

A Project Report

On

**App Locker**

Submitted To

Jaypee University Of Information Technology, Solan



In fulfillment Of Project Part

Of

Bachelor of Technology In Computer

Science

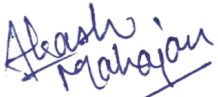
By

**Akash Mahajan (161250) & Rishi Rai (161259)**

Under the guidance of **Mr. Amol Vasudeva**

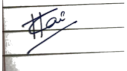
## CERTIFICATE

I hereby declare that the work presented in this report entitled APP LOCKER in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of our work carried out over a period from August 2019 to December 2019 under the supervision of Mr. Amol Vasudeva, Professor of Computer Science & Engineering Department.

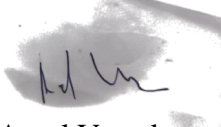


Akash Mahajan, 161250

Rishi Rai, 161259



This is to certify that the above statement made by the candidate is true to the best of my knowledge.



Mr. Amol Vasudeva

Asst. Professor (Senior Grade) Computer and Science Engineering Dated:

## **ACKNOWLEDGEMENT**

Keep away from people trying to demean your ambitions. Small people do that all the time, but the really great thing makes you feel like you too can be great. We are taking this opportunity to share my heartfelt appreciation and profound gratitude to all those people who have extended their wholehearted support and helped me effectively achieve this mission. Special thanks to Mr. Amol Vasudeva for all the support and encouragement which he has provided me during my training at every level. I am very grateful and deeply indebted to Mr Amol Vasudeva for their strict supervision, constant encouragement, inspiration and guidance. Serving with him had been an enriching experience.

## INDEX

<u>CONTENTS</u>	<u>PAGE NO.</u>
CERTIFICATE	i
ACKNOWLEDGEMENT	ii
CERTIFICATE	iii
LIST OF ABBREVIATIONS	iv
LIST OF FIGURES	v
LIST OF TABLES	vi
ABSTRACT	vii
<b><u>CHAPTER 1. INTRODUCTION</u></b>	<b><u>1</u></b>
1.1 INTRODUCTION	1
1.2 PROBLEM STATEMENT	16
1.3 OBJECTIVES	17
1.4 METHODOLOGIES	18
1.5 ORGANISATION	19
<b><u>CHAPTER 2. LITERATURE SURVEY</u></b>	<b><u>20</u></b>
2.1 ANDROID PRIVACY	20
2.2 ANDROID DEVELOPMENT	21
2.3 ANDROID SECURITY	22
<b><u>CHAPTER 3. SYSTEM DEVELOPMENT</u></b>	<b><u>23</u></b>
3.1 SOFTWARE USED	23
3.2 HARDWARE REQUIREMENTS	23
3.3 ABOUT SOFTWARE AND TOOLS	23
3.4 DESIGNING AND IMPLEMENTATION	27

**CHAPTER 4. PERFORMANCE ANALYSIS** **29**

---

4.1 HARDWARE REQUIREMENTS 29

4.2 RESULT ANALYSIS 29

**CHAPTER 5. CONCLUSIONS** **38**

---

5.1 CONCLUSION 38

5.2 FUTURE PROSPECTUS 39

REFERENCES 40

## **LIST OF ABBREVIATIONS**

- **ASCII** - American Standard Code for Information Interchange
- **SDK** - Software Development Kit
- **APP** - Application
- **PDE** - Plug-ins Development Environment

## LIST OF FIGURES

<b>FIGURE NUMBER</b>	<b>FIGURE NAME</b>	<b>PAGE NUMBER</b>
1.	Architecture Of Android System	10
2.	Linux kernel	10
3.	Libraries	11
4.	Android Runtime	12
5.	Application Framework	12
6.	Applications	12
7.	Lifecycle Of An Activity	13
8.	PDE	25
9.	Eclipse Versions	26
10.	Use Case Diagram	27
11.	Flow Chart	28
12.	App Info	30
13.	Homepage	31
14.	Enabling Pin Lock	32
15.	Setting Pin	33
16.	Deleting/Updating Pin	34
17.	Pattern Lock	35
18.	Entering Pin	36
19.	Lock screen	37

## LIST OF TABLES

<b>TABLE NUMBER</b>	<b>TABLE NAME</b>	<b>PAGE NUMBER</b>
1.	Android Versions	13
2.	Eclipse Versions	26



## **ABSTRACT**

The App Locker project is an android-based application that allows individuals to stop unauthorized use of their smartphone powered by android. The project's aim was to make user-friendly method for the safety of their smart-phone powered by android. Within this project report we begin with the introduction in which we describe the background and then clarify design gives you specifics of platform in which android operates. Then we inserted device screenshots such as sending pin to the cell phone, question & response on protection, etc. After that comes the Test Component, as we know the application doesn't have a complicated or high end feature, we do the application's manual testing on the correct computer and on the Android Virtual System, where the application runs with ease successfully. Then we discuss the results in which we try to remind ourselves of all the function we have said to develop. After wards we discuss the conclusion that we get following the development of this application with the application's future prospects. Then we write down all the names of places where we get resources, books.

# **CHAPTER 1. INTRODUCTION**

## **1.1 INTRODUCTION**

### **1.1.1 HISTORY**

Android, Inc. was founded in October 2003 by Andy Rubin, a fellow Danger benefactor, Rich Miner, a leading supporter of Wildfire Communications, Inc., Nick Sears once served as VP at T-Mobile, and Chris White, who headed WebTV 's plan and interface enhancement.

The organization's goal was to build up an advanced working framework that could be used for computerized cameras, but when they realized that the demand required for that was not big enough to grab, so they received their efforts to produce the working framework that would be equivalent to Symbian and others.

Despite past achievements by the organizers, Android Inc. has worked furtively, uncovering that it was taking aim at programming cell phones. On August 17, 2005, Google acquires Android Inc. After Google Securing, Android Inc. key workers Rubin Miner and White remained in the organization. Many agreed that Android Inc. had decided to enter the mobile phone industry at that time. The Linux part controlled cell phone working framework was created by Rubin at Android group drive.

With the goal of providing an adaptable and upgradable approach, Google demonstrated the working framework for handset producers and transporters. Google has made progress with h / w segment and s / w accomplices and moved carriers to open up more co-activity. The managed mobile phone stage of Android based on the Linux component rendition 2.6.25 is propelled on 5 November 2007. HTC Dream was first discharged on October 22, 2008 as a monetarily accessible android-controlled Smartphone. In 2010 Google propelled gadget arrangement named-Nexus

