

Project Name: Advanced App Protector from Unauthorized Access

Author Details

Name		Mail
	Sayeed Hossain	me@sayeedhossain.com

Introduction

A smartphone is an extremely personal device. In the social media apps, we have personal messages and information. Then there are financial apps, which keep a lot of our personal information. We also take a lot of personal images and movies, which are saved in our smartphone's gallery app. So, clearly, you may use our Advanced App Access Protector to protect all confidential information. The App Protector is one of the most basic security tools available. It operates by encrypting our other programs to protect them from prying eyes. We won't have to worry about someone gaining access to our Facebook, gallery, or banking apps this way. It's vital to remember that this isn't a complete security solution. It's appropriate for children or snoopypals. These, on the other hand, offer very little genuine protection against things like stolen phones or even our more tech-savvy acquaintances.

Actually, App Protector is an Android app that allows us to lock apps that we don't want others to have access to without our permission. It will assist us in preventing data loss by inexperienced users or youngsters. This App Protector is the ideal tool for protecting our apps, allowing us to set or disable the lock as needed, as well as toggle between lock pattern mode. One of the key advantages of utilizing App Protector is that it allows us to have complete control over which applications we use and how we use them while also providing extensive customization choices and meticulous attention to detail.



Problem Domain

The problem domain of this project is Security, Pattern Lock, Information Protection, Unauthorized Access Point, Detection and Prevention.

Motivations

Smartphones now hold a lot of sensitive information. Banking apps, note-taking apps, gallery apps, and other similar apps must be protected from prying eyes. As a result, I am driven to work with it in order to preserve all confidential information.

Many parents occasionally lend their smartphones to their children. However, parents do not want their children to have access to all Applications. So I'm motivated to work on it in order to solve it.

Also Keeping our phones safe from unwanted visitors is essential nowadays, especially when we have a lot of personal information on them. Whether we need to keep our children from seeing our photos or our flatmate from seeing our Facebook account, an application bolt can ensure that only we control who sees our information.

We utilize our Smartphones in our daily lives. We also engage in various activities using Smartphone Apps. Sometimes we need to focus on our work or studies. However, some of us are so engrossed in our Smartphones that we are unable to focus fully on our work or studies. In this case, we can disable the apps to which we are attached and focus on our work or studies. In this case, our App Protector could be a fantastic solution.

So all of those things encourage me to work on this project.

Aims/Objectives

The development of this type of system has numerous objectives/goals.

To provide the highest level of security, we wish to give pattern lock and Prevent illegal access by capturing photos features.

Children of all ages accessing financial data or app shops, or accidentally purchasing in-app stuff, is one of the most important challenges that guardians face nowadays. The only method to avoid this is to use an App Access Protector to ensure that we are protected from unwanted costs.

Also Because app lock is a security feature that allows us to determine who has access to what on our phone or tablet, many frameworks will have additional security features that monitor for application-related viruses or unwanted phone snoops.

There's nothing worse than worrying that someone is looking through our belongings, therefore an App Access Protector will help us feel more secure about what's on our phones.

So all of those things are the main objective or goal for this work. So all of those things are the main

Tools & Technologies

To develop this system, we need some TOOLS & TECHNOLOGIES. Some of them are mentioned below:

1. Android Studio (IDE)
2. SDK tools
3. JAVA
4. Android Emulator
5. Real Android Device
6. Linux Kernel
7. ADB
8. AVD
9. DB
10. Proguard-Rules

Challenges

When an engineer is required to think in a different way, he or she is constantly confronted with a variety of obstacles, rivalries, and hindrances. In our task, there are various obstacles, challenges, challenges, and barriers, just as there are in this situation. Our project is beneficial to members of the general public. However, we now add to Facebook, what apps, Viber, IMO, Instagram, snap-talk, and a slew of other web-based life applications. As a result, doing so in our country will be more difficult.

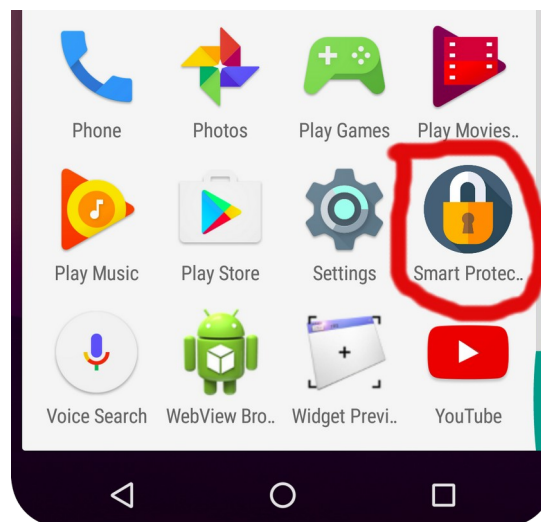
We frequently waste our time on internet-based life applications; this application will fill a critical need that is now being overlooked. I can obstruct my site with this program, which is incredibly harmful to us. Furthermore, we strive to provide complete customer satisfaction. If we encounter any difficulties with a catch's or our application's security, we will work to resolve them.

