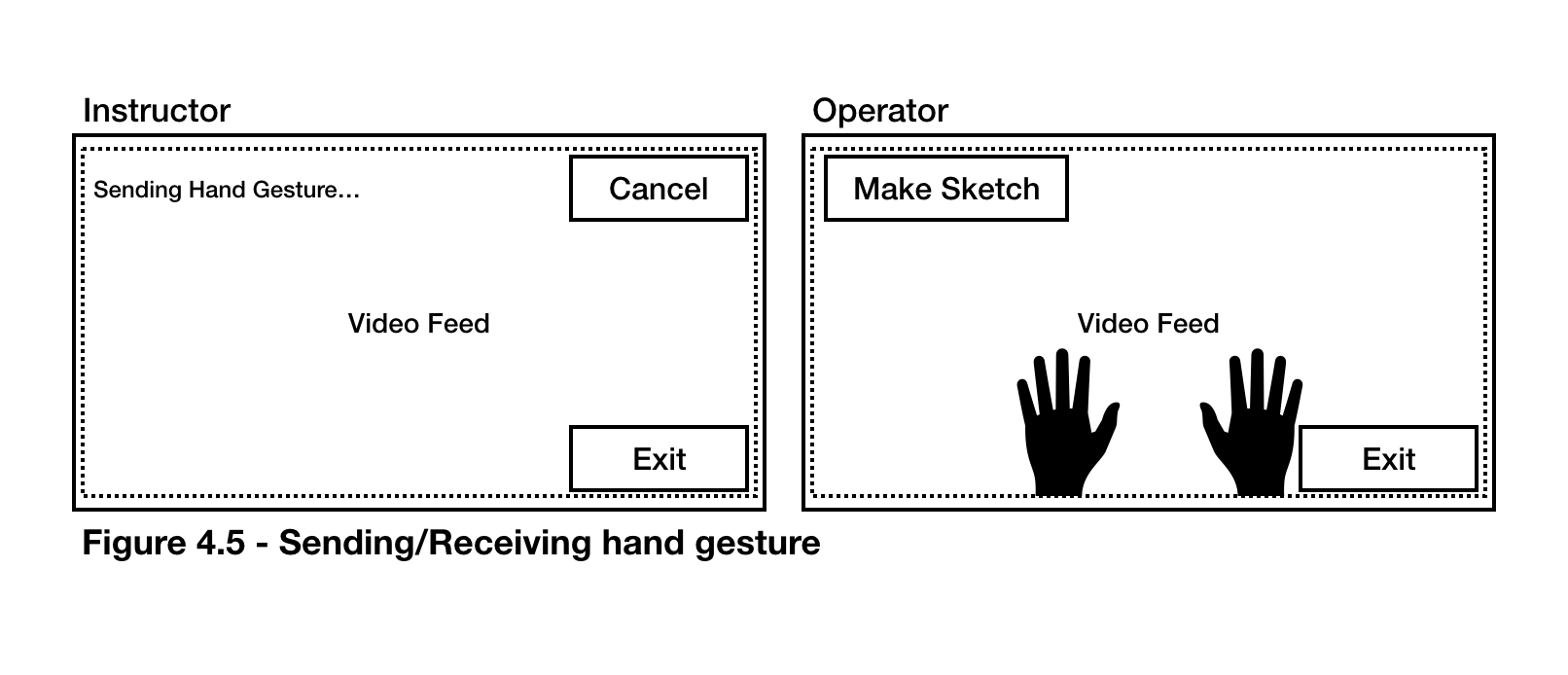
When the user opens the application, they will be prompted with the choices of role selection between Instructor or Operator, see Figure 4.1.

Once the user has selected a role, the user should see a list of users whom are available for connection, see Figure 4.2. Users will then be shown a list of users with the opposing role. The user will be able to return to the sign in screen by clicking “Cancel”. The listed users are identified by their role and a identification number, which will be generated when they sign in. The user can connect with another user by clicking “connect”.

