ANDROID APP-LOCKER FOR SECURITY AND APPLICATION MANAGEMENT

SIDDHARTH GUPTA
School of Computer Science And Engineering
GALGOTIAS UNIVERSITY
Greater Noida,U.P. ,INDIA.
siddharth_gupta.scsebtech@galgotiasuniversity.edu.in

PETERSON WINNY BABU
School of Computer Science And Engineering
GALGOTIAS UNIVERSITY
Greater Noida,U.P. ,INDIA.
peterson_winny.scsebtech@galgotiasuniversity.edu.in

MR.SHAILENDRA PRATAP SINGH
School of Computer Science And Engineering
GALGOTIAS UNIVERSITY
Greater Noida,U.P. ,INDIA.
shailendra.singh@galgotiasuniversity.edu.in

Abstract—**Mobile systems are emerging more and more as we interact with them everyday.Providing security and protection are one of the primary concerns raised by these systems.**

Android App-Locker systems are applications that allow users to lock specific apps or certain features of their Android devices. These systems provide an additional layer of security and privacy, especially for users who share their devices with others. Furthermore, overuse of certain applications are detrimental to one's mental health. These systems typically require users to set a password, PIN, pattern, or fingerprint as a security measure to access the locked apps or features. Users can then choose which apps or features to lock and customize the lock settings for each app or feature.

Android App-Locker systems also often come with additional security features, such as intruder selfies, which capture a photo of anyone who tries to access a locked app or feature with an incorrect password. Some systems also allow users to remotely lock or unlock their devices.

Overall Android App-Locker systems provide an easy and effective way for users to secure their devices and protect their sensitive data from unauthorized access.In this project we aim to make a user-friendly Android Based App-Locker Application using ANDROID STUDIO with features like Fake lock screen and intruder capture etc.,.

Keywords—

Androiddevelopment,android,Apps,AppLocker,Androidsecurity, Android SDK,Android Studio.

I. INTRODUCTION

HISTORY OF ANDROID

Andy Rubin,Rich Miner, a prominent backer of Wildfire Communications, Inc., and Nick Sears, who had served as VP, formed Android, Inc. in October 2003 at T-Mobile, and Chris White, who was in charge of Web-TV's strategy and interface improvements.
The organization's purpose was to create a sophisticated working framework that could be used for computerized cameras, but when they saw that the demand for that was insufficient, they focused their efforts on creating a working framework that would be comparable to Symbian and others.

Despite past accomplishments by the organizers, Android Inc. has worked covertly, revealing that it was attempting to programme cell phones.

Google purchased Android Inc. on August 17, 2005. Several people agreed that Android Inc. made the decision to enter the mobile phone market at that time. Rubin designed the
Linux-based mobile phone functioning foundation at the Android group drive.
The first Android phone developed was released in 2008 as **HTC Dream** which runs on **Android version 1.0** with features like web browser,email and Google Maps.
Later on Android is being developed by Google releasing different versions with new features and updates for users.

