AN INTELLIGENT SENSOR MOBILE PHONE ASSISTING SYSTEM USING AI AND MACHINE LEARNING

Ruilang Liang[1] and Yu Sun[2]

[1]Brea Olinda High School, 789 N Wildcat Way, Brea, CA 92821
[2]California State Polytechnic University, Pomona, CA, 91768, Irvine, CA 92620

ABSTRACT

Technology is taking over the world [7]. Thus, how elderly people can request for help when they use a mobile device if there is anybody around them? In this paper, we address this issue by providing a system that can share and remote control a mobile device in real time [8]. An Android mobile app has been developed as an assistant tool. Thus, when a user needs help, she/he uses an unique ID, sends a request and shares the mobile screen, so the helper sees the sharing screen in his/her device and assists the person who needs help. We applied our application to data analysis and accurate measurements. For the accurate measurement, we conducted diverse experiments to observe the stability of use in different devices, and the influence of geographic, environmental, and network factors. The result shows there are no interrupts during the 30 experiments, which means that the system is stable for use and the network speed is the main factor which affects the average connection delay. For the data analysis, we advertised the Mobile App in communities and schools and received a total of 20 feedback questionnaires. We observe that users from 66 - 70 yield the highest positive score.

KEYWORDS

Machine Learning, Screen Remote Sharing, Mobile APP.

1. INTRODUCTION

Working and living outside all year round, not around their parents. In addition to the phone, we also want to communicate with our parents in time, so that they can understand our state and life. The rich and colorful social software is easy for us to operate [9]. For parents who are not good at playing with mobile phones, the difficulty is sometimes no less than doing a high number of problems without solutions. For this reason, we came up with the idea of remote screen control software that would allow us to use another phone to control the original phone over the network. This is a function to support remote assistance of mobile phones [10]. With a tap on the mobile phone, you can ask for help or help others to solve their mobile phone problems remotely, just as easy and convenient as operating your own mobile phone. Ask a contact in the address book for help. After the contact accepts the request, the contact can control your mobile phone. Accept the help of the other party, remotely view and control the other party's mobile phone, help the other party to modify mobile phone Settings, download and install applications, or remotely doodle on the other party's mobile phone, direct operation. It also supports voice calls between two mobile phones.

There exists multiple software that help to share the screen with other users and might or not control other mobile devices. Some of the software are Skype, TeamViewer and Inkwire Screen Share + Assist [3]. Skype is a very popular social software that people use to communicate and share their screen. Skype users can use Skype through the internet; however it doesn't provide the feature to remote control other mobile devices [11].

The other software is TeamViewer. TeamViewer is a software for personal use that can be used in Android devices. Similarly to Skype it can be used through the internet and share the screen. It allows others to control mobile devices after a user enters the unique ID for the device that needs help. However, to use this remote control feature the helper device and the “help” device need to install 2 different software, TeamViewer Quick Support and TeamViewer for Remote Control. Also, if a user who needs help has an Android device, he/she needs to install an add-on depending on your Android device manufacturer.

Finally, Inkwire Screen Share + Assist is software that is only for Android devices. Like Skype and TeamViewer, it allows users to share their screen. The app does not have the ability to control other devices, but it has a feature to draw on other screens. To use this app the user who shares the screen needs to share his/her unique ID. Even though this app is easy to use, there exists significant lag during the communication.

In this paper, we follow similar approaches that Skype, TeamViewer and Inkwire Screen Share + Assist do in their apps. Our goal is to allow users to share their screen and ask for help when they need it by allowing other devices to control their device screen. We provide an Android app that users can utilize to share their screen and use through the internet.. As different from Skype, our app has the ability to remote control other devices which is helpful when others need assistance. There are good features in TeamViewer and Inkwire Screen Share + Assist like users can remote control other devices to assist others when they need it. Secondly, these apps use a unique ID feature to make the sharing screen and remote control easier and secure between 2 or more devices. Therefore we believe that using a unique ID adds more security when they share their screen since only the users that know the unique ID will be allowed to see the screen of the mobile device that needs help and at the same time make the app easy to use. In our app as different from TeamViewer, users do not need to install 2 different soft wares and additional Add- on to allow the remote control on Android devices.