### 1.1.2 ARCHITECTURE OF ANDROID SYSTEM

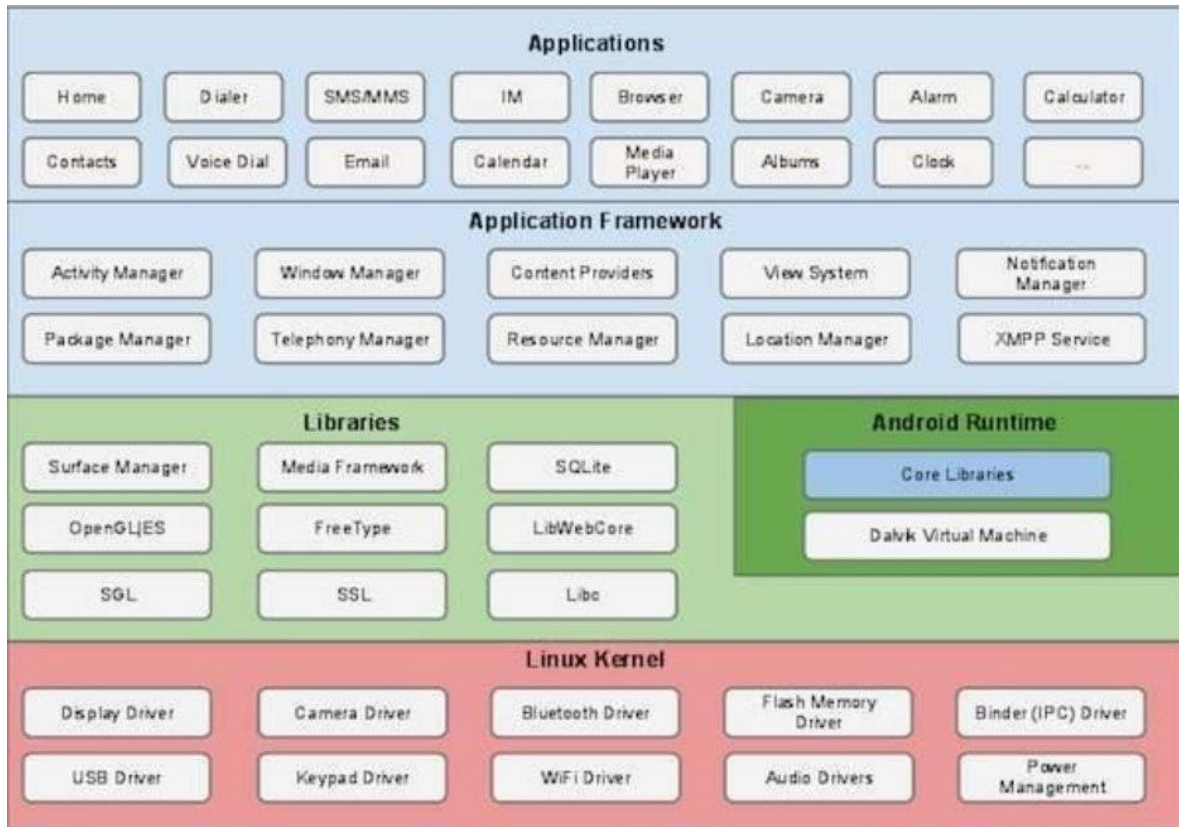


Fig.1 ARCHITECTURE OF ANDROID SYSTEM

### 1.1.3 LINUX KERNEL



Fig.2 LINUX KERNEL

### 1.1.4 LIBRARIES

These are some libraries in android OS.



**Fig.3 LIBRARIES**

### 1.1.5 ANDROID LIBRARIES

Android.app – Gives access to the model of the application and is the cornerstone of all Android applications.

- Android.content – It provides the content accessible throughout the app and app components.
- Android.database – It is used to access the SQLite.
- Android.opengl – Provides framework for 3D graphics.
- Android.os – Provides the apps like interprocess communication and system services.
- Android.text – Used on a devices computer to render and manipulate data.
- Android.view – Fundamental application user interface building block.

### 1.1.6 ANDROID RUNTIME

It is basically a application runtime environment.



**Fig.4 ANDROID RUNTIME**

**1.1.7 APPLICATION FRAMEWORK**

Main Resources –

- Activity Manager -- It controls lifecycle and activity stack aspects of the application.
- Resource Manager – It provides the tools like strings , color settings etc.



**Fig.5 APPLICATION FRAMEWORK**

**1.1.8 APPLICATIONS**

Basic applications for android OS.



**Fig.5 APPLICATIONS**

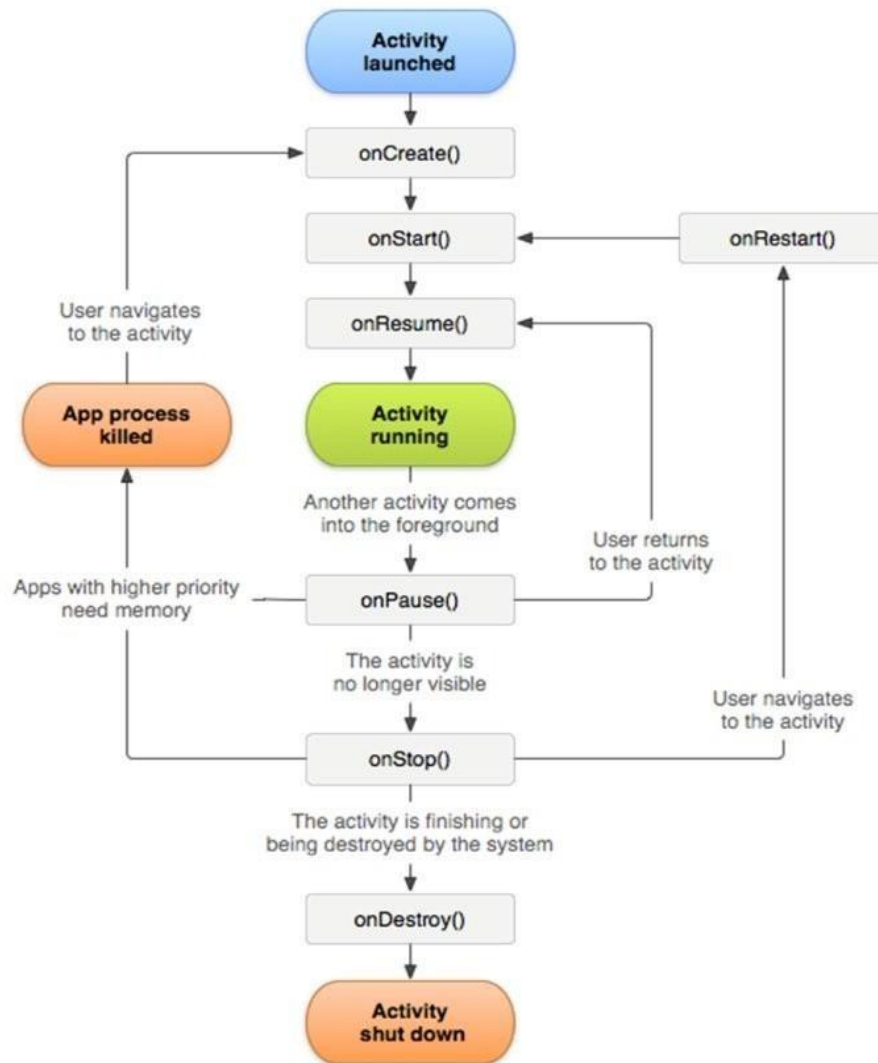
### 1.1.9 ANDROID BUILDINGBLOCKS

There are many building blocks which is very helpful to run any application by which the application enter in the runtime environment.

#### ACTIVITIES

Activity is basically a single screen where user interface is there .