ANDROID ARCHITECTURE

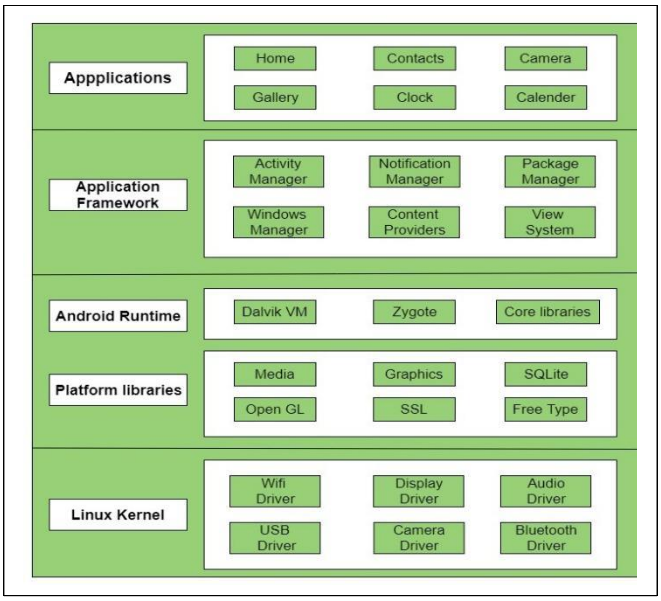


FIGURE-1:ANDROID ARCHITECTURE BLOCK DIAGRAM

ANDROID LIBRARIES

S.no	LIBRARY	DESCRIPTION
1	**Android.app**	responsible for accessing the model of application and is important for all Android apps.
2	**Android.content**	provides content for app and app components.
3	**Android.database**	responsible for accessing SQLite.
4	**Android.os**	responsible for internal process communications and system services for apps.
5	**Android.opengl**	provides a 3D graphics framework.
6	**Android.text**	used to manipulate data and render computer devices.
7	**Android.view**	used as the user interface's fundamental application.

TABLE-1:ANDROID LIBRARIES

ANDROID RUNTIME

Android runtime,a java virtual machine called **Dalvik Virtual Machine** optimized for android.
It enables every application to run its own process using its own instance of Dalvik Machine.
It also provides different libraries which are used in making android applications using standard **Java**

ANDROID APPLICATION COMPONENTS

Activities: These are basically single screen user interfaces which perform activities on screen.
Services: These are components that run in the background to perform long-runtime operations.
Broadcast Receivers: These components respond to broadcast messages from other application components
Content Providers: These provide data from one application to another when requested.

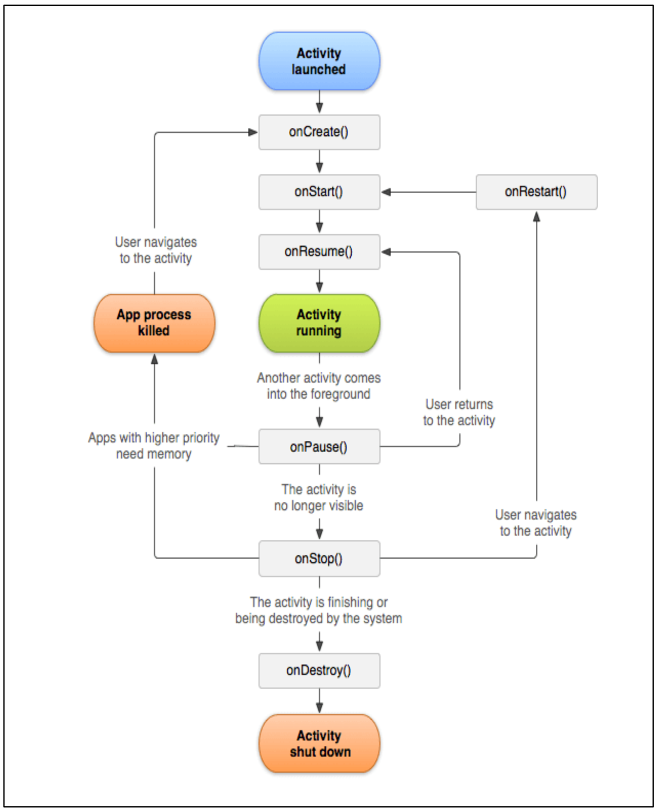

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

FIGURE-2:ANDROID ACTIVITY LIFECYCLE

ANDROID APPLICATION FRAMEWORK

S.no	SERVICE	DESCRIPTION
1	**Activity Manager**	Manages all parts of the application lifecycle and activity stack.
2	**Content Providers**	Enable apps to publish and share data with other applications.
3	**Resource Manager**	Provides access to non-code embedded resources such as strings, colour settings, and user interface layouts
4	**Notifications Manager**	Enables programs to show user alerts and notifications.
5	**View System**	An extendable set of views used to develop application user interfaces.