Front-End Design

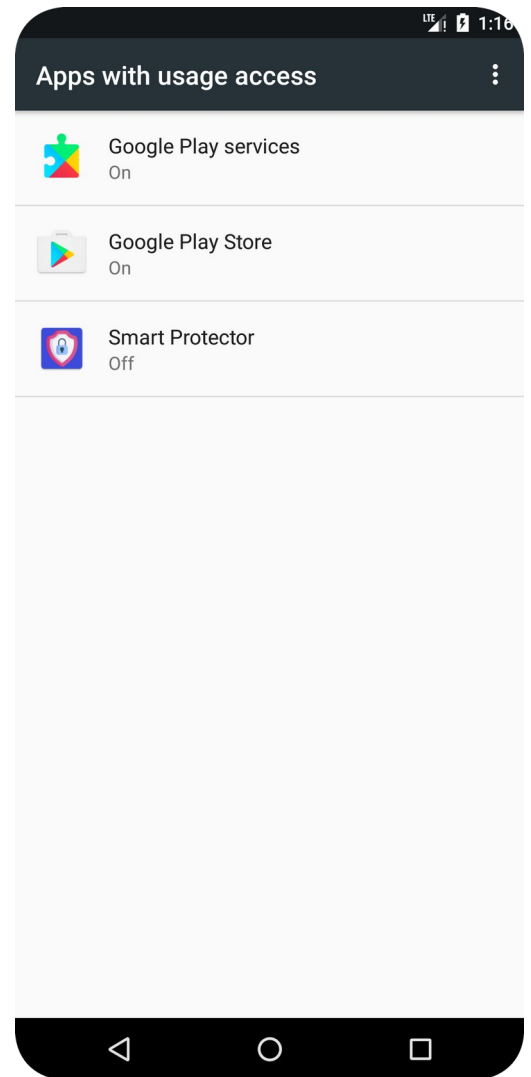
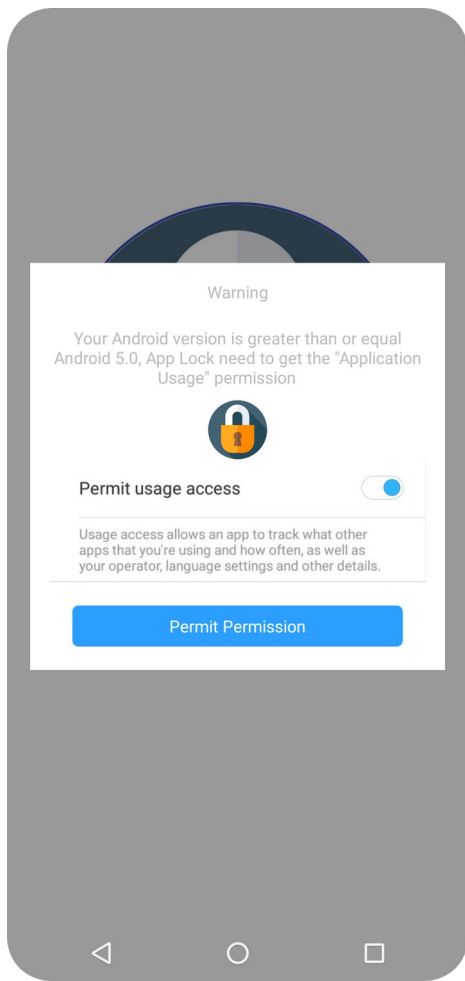
The front-end encompasses everything that the user sees, including design and some programming languages. The user interface design that the user sees when he or she launches the program or website is called front-end design. That means the front-end design is the most crucial aspect of a project. When a user wants to open an application or a website, most people anticipate a straightforward User Interface or Graphical User Interface from the developer. The program fails to attract the user if the front-end design is too complicated.