Android activity is the subclass of ContextThemeWrapper class.



**Fig.6 LIFECYCLE OF AN ACTIVITY**

## **SERVICES**

It is also a part of android building block where the service is an entity of activity. It starts when an activity starts by calling the function startService().

It runs in the background of an OS until we call bind function to terminate the service.

### **Started Service**

It starts when we call the function as I mentioned above in services.

### **Bound Service**

To bind the service we have to call the bind function otherwise it will continue to run in the background.

#### **1.1.10 CONTENT PROVIDERS**

A manufacturer of the material works with a growing use knowledge agreement. A manufacturer of a product supplies the client knowledge with an structured gui. Your program will distribute knowledge to specific applications through a material supplier. Android comprises a SQLite database that is commonly used in connection.

#### **1.1.11 BROADCAST RECEIVERS**

Broadcast receivers respond fundamentally to communication signals from different applications or from the system itself. Throughout every level, such communications are named occasions or targets. Of example, apps will often start interacting and notify different applications that any information has been transmitted to the gadget and is available for them to use, and there is a exchanged collector that can grab that communication and start fitting operation.

### **1.1.12 INTENT**

Aims are offbeat messages that allow the device to claim functionality from certain sections of Android, such as from administrations or exercises. An application can call a part directly (express Intent) or request that the enlisted segments be assessed by the Android framework depending on expectation information (some objectives). For example, the application could execute information sharing through a plan, and all segments that would allow information sharing would be accessible for the client to choose from. Applications file with an assumption on a schedule. Channel Intents helps an Android app to launch and function with segments from other Android applications.

#### **USING INTENT TO LAUNCH THE ACTIVITIES**

- Passing Intent to startActivity) (or startActivityForResult) (will start an activity (or give it something fresh to do).
- You can start a service by passing an Intent tostartService) ((or give new instructions to an ongoing service). And you may connect to the provider by forwarding bindService) (to an Objective.
- Use methods such as sendBroadcast), (sendOrderedBroadcast), (or sendStickyBroadcast) (to initiate a broadcast.
- You can do a content provider query by calling a Content Resolver query.



## **1.2 PROBLEM STATEMENT**

Our main aim is to develop a android app named **APP LOCKER** that can protect our devices from unauthorized access. The project's goal was to make user-friendly method to secure their smartphone powered by android. Also we discussed that we will integrate a Advertisement system with this app which able us help business communities. This app allows emerging brands and products to promote themselves in their LOCALE economically & directly through people while the people earn money through the ads.

## **1.3 OBJECTIVES**

The project's primary purpose is to plan, build and evaluate an Android "Lock Screen" program. The program suggested is to encourage citizens to avoid illegal usage of their smart-phone powered by android. The project's goal was to make user-friendly method to secure their smartphone powered by android. Our priorities in the production of the proposed Android framework are to:

- Precise: The degree of precision would be better under the new method. Each procedure should be performed properly, meaning that all input from the center is reliable.
- Access: The Lock Screen has a special gui designed to enable you to work quickly and efficiently.
- Immediate information retrieval: the key purpose of the new program is to ensure fast and effective information retrieval.

## **SCOPE**

- Lock screen Guide, simple to use.
- Complete exposure to all the Lock processes.
- Big friendly button, great to get some operation running quickly. Big email, genius in a hurry for bolt.
- Locking the screen just with a touch. Orientation click. Easily lock screen without the screen lock button hurting. Options easy to access with just a touch on the screen
- Shiny, reflective, catchy app screen. Horizontal mode orientation.
- Provide Security to your Android Smart-Phone.
- Use low battery in running since it is not using most of service.

## **WHY**

### For Brands:

The most important factor for the success of an emerging brand is recognition. For people to get to know a particular brand or a product it is necessary for them to advertise it. The cheapest way to advertise a product is through social media. But just to create a page on an online forum is not enough, unless PEOPLE endorse it. The advertising through THIS APP is not through a static page but through PEOPLE itself & not just through one platform but as many as possible to give a wider coverage.

### For People:

WHY? WHY NOT? Why not earn some extra bucks when you can? Why not be in the spotlight? Why not stand for something & be recognized?

## **1.4 METHODOLOGY**

This section is about the working of our application step by step from the beginning and How we collect the data from research papers and what approach we select? After reading several research papers, first we learnt about the introduction about the Android operating system. Then we learnt about the tools and software we are going to use in our project. And then we read research papers on Android security and know about the implementation of security in our application. Afterwards we learnt about the Eclipse Juno which is a software used to develop different types of applications. And then we learn about the tool we used for this project which is Android SDK(Software Development Kit). And then there is the first part of software development phase which is Requirements Gathering. After that we design UML diagrams like Use Case Diagram and Flow Chart Diagram. After the implementation of our project successfully we are ready to test our application.

### **STEP BY STEP WORKING OF APP LOCKER:**

**Step 1:** First user will install the application from a open source or a link.

**Step 2:** After installation user will set the lock based on the methods like pin, password or pattern.

**Step 3:** After that we can set that password on selective applications on smartphone.

**Step 4:** User is then ask to confirm the password or the pattern.

**Step 5:** After user confirmation user is ready to use the application.

**Step 6:** Whenever the phone of user get on the advertisement comes on the background of the lock screen.

**Step 7:** The advertisement is in such a way that it does not blur then dots for pattern.

**Step 8:** User draws then pattern or enter the password to unlock the screen.

## **1.5 ORGANISATION**

- Chapter 1 is about the Introduction of our project. It includes the history of the android applications, android development. It is about the architecture of android system. Then we mention about the problem statement and the objectives of our project. After that we discuss about the methodologies we have used in developing the application.
- Chapter2 is about the Literature Survey in which we include the research papers we have read and the several article about the Android SDK, Android Security, Android App Development.
- Chapter 3 is about the System Development in which we specify the requirements, Designing, Modeling And implementation. Also we have mentioned about the tools and software we have used in our project.
- Chapter 4 is about the Performance Analysis part in which we have mentioned the result we have obtained after implementation. We also detailed here about the various aspects about our project. Here the performance analysis have been discussed.
- Chapter 5 is conclusion about the project. Here we derived many conclusions about the project. Here we also put together many things which we learned during the making of project.
- Reference section is about mentioning various references and inspirations. We did this section to acknowledge the various works by the different people in the field of android development,
- Security of applications, advertising etc. In making of this project we came across various books which are essential for making of this project. We are vary thankful to the various resources which enable to think and implement deeply.

