Video Based Guidance Application

Nasrulla Khan Haris, Prashant Narayan Kulkarni, Sowjanya Deenadayal Reddy, Ujwal Komarla
Department of Electronics and Computer Science
North Carolina State University
nkhanha@ncsu.edu, pkulkar4@ncsu.edu, sreddy3@ncsu.edu, uskomarl@ncsu.edu
Under the guidance of Dr Mihail L. Sichitiu for the course ECE/CSC 575

I. ABSTRACT

The increase in necessity of seeking any kind of help through our smart phones and the ever increasing dependencies on applications to help us out in various situation has given us impetus to implement the Video based Guidance Application presented in this paper. We have presented–the design and implementation of an asymmetric application that runs between two android phones over WiFi, along with the illustration on how a person (client) with the application can choose any other person (guide) with the same application and seek any kind of guidance. The client can connect to a particular guide, by choosing from a list of guides. The client can stream a video using the Camera to the guide and request guidance. The guide can draw directions or instructions on a canvas and send it to the client. The client can view the drawn instructions over the video. This application also allows you to send text messages to interact with the guide and get instructions. This paper helps us discover how the application was implemented and understand technologies used for the same. It also illustrates how the login details of the clients and guides are stored in the database.

II. INTRODUCTION

The ubiquitous nature of smart phones along with their software development tools made available to the masses has given rise to countless new ways for the everyday user to digitize their life. As such, it was only a matter of time before people started writing applications to seek customized guidance from a person they wish. Several applications have been developed to make services available at everyone's fingertips. The Video based guidance application discussed in this paper is one amongst such application.

This application helps in providing guidance to android device owner (client) who is using his device as a remote camera for a computer/smart device (guide). The guide then sends a few signals to the client. If these signals are predefined with some meanings then they can be used as perfect guidance mechanisms. This application supports two ways of guidance, using streaming video from client to guide as reference:

1. Text commands as a response from guide
2. Gestures over the video as a feedback mechanism

For an improved understanding of the application let's consider a client is lost in a place and requires guidance from person *A* among her group of friends (*A, B, C, D, E* and *F*). The client first downloads the application and so do all the people among the group. They individually register in the application and create their login details, which is in turn stored in our MySQL database. Using its registered login details a client or a guide can login to the application. The client then chooses a guide (person *A* in this case) from the list of active guides available at that point of time. Once the choice of the guide is made, a connection to a guide is established. The client then streams in the video from its phone camera to *A*. *A* receives the video stream and draws the instructions (navigation in this case) over the received Video stream. These instructions are simultaneously registered in backend service (Firebase) which in turn makes the drawn instructions appear over the client's video. Hence, the client has been guided successfully from the chosen guide.

A few of the use cases of our application could be:

1. A deaf person might require a sign guidance to assist him through a place or assist him with any other work that requires perspective guidance.

2. A network administrator can guide the lab technician/support by gesturing him to make/remove connections remotely.
3. Medical procedures (emergencies) can be guided by a doctor remotely.
4. This application can act as a guiding mechanism when established features like GPS cannot give accurate details. For example it helps in navigation inside a building.

This application can be used for any other similar cases that require remote assistance and can be extended to more advanced applications as well. This paper is organized as follows. The background check of the related work has been presented in Section 3. Section 4 presents the Model of the design in detail. We address the experimental results in Section 5. We conclude this paper along with our future work in Section 6.

III. RELATED WORK

There are several applications that allow us to stream video in real-time. One such application is "Spydroid" which helps us stream real-time videos to other phones. Android developers are making use of this app to develop exciting new apps in related field. This application does not provide features like providing guidance to the client but just helps us stream video in real time. There is an app named "Ustream" that supports broadcasting live to any number of device using camera of the device.

All the existing apps have managed to achieve live streaming but haven't supported Video based guidance. The application presented in our paper not only allows gesture guidance but supports message guidance as well.

