Privlok+ (Password keeper App)

A PROJECT REPORT
Submitted By

Sachin Rathi
University Roll No 1900290140029

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of
Dr. Vipin Kumar
(ASSOCIATE PROFESSOR)

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad Uttar Pradesh-201206
(NOVEMBER 2021)

CERTIFICATE

Certified that **Sachin Rathi (Enrolment No. 190029014005172)** have carried out the project work having "**Privlok+ (Password keeper app)" for** Master of Computer

Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:24-11-2021

Sachin Rathi
University Roll No. 1900290140029

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

Date:

Dr. Vipin Kumar
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Ajay Shrivastava
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

The project is based on android technology whose name is Privlok+. We are just made this application to store passwords of different platform. The technology which we are using in our application are as follows i.e., Android studio and SQL. And the benefits of our application in real world is that user no need to remember all the passwords all the time. User can access our application to retrieve all the passwords which have been stored.

ACKNOWLEDGEMENTS

Success in life is never attained single headedly. My deepest gratitude goes to my thesis supervisor, **Dr. Vipin Kumar** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions. Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Sachin Rathi

CHAPTER-1

INTRODUCTION

1.1 PROJECT DESCRIPTION

The project is based on android technology whose name is Pirvlok+. We are just made this application to store passwords of different platform. The technology which we are using in our application are as follows i.e., Android studio and SQL. And the benefits of our application in real world is that user no need to remember all the passwords all the time. User can access our application to retrieve all the passwords which have been stored.

Chapter 2

PROJECT CATEGORY

2.1 TECHNOLOGIES USED

2.1.1 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for
Android app development, based on IntelliJ idea. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks

TECHNOLOGIES USED

APPLICATION	: Android Studio
DESIGNING	: HTML, xml, Java
Backend	: SQLite

2.2 Language Used (Designing and Development)

2.2.1 HTML:-

1. HTML stands for Hyper Text Markup Language o HTML is the standard markup language for creating Web pages o HTML describes the structure of a Web page o HTML consists of a series of elements.
2. HTML elements tell the browser how to display the content o HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

2.2.2 XML:

XML stands for Extensible Markup Language. XML is a markup language much like HTML used to describe data. It is derived from Standard Generalized Markup Language (SMGL). Basically, the XML tags are not predefined in XML. We need to implement and define the tags in XML. XML tags define the data and used to store and organize data. It's easily scalable and simple to develop. In Android, the XML is used to implement UI-related data, and it's a lightweight markup language that doesn't make layout heavy. XML only contains tags, while implementing they need to be just invoked.

2.2.3 Java

Firstly Java was the official language for Android App Development (but now it was replaced by Kotlin) and consequently, it is the most used language as well. Many of the apps in the Play Store are built with Java, and it is also the most supported language by Google. In addition to all this, Java has a great online community for support in case of any problems (And trust me, there will be problems!).
However, Java is a complicated language for a beginner to use as it contains complex topics like constructors, null pointer exceptions, concurrency, checked exceptions, etc. Also, The Android Software Development Kit (SDK) increases the complexity to a new level!

2.2.4 SQLite:

It is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. It is the most used database engine in the world. It is an in-process library, and its code is publicly available. It is free for use for any purpose, commercial or private. It is basically an embedded SQL database engine. Ordinary disk files can be easily read and write by SQLite because it does not have any separate server like SQL. The SQLite database file format is cross-platform so that anyone can easily copy a database between 32-bit and 64-bit systems. Due to all these features, it is a popular choice as an Application File Format.
It was designed by D. Richard Hip for the purpose of no administration required for operating a program. in August 2000. As it is very lightweight compared to others like MySQL and Oracle, it is called SQLite. Different versions of SQLite are released since 2000.

Chapter 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 GENERAL DESCRIPTION

This App gives the flexibility to the user to save their critical information to store internally i.e., (Mobile) because in today life hacking is most critical situation for users privacy. It is easy to use and user can remember their password all time without loosing or worries of their data losing or forgetting issue.

3.1.1 PROBLEM STATEMENT:

The problem is occurred in different situations.

- If user forget their password, then it helps to remembering that.
- Every time when user forget their password or any other information without using rest their password.
- Saves the time of user because we all know time is money!

3.2 SYSTEM OBJECTIVES

3.2.1 Improvement in control and performance

The system is developed to cope up with the current issues and problems of forgetting the traditional mechanism. The system gives the security for protecting their password of critical information with biometric and special password.

Save cost

The existing system is based on the pen paper mode and several in the digital mode.

Save Time

People at any location will be able to perform or know their passwords or other sensitive information in just seconds team.

3.2.2 Requirement Specification:

The application requirement specification is produced at the analysis task. The function and performance allocated to application as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioural description, an indication of performance requirements and design constraints.