## **CHAPTER 2 .LITERATURE SURVEY**

### **2.1 ANDROID PRIVACY**

**Title:** MOBILE APPLICATION SECURITY ON ANDROID

**Author(s):** Jesse Burns

**Date of Conference:** June, 2009

**Published in:** Black Hat USA talk

**Conference Location:** United States of America

Android has an extraordinary model of protection which focuses on putting the client in control of the gadget. Android gadgets may be that not all come from one spot, but the open idea of the stage takes restrictive expansions and changes into consideration. These expansions can help or could interfere with safety. Having the option of breaking down Android circulation is a significant step forward in ensuring data on that framework in this manner. This paper brings the peruser across Android's protection model, covering a large amount of core security elements and how they protect assets. This foundation data is basic to having the option to comprehend the instruments Jesse will introduce at Black Hat, and the kind of data you can gather from the apparatuses, and from any running Android dissemination or application you wish to examine. Before perusing this paper you should as of now be comfortable with Android's essential design and significant reflections including: Intents, Activities, Broadcast Receivers, Services, Content Providers and Binder<sup>1</sup>. On the off chance that you haven't yet experienced the greater part of these stage highlights you should begin with the Notepad<sup>2</sup> instructional exercise. As Android is open source you ought to likewise have this code accessible to you. Both the java and C code is basic for understanding how Android functions, and is definitely more point by point than any of the stage documentation. As examined operating at a profit Hat talk, the distinction between what the documentation and the source say is regularly basic to understanding.

## **2.2 ANDROID DEVELOPMENT**

**Title:** A Study of Android Application Security

**Author(s):** William Enck, Damien Ocateau, Patrick McDaniel, and Swarat Chaudhuri

**Date of Conference:** 13-15 August 2014

**Published in:** The Pennsylvania State University

**Conference Location:** Pennsylvania, United States of America

### **Summary:**

This paper aims to better understand mobile device protection through the analysis of 1,100 common free Android applications. Paper introduces the dedde compiler, which retrieves source code directly from its install image from the Android app. The study shows the creation and execution of a horizontal sample of 21 million lines of recovered code based on mobile app static analysis.

Analysis of widespread and extensive use / misuse of personal / phone identifiers

Networks of advertising and analytics are breaking through. There is no evidence in paper, however, of malware or exploitable vulnerabilities in the applications being studied. By providing guidance for future research, Paper concludes by considering the consequences of these preliminary results.

The exponential development of Smartphone has triggered a revival of telecom networks. Go-anywhere devices deliver a wide variety of media, financial and business tools to any Smartphone with a cellular data plan. Software markets such as Apple's App Store and Google's Android Market offer point and click on access to hundreds of thousands of free and paid apps.

## **2.3 ANDROID SECURITY**

**Title:** Android Security Review

**Date of Conference:** 15 December 2014

**Published by:** Google Inc.

**Conference Location:** Mountain View, California, United States

### **Summary:**

The Android framework made several major changes in device protection technologies in 2014, including allowing complete disk encryption deployment, extending the usage of hardware-protected authentication, and improving the Android sandbox functionality with a Compulsory Access Control System (MAC) built on SE Linux. Developers were equipped with improved resources to identify and respond to weaknesses in infrastructure, including the no go to project failure and the protection provider. We also given continuous help to software makers to repair security vulnerabilities in products, including the creation of 79 security fixes, and improved the opportunity to resolve possible vulnerabilities in key areas, As in the case of

Google's Android security services increased user protection, and increased visibility in attempts to exploit Android. Ongoing analysis by Verify Apps showed that attempts to offer potentially dangerous applications (PHAs) remained at low rates during 2014, fewer than 1% of all smartphones had a PHA installed Less than 0.15% of Google Play-only activated smartphones had a PHA installed.

Expanded protection in Verify Apps and safe browsing now also gives insight into vulnerabilities in the platform, network, and browser that affect Android devices. Exploitation attempts have been tracked for multiple vulnerabilities, and the data show no evidence of widespread use of Android devices.

## **CHAPTER 3 .SYSTEM DEVELOPMENT**

### **3.1 SOFTWARE USED**

S/W: Eclipse Juno

Operating System: Windows 10 (x64)

Platform: Android SDK

### **3.2 HARDWARE REQUIREMENTS**

Android Operating System (2.3.3-8.3.3)

RAM: 512 MB

ROM: 4 GB

### **3.3 ABOUT SOFTWARE AND TOOLS USED**

#### **ECLIPSE JUNO**

Eclipse is an IDE used in computer science technology as an open computing interface. It has a major workspace, and an extensible application configuration plug-in framework.

This IDE is written mostly in programming language JAVA. And it is mostly used for Java applications development, but it can also develop other applications in other programming languages.



It can also be used to develop LaTeX docs, and s / w mathematics packages.

The origin codebase is IBM VisualAge. The Eclipse SDK, a Software Development Package with JAVA Programming Tools.

Users have many options that include plug-in extension by installing them in the Eclipse Platform itself. In this program, there's another function in which we can dynamically interrupt plug-ins and label them bundles (OSGI).

It is free and open-source software on the Eclipse Software Development Kit, and released under the terms of the Eclipse Public License.

### **Platform Architecture of Eclipse:**

Plug-in definition is the fundamental framework for Eclipse platform. Well plug-ins are standardized programming / data packages that are primarily responsible for the operation of the program. These functionalities are addressed as code libraries, extensions to platforms or even as documentation.

Eclipse SDK is a simple framework with a package of two main resources that are helpful in designing plug-ins.

### **JDT**

Stands as Java Development Tools. It helps the implementation of Java development environment.

### **PDE**

Stands as Plug-in Developer Environment. It adds specialized tools that streamline the development of plug-ins and extensions.