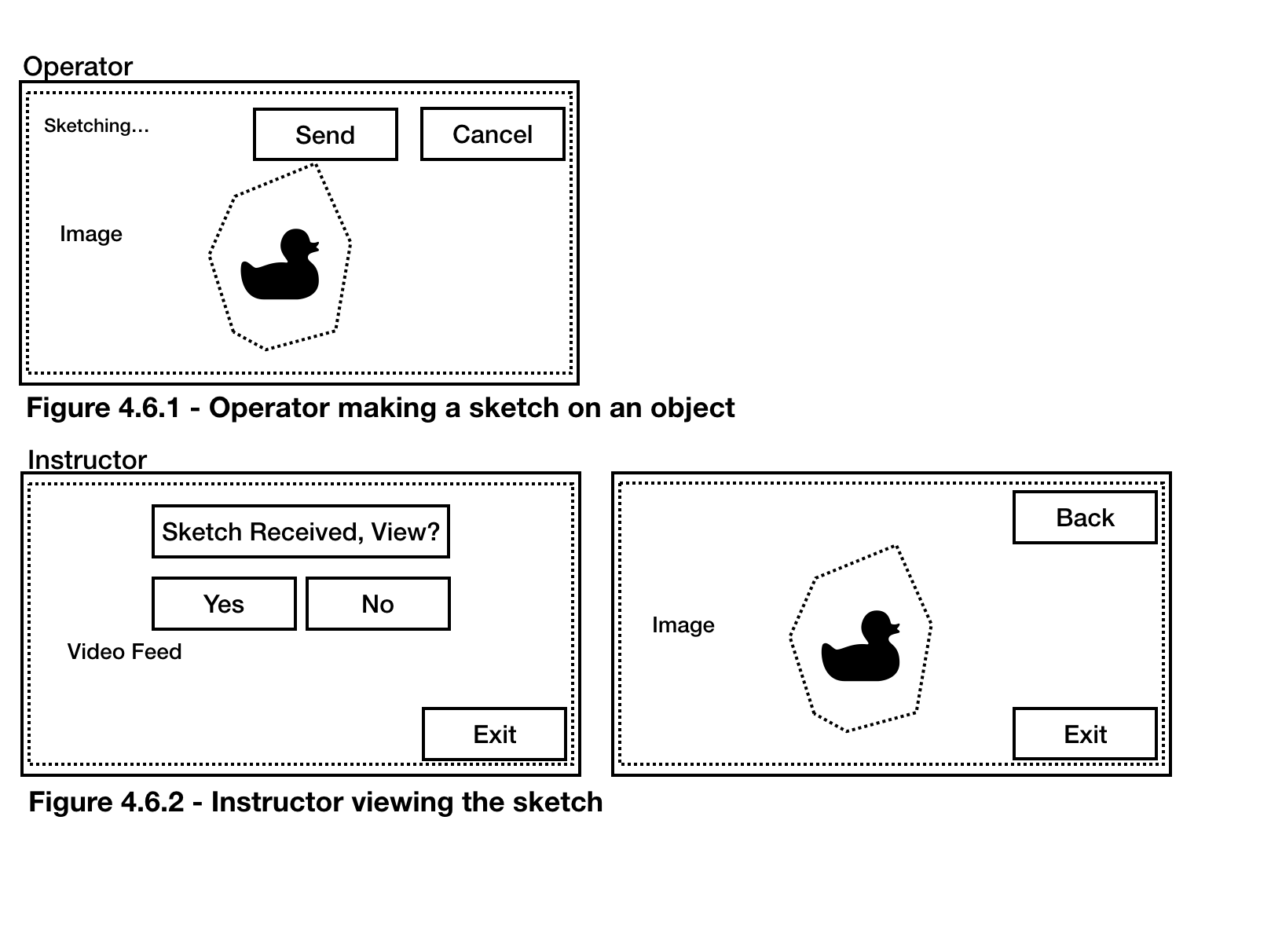


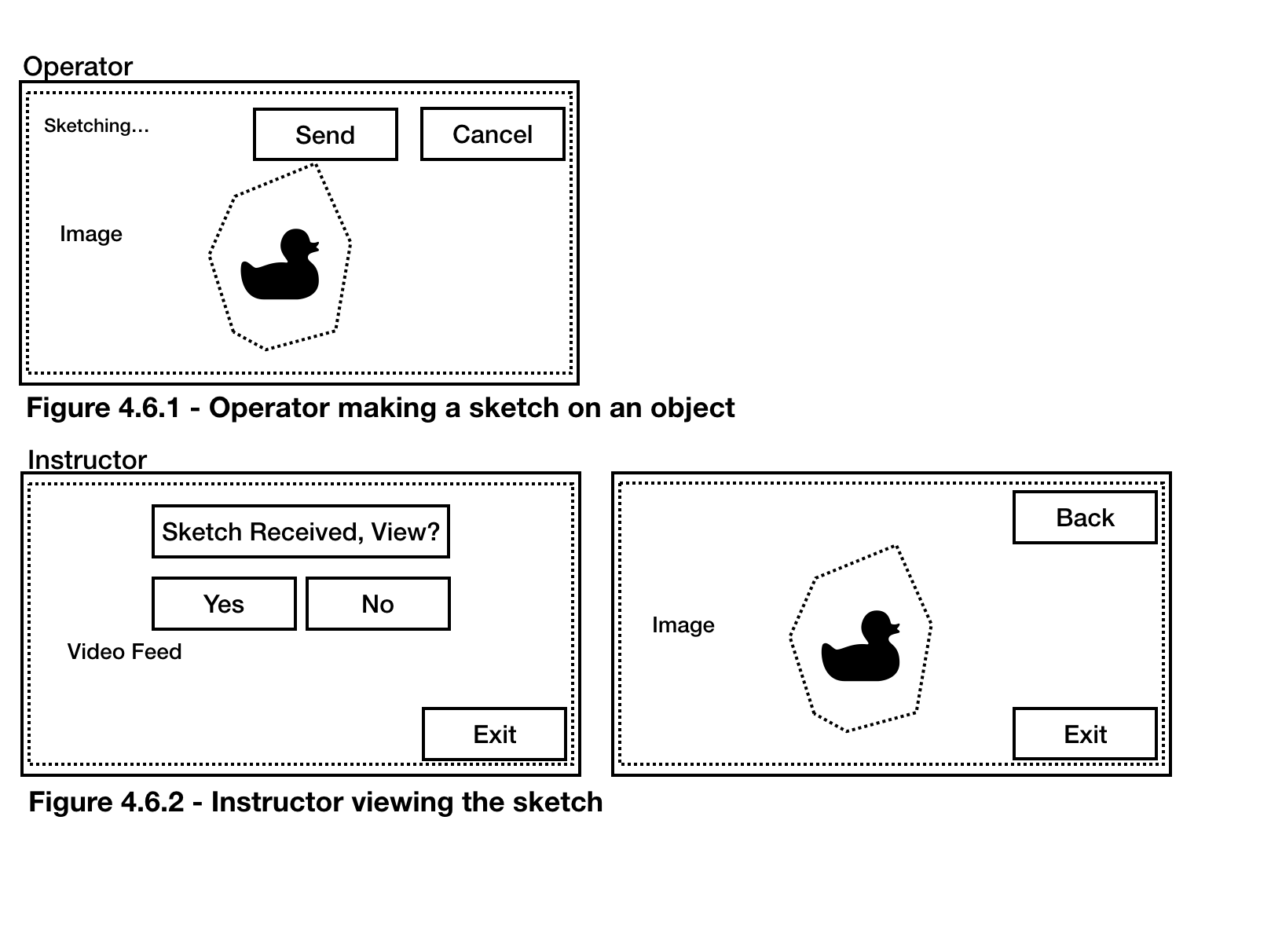


Both users, either the user initiating the connection or the user receiving the connection, will be connected and moved to the main screen, see Figure 4.3. A user will be able to initiate video communication by clicking “Start Video Communication”. The first user who initiates this action will move both users to the video communication screen, see Figure 4.4. Both users are able to see the video feed captured by the camera on the Operator’s device, and the video feed will be displayed on the main display within the dotted lines. Exclusive actions which are available to each of the users will be overlaid on top of the main display. Both users can terminate the current video communication session by clicking “Exit”, in which case both users will return to the menu in Figure 4.3.