3.3 Functional Requirements:

Admin Generated Username and Password

The application will work with email or username and password generated by the admin after using the App.

Security:

For using the application user need biometric scan (fingerprint) and also the master password for use the application.

3.4 Non-functional Requirements:

Performance Requirements

- User friendly**:** The system should be user friendly so that it can easily be understand by the user without any difficulty.
- Ease of maintenance: - System should be easy to maintain and use.
- Less time consuming: The system should be less time consuming which could be achieved by good programming.
- Error free: The system should easily handle the user error in any case.
- Static: Application runs on standalone machine. Support only single user.

3.5 SOFTWARE AND HARDWARE REQUIREMENTS

This section describes the software and hardware requirements of the system.

3.5.1 SOFTWARE REQUIREMENTS

Operating system- Android operating system is required at least android version 7 or above.

Database: SQLite is used in storing the data in structured manner.

Development tools and Programming language-HTML, Java, Xml is used to design the whole interface and functionality of Privlok+. SQLite is used for Database.

3.5.2 HARDWARE REQUIREMENTS

• Desktop/Laptop (Processor-i5, 8th gen, RAM (Min)-8Gb.)

3.6 Software System Attributes

- **Portability**: - The system should be machine independent.
- **Security: -** The system is designed in such a way that it will store the recorded data in the system of the owner. The system will be secure from unauthorized access of the application.
- **Maintainability**: The system will be designed in a maintainable order. The system can be easily modified and renewed according to the need of the organization.

3.8. Feature of Privlok+

- No internet connection required.
- People can register and login in the system.
- Secure Password Generator
- Automatically Backup
- Recovery Option
- Save and Autofill Passwords
- Item Storage
- Sharing Support

3.9 Model used: Incremental Model

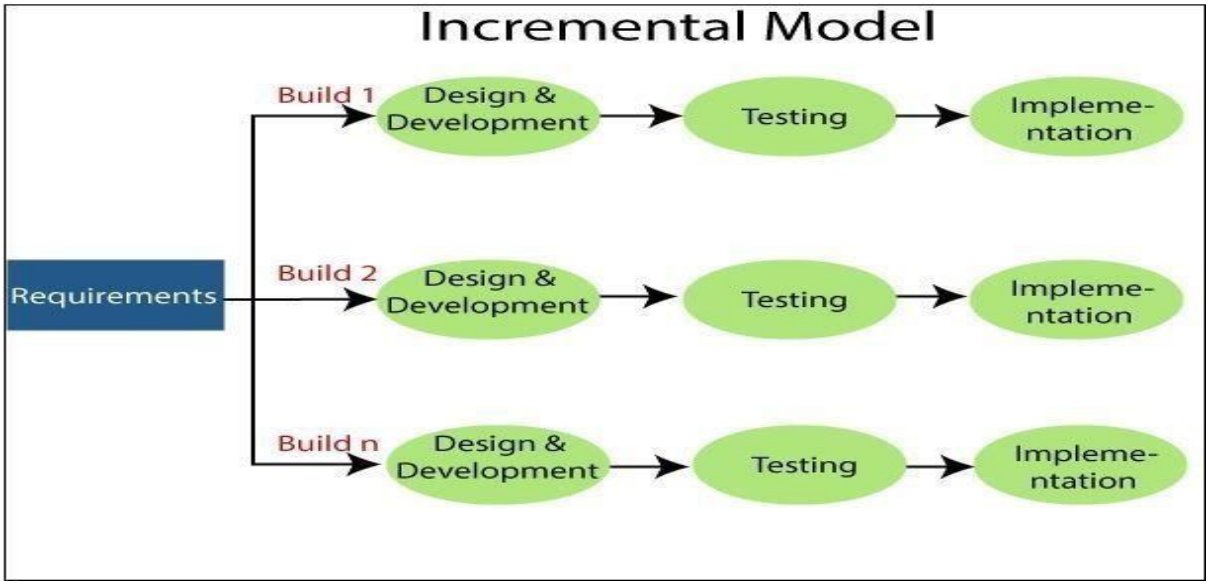


Fig 1.0: Incremental Model

Incremental Model is a software development process where requirements are divided into several stand-alone software development modules. In this project the first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments.

3.10 Preliminary Description:

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the system. The purpose of preliminary investigation is to evaluate project requests. It is not a design study, nor does it include the collection of details to describe the system in all respect. Rather, it is the collecting of information that helps committee members to evaluate the merits of project request and make an informed judgement about the feasibility of the proposed project.

Analyst working on the preliminary investigation should accomplish the following objectives:

- Clarify and understand the project request.
- Determine the size of the project.
- Access costs and benefits of alternative approaches.