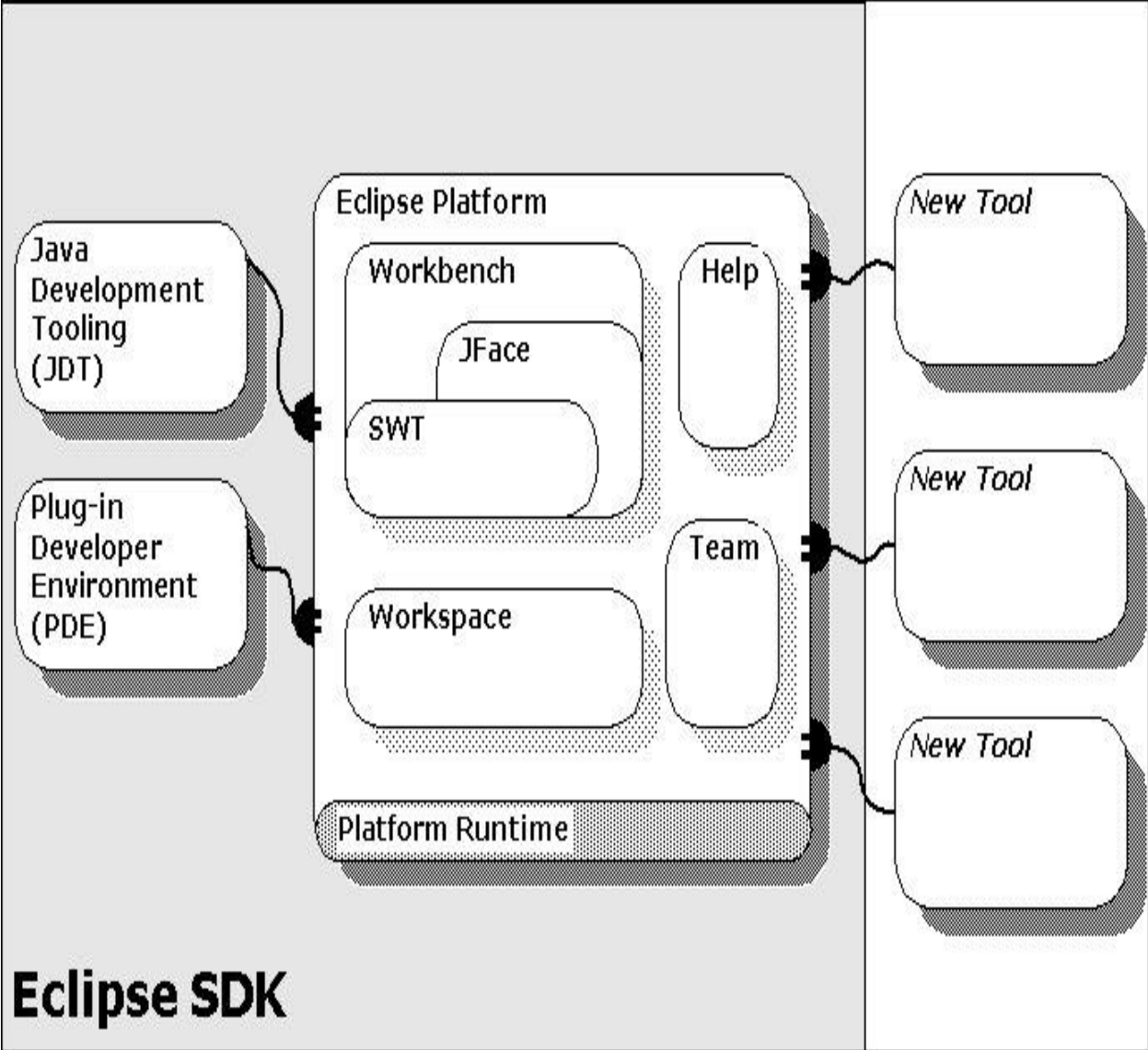


Fig.7 PDE

## Eclipse Versions List:

Codename ↕	Date ↕	Platform version ↕	Projects ↕
N/A	21 June 2004	3.0 <sup>[14]</sup>	
N/A	28 June 2005	3.1	
Callisto	30 June 2006	3.2	<a href="#">Callisto projects</a> ↗
Europa	29 June 2007	3.3	<a href="#">Europa projects</a> ↗
Ganymede	25 June 2008	3.4	<a href="#">Ganymede projects</a> ↗
Galileo	24 June 2009	3.5	<a href="#">Galileo projects</a> ↗
Helios	23 June 2010	3.6	<a href="#">Helios projects</a> ↗
Indigo	22 June 2011	3.7 <sup>[Notes 1]</sup>	<a href="#">Indigo projects</a> ↗
Juno	27 June 2012	4.2 <sup>[15]</sup>	<a href="#">Juno projects</a> ↗
Kepler	26 June 2013	4.3	<a href="#">Kepler projects</a> ↗
Luna	25 June 2014 (planned)	4.4	<a href="#">Luna projects</a> ↗

■ Old version   ■ Older version, still supported   ■ Latest version   ■ Future release

**Fig.8 ECLIPSE VERSIONS**

## ANDROID SDK

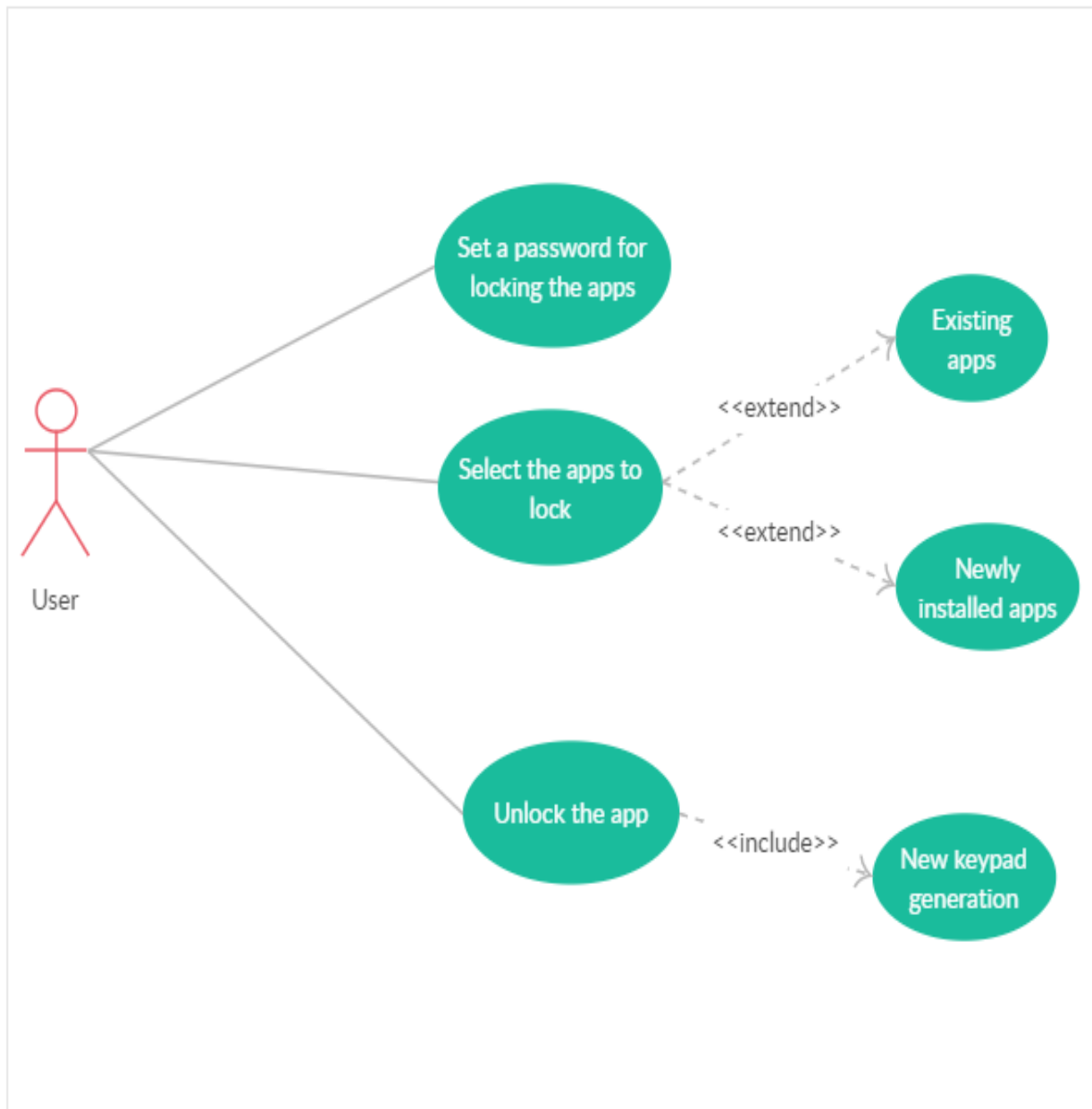
As I described above, Android SDK is a package for mobile development and is a collection of tools used to build Android platform applications.