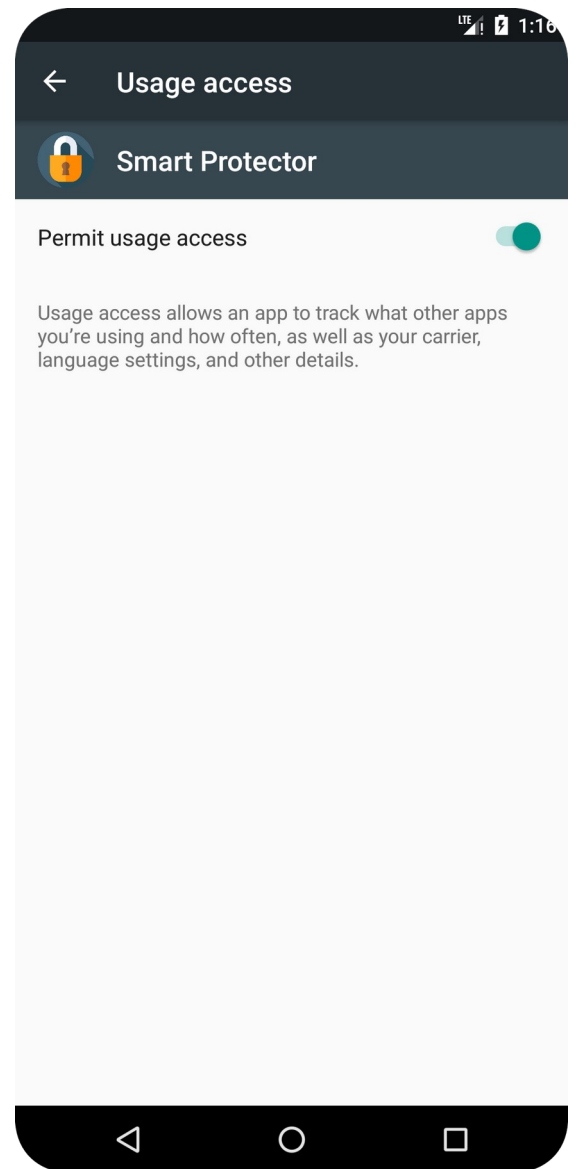
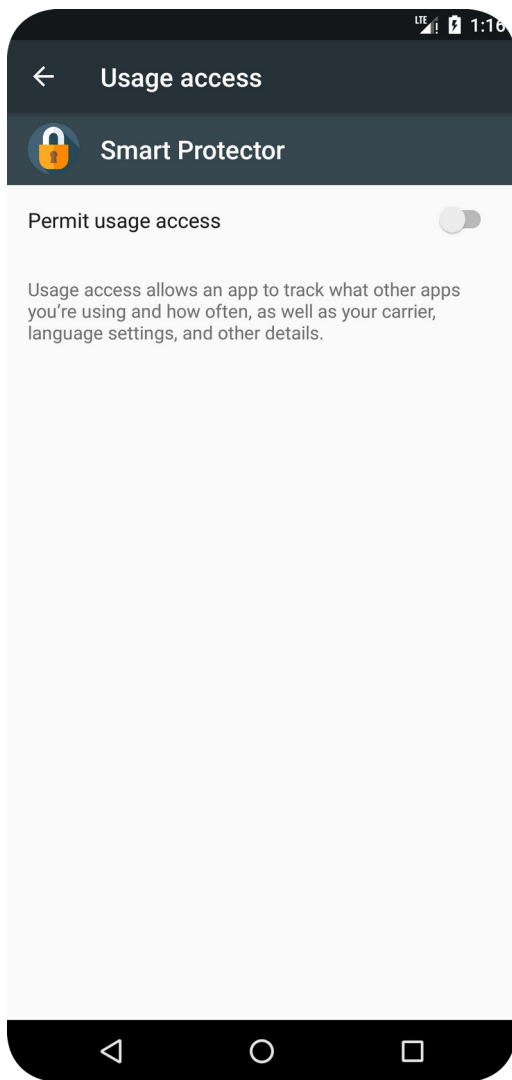
As a result, we attempted to keep the front-end of our program as simple as feasible. We also attempted to create a user-friendly graphical user interface. However, designing a graphical user interface for providing these types of services to users is a difficult task. However, we made every effort to make our application as simple as possible. Hopefully, our program is very user-friendly and that the user can easily access it. The front-end design for our Android application is as follows:

A Screenshot to show the app icon :

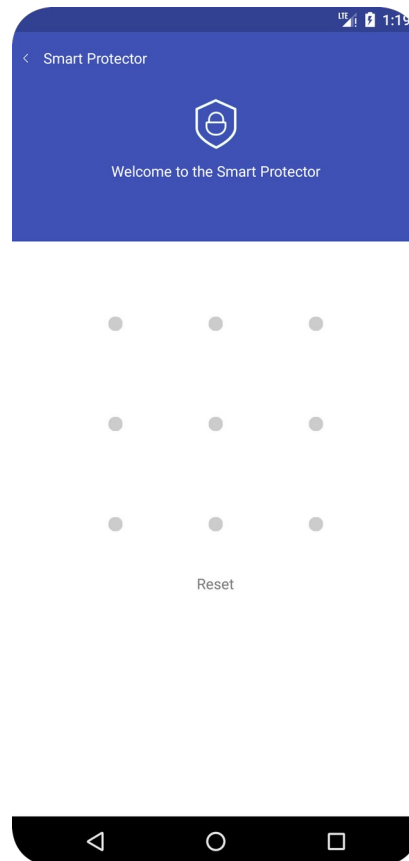


A screenshot for dialog_permission.xml file to take the permission from user.