IV. PROBLEM FORMULATIONS

Android is an open source mobile platform offering a bunch of APIs for application development. We have chosen to use this mobile platform to develop our Video based Guidance application. This section describes the functionalities of the application and how it is achieved. Our project can be subdivided into four main components:

a) Google Cloud Messaging

b) Backend Server

c) Database

d) Firebase

e) Libstream

f) Android Application

These components have helped us develop all the other features like registering a user, providing login portal so that the user can login as a client or a guide. It provides us with a platform to set up the client-server model and stream a video from the client's camera. A few of the components help in sending the gestures (guidance/instruction) from the guide to the client. These components individually contribute to the application in one or more functionalities. The layout of the project is as shown in Fig1. The functionality of each component is described as follows:

1] Google Cloud Messaging

It is a cloud service provided by google for Android that allows you to send and receive messages from the android devices. This service is used in our project to help achieve and maintain connectivity with 'mobile' guides without using the naïve polling version and then get the IP address of the guide. This connection allows us to send data from backend server to the mobile Android-powered device application which is described clearly in the below sections. We utilize the service of GCM to track of all the guides and their status, as in active or inactive. Every user has to be registered with the GCM which in turn provides them a GCM ID which is used for future communication.

2] Backend Server

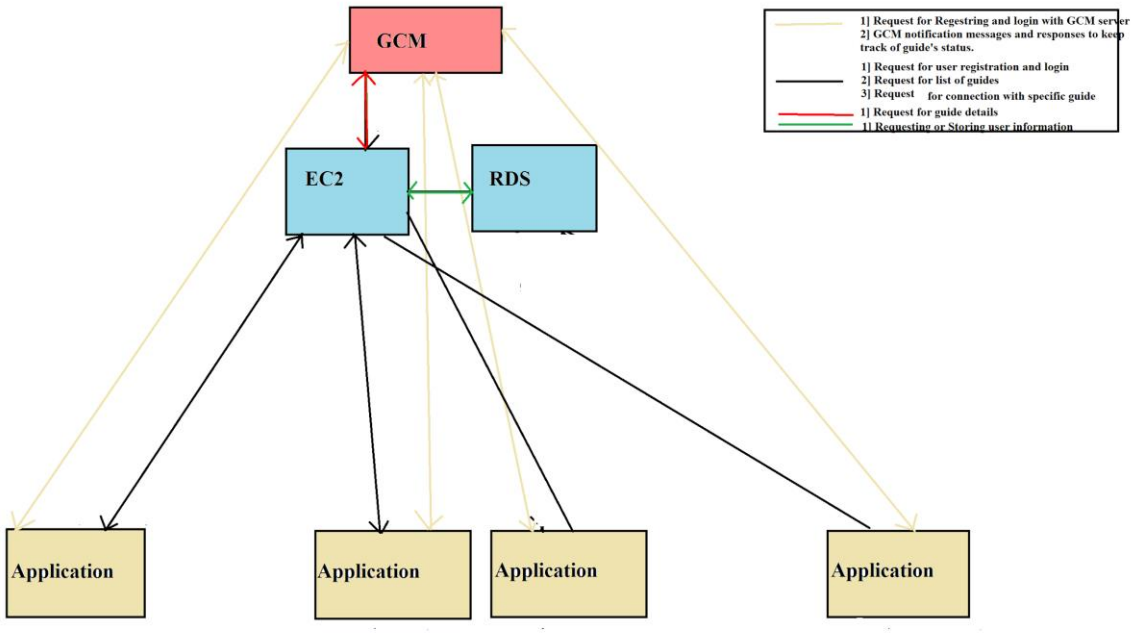

Figure 1: Layout of the Video Based Guidance Application

Our project makes use of the Amazon Web Server – Elastic Cloud Computing (AWS- EC2) to run the webserver. AWS- EC2 is a web service that provides resizable compute capacity in the cloud. The web server running on AWS-EC2 with a capability to respond to php requests. The php requests the backend server caters to are:

a) Register and login of users

During login, the GCM ID on the device is marked against the given application user by server, which is used in late stages for communication from server to particular application user. If the GCM ID has not been dispatched to the device then it is first made to register with the GCM to obtain a GCM ID and then proceeds with the login.

b) Request list of active guides

It frequently updates the list of active guides by consulting the GCM. If the request for a list of guides was very recent then it replies with the previous list else consults the GCM again and sends the updated list.

c) Connect to a guide and communicate with the same

Once the client chooses to connect to a specific guide. The backend server collects the IP address of the requested guide from the GCM by providing the GCM ID and then establishes the connection between the client and the guide.

