PROXUNLOCK REPORT

Emre Ozbodur
eozbodu1@binghamton.edu

Kaan Demirel
kdemire1@binghamton.edu

ABSTRACT

In this paper, we describe our project ProxUnlock in 4 main topics . Paper starts with project topic description and continues with the explanations of project design and implementation. Last part of paper includes our overall evaluation of the project.

1. PROJECT TOPIC

ProxUnlock is an Android project, which allows faster unlocking systems for Android smart phone users. People believed to be unlocking their phones approximately 110 times a day. Some of them for no reason and some of them to actually use their phone. With ProxUnlock, we aimed to decrease the amount of time spent on unlocking a smart phone. In addition to that, we also set sight on adding a cool feature to smart phone unlocking systems. To accomplish our goal, we decided to build an algorithm that uses Orientation and Proximity sensors within smart phones. We also added some other features that can provide a better user experience while using the application. ProxUnlock has three main objectives: easiness, reliability and speed. To build a good unlocking system for smart phones, we believe these three features should be our main target. While building ProxUnlock, we always considered application's overall easiness, reliability and speed.

2. PROJECT DESIGN

2.1 Design Dependencies and Constraints

On our project, we identified and defined some dependencies and constraints. Minimum SDK requirement for ProxUnlock is Android 5.0 Lollipop. For the sensors, we require the phone to have proximity and an orientation sensor. For proximity sensor, sensor value outputs should be numerical values in centimeters. On some phones, proximity sensor only returns two states like close and far. In our design, we use a proximity sensor that detects how close object is in centimeters and returns that value. For orientation sensors, we require first output of the sensor to be a orientation degree value between 0 and 360. Since we only care about the orientation degree value, the other outputs of the sensor are irrelevant.

2.2 Design Goals

We had three main objectives prior to start of the project: easiness, reliability and speed. Since phone screens are getting bigger and bigger every year, unlocking your phone easily is getting harder than ever. Unless the user has big hands that can reach all around the screen it is now almost impossible to unlock your phone using only one hand. With our design, we aimed our application to have the easiest solution comparing with other unlocking systems. We also targeted to be less touch-dependent and more gesture-dependent while building our algorithm. As a result, we decided on using a system that requires no touching to the screen or any buttons and just uses a quick rotation with the selected degree to unlock the phone. Our second objective for the project is the reliability objective. On the latest part, we explained by we needed an easy and user-experience friendly application. Easiness is a key objective but we figured we needed to design our application to be reliable as well to ensure its easiness. Although, unlocking, itself is not a very complex system that can cause problems on smart phones, sensors are unpredictable and their outputs vary on a lot of different situations.

On Android phones proximity sensors are extremely reliable with some limitations. Most of the Android phones get three different values as the first output of their proximity sensor: 1, 3 and 100 centimeters. As far as we tested, on normal conditions proximity sensor is extremely reliable. On some abnormal conditions like phone being in a bag or a pocket, proximity sensor's accuracy decreases. For orientation sensor, the accuracy with phone on the hand and phone on a hard surface differs. From our experiments, we find out that orientation sensors are more reliable when phone is used on hand, and its accuracy can decrease up to 5 percent when it is used on hard surface like tables, desks etc. In overall, using proximity and orientation combined with each other gives us an unlocking accuracy of 95 percent to 100 percent depending on circumstances. The third objective of our project is ensuring a high speed. We calculated average unlocking times on each available unlocking system such as unlocking through a pattern, unlocking with a PIN, unlocking with face recognition. After experimenting on these different methods, we found out that our solution is 52 percent faster than the current fastest unlocking system. Designing our system as independent from buttons and the screen helped us designing and building a faster unlocking solution.

2.3 Description of Components

While planning on our project, we identified some main and sub components that we can use on our design. Therefore, we created two main components for our project: front-end and back-end features. On our front-end features we used buttons, switches, spinners and image holders. On our back-end features we used background service, helper activities and special functions such as a function that checks running services, a function that checks screen statuses, a function that checks sensor values, a function that checks phone's lock status. On the implementation part of this document, each component is explained in detail and their connections are mentioned.

2.4 User Interface Design

For our user interface we chose to go with a simple user interface design that provides a better user-experience. On our main page, we have two options: Set Unlocking Pattern and About. On Set Unlocking Pattern option, we added three features such as a spinner for selecting a rotation degree, a switch to start or stop the service, a switch to activate or deactivate vibration. On our About page, we provided contact information about ourselves and we also provided the steps to use the application. In addition, we

created a view which serves as the view where unlocking happens. On this screen, we provided a button to return to application main page.

3. PROJECT IMPLEMENTATION

First of all, we used background service, sensor event listener, a function to check whether service is running, screen receiver, spinner for selecting rotation degree, and algorithm for unlocking phone in the project. All of these items are described below with more detailed.