Here are few things Android SDK includes:

- Libraries Needed
- Serial emulator
- An emulsifier
- Applicable Android application software interfaces (APIs) documentation;
- Example Quell Script
- Android OS-Tutorials

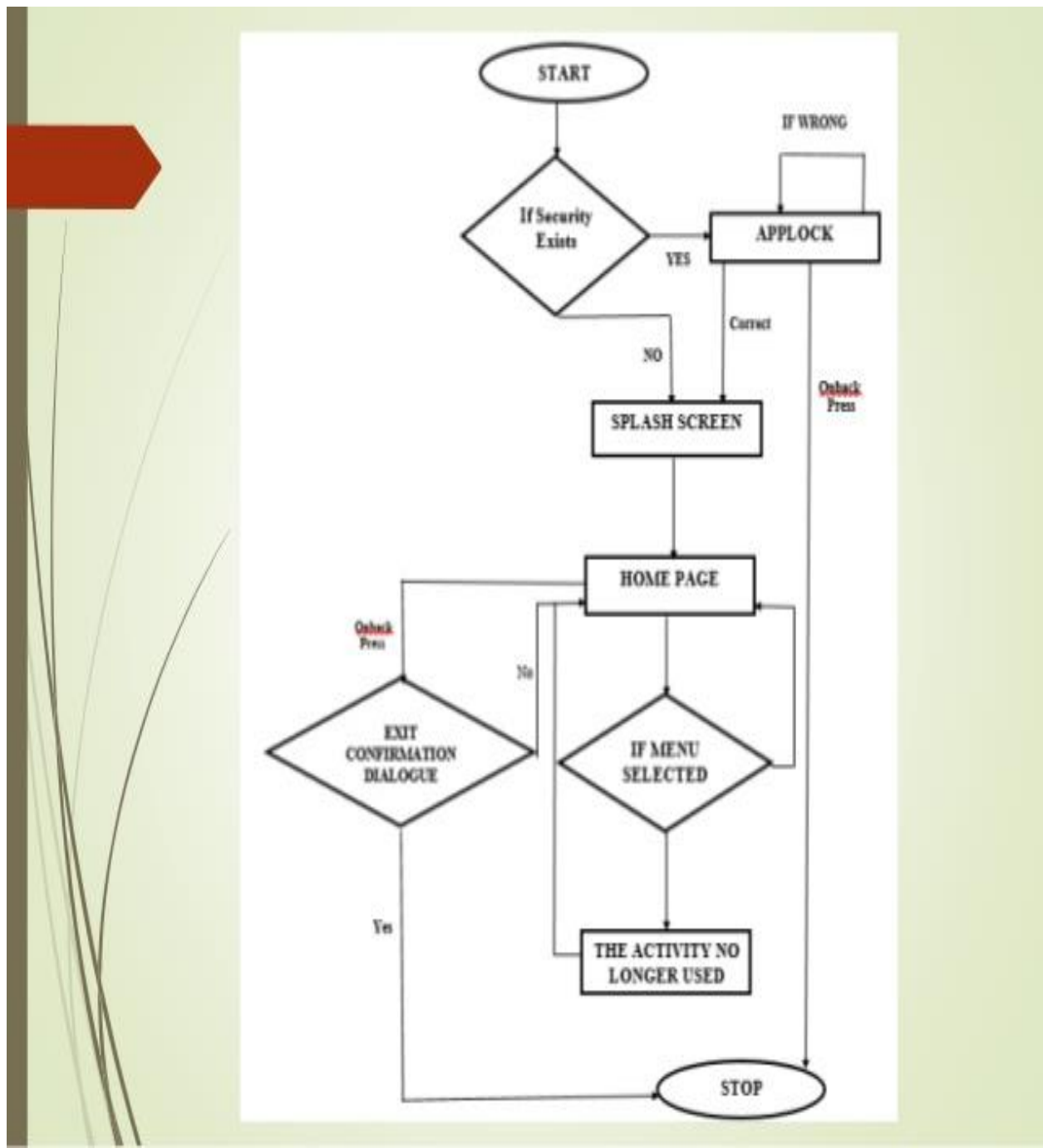
### 3.4 DESIGNING AND IMPLEMENTATION

#### USE CASE DIAGRAM



**Fig 9 .USE CASE DIAGRAM**

## FLOW CHART



**Fig.9 FLOW CHART DIAGRAM**

## **CHAPTER 6 .PERFORMANCE ANALYSIS**

This section is about the result we have obtained during the development period.

We will discuss about the various results we have received by the help of figures.

### **6.1 HARDWARE REQUIREMENTS FOR THE APPLICATION**

Operating System: Android 4 or above

RAM: 256 MB or above

ROM: 1 GB or above

### **6.2 RESULT ANALYSIS**

We have facing problem in stable mode for this application, so we have to switch to developer mode to use this application efficiently.

Even after switching into the developer mode the running of the application is not that smooth.