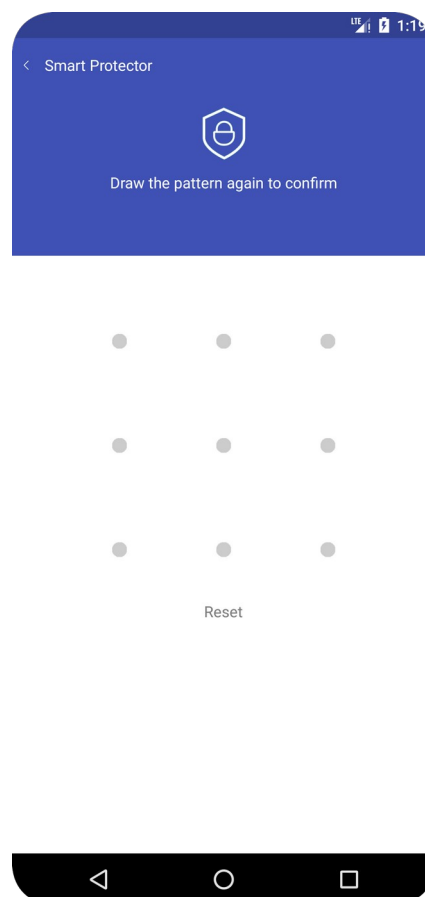




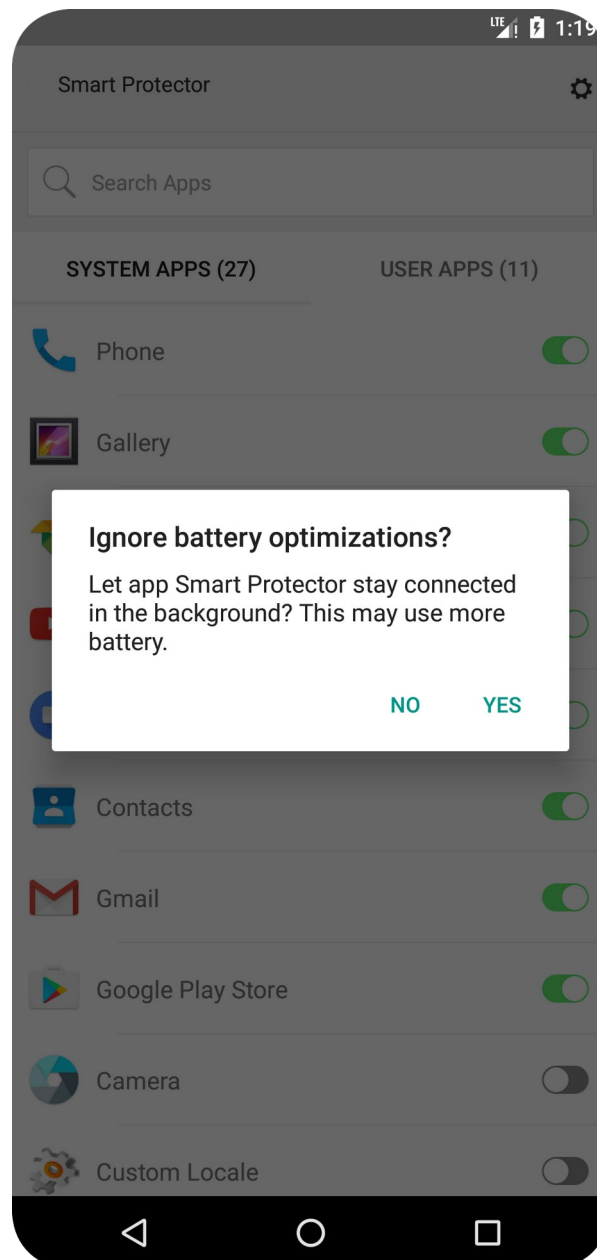
A screenshot for activity_gesture_self_unlock.xml file to setup the pattern lock for the first time. It is only for authors. And obviously we have to remember it for the next access.



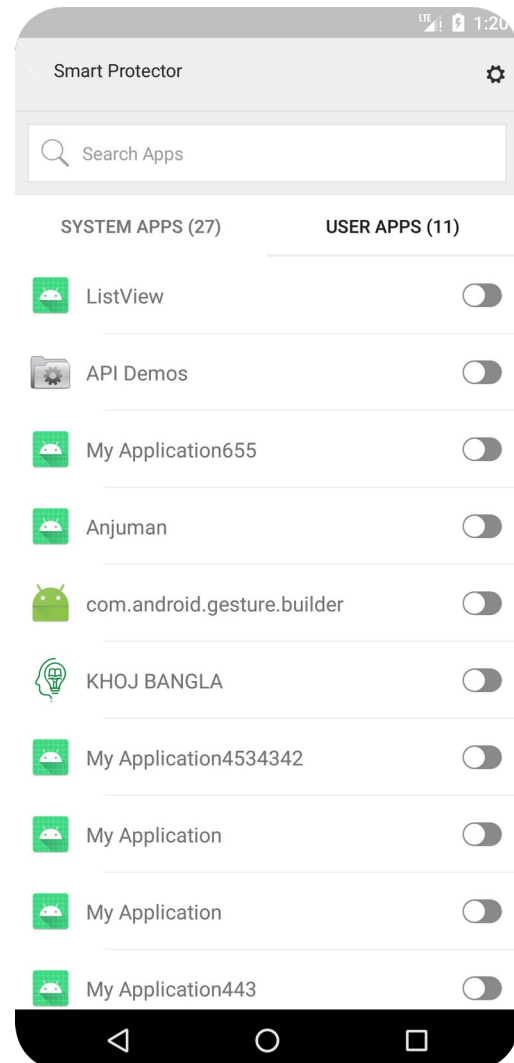
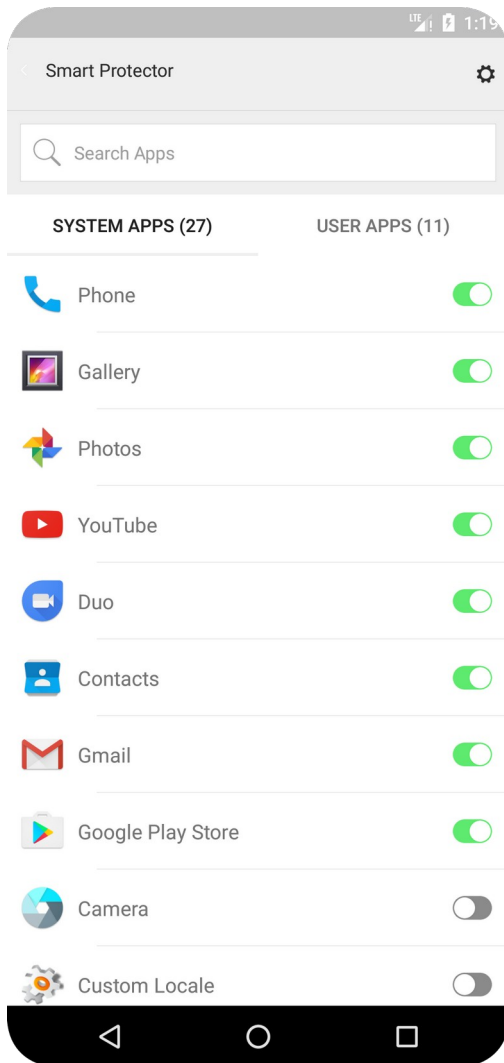
After giving the pattern lock for the first time. It is required for confirmation. So we have to give the same pattern lock again for confirmation.