3.1 Background Service

```
mySwitch = (Switch) findViewById(R.id.switch1);
if(isMyServiceRunning(StartStopService.class)){
    mySwitch.setChecked(true);
}
mySwitch.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            Intent intent = new Intent(CheckActivity.this, StartStopService.class);
            startService(intent);
        } else {
            Intent intent = new Intent(CheckActivity.this, StartStopService.class);
            stopService(intent);
        }
    }
});
```

```
public void startService(View view){
    Intent intent =new Intent(this,StartStopService.class);
    startService(intent);
}

public void stopService(View view){
    Intent intent =new Intent(this,StartStopService.class);
    stopService(intent);
}
```

Background service was created inside of the Set Unlocking Pattern page. Start/Stop Service is the heart of our application. We used a basic switch button to start or stop the service. When the button is switched on, the application starts to run on background and unlocking system continues to work until the service is switched off.

3.2 Sensor Event Listener

```
@Override
public void onSensorChanged(SensorEvent event) {
    Sensor sensor = event.sensor;
    if (sensor.getType() == Sensor.TYPE_PROXIMITY) {
        proxValue = event.values[0];
        Log.v("Prox Value:", String.valueOf(proxValue));
    } else if (sensor.getType() == Sensor.TYPE_ORIENTATION) {
        if (proxValue>0.5 && proxValue<4) {
            angleValue = Math.round(event.values[0]);
            Log.v("Angle Value:", String.valueOf(angleValue));
        } else {
            angleValueNew = Math.round(event.values[0]);
        }
    }
```

Sensor Event and Sensor Event Listener was used because it allowed us to access the device's sensors and if the sensor's value is changed, we receive a notification. With the change of proximity or orientation sensor's values, program decides whether the requirements for unlocking the phone are satisfied or not.

3.3 Checking Whether Service Is Running or Not

```
private boolean isMyServiceRunning(Class<?> serviceClass) {
    ActivityManager manager = (ActivityManager) getSystemService(Context.ACTIVITY_SERVICE);
    for (ActivityManager.RunningServiceInfo service : manager.getRunningServices(Integer.MAX_VALUE)) {
        if (serviceClass.getName().equals(service.service.getClassName())) {
            return true;
        }
    }
    return false;
}
```

In the project, we needed a function that constantly checks the service whether or not running on the background. If the service is switched on, it means service is running, otherwise the service is not running on the background.

3.4 Screen Receiver

```
public class ScreenReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {

        if((intent.getAction().equals(Intent.ACTION_SCREEN_OFF)))
        {
            lockStatus = true;
            if(keyguardDisable){
                keyguardDisable =false;
                lock=null;
            }
        }
        else
        {
            lockStatus = false;
        }

    }
}
```

We used Screen receiver to decide if the screen is either off or on. To be able to unlock the phone, smartphone's screen must be closed. That explains why we used Screen receiver in the project.

3.5 Spinner

```
Spinner dropdown = (Spinner)findViewById(R.id.spinner1);
String[] items = new String[]{"90 degrees left", "90 degrees right","180 degrees turn"};
ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_spinner_dropdown_item, items);
dropdown.setAdapter(adapter);
```

We used spinner in the Set Unlocking Pattern page that allows users to choose the rotation degree for unlocking the phone. With the selecting rotation degree from the spinner, user unlocks his phone with the selected degree unless he selects another rotation degree on the spinner.

3.6 Algorithm for Unlocking Phone

```
if (proxValue2 == 2 && isMyServiceRunning(StartStopService.class) &&
        lockStatus == true && ((angleValue + Integer.parseInt(spinnerData[0])) % 360) + 20 >= angleV
        && (angleValue + Integer.parseInt(spinnerData[0])) % 360 - 20 <= angleValueNew) {
```

For unlocking the phone, there are some requirements to be satisfied. One of the requirements is that the distance of an object to proximity sensor should be closer than 3 centimeters. Secondly, the service should be running on the background. It means that the start/stop service must be switched on. Another requirement is that the phone must be locked. Lastly, phone must be rotated as much as the degree selected on the spinner in order to unlock the phone.

3.7 Unlocking Phone

```
getWindow().addFlags(WindowManager.LayoutParams.FLAG_SHOW_WHEN_LOCKED | WindowManager.LayoutParams.FLAG_DISMISS_KEYGUARD
        WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON | WindowManager.LayoutParams.FLAG_TURN_SCREEN_ON |
        WindowManager.LayoutParams.FLAG_ALLOW_LOCK_WHILE_SCREEN_ON);
Button returnButton = (Button) findViewById(R.id.button3);
returnButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(StartStopActivity.this, MainActivity.class);
        startActivity(intent);
    }
});

moveTaskToBack(true);
```

The line starting with getWindow opens the lock on the phone and it opens the screen with those lines. Also, we have a line starting with moveTaskToBack. It provides the users to turn back to main screen on the phone.

4. PROJECT EVALUATION

For our project, our aim was to unlock the smartphone by using wireless sensors with high accuracy. In order to achieve this goal, we used both proximity and orientation sensors to unlock the smartphone, which provides the users faster, and easier unlocking system. As a team, we reached our goal that we determined at the beginning of the semester. When we completed the project, the application works with high accuracy and it uses two different wireless sensors. Sensors are used together to build a better example on wireless sensor networks. Also, our application's accuracy is in the sufficient level that we expected before starting the project. To take everything into consideration, we believe that we created an application that can be different solution for the smartphone users in order to unlock their phones.