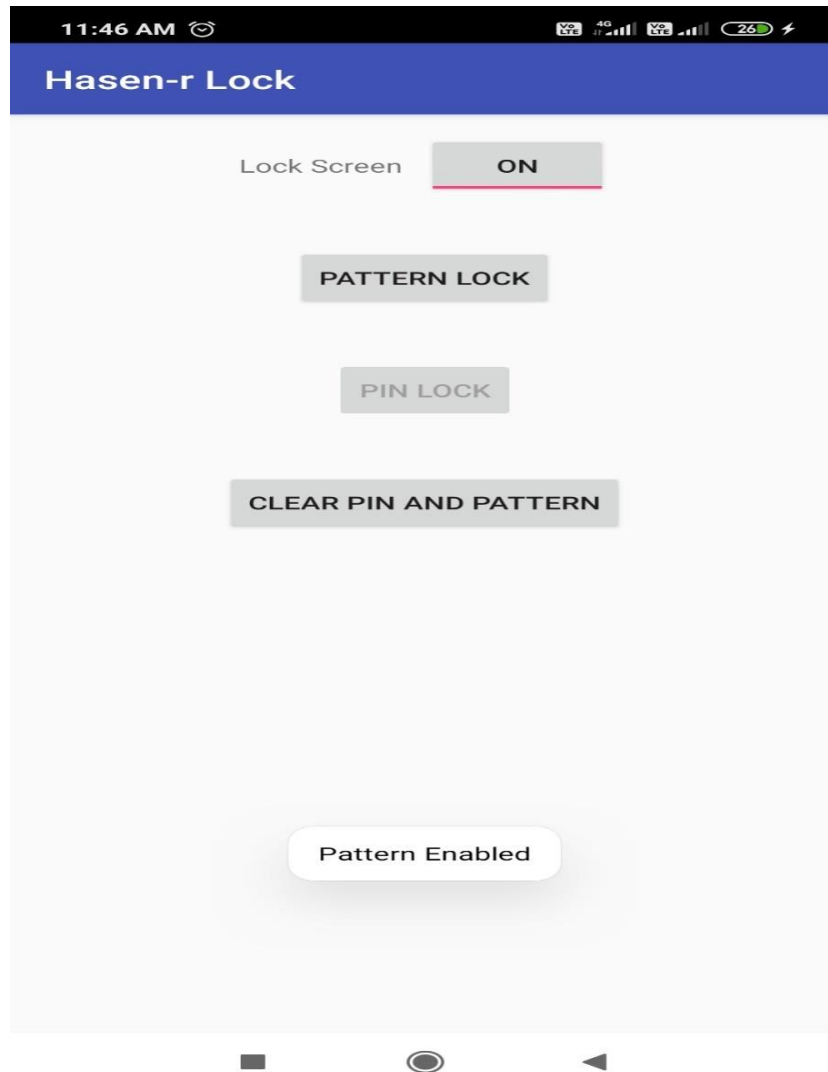


**Fig.10 APP INFO**

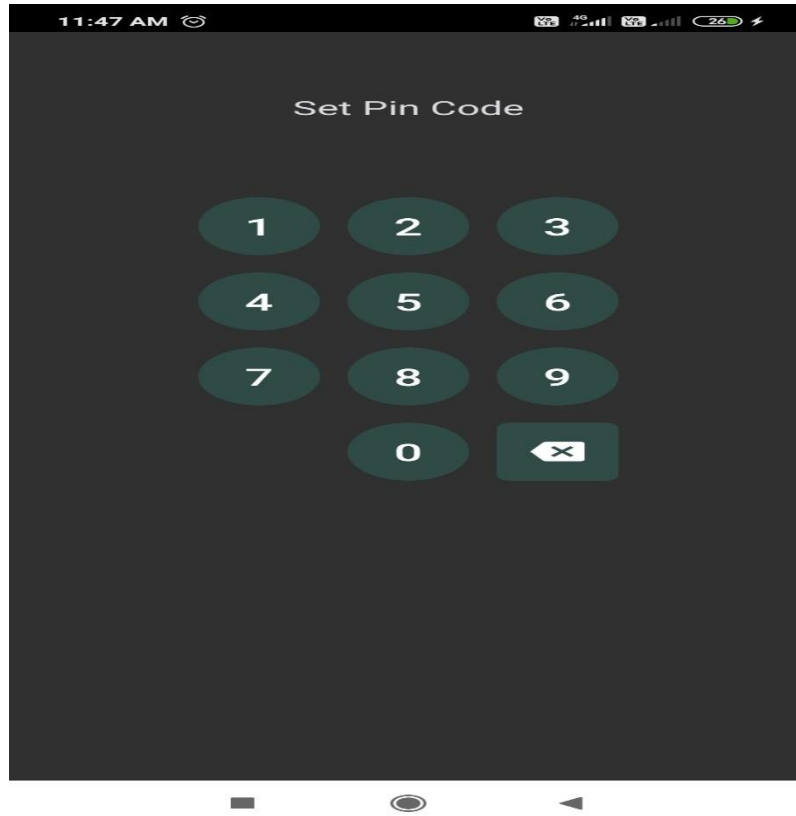


**Fig.11 HOMEPAGE**

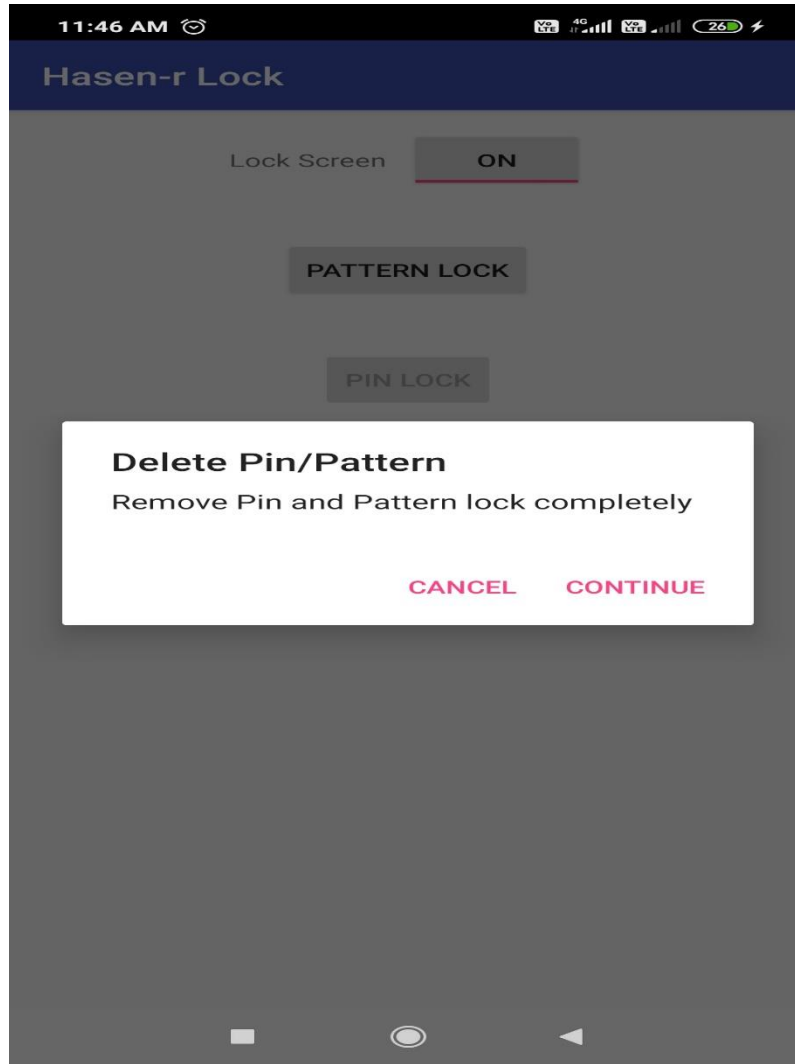




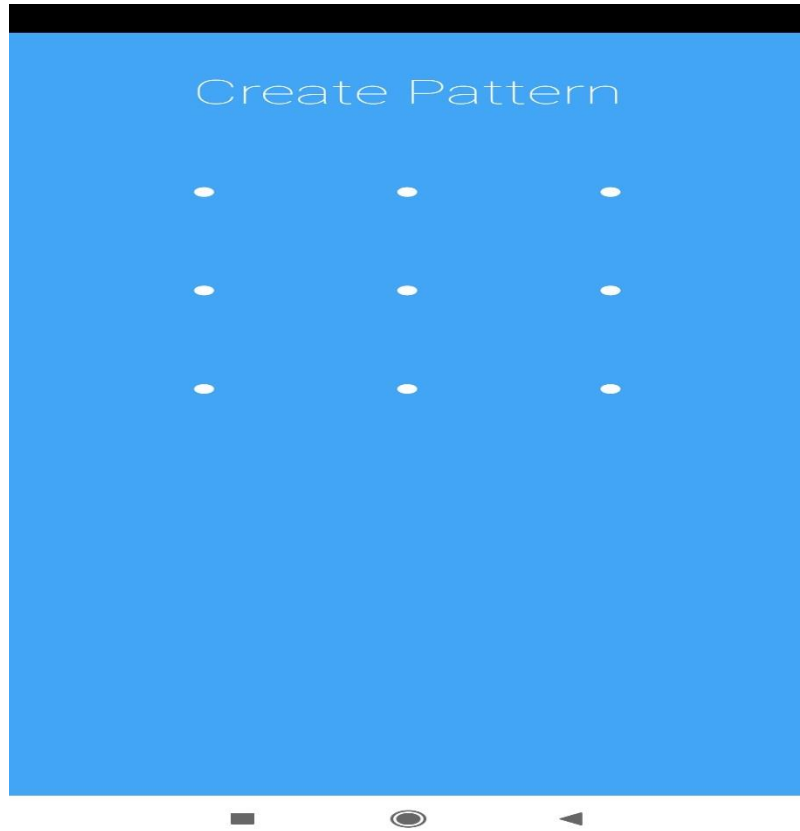
**Fig.12 ENABLING PIN LOCK**



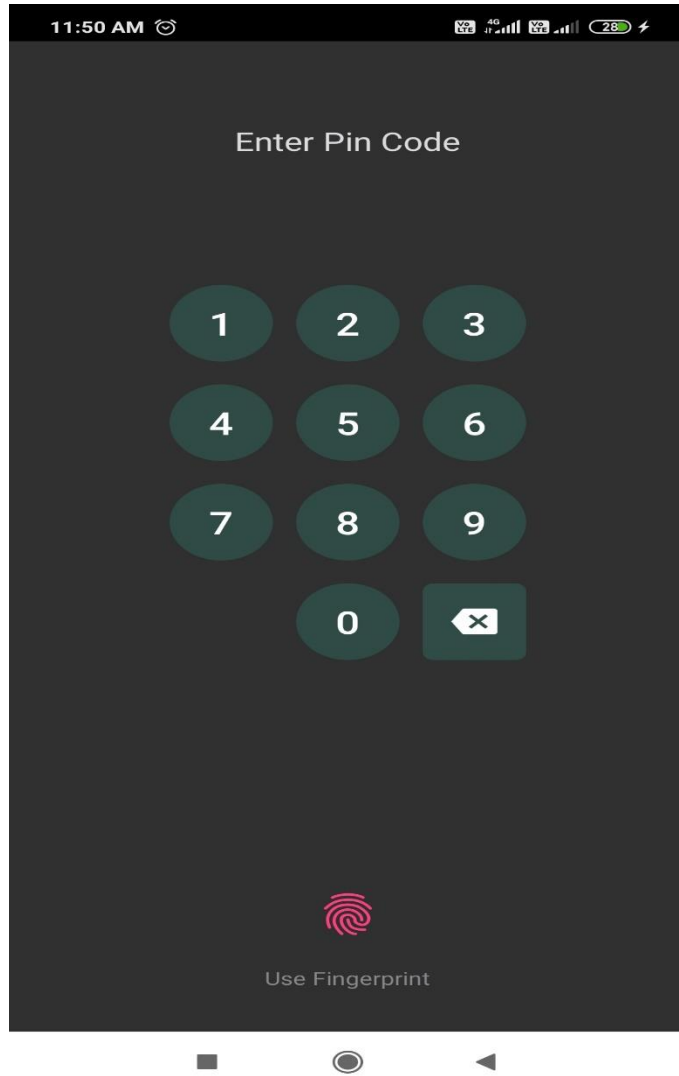
**Fig.13 SETTING PIN**



**Fig.14 DELETING/UPDATING**



**Fig.15 PATTERN LOCK**



**Fig.16 ENTERING PIN**



**Fig.17 LOCKSCREEN**

## **CHAPTER 5.CONCLUSION**

### **5.1 CONCLUSIONS**

The project titled APP LOCKER is an application based on android. The project's aim was to make your smart-phone powered by android secure.

The project was partially completed which covered Industry Mentor's maximum requirement. The constraints have been met, and successfully overcome. The system is designed as decided in the design phase earlier. The project gives a good idea to develop a user-friendly application that would satisfy different user types.

The system is made flexible, secure , reliable and versatile. This application has a user-friendly screen, allowing the user to use the lock screen application without any problems or inconvenience. The induced validation checks have reduced errors in the app locker application considerably. App locker application was upgraded with provisions. The application has been tested with live user and has provided a successful result. Hence the application has proved to work efficiently.

The program developed, by being simple to use with every form of consumer, achieved its objectives. Easily available from the site where we published it. With scalability in mind, this software is built. You can easily add additional modules whenever we feel it's necessary. However, there is still a lot of scope for future enhancement and functionality add on. Some of the main ones that are being developed for other mobile operating systems such as windows, IOS etc.

## **5.2 FUTURE PROSPECTUS**

Application is increasing at an unprecedented fast speed in Android application. Android is fast becoming the fastest-growing mobile operating system with the largest installed base. More than 900,000 new Android devices are turned on worldwide every day. Android's open-mindedness has made it a favorite for both consumers and developers, driving strong growth in app usage.

Android users download more than 1 billion Google Play Store apps and games every month. Android is continually pushing the hardware and software boundaries forward with its partners to bring new capabilities to users and developers.