3] Database

MySQL database using Amazon Web Server – Relational Database Server (AWS-RDS) is used to store information relating to each user, such as, USER ID, Password, GCM ID, IP address, Flag to see if Client/Guide. This is consulted for any information regarding any user.

4] Firebase

Firebase is a third-party service used to store and sync data in real time. In our project we use Firebase to sync the gestures drawn over a canvas at the guide end with the video at the client end.

5] Libstream

This governs the streaming of the video and message by using a few protocols such as RTSP and RTMP. Real Time Streaming Protocol (RTSP) is a networking protocol mainly used to stream real time media data like audio or video. In our project we use RTSP streams for streaming the video. This hence, helps us establish a streaming session between client and server. Real time Messaging Protocol is used to send a text message from the client to the guide in our project.

6] Android Application

The functionality of the complete application using the above components is described here. This application has the ability to register a new user (Client/Guide) by communicating with the backend server (Explained in Backend section). This communication is done using a JSON format request/response method. Once, a successful registration is achieved the user can login either as a Client or as a guide with communication occurring on similar lines as the registration.

During the login process, we check whether the application on this device has a registered Google Cloud Messaging (GSM) ID, if not, we do the GCM registration. We send this GCM ID to the backend server, used later for communication with the user on this particular device. This application has the ability to save the user information on the device until explicit logout, using 'shared-preferences' android feature.

When the client requests the server for list of guides and receives a response from the server with Guide information (used ID). Then, we provide with an option for the client to request the server to setup a connection with a particular guide.

In this step, the server would send a message to the guide to start with the listening service and give back the server, the port it is listening on. The server will then send a message to the client with the guide's IP and port number, so that the client can directly communicate with the guide. If the guide responds to GCM notification messages by sending 'Currently active' message to backend server, the backend would get the guides IP in this message.

Once the connection is established between the client and guide the Video is streamed from the client's camera. This RTSP video stream is received by the guide at the mentioned IP and port. This video is played at the guide end as it is received and the guide is allowed to draw gestures over the video. These gestures/ instructions are actually drawn over a canvas on top of the video at the guide-end. As and when the guide draws a gesture over the canvas i.e over the video, it is registered at the Firebase Server which syncs the data to the client. These gestures are overlaid on top of the video by making the background of the synced canvas transparent. While this video streaming is happening, on choosing the appropriate button the guide can choose to guide through a text message well. Hence, the guidance is successfully transmitted to the client in real time.

V. RESULTS

This section mainly discusses about how our application looks and the outputs of the same based on the guide's request. The user is greeted with a login page as shown in Fig 2. The login page provides two options namely, Login and Register. The client may choose to login if he has already registered.

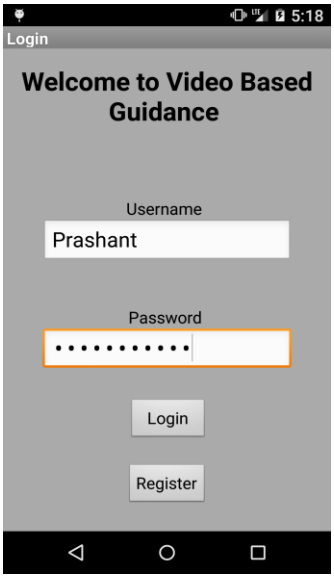

Figure 2. Login Page: Start of application

If the user chooses to Register then a Registration screen is displayed which allows the user to create an User ID and password which governs its future logins as shown in Fig. 3. When the user chooses to login at a later point of time. He is prompted to enter the Username and Password as shown in Fig. 2. A successful match of the Username and the Password to the registered details gives the user the access to our application. The user is given a choice to be a Client or a Guide on every login as shown in Fig. 4.