In two application scenarios, we demonstrate how the above combination of techniques increases the convenience of screen sharing:

1. Experiment 1: In order to have sufficient experimental data, we pick up 10 different groups of phone screens to test if the remote sharing function works well. To prove that the program can run stably, we conducted 3 experiments on each experimental group. In order to detect the influence of geographic, environmental, and network factors on APP performance, We have adopted different control groups for the above factors. Experiments results prove that our application can run stably, The influence of the network environment and geographic location is not obvious.

2. Experiment 2: To investigate our user experience and user satisfaction in UI design, we promoted our products to communities and schools. A total of more than 1,000 students and parents have used our app, We received a total of 20 feedback questionnaires. The questionnaire shows that families with parents in the age group 66 - 70 years of age have the most significant effect and give the highest feedback scores.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Lacking knowledge in Java

At the beginning, I found that my knowledge of Java was far from enough for me to complete the program. As a result, I had to learn more Java while planning my software in order to keep the project going. However, exceptions are my unavoidable problem. This includes Error, Runtime Exception, Exception, throw custom Exception, and so on. Exception encountered before will be hurry-scurry, most anomaly can be solved through mode, there are also many exceptions are due to the developers coding errors caused by, therefore encounters abnormal must first to analyze the causes of abnormal, step by step to the position of the mode for throwing an exception, and then constantly sums up the various reasons of throwing an exception, In the study and work to continuously improve their ability to solve problems. There are two ways to learn exceptions. One is to systematically understand the types of exceptions and understand the causes of the exceptions, apply the methods to the actual problems, and then look for different solutions. Another method is to do a lot of practice in learning. After encountering abnormalities in the process of practice, check the causes of abnormalities and summarize them according to the actual situation.

2.2. Platform problem

Then I realized that the app I wanted to create had to be tested on Android [12]. Since I needed to create an Android app, the best program for me was Android Studio. But when I needed to install virtual machines on Android Studio, I had problems [13]. The two virtual Android phones I needed couldn't exist on my Android Studio at the same time. In the initial learning stage of Android, there are usually many problems. In the learning process of Android, there are many knowledge points and it is difficult to skillfully apply them. It is difficult to skillfully apply what you have learned without a long period of time. There are two ways to use ragmen: statically loaded and dynamically loaded. I usually use the dynamic approach. There are three ways to write fragments. List Fragment Dialog fragment Load the layout object in the on Create View method and set the value in the on View Created method. Finally, in the main method you can get the Fragment Manager submission and load the fragment Layout from mian.xml and fragment into the layout file.

2.3. Mastering Java programming ability

To sum up, in order to develop this APP, I need to master some JAVA programming ability and be able to make use of JAVA programming smoothly. In addition, you must learn to master some basic knowledge of Linux, which is based on the android system design foundation. In addition, we should also learn some basic knowledge of database and network protocol, which will be involved in the design of mobile app. Of course, it is also important to master the operation of some development platforms, such as AppmakrAppMakr, AppCanAppCan and Ling. The whole app may be simple or complex, and the difference of application functions of different apps also leads to different technical implementation or algorithm model. Generally speaking, I need to know the following essential aspects from design to final implementation of this app:

1. Preliminary requirements planning and information, interaction design -- you need to develop a complete requirements document, function document, flow chart, sequence diagram.

2. Interaction design and UI design -- Design a basic and perfect prototype diagram and the basic interaction design effect of app, then design a complete UI interface based on these and learn to cut diagrams. Some adaptive material pictures need to be made with 9patch. You also need to understand the conversion between PX, PT and DP, screen density conversion and coefficients between each other, so that your app can adapt to different resolutions. Interaction design requires you to know a lot of man-machine operation skills and experience, master the use of Axure and other interactive tools, UI design requires you to master Photoshop and Illustrator and other operations.

3. Using the DEVELOPMENT environment such as ADT for APP development, you have to master the Java language, familiar with the Android environment and mechanism, which involves a wide range of areas, please learn relevant knowledge according to the project [15].

4. if it is not a stand-alone version of the app, you need to use the server, then you have to master Web Service related knowledge and development language, commonly used ASP.Net, PHP, JSP and so on.

5. Familiar with and able to develop databases.

6. some functions need to do algorithms, which also need certain professional knowledge, especially mathematical basis.

7. Be familiar with API development, including your ability to develop your own API and experience in calling third-party apis.