While both users are engaged in a video communication session, the Instructor can send hand gestures by clicking the “Send hand gesture” button, shown in Figure 4.4. The instructor should be able to see that hand gestures are currently being recorded, see Figure 4.5. The hand gestures of the Instructor should be recorded by the Instructor’s device camera, once processed, the hand gestures are over-layed on top the main display of the Operator, see Figure 4.5. The instructor can cancel the operation by clicking “Cancel”, in which case they will return to normal video communication (without hand gestures).





While both users are engaged in a video communication session, the operator can create a sketch of the current camera view by clicking the “make sketch” button, shown in Figure 4.4. Once selected, the operator will no longer see their own video feed, instead an image will be taken and displayed from the time when the user clicked the “make sketch” button. The image is captured with the Operator’s camera, see Figure 4.6.1. The user can sketch on the image around an object using their finger, and the finger movements will be captured by the device’s camera and overlaid on top of the image, shown in dotted lines in Figure 4.6.1. The operator should be able to send the modified image to the instructor by clicking “Send” or cancel the operation by clicking “Cancel”, clicking either button will return the operator to the previous screen, shown in Figure 4.4.

If the operator has chosen to send the modified image, the instructor will be notified of a sketch, see Figure 4.6.2. The instructor should be able to view the image by clicking “Yes”, in which case the image will be shown on their main display. If the instructor chose not to see the image by clicking “No”, the notification will be dismissed and the instructor should be returned to the previous screen shown in Figure 4.5.

## 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. The gesture sensor (3 DOF gesture engine) will be used to handle all interactions between the user and the system. Video streaming can be displayed on its near-eye display. Wireless communication between the Smart Glass and the server will be transmitted using its built-in wifi module.

The server will be hosted locally on a laptop capable of sending and receiving wireless traffic from multiple devices.