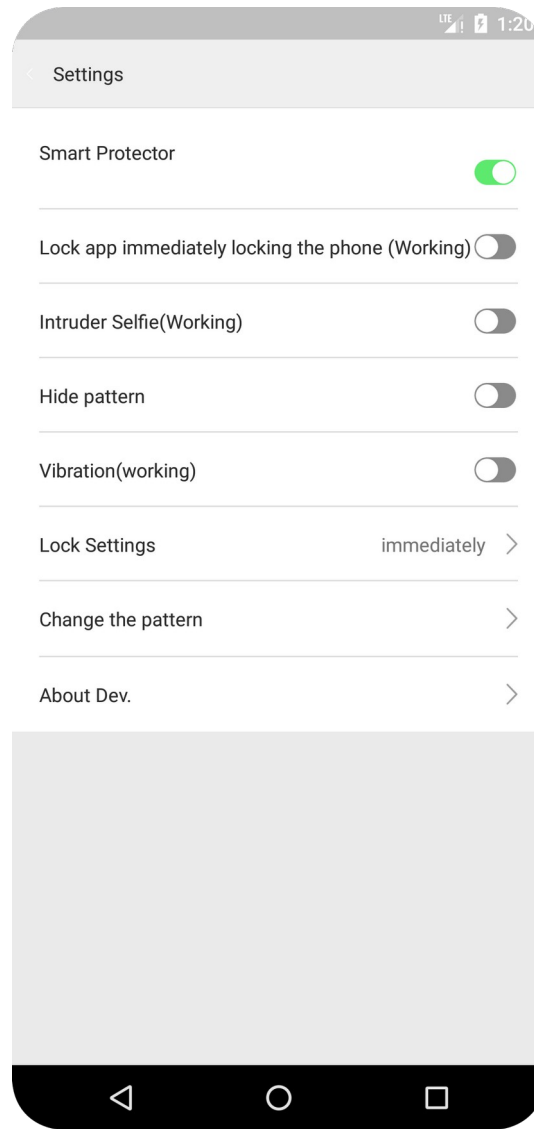
Since our smart protector app stays connected in the background, we should ignore battery optimizations. So I keep a function in my app to ignore battery optimizations so that our app can work properly. Check out the attached image for better understanding.



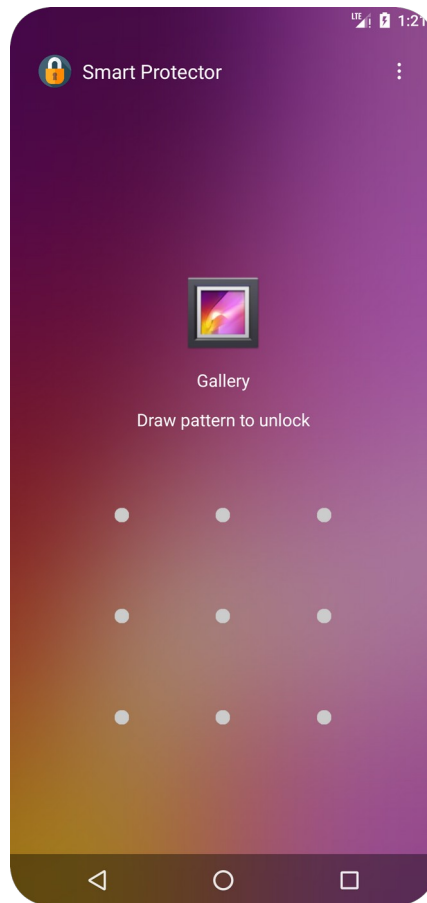
Screenshot of Main Home Page UI :



A Screenshot of Settings Page UI :

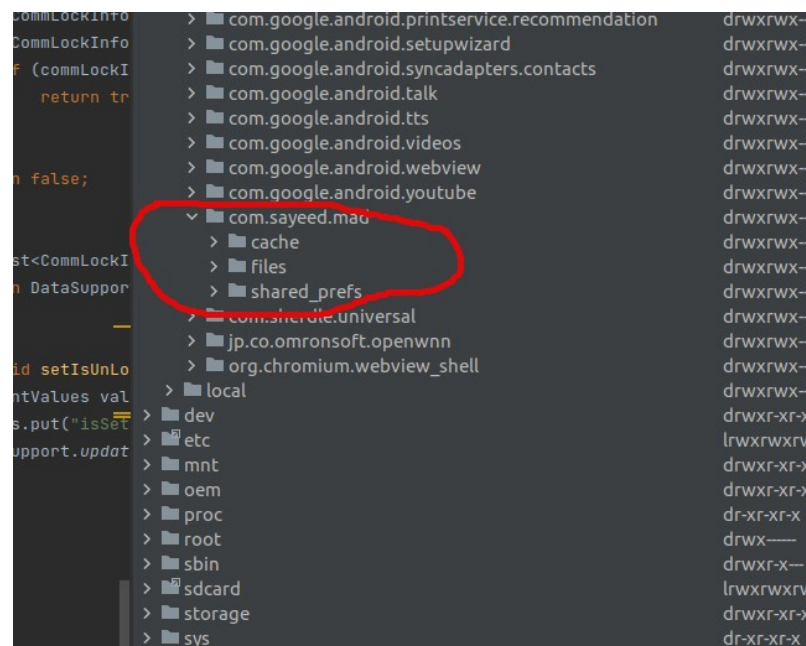


Then after setting up all of the things, when I want to open any app which is locked by our application then it will show the following UI.