Android creativity helps us to create strong, customized apps for developers that use the new mobile technologies.

Google Play is the premiere Android app distribution marketplace. I'm proposing that my Android development activities continue. JAVA to be part of the industry's ongoing innovations.

I'd also look for a business to gradually develop on Android, using Google App Engine.

I hope to function under the guidance and facilitation of the faculty of Jaypee University to engage with industry (both developer and hardware manufacturer) to allow me to pursue live projects in accordance with industry requirements..



## **REFERENCES:**

- 1. Java: A beginner's guide**
- 2. Learning Java by Building Android Games**
- 3. Thinking in Java**
- 4. Android Programming for Beginners**
- 5. Hello, Android: Introducing Google's Mobile Development Platform Book**

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT**

**PLAGIARISM VERIFICATION REPORT**

Date: .....03-10-2020.....

Type of Document (Tick):  PhD Thesis  M.Tech Dissertation/ Report  B.Tech Project Report  Paper

Name: \_\_Akash Mahajan,Rishi Rai\_\_\_\_\_ Department: \_\_\_\_\_CSE/

IT\_\_\_\_\_ Enrolment No. \_161250,161259\_\_\_\_\_ Contact No.

\_\_\_\_\_E-mail. \_\_\_\_\_ Name of

theSupervisor: \_\_Dr.AmolVasudeva

Title of the Thesis/


Dissertation/Project Report/Paper (In Capital letters): \_\_\_\_\_App Locker \_\_\_\_\_

**UNDERTAKING**

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

**Complete Thesis/Report Pages Detail:**

- Total No. of Pages =41
- Total No. of Preliminary pages =10
- Total No. of pages accommodate bibliography/references =

  
 (Signature of Student)

**FOR DEPARTMENT USE**

We have checked the thesis/report as per norms and found **Similarity Index** at ...13.....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

**FOR LRC USE**

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none"> <li>• All Preliminary Pages</li> <li>• Bibliography/Images/Quotes</li> <li>• 14 Words String</li> </ul>		Word Counts	
<b>Report Generated on</b>			Character Counts	
		<b>Submission ID</b>	Total Pages Scanned	
			File Size	

Checked by  
Name & Signature

Librarian

**Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at [plagcheck.juit@gmail.com](mailto:plagcheck.juit@gmail.com)**