Chapter- 4

4.1 Feasibility study:

After studying and analysing all the existing and requires functionalities of the system, the next task is to do the feasibility study for the project. Feasibility study includes consideration of all the possible ways to provide a solution to a given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

4.1.1 Economical Feasibility:

For the economic feasibility, Economic analysis or cost/benefits analysis is most frequently used technique the effectiveness of a proposed system. it is a procedure to determine the benefits and saving those are expected from the proposes system and compare them with cost. if the benefits outweigh the costs, a decision is taken to design and implement the system. otherwise, further justification or alternative in proposed system will have to be made if it is to have a chance of being approved this is ongoing effort that improves in accuracy at each phase of a system life cycle

4.1.2 Technical feasibility:

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionalities to be provided in the system, as described in the System Requirement Specification (SRS) and checked if everything was possible using different type of front end and back-end platform.

4.1.3 Operational Feasibility:

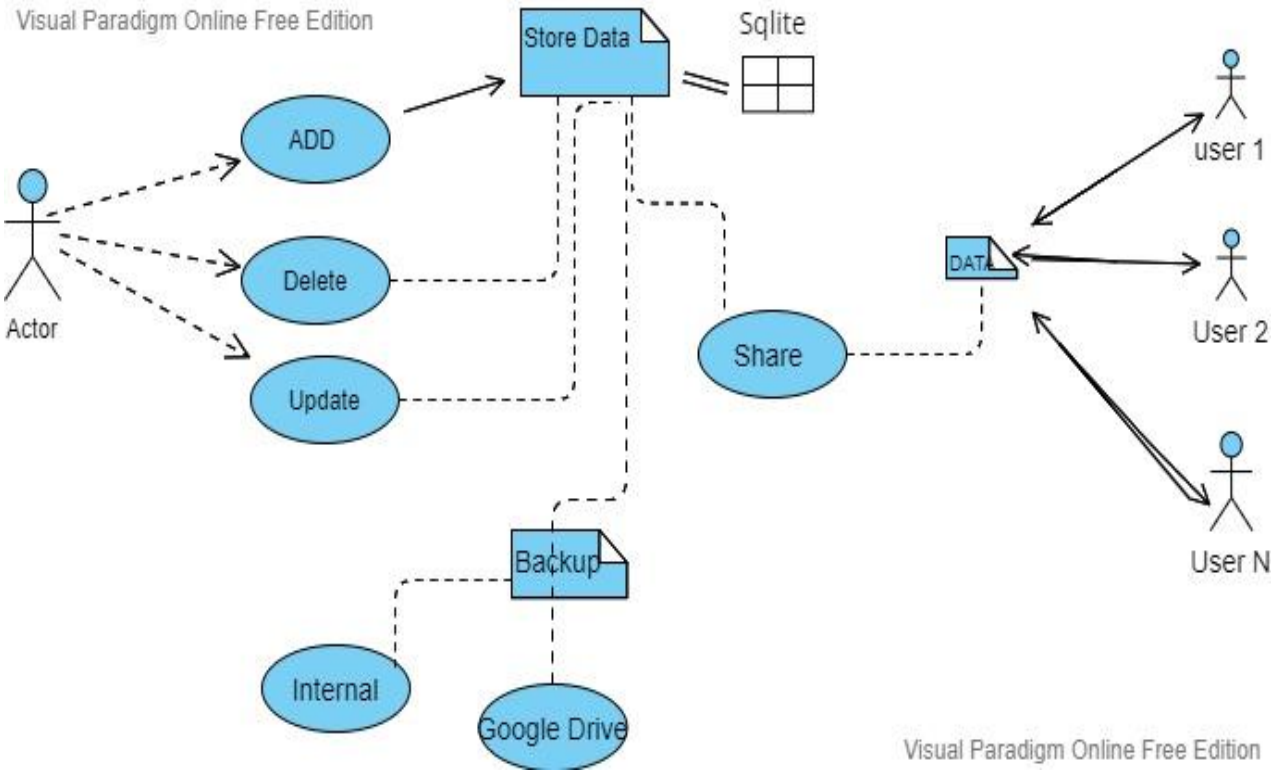
No doubt the technically growing world needs more enhancement in technology, this apps is very user friendly and all inputs to be taken all self- explanatory even to a layman. As far our study is concerned, the clients will be comfortable and happy as the system has cut down their loads and bring the young generation to the same virtual world they are growing drastically. Operational feasibility cover two aspects.one technical performance aspects and the other is acceptance within the organization. Operation feasibility determine how the proposed the system will fit in with the current operation and what needs to implement the system

Chapter 5

5.1 Planning and scheduling

5.1.1 Use Case

A use case diagram is used to represent the dynamic behaviour of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and tells how the user handles a system.