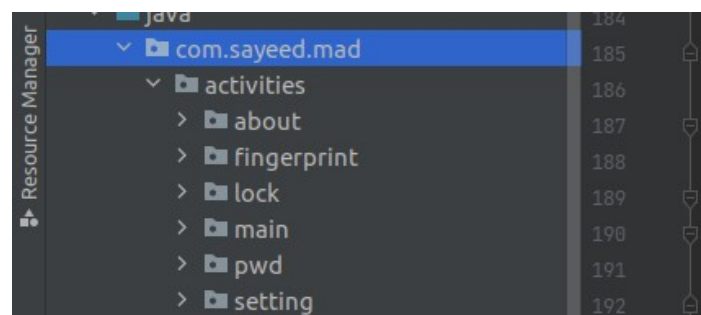
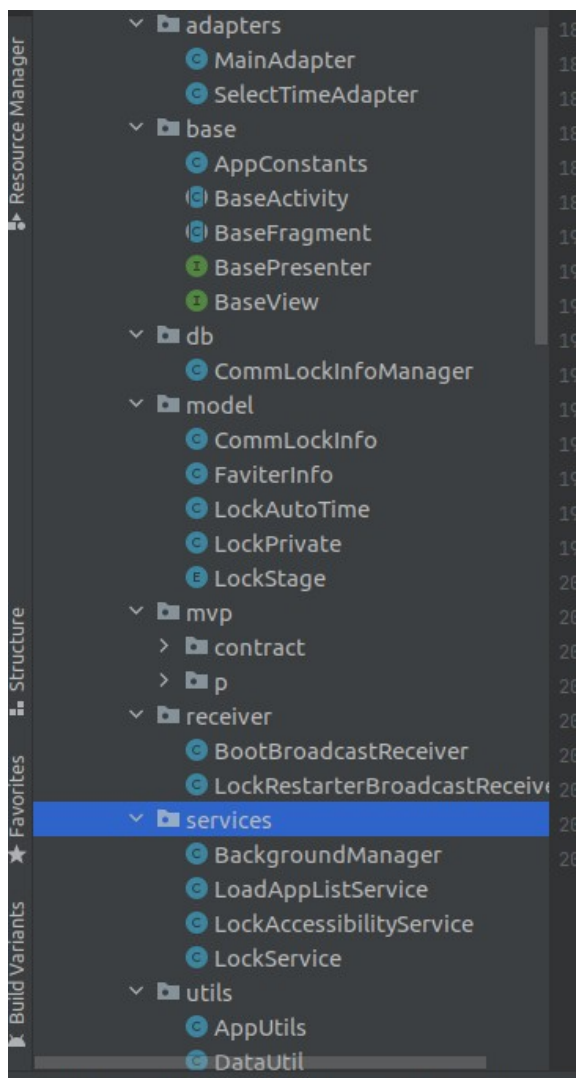
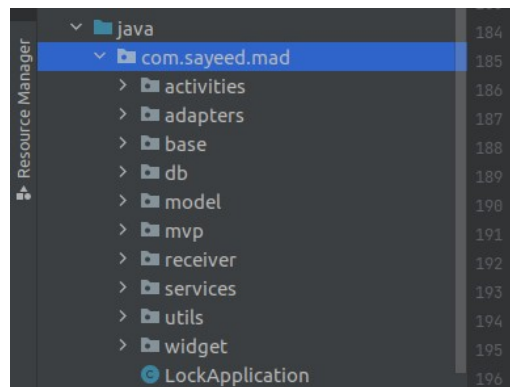