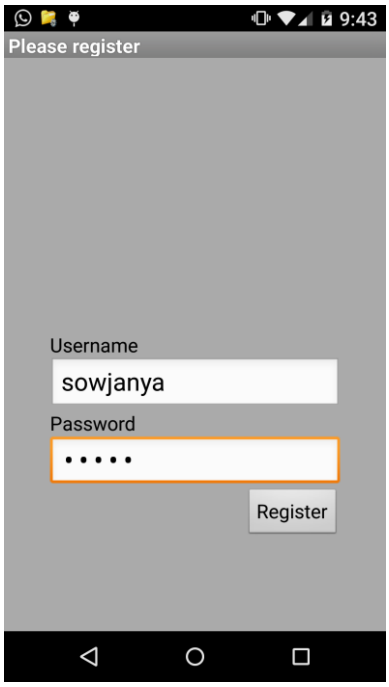

Figure 3: Registration Page

Figure 4. Login Page with options to choose the kind of user

If the user chooses to be a client he can choose from the List of active guides at the time of request for the list. The list is displayes as shown in Fig. 5 . Once the guide is chosen from the list the video streaming starts at the client end using the camera of the client's android phone. .

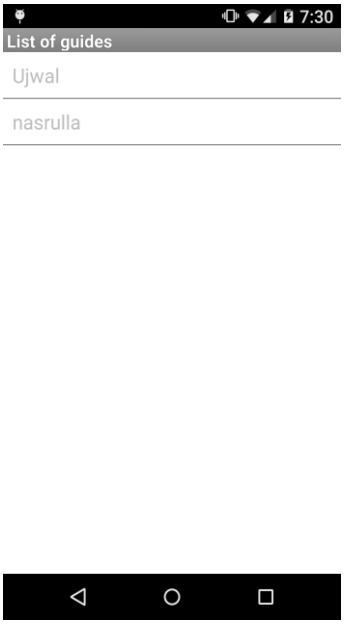

Figure 5. Logged in as client: List of active guides

At the guide end the video is received and is available for any instructions or gestures as shown in Fig.6. The guide is allowed to draw gesture over the video as well as type out a message at the bottom of the screen. The guide can also choose to clear the gestures by using the Clear screen button at the top right corner of the screen. In which case the instructions displayed over the client video is also cleared giving way to the next request. The gestures/ instructions are then synced to the video at the client side and it appears as demonstrated in Fig. 7. The gestures appear over the video where as the message instruction is displayed at the bottom of the screen. There is also a button available at the bottom of the screen that provides the client with a choice to stop streaming.

Hence, this application was successfully implemented as described in the previous sections and demonstrated clearly in this section.