8. Familiar with TCP/IP, socket and other network protocols and related knowledge.

9. Proficient in App release process, real machine debugging skills, certificates, packaging and shelves.

3. SOLUTION

Mobile Phone Remote Control Software is an application for one phone to control the other phone. We first need to download the app on two different Android phones, and then open it on both phones. Once in the app, click the "Give Help" button on one phone, click "Accept Help" on the other, and enter the same numeric key on both phones (you can use any number as your key, but both phones must enter the same numeric key). At this time click "accept help" mobile phone will jump to your mobile phone desktop, you can operate the phone normally. After clicking the "Help" button, the phone will see the screen of the "Help" phone. The operator can not only see the screen of the "help" phone, but also guide the user of the "help" phone to use the phone by clicking the screen of the "help" phone. Mobile phone screen clicks on the "help", "accept help" to see "help" on the phone's screen mobile phone operation, then users only need to apply these operations themselves also to "accept help" on the phone, give the helper can be accomplished by a mobile phone to guide the other mobile phone users of mobile phone use this action.

We also provide a set of navigation techniques for our system. The following sections describe these components in detail.

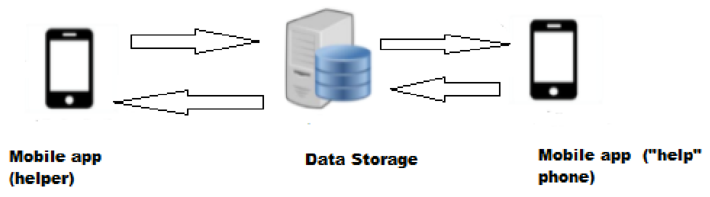
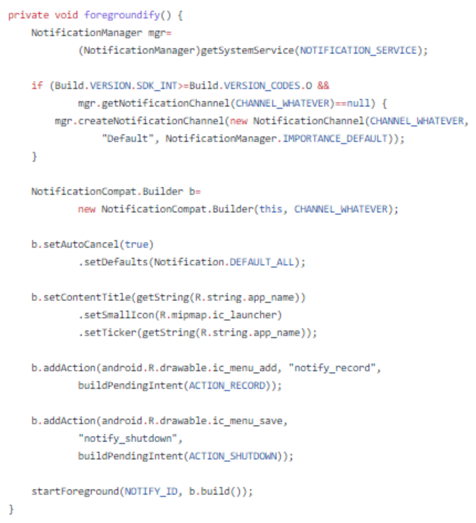


Figure 1. Overview of the solution

The frontend mobile application is developed using Android Studio. Android Studio is an Android integrated development environment (IDE) that is developed specially for Android App[1]. It is written in Java, Kotlin and C++.

To keep tracking the screen of the "help" phone, we implement a foreground service, so that the app can continue to perform a task in the foreground while the user navigates to the screen that he/she needs help. As shown in Fig.2, we developed a function that pops up a notification to the user, so that he/she knows that the app is running in the foreground. In this notification, we created a NotificationManager object to create a channel. Then we use a Notification Builder to add all the widgets that are needed for the notification.