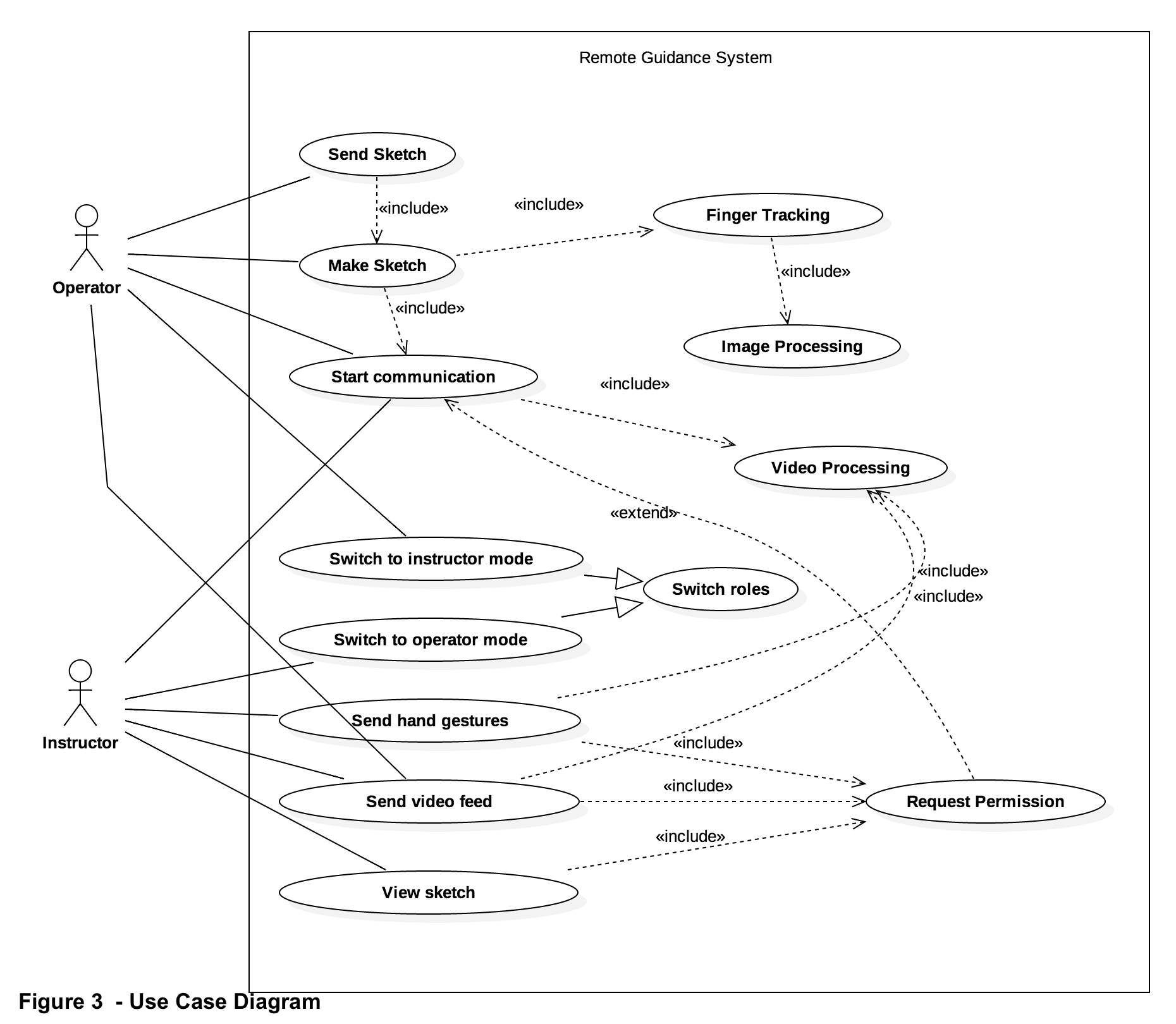
## System Interfaces

Data from the device software must be transformed into a format that is supported by the server. Video data that has been streamed into the server will be required to be processed by the device software and transformed into RTP packets prior to transmission as the server will be expecting RTP packets. On the other hand, video data that has been streamed into the device software will translate RTP packets into video data.

## Communication Interfaces

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

## Functional Requirements



## Use Cases

### Select Role

|  |  |
| --- | --- |
| **Use case name** | Select Role |
| **Actor(s)** | Any user |
| **Precondition(s)** | Device is switched on |
| **Trigger** | User opens the application, select role |
| **Procedure** | 1. User opens the application 2. User is prompted with the role selection screen 3. User selects between two roles “Instructor” and “Operator” |
| **Post Condition(s)** | User is in Operator / Instructor mode  User is waiting for connection |
| **Exception(s)** | Device is powered off  Application crashed |

### Connect with user

|  |  |
| --- | --- |
| **Use case name** | Connect with user |
| **Actor(s)** | Instructor, Operator, System |
| **Precondition(s)** | Device is switched on  User is in Operator Mode or User is in Instructor Mode |
| **Trigger** | User opens the application, select role |
| **Procedure** | 1. In the main menu, user selects “connect with user” 2. Connect with user |
| **Extensions** | 2a. If the user is in Instructor Mode, the user is displayed a list of users in Operator mode who are waiting for a connection  2b. If the user is in Operator Mode, the user is displayed a list of users in Instructor mode who are waiting for a connection |
| **Post Condition(s)** | User is in “Operator” mode  User is in “Instructor” mode |
| **Exception(s)** | Device is powered off  Application crashed |

### Send video feed

|  |  |
| --- | --- |
| **Use case name** | Send video feed |
| **Actor(s)** | Any user |
| **Precondition(s)** | Device is switched on  User has selected a role  User is connected with another user |
| **Trigger** | User opens the application, select role, connect with another user, send video feed |
| **Procedure** | 1. User selects the option to send video feed 2. System begin processing video feed captured by the user’s camera 3. System broadcast video feed to the connected user |
| **Post Condition(s)** | User enters broadcast mode  User can stop broadcasting |
| **Exception(s)** | Device is powered off  Application crashed  User has selected a role |

### View video feed

|  |  |
| --- | --- |
| **Use case name** | View video feed |
| **Actor(s)** | Any user |
| **Precondition(s)** | Video from the connected user is been broadcasted |
| **Trigger** | User opens the application, select role, connect with another user, user is notified that a video is been broadcasted, user view video feed |
| **Procedure** | 1. User selects view video 2. System gets a video feed from the server 3. System displays the video feed on the user’s display |
| **Post Condition(s)** | User is viewing video  User can stop viewing video |
| **Exception(s)** | Device is powered off  Application crashed  No video is been broadcasted by the server/other user |