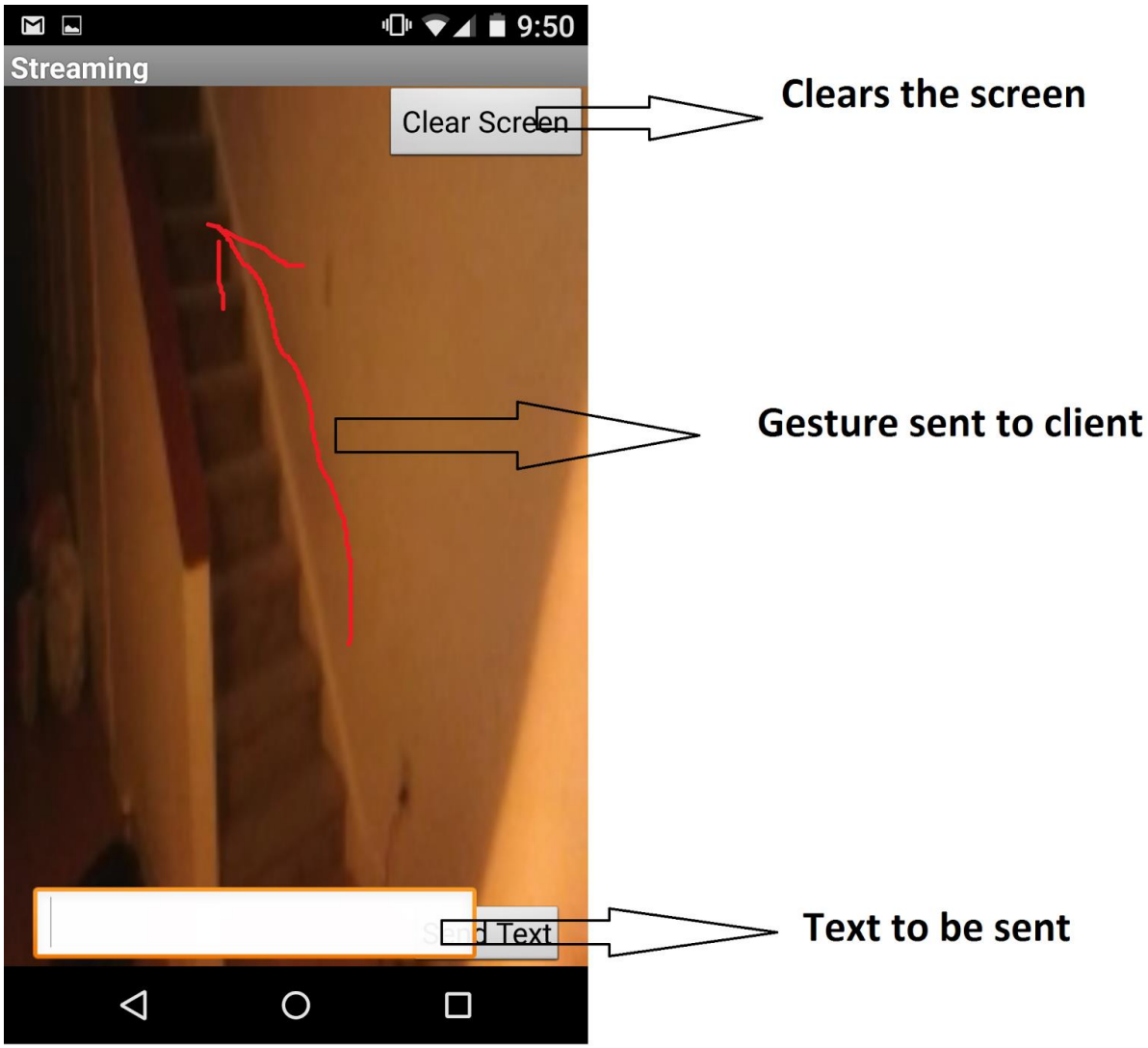

Figure 6. The screen available at the guide end to allow him draw gestures or write message instructions