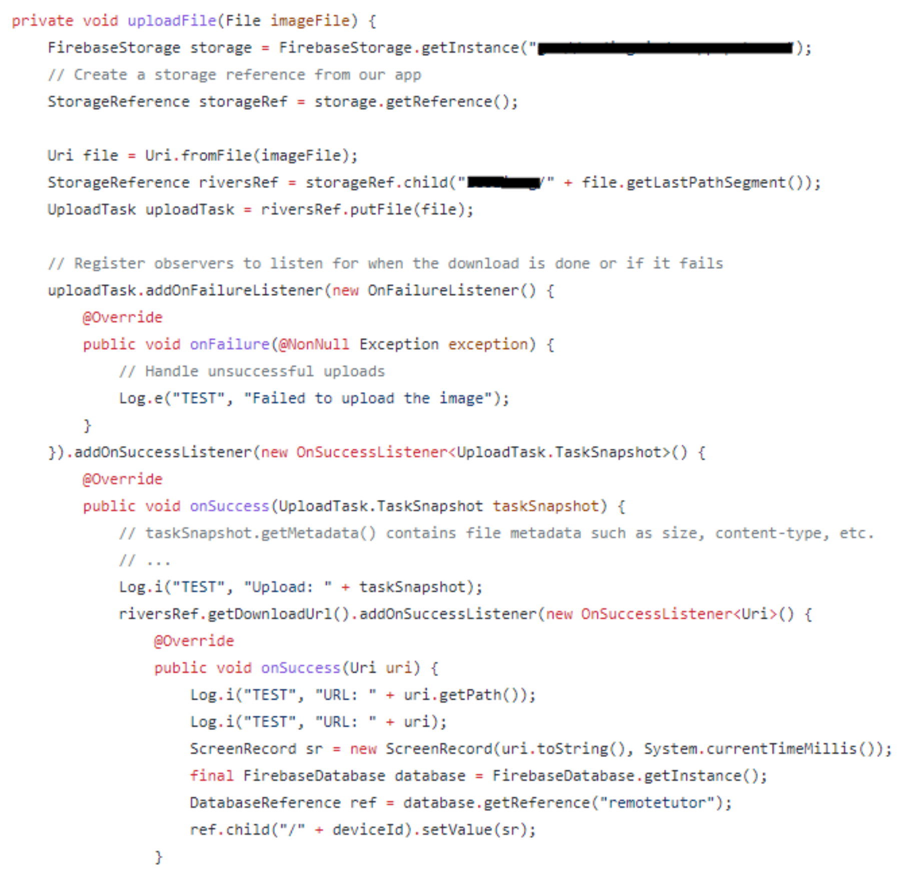


```
private void foregroundify() {
    NotificationManager mgr=
            (NotificationManager)getSystemService(NOTIFICATION_SERVICE);

    if (Build.VERSION.SDK_INT>=Build.VERSION_CODES.O &&
            mgr.getNotificationChannel(CHANNEL_WHATEVER)==null) {
        mgr.createNotificationChannel(new NotificationChannel(CHANNEL_WHATEVER,
                "Default", NotificationManager.IMPORTANCE_DEFAULT));
    }

    NotificationCompat.Builder b=
            new NotificationCompat.Builder(this, CHANNEL_WHATEVER);

    b.setAutoCancel(true)
            .setDefaults(Notification.DEFAULT_ALL);

    b.setContentTitle(getString(R.string.app_name))
            .setSmallIcon(R.mipmap.ic_launcher)
            .setTicker(getString(R.string.app_name));

    b.addAction(android.R.drawable.ic_menu_add, "notify_record",
            buildPendingIntent(ACTION_RECORD));

    b.addAction(android.R.drawable.ic_menu_save,
            "notify_shutdown",
            buildPendingIntent(ACTION_SHUTDOWN));

    startForeground(NOTIFY_ID, b.build());
}
```

Figure 2. Foreground Feature Code

For the database storage, we use Firebase. Firebase is a platform developed by Google that allows you to store files and text information in real time[2]. It provides different features that help to build applications.

In order to share the screen of the "help" device, we implement ImageReader to get the screenshot and send it to Firebase. As shown in Fig. 3, we use FirebaseStorage to store the screenshot of the "help" device. After the screenshot is stored in Firebase Storage, we fetch the url of FirebaseStorage location of the screenshot and send it to FirebaseDatabase; thus the helper device can get the screenshot of the “help” device.



```
private void uploadFile(File imageFile) {
    FirebaseStorage storage = FirebaseStorage.getInstance("[redacted]");
    // Create a storage reference from our app
    StorageReference storageRef = storage.getReference();

    Uri file = Uri.fromFile(imageFile);
    StorageReference riversRef = storageRef.child("[redacted]/" + file.getLastPathSegment());
    UploadTask uploadTask = riversRef.putFile(file);

    // Register observers to listen for when the download is done or if it fails
    uploadTask.addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception exception) {
            // Handle unsuccessful uploads
            Log.e("TEST", "Failed to upload the image");
        }
    }).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
            // taskSnapshot.getMetadata() contains file metadata such as size, content-type, etc.
            // ...
            Log.i("TEST", "Upload: " + taskSnapshot);
            riversRef.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                @Override
                public void onSuccess(Uri uri) {
                    Log.i("TEST", "URL: " + uri.getPath());
                    Log.i("TEST", "URL: " + uri);
                    ScreenRecord sr = new ScreenRecord(uri.toString(), System.currentTimeMillis());
                    final FirebaseDatabase database = FirebaseDatabase.getInstance();
                    DatabaseReference ref = database.getReference("remotetutor");
                    ref.child("/" + deviceId).setValue(sr);
                }
```

Figure 3. Capture Device Screenshot

To develop the screen for the helper, we show the screen of the other device and track the event when the screen is touched. Thus, when the user requests help, the helper sees the screen of the other device. As shown in Fig. 4 the helper device can see the screenshot of the “help” device, so that the helper instructs the other user where she/he needs to click on the screen. In order to assist the “help” device, the helper clicks on the screen; thus the coordinates of the screen are recorded and sent to the database. In Fig. 5 we fetch the coordinate of the helper screen and send the coordinates to the Firebase.