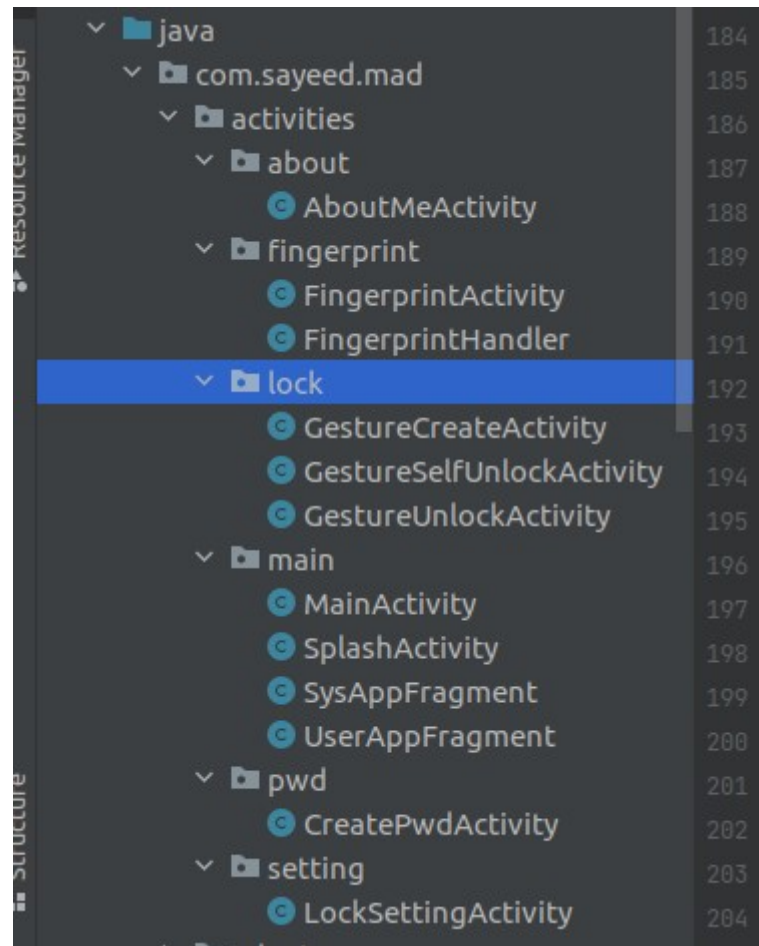
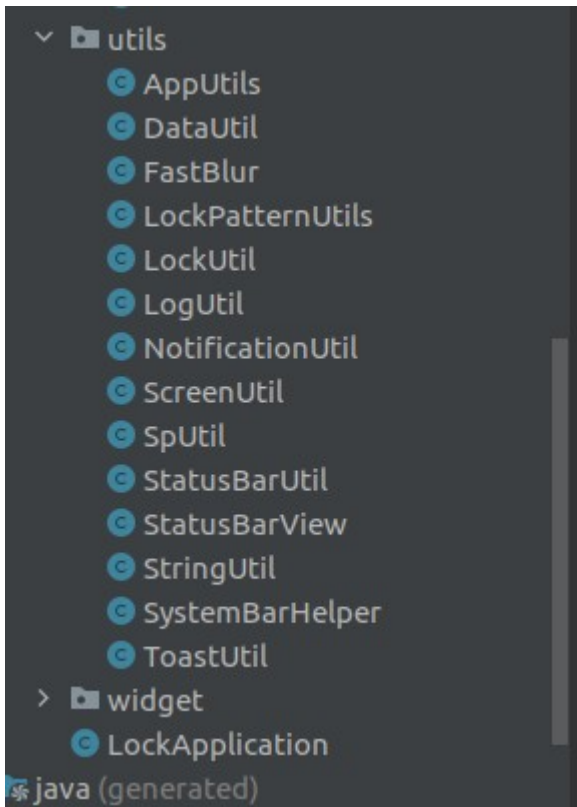
Back-End Development

Back end development refers to an application's server side and anything that communicates between the database and the application. AppLocker is an entirely offline application, however it still need the gesture.key file to save Pattern for later usage.



To build up all the functionality for this system, we have to write down some java code. The screenshot of the file structure for the java file are attached below:





App Permission Setup

Mainly we are use 3 permission to make the app properly workable. All of those permission are listed below:

```
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
```

```
<uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
```

```
<uses-permission  
android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
```

RECEIVE_BOOT_COMPLETED:

Allows an application to receive the Intent.ACTION_BOOT_COMPLETED that is broadcast after the system finishes booting. If you don't request this permission, you will not receive the broadcast at that time. Though holding this permission does not have any security implications, it can have a negative impact on the user experience by increasing the amount of time it takes the system to start and allowing applications to have themselves running without the user being aware of them. As such, you must explicitly declare your use of this facility to make that visible to the user.

Protection level: normal

Constant Value: "android.permission.RECEIVE_BOOT_COMPLETED"

FOREGROUND_SERVICE:

Allows a regular application to use Service.startForeground.

Protection level: normal

Constant Value: "android.permission.FOREGROUND_SERVICE"

REQUEST_IGNORE_BATTERY_OPTIMIZATIONS:

Permission an application must hold in order to use

Settings.ACTION_REQUEST_IGNORE_BATTERY_OPTIMIZATIONS.

Protection level: normal

Constant Value: "android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"

Simulation Environment/ Simulation Procedure

There are the steps to simulate our project in android environment, First of all,

1. Open Android Studio,
2. select File >New >Import Project.
3. Locate our project, select the project's root directory and click OK.
4. Select project from existing sources and click Ok. Your project will be open in the android studio.

Then it's time to Connect our device.

Note: Your USB debugging must "on" from developer option to run your app on your real device. And also you can easily use virtual device by installing any android emulator.

How you can get the developer option in your android device [in the case of USB debugging]:

1. Go to Settings > General > About phone.
2. Then scroll and select Software information > Build number. Now rapidly tap on 'Build Number' five times and you will see the message 'You are now a developer!' The critical thing is to get to 'Build number', and then tap rapidly until you see the message.

Now in this stage, the quesiton is, How to enable the USB Debugging [in the case of USB debugging]:

Go back and now access the Developer options menu, check 'USB debugging' and click OK on the prompt.