### Send hand gestures

|  |  |
| --- | --- |
| **Use case name** | Send hand gestures |
| **Actor(s)** | Instructor |
| **Precondition(s)** | User is connected with another user  Device is switched on  User is in “Instructor” mode |
| **Trigger** | User opens the application, select role as instructor, select send hand gestures |
| **Procedure** | 1. User selects “send hand gestures” 2. System starts recording video using device 3. System extracts hand gestures from video 4. System sends processed video to connected user(Operator) |
| **Post Condition(s)** | User(operator) can view hand gestures(video) |
| **Exception(s)** | Device is powered off  User(s) disconnected  Application crashed  Network crashed |

### Make sketch

|  |  |
| --- | --- |
| **Use case name** | Make sketch |
| **Actor(s)** | Operator |
| **Precondition(s)** | Device is switched on  User is in operator mode |
| **Trigger** | User opens the application, select role as operator, select make sketch |
| **Procedure** | 1. User selects “make sketch” 2. System turns on the device camera, preparing to capture image 3. User selects “capture” 4. System captures image from device camera 5. User is prompted with the captured image 6. User selects “draw with finger” 7. System tracks user’s finger movements 8. System displays path of finger movements over the image |
| **Post Condition(s)** | User(operator) can now send sketch to the instructor |
| **Exception(s)** | Device is powered off  Application crashed |

### 

### Send sketch

|  |  |
| --- | --- |
| **Use case name** | Operator |
| **Actor(s)** | Any user |
| **Precondition(s)** | Device is switched on  User is in operator mode  User was in “make sketch” function |
| **Trigger** | User makes sketch, user selects send sketch |
| **Procedure** | 1. User select “send to instructor” 2. System prepares the image 3. System sends the image to connected user (instructor) |
| **Post Condition(s)** | User(instructor) can view sketch |
| **Exception(s)** | Device is powered off  Application crashed  Network crashed |

### View sketch

|  |  |
| --- | --- |
| **Use case name** | View sketch |
| **Actor(s)** | Instructor |
| **Precondition(s)** | Device is switched on  User is in Instructor mode  User(instructor) sends sketch |
| **Trigger** | User is prompted with a notification of “new sketch received”, user selects view sketch |
| **Procedure** | 1. User selects view sketch 2. System displays the image |
| **Post Condition(s)** | User can dismiss the sketch(image) |
| **Exception(s)** | Device is powered off  Application crashed  Network crashed |

### Switch to instructor

|  |  |
| --- | --- |
| **Use case name** | Switch to instructor mode |
| **Actor(s)** | Operator, Instructor |
| **Precondition(s)** | Device is switched on  User is in operator mode  Both users, instructor and operator, are connected |
| **Trigger** | User(operator) selects switch to instructor |
| **Procedure** | 1. User(operator) selects “switch to instructor mode” 2. System notifies connected user(instructor) 3. User(instructor) selects “accept” 4. System relays permission to user(operator) 5. User(operator) is switched to instructor mode 6. User(instructor) is switched to operator mode |
| **Post Condition(s)** | User(instructor) is in “Operator” mode  User(operator) is in “Instructor” mode  Users switched roles |
| **Exception(s)** | Device is powered off  Application crashed  Network crashed  Permission denied |

### Switch to operator

|  |  |
| --- | --- |
| **Use case name** | Switch to operator mode |
| **Actor(s)** | Operator, Instructor |
| **Precondition(s)** | Device is switched on  User is in operator mode  Both users, instructor and operator, are connected |
| **Trigger** | User(instructor) selects switch to operator |
| **Procedure** | 1. User(instructor) selects “switch to operator mode” 2. System notifies connected user(operator) 3. User(operator) selects “accept” 4. System relays permission to user(instructor) 5. User(operator) is switched to instructor mode 6. User(instructor) is switched to operator mode |
| **Post Condition(s)** | User(instructor) is in “Operator” mode  User(operator) is in “Instructor” mode  Users switched roles |
| **Exception(s)** | Device is powered off  Application crashed  Network crashed  Permission denied |

## 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 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