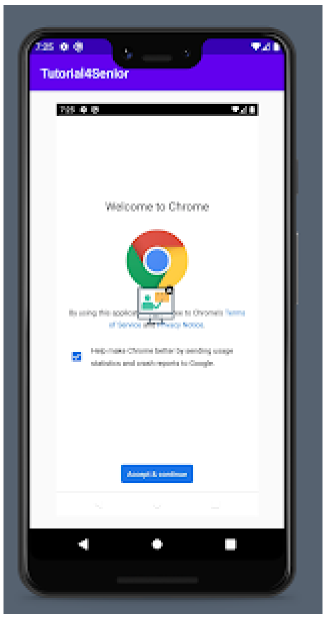


Figure 4. Helper Screen

```
private void post_coords(double x, double y, double xMargin, double yMargin){
    if (x < width / 2 && x - xMargin >0 )
        x= x-xMargin;
    else if (x + xMargin > width)
        x = x + xMargin;
    if (y < height / 2 && y - yMargin < 0 )
        y= y-yMargin;
    else if (y + yMargin > height)
        y = y + yMargin;
    x = (x / width *100);
    y = (y / height *100);
    Log.i("TEST", "post: x:" +  x + ", y: " +  y);
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference ref = database.getReference("remotecontrol");
    ClickRecord cr = new ClickRecord((int)x, (int) y);
    ref.child("/" + deviceId).setValue(cr);
}
```

Figure 5. Coordinate of Helper Screen

Fig. 6 shows the “help” device screen. We can observe an icon with a red target and arrow that indicates where the user needs to click in order to perform the target task. To implement this feature, we use a listener that notify when there is a change in the Firebase Database [14]. Thus, when there is any change in the Firebase Database coordinates of the icon with the red and target and arrow are updated and placed in the corresponding position (See Fig. 7)

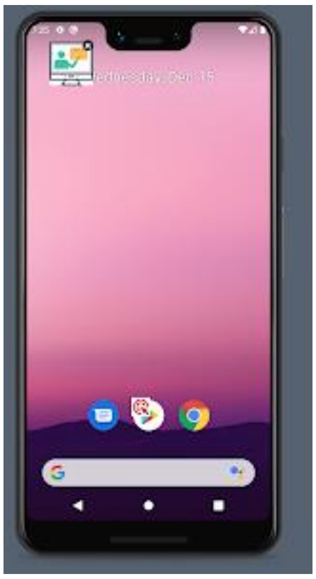
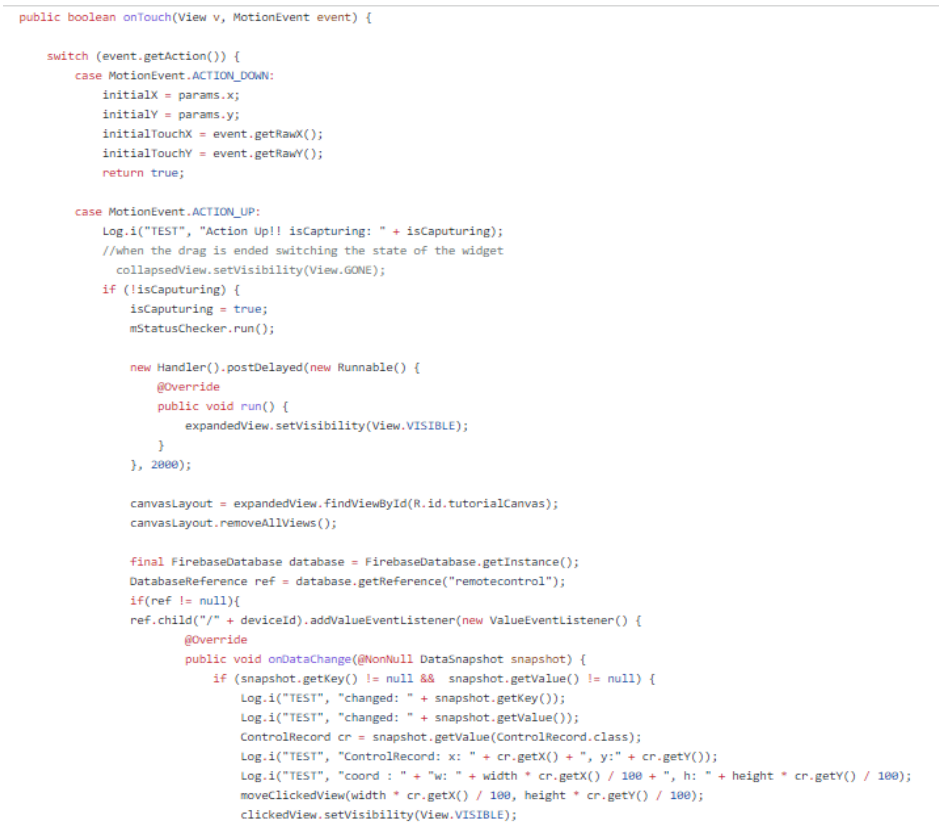