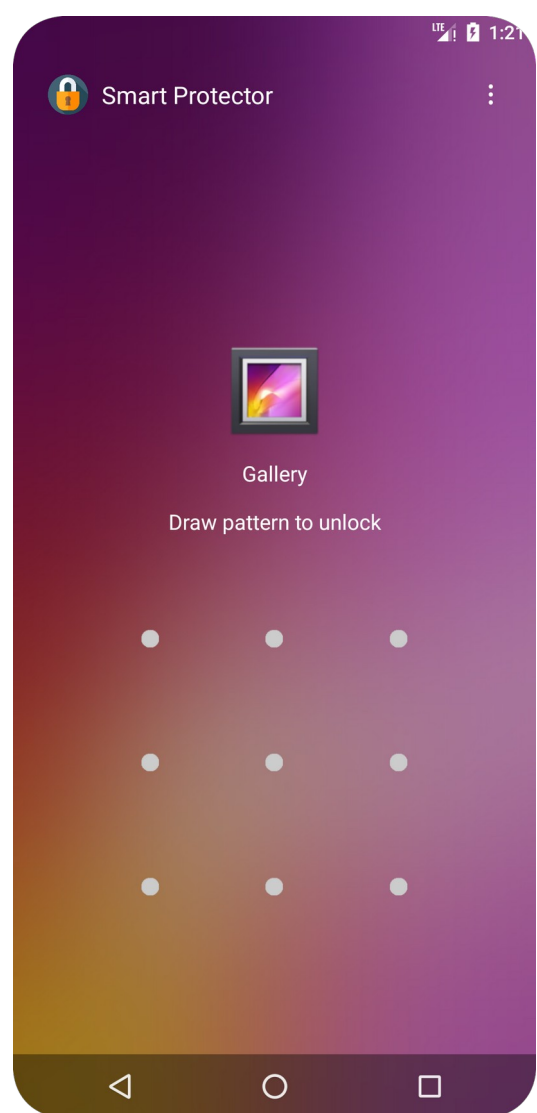
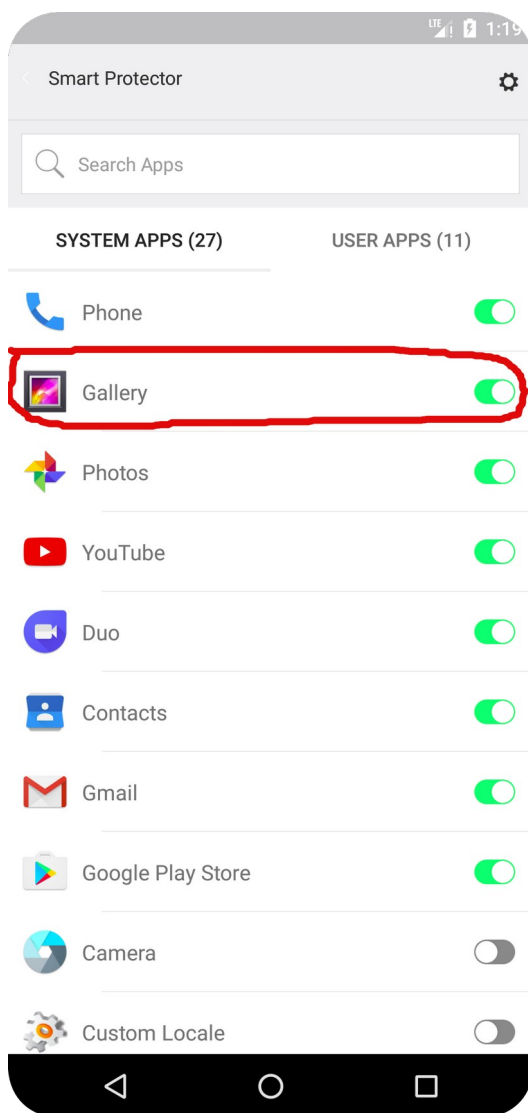
After connecting your device with enabling the USB debugging, Now you can run the app on your android device by clicking the run button of android studio,

Now select your device to run the app, Done!!! It time to Enjoy the app.

Results and Discussions

Now we must test the app to determine whether it is functioning properly. I used a structured approach to the testing. I put our Apps through their paces on a variety of cellphones. The results of all of the tests are positive.

For my application, I find the expected outcomes. The test result was quite positive. My anticipation is that our application will be simple to use and comprehend, with a better user interface.



Conclusion

Smartphones have now become a necessity, and as a result, they hold a great deal of sensitive information. As a result, it is now vital to keep data more secure. We've all been in circumstances when we've had to share our phone with people for various reasons. It may be to show them photographs or make a phone call, for example. However, we are hesitant to lend our phone to others because of the personal information it contains. To address such scenarios, we created our App Access Protector, which locks our key apps so that no one else may access them.

This Android Smartphone application was completed successfully. Our application work is finally finished. For the purpose of executing our program, we tried it on a variety of cellphones and found that it worked flawlessly in every way.

Our application has a very user-friendly design. The user interface is straightforward. Our application is completely trouble-free to use. We did our utmost to meet all of the standards. We hope that people will utilize our software and receive the best results possible.

Practical Implications

There are a several Practical Implications is possible for this work. Some of them are mentioned below:

1. It can be very useful for those parents who do not want their children to have access to all Applications.
2. Some of us are so engrossed in our Smartphones that we are unable to focus fully on our work or studies. In this case, we can disable the apps to which we are attached and focus on our work or studies. In this case, our App Protector could be a fantastic solution.
3. It will be effective for employees to prevent them from using unnecessary apps.

Scope of Future Work

We are currently working on our application. We wish to develop and tweak something about our application if we locate any future development opportunities. The following are the scopes of our planned developments:

1. In future, We will make every effort to make this application compatible with all smart phone platforms. (Android & IOS)
2. As you know that, in this time , We can just implement this App in our own device, But we can't lock Apps from any other device. So in future, we want to implement this functionality.