TABLE-2:APPLICATION FRAMEWORK

II. FORMULATION OF PROBLEM

As the technologies emerging,the usage of smart phones is increasing, raising a concern for security.One of the common security concerns arose was unauthorized access to applications on the smart phone like social media, gallery etc..,

Our aim is to make an application for Android devices which enables users to lock specific applications,to avoid unauthorized access to that application.We also integrated an observer feature for capturing the face of the intruder,and also some additional securities like fingerprint unlock and facial unlock features.To manipulate the intruders we created a fake lock screen feature to this application.

III. TOOLS AND TECHNOLOGIES USED

ANDROID SDK:Software Development Kit is a set of important tools required to build an android application.
We will be utilizing API 24: Android 7.0 (Nougat) so that we can cover 94.4% of devices.
Android SDK tools:It consists of a complete set of tools used for debugging and development of Android Applications.
Android SDK BUILD TOOLS:These tools are used to build the Android application's Binaries.These tools are used to build,run,test and debug android applications.
Android Emulator: An Emulator is a device that is used for simulation purposes.Android Emulator simulates an virtual Android Device on the system.
Android SDK platform tools consist of a command line tool called **ANDROID DEBUG BRIDGE** that helps to communicate with the android device and helps in installing and debugging Android applications.
FASTBOOT:It helps us to boot a device using system image and can also be used to flash a device with new image
SYSTRACE TOOLS are used to inspect and collect information about timing of the processes.

ANDROID STUDIO: It is an IDE for Android operating systems designed for android development projects**.**

IV. LITERATURE SURVEY

Android architecture has an extraordinary protection which made android devices popular and increased their demand exponentially.More than 64% of the worlds smartphone market was based on android followed by iOS (16%) and windows mobile(16%).With the increase in demand the security risks were also increased for Android devices (49% as of 2013).This is because of the usage of Android devices as a media for everyday tasks.The most attracted targets of risks were the Mobile Web Browser Applications on different devices of different operating systems[1].

In 2014 the Android made several changes to their device protection technologies like extended usage of hardware for protected authentication and Android sandbox functionality with a Access Control System built on Linux.
Android sandbox helps applications to run,the resources used by any application in android were limited.Android supports user-authentications like PIN,Pattern,Password,Fingerprint and Facial recognition.Android increased the security for application installations which warns the users before installing apps from unknown sources.[2]

Android supports the concepts of gated cryptographic keys for user authentication.These keys require key storage and service provider and user authenticators.Android's Gatekeeper supports user authentication through PIN/PATTERN/PASSWORD/FINGERPRINT.Gatekeeper works with keystore to support use of AuthTOKENS. Authentication in Android can be done by using different factors by combining them so as to increase the security level.Android uses Enrollment on the first boot to create AuthTokens for the initial user.It uses TRUSTED EXECUTION ENVIRONMENT (TEE) to authenticate the tokens.
The HMAC key was generated on successful enrollment of user and the key was shared with the authenticators.[3]

Marian Harbach presented a detailed review about smartphone locking systems,they proposed that the users who access their smartphones using PIN as their security take longer to unlock the phones but commit less errors than the users using pattern as their security.The data in their paper was limited because participants were all students and they were using the same device[4].

Biometrics in Android is one of the active researches going on.Use of biometrics for authentication is more effective than traditional securities like Passwords etc..,.[5]

As Android is used widely around the world the updates for new features are done every six to nine months by GOOGLE.The development in hardware and software features took an exponential turn.The new technologies were introduced as software updates for the Android devices.The growth in AI and Machine Learning did impact the Android device development making the devices capable to run programs at faster speeds.Linux kernel is the most important block of Android OS[6].

William Enck, Damien Octeau, Patrick McDaniel, and Swarat Chaudhuri in their article proposed better understanding of mobile device security by analyzing 1,100 popular free Android applications. The paper introduces the DED compiler, which obtains source code directly from the Android app's install image. Based on mobile app static analysis, the study illustrates the production and execution of a horizontal sample of 21 million lines of recoverable code.The rapid popularity of smartphones has resulted in the exponential growth of telecom networks. Any Smartphone with a cellular data plan can use mobile devices to access a wide range of media, financial, and business features[7].