Figure 6. “Help” Device Screen



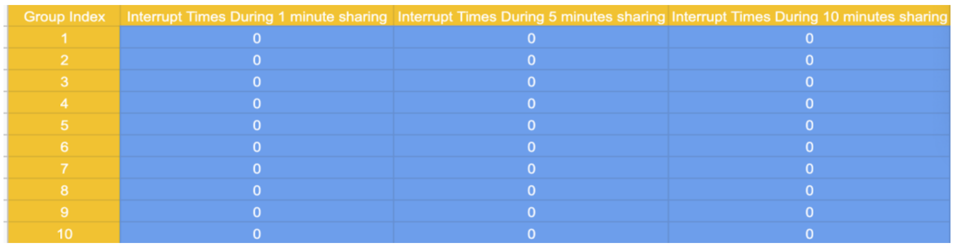
```
public boolean onTouch(View v, MotionEvent event) {

    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
            initialX = params.x;
            initialY = params.y;
            initialTouchX = event.getRawX();
            initialTouchY = event.getRawY();
            return true;

        case MotionEvent.ACTION_UP:
            Log.i("TEST", "Action Up!! isCapturing: " + isCaputuring);
            //when the drag is ended switching the state of the widget
              collapsedView.setVisibility(View.GONE);
            if (!isCaputuring) {
                isCaputuring = true;
                mStatusChecker.run();

                new Handler().postDelayed(new Runnable() {
                    @Override
                    public void run() {
                        expandedView.setVisibility(View.VISIBLE);
                    }
                }, 2000);

                canvasLayout = expandedView.findViewById(R.id.tutorialCanvas);
                canvasLayout.removeAllViews();

                final FirebaseDatabase database = FirebaseDatabase.getInstance();
                DatabaseReference ref = database.getReference("remotecontrol");
                if(ref != null){
                ref.child("/" + deviceId).addValueEventListener(new ValueEventListener() {
                        @Override
                        public void onDataChange(@NonNull DataSnapshot snapshot) {
                            if (snapshot.getKey() != null &&  snapshot.getValue() != null) {
                                Log.i("TEST", "changed: " + snapshot.getKey());
                                Log.i("TEST", "changed: " + snapshot.getValue());
                                ControlRecord cr = snapshot.getValue(ControlRecord.class);
                                Log.i("TEST", "ControlRecord: x: " + cr.getX() + ", y:" + cr.getY());
                                Log.i("TEST", "coord : " + "w: " + width * cr.getX() / 100 + ", h: " + height * cr.getY() / 100);
                                moveClickedView(width * cr.getX() / 100, height * cr.getY() / 100);
                                clickedView.setVisibility(View.VISIBLE);
```

Figure 7. Update Coordinates of the target Icon

4. EXPERIMENT

4.1. Experiment 1

To evaluate the accuracy of our approach, we have collected 30 real dataset from 10 Sensor and Students Group. In order to compare the approaches, we conducted experiments to verify two aspects: the stability of use in different devices, and the influence of geographic, environmental, and network factors. To test the stability of use in different devices, we ask the 10 different groups to test the sharing function 3 times. The sharing duration of each time is 1 minute, 5 minutes and 10 minutes. The data table shows below:

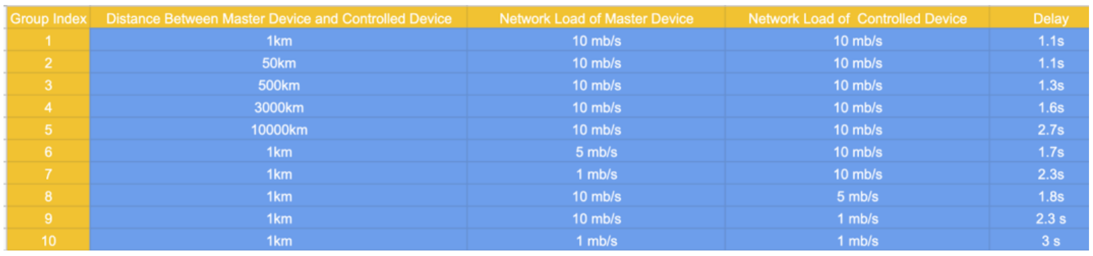


Group Index	Interrupt Times During 1 minute sharing	Interrupt Times During 5 minutes sharing	Interrupt Times During 10 minutes sharing
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0

Figure 8. Result of experiment 1 (1)

The result shows there are no any interrupts during the 30 experiments, which means that the system is stable for use.

To find the influence of geographic, environmental, and network factors, we collect the connecting time and the average delay of each connection, the result shows below:



Group Index	Distance Between Master Device and Controlled Device	Network Load of Master Device	Network Load of Controlled Device	Delay
1	1km	10 mb/s	10 mb/s	1.1s
2	50km	10 mb/s	10 mb/s	1.1s
3	500km	10 mb/s	10 mb/s	1.3s
4	3000km	10 mb/s	10 mb/s	1.6s
5	10000km	10 mb/s	10 mb/s	2.7s
6	1km	5 mb/s	10 mb/s	1.7s
7	1km	1 mb/s	10 mb/s	2.3s
8	1km	10 mb/s	5 mb/s	1.8s
9	1km	10 mb/s	1 mb/s	2.3 s
10	1km	1 mb/s	1 mb/s	3 s

Figure 9. Result of experiment 1 (2)

The result shows that the network speed is the main factor which affect the average connection delay.

4.2. Experiment 2

To know if our user experience is good and if the user is satisfied with the UI design, we publish our app in the market and advertise the production in communities and schools. A total of more than 1,000 students used our app to help their parents, and we received a total of 20 feedback questionnaires. We collect the data and make a diagram to show the feedback result.

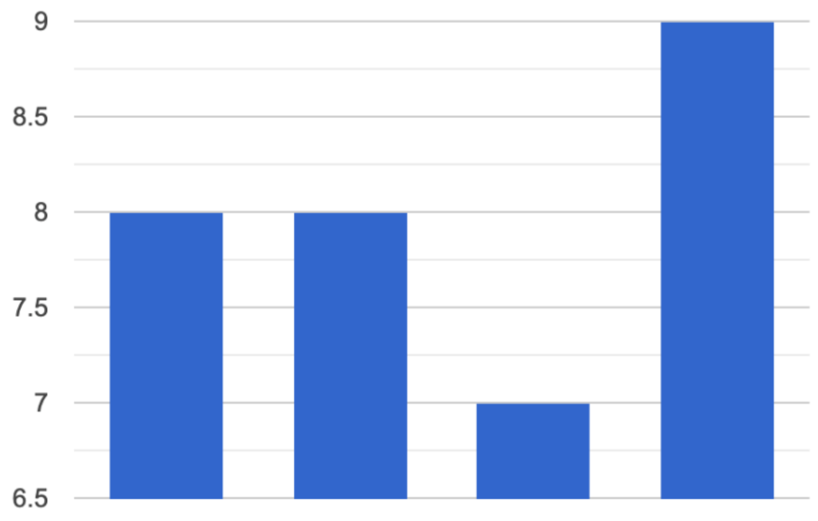


Figure 10. Result of experiment 2

The first from left bar shows the average score while the parents age is between 50 - 55, the second from left bar shows the average score while the parents age is between 56 - 60, the third from left bar shows the average score while the parents age is between 60 - 65, the third from left bar shows the average score while the parents age is between 66 - 70. The result shows the age with 66 - 70 has the highest score.

The first experimental results show that network speed is the main factor affecting the average connection latency. We can solve this problem by increasing the speed of the network. The second experimental result shows that when people aged 66-70 get the highest average score in using this software, we can know our main target group through this result. The results of these two experiments are the same as what I guess, which also meets my expectation for this software.