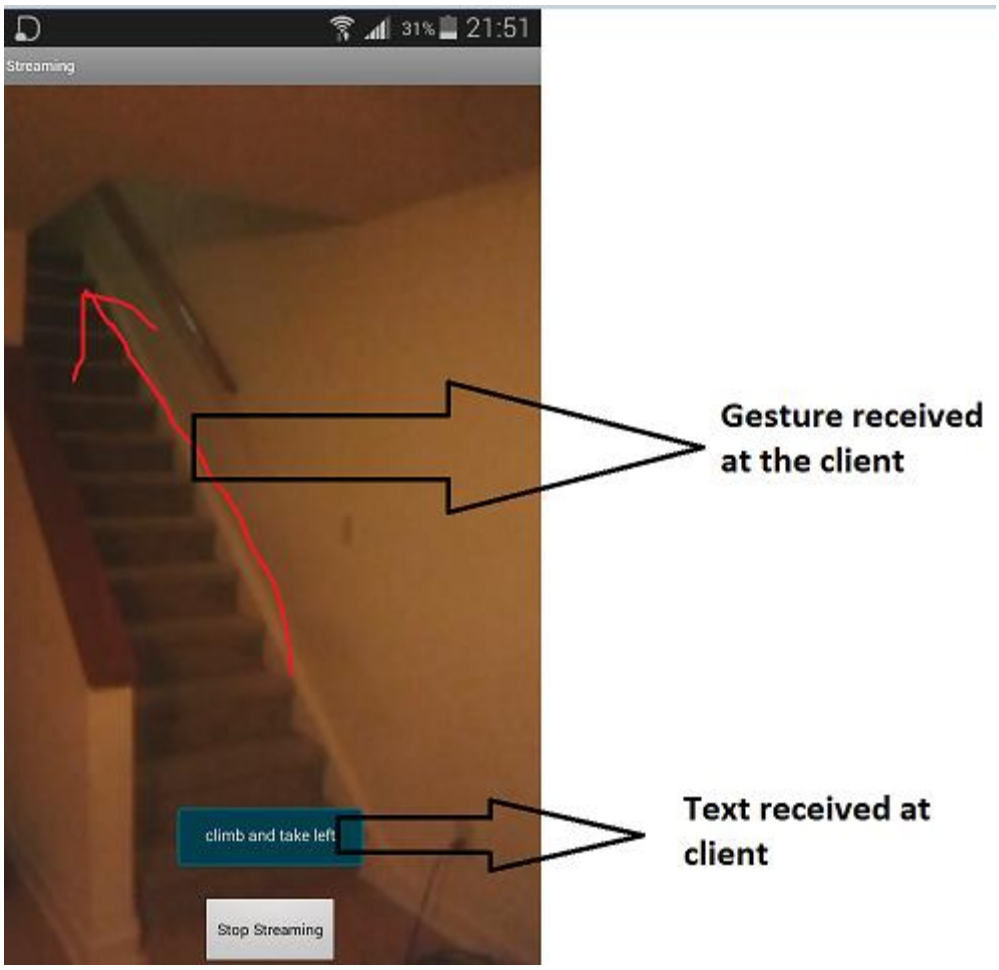

Figure 7. Video Guidance and Message guidance received at the client end

VI. CONCLUSION

To conclude this application is estimated to be very useful in a layman's day to day routine. It probably would emerge to be as necessary as today's GPS guidance system. There are few assumptions made in the development of the project. They are: The guide and client are in the same network .i.e both the addresses are private addresses.

As a part of future enhancement we plan to extend this application to support scenarios where the guide and client are a part of different networks i.e considering their public addresses. Another direction to improve would be that the guide is also allowed to send voice messages back to the client on the RTSP stream as part of the guidance. We also think making the application adapt the quality of the video to the available speed of the network makes it complete in all aspects.

VII. REFERENCES

1] Spydroid application: https://github.com/fyhertz/spydroid-ipcamera
2] Live Streaming: http://code.tutsplus.com/tutorials/streaming-video-in-android-apps--cms-19888
3] Overlay: https://stackoverflow.com/questions/10563534/linearlayout-translucent-but-not-transparent
4] Using firebase for real-time database: https://github.com/firebase/AndroidDrawing
5] Video Capture: https://developer.android.com/training/camera/videobasics.html
6] WebRTC: http://www.webrtc.org/native-code/android

7] Backend database: http://www.mybringback.com/android-sdk/12924/android-tutorial-using-remote-databases-php-and-mysql-part-1/
8] JSON http://www.javacodegeeks.com/2013/10/android-json-tutorial-create-and-parse-json-data.html
9} Google Cloud Messaging : https://developer.android.com/google/gcm/index.html
10] Display from RTSP link: http://code.tutsplus.com/tutorials/streaming-video-in-android-apps--cms-19888
11] SQL commands: http://www.sqlteaching.com/

VI. APPENDICES

Implementation Considerations:

1] The phones are assumed to be in the same subnet.: In our application, it is assumed that both the guide and client are in the same network. We have not considered the case where the users are in two different networks

2] We have overlooked NPAT problems. These can be improved by implementing hole-punching.

3] System is tested in a WiFi environment.