V. PROJECT DESIGN

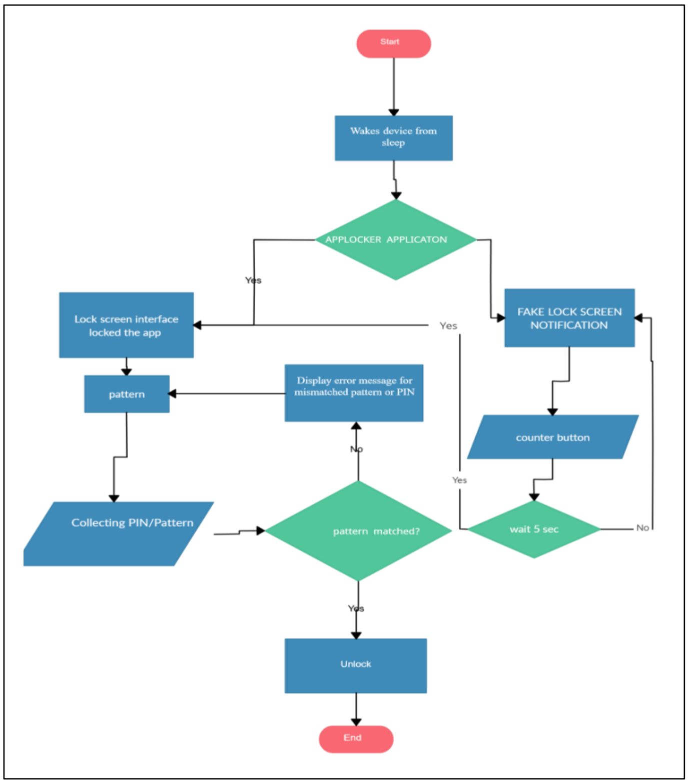


FIGURE-3:FLOWCHART FOR APPLOCKER

This flowchart describes the flow of activities and decisions of the working App-Locker application with the Fake Lock screen notification which is an application crash notification appears when the specified application which is already locked was accessed.

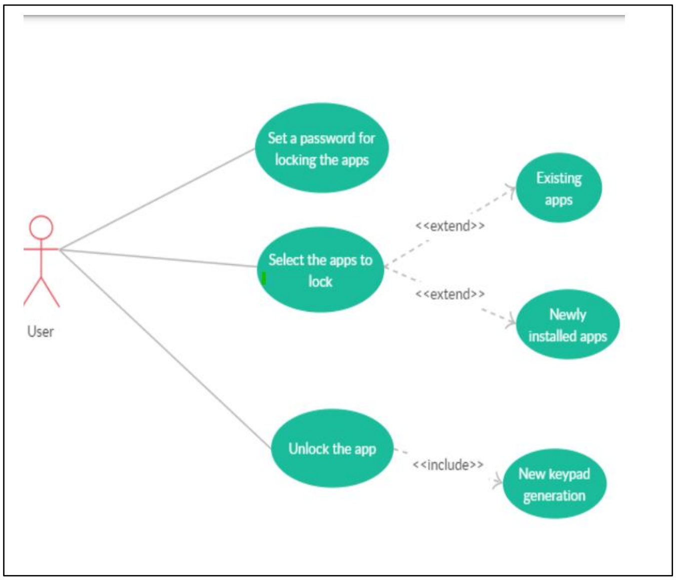


FIGURE-4:USE CASE DIAGRAM

This use case diagram describes the basic features of an App-Locker application and user’s interaction with it .

VI. MODULE DESCRIPTION

Passcode Module:

This module includes a keypad for entering the passcode and at the beginning of app setup the passcode will be set and stored.Later on the passcode can be altered or can be changed to a password using the application settings.

Utilizing shared preferences, we are able to store passwords within our app. Also, for resetting, we delete this same data. This is our requirement for memory management.

Fingerprint Module:

This module includes a scan for the fingerprint,the fingerprint stored in the device settings is used to authenticate the lock.

The biometric data stored in the system is directly accessed to provide authentication.

Default:

By creating an object of the FingerprintManager Class, we were able to handle STRONG and WEAK biometric authentication.