5. Related Work

Yuanyi Chen presented a system to remote control Android Mobile Phone by using a computer[4]. In the paper, the author explained that developers need to understand and identify the relationship between the four components, active page, service, content provider and broadcast receiver, of the Android system in order to create a remote control application. Also, the author described how the remote control system works from PC device to the server and mobile device. Our application has a similar approach. We use Wifi to send the information from one device to the other. However, our app uses another mobile device as a controller instead of PC, which makes the controller device be more efficient at the time to help another user since most of the people carry the mobile device.

Sørensen H. et al presented a wireless system to share screens in video calls [5]. They proposed a system that can share both digital content as well as physical artifacts in a video call. Our app is similar to this system, our system mirroring the screen in realtime. However, our system is not for a video call and not only for screen share; it also provides remote control.

Bi L. et al proposed a system to remote control power point play in computers without installing any program in mobile devices. It uses Java Native Interface (JNI) technology to control the windows system's function [6]. In our research, we use Android Studio that uses JNI in order to compile our code. As different from this paper, we control the screen of other mobile devices to provide help instead of remote control power point play.

6. CONCLUSIONS

What if older people need our help to use their phones, but we're not around? That way, when the aforementioned emergency actually happens, they just need to open a simple app on their phone, and the person on the other side can use the app to control the older person's phone with their phone and guide them step by step through the phone to use the phone remotely. So I created a mobile remote control app that allowed us to display a page on a mobile phone on another mobile phone. Through experiments, we know that this method needs to run under good network conditions. And through experiments, we can know that this software is very suitable for the elderly, which is also what I hope this software can solve.

Currently, the app is only available on Android phones, which is its limitation. Then there's the latency of running the software on both phones, so when one phone enters a command, the other phone takes a long time to sync the command and has to manually refresh the screen. Finally, we cannot directly control all operations of another mobile phone through one mobile phone, but can only use one mobile phone to guide another mobile phone, and then the operator needs to operate the mobile phone according to the instructions.

In the future, I will optimize the program to reduce latency when using the software. Updates will be pushed to upgrade the functionality of the software to directly control all operations from one phone to the other. Finally, the compatibility of software on the Apple system is solved.

REFERENCES

[1] Developers, Android. "Download Android Studio and SDK tools." linha]. Disponível em: https://developer. android. com/studio/index. html.[Acedido: 03-Mai-2019] (2015).
[2] Ayewah, Nathaniel, and William Pugh. "The google findbugs fixit." Proceedings of the 19th international symposium on Software testing and analysis. 2010.
[3] Machiry, Aravind, Rohan Tahiliani, and Mayur Naik. "Dynodroid: An input generation system for android apps." Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering. 2013.
[4] Chen, Yuanyi. "Research on Application System of Remote-Control Computer of Android Mobile Phone." Journal of Physics: Conference Series. Vol. 1992. No. 2. IOP Publishing, 2021.
[5] Sørensen, Henrik, et al. "Wireless smartphone mirroring in video calls." IFIP Conference on Human-Computer Interaction. Springer, Cham, 2015.
[6] Tan, Gang, et al. "Safe Java native interface." Proceedings of IEEE International Symposium on Secure Software Engineering. Vol. 97. 2006.
[7] Burke, Andrew. "Ultracapacitors: why, how, and where is the technology." Journal of power sources 91.1 (2000): 37-50.
[8] Punja, Shafik G., and Richard P. Mislan. "Mobile device analysis." Small scale digital device forensics journal 2.1 (2008): 1-16.
[9] Osterweil, Leon. "Software processes are software too." Engineering of Software. Springer, Berlin, Heidelberg, 2011. 323-344.
[10] Fakourfar, Omid, et al. "Stabilized annotations for mobile remote assistance." Proceedings of the 2016 CHI conference on human factors in computing systems. 2016.
[11] Chen, Kuan-Ta, et al. "Quantifying skype user satisfaction." ACM SIGCOMM Computer Communication Review 36.4 (2006): 399-410.
[12] Developers, Android. "What is android?." Dosegljivo: http://www. academia. edu/download/30551848/andoid--tech. pdf (2011).
[13] Enck, William, Machigar Ongtang, and Patrick McDaniel. "Understanding android security." IEEE security & privacy 7.1 (2009): 50-57.

[14] Moroney, Laurence. "The firebase realtime database." The Definitive Guide to Firebase. Apress, Berkeley, CA, 2017. 51-71.
[15] Joorabchi, Mona Erfani, Ali Mesbah, and Philippe Kruchten. "Real challenges in mobile app development." 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. IEEE, 2013.