Brightness settings:

This module controls the screen’s brightness and screen auto rotation properties.

Using the Settings.System class we inherited extends Settings.NameValueTable

Lock apps,select:

The toggle switch which is used select the apps to lock and unlock.One button to disable app for ease of use.

System settings lock:

This is the module which locks the system settings so that it cannot be accessed to uninstall or stop the App-Locker from functioning.The System lock can also restricts the unwanted access to services like Wi-Fi, Bluetooth etc..,

Using the intent class, we created an object of the ACTION_SECURITY_SETTINGS to handle the use cases pertaining to system locking. Eg: User wants to fend off unwanted use of any application/service, they will have the control to do so. From a security standpoint, this gives them granular control of settings like connecting to the internet, Bluetooth, Aeroplane mode etc.

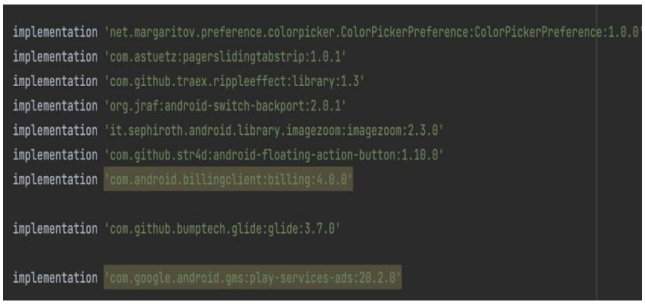
App manager settings:

Backup of apps on phone, delete them. Helps users reduce friction and manage preferences from one place.

Lock types:

This App-locker supports different kinds of lock types available in Android devices ,like standard features like PIN/PATTERN/PASSWORD and biometric features like Face-unlock and fingerprint unlock and other features like gesture unlock .

Libraries/Dependencies:



```
implementation 'net.margaritov.preference.colorpicker.ColorPickerPreference:ColorPickerPreference:1.0.0'
implementation 'com.astuetz:pagerslidingtabstrip:1.0.1'
implementation 'com.github.traex.rippleeffect:library:1.3'
implementation 'org.jraf:android-switch-backport:2.0.1'
implementation 'it.sephiroth.android.library.imagezoom:imagezoom:2.3.0'
implementation 'com.github.str4d:android-floating-action-button:1.10.0'
implementation 'com.android.billingclient:billing:4.0.0'

implementation 'com.github.bumptech.glide:glide:3.7.0'

implementation 'com.google.android.gms:play-services-ads:20.2.0'
```

FIGURE-5:LIBRARIES USED IN DEVELOPMENT

These are the libraries that are used for the development of different modules and functions in the application to make it work .

VII. RESULTS/CONCLUSION

This project named "ANDROID APP LOCKER FOR SECURITY AND APPLICATION MANAGEMENT " is an android operating system based application which is useful in securing devices based on AndroidOS from unauthorized access to certain apps which may contain sensitive information and secure the device by restricting the access to those applications.

These are the screenshots of the app-locker working on an android device.The application is stable and the modules works accordingly and the app as a whole is working properly .

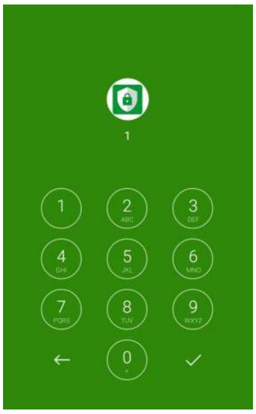
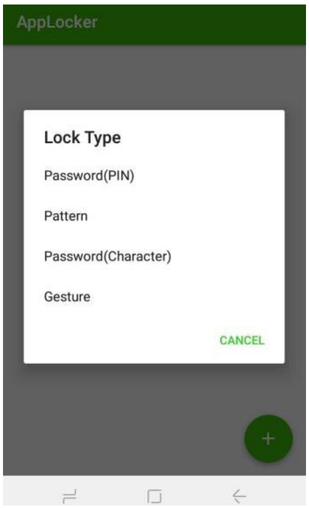


FIGURE-6:HOME SCREEN OF THE APPLICATION



AppLocker

Lock Type

Password(PIN)

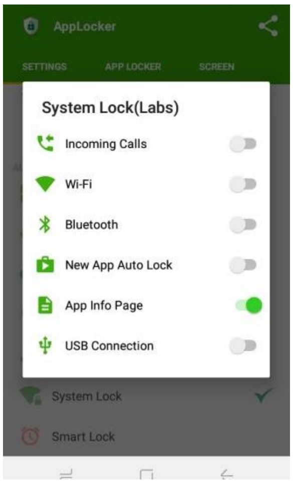
Pattern

Password(Character)

Gesture

CANCEL

FIGURE-7:LOCK TYPE SELECTION WINDOW

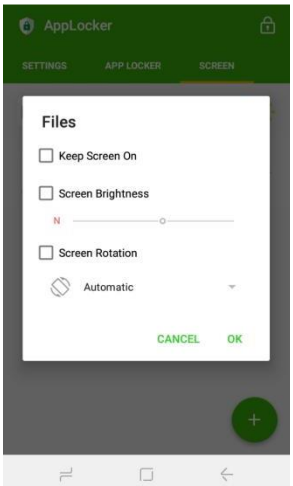


System Lock(Labs)

Incoming Calls ○
Wi-Fi ○
Bluetooth ○
New App Auto Lock ○
App Info Page ●
USB Connection ○

System Lock ✓
Smart Lock

FIGURE-8:SYSTEM LOCK SELECTION TOGGLE SCREEN



Files

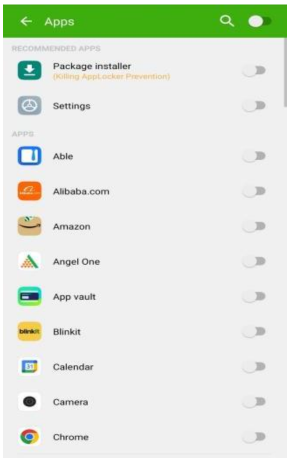
□ Keep Screen On

□ Screen Brightness

□ Screen Rotation

Automatic

FIGURE-9:BRIGHTNESS SETTING WINDOW



Package installer (Killing AppLocker Prevention)

Settings

Able

Alibaba.com

Amazon

Angel One

App vault

Blinkit

Calendar

Camera

Chrome

FIGURE-10:THE APPLICATION SELECTION TOGGLE SCREEN

The system was made convenient and reliable.The application allows user-friendly screens and easy to access options with out complex issues.This app was set to upgrade and addition of different modules will be done in the future.
The application was tested on an android device and works efficiently.

REFERENCES

[1]MAHMOOD, SARDASHT & AMEN, BAKHTIAR & NABI, REBWAR. (2016). MOBILE APPLICATION SECURITY PLATFORMS SURVEY. INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS. 133. 40-46. 10.5120/IJCA2016907736.

[2]https://www.android.com.

[3]https://source.android.com/docs/security/features/authentication

[4]Harbach, Marian; De Luca, Alexander; Egelman, Serge (2016). *[ACM Press the 2016 CHI Conference - Santa Clara, California, USA (2016.05.07-2016.05.12)] Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16 - The Anatomy of Smartphone Unlocking. , (), 4806–4817.* doi:10.1145/2858036.2858267

[5]F-LOCKER: AN ANDROID FACE RECOGNITION APPLOCKER USING LOCAL BINARY PATTERN HISTOGRAM ALGORITHM ALA Ramos, MAM Anasao, DB Mercado, JA Villanueva, CJA Ramos, AAT Lara, CNA Margelino Institute of Computer Studies, Saint Michael's College of Laguna, Philippines

[6]https://digitalscholarship.unlv.edu/thesesdissertations/2959

[7]A Study of Android Application Security William Enck, Damien Octeau, Patrick McDaniel, and Swarat Chaudhuri Systems and Internet Infrastructure Security Laboratory Department of Computer Science and Engineering The Pennsylvania State University.