5.1.2 Software Requirements with specifications:

Name of Components	Specifications
Operating system	Windows
Language	HTML, XML, Java, SQLite
Software Development kit	Android Studio

5.1.3 Hardware Requirements with specifications:

Name of Components	Specifications
Laptop/Desktop	Processor-i5, 8th Gen
RAM	Minimum 8Gb.

5.2: DATA FLOW DIAGRAM

Are used to graphically represent the flow of data in a Privlok+ (Password keeper App). DFD describes the processes that are involved in an adding data, update data and delete Data.

Fig 3.1 Privlok+ DFD

Chapter-6

6.1. Coding:

6.1.1 Add Activity:

```
public class AddPasswords extends AppCompatActivity {
    private TextInputLayout textInputTitle;
    private TextInputLayout textInputPassword;
    private TextInputLayout textInputAccount;
    private TextInputLayout textInputUsername;
    Button save;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_passwords);

        textInputTitle = findViewById(R.id.text_input_title);
        textInputPassword = findViewById(R.id.text_input_password);
        textInputAccount = findViewById(R.id.text_input_account);
        textInputUsername = findViewById(R.id.text_input_username);
        save = findViewById(R.id.btnInsert);
    }

    private boolean validateTitle() {
        String Title = textInputTitle.getEditText().getText().toString().trim();

        if(Title.isEmpty()){
            textInputTitle.setError("Field can't be empty");
            return false;
        } else {
            textInputTitle.setError(null);
            return true;
        }
    }
```

6.1.2 Add Activity.xml

```
<com.google.android.material.textfield.TextInputLayout
        android:id="@+id/text_input_title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        app:errorEnabled="true">


<com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Title"
            android:textSize="30dp"
            android:background="#00000000"
            android:inputType="text"/>


</com.google.android.material.textfield.TextInputLayout>


<com.google.android.material.textfield.TextInputLayout
        android:id="@+id/text_input_username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:counterEnabled="true"
        app:counterMaxLength="20"
        android:layout_margin="20dp"
        app:errorEnabled="true">


<com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Username"
            android:textSize="20dp"
            android:background="#00000000"
            android:inputType="text"/>


</com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.material.textfield.TextInputLayout
        android:id="@+id/text_input_account"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        app:counterEnabled="true"
        app:counterMaxLength="16"
        app:errorEnabled="true">


<com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Account Number"
            android:textSize="20dp"
            android:background="#00000000"
            android:inputType="number"/>


</com.google.android.material.textfield.TextInputLayout>


<com.google.android.material.textfield.TextInputLayout
        android:id="@+id/text_input_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        app:errorEnabled="true"
        app:passwordToggleEnabled="true">


<com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Password"
            android:background="#00000000"
            android:textSize="20dp"
            android:inputType="textPassword"/>


</com.google.android.material.textfield.TextInputLayout>
```

```
    <Button
        android:id="@+id/btnInsert"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:text="Save"
        android:textSize="20dp"
        android:onClick="saveInput" />
```

6.1.3 Bottom Activity.xml

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/androi
d"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BottomNavigationActivity">

    <ListView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <androidx.coordinatorlayout.widget.CoordinatorLayout
        android:id="@+id/containerLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.8"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="1.0">
```

```
<com.google.android.material.floatingactionbutton.Floati
ngActionButton
            android:id="@+id/floating_action_button"
            android:layout_width="68dp"
            android:layout_height="wrap_content"
            android:layout_gravity="bottom|right"
            android:layout_margin="32dp"
            android:src="@drawable/ic_baseline_add_24"
            app:backgroundTint="@color/teal_200"
            app:fabSize="normal"
            app:rippleColor="@color/teal_700" />


</androidx.coordinatorlayout.widget.CoordinatorLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

6.1.4 Main.Activity.Java

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ImageView imageview
=findViewById(R.id.imageView);
        Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(),R.a
nim.fade);
        imageview.startAnimation(animation);

        Thread timer = new Thread(){

            @Override
            public void run() {
                try{
                    sleep(8000);
                    Intent i = new
Intent(getApplicationContext(),BottomNavigationActivity.
class);
                    startActivity(i);
```

```
                    finish();
                    super.run();
                }catch (InterruptedException e){
                    e.printStackTrace();
                }
            }
        };
        timer.start();
    }

}
```

6.1.5 Main Activity.xml

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/androi
d"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="@drawable/gradient_background"

android:theme="@style/Theme.MaterialComponents.DayNight.
NoActionBar">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="180dp"
        android:layout_centerInParent="true"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/lock" />

</RelativeLayout>
```

6.2: Update Coding:

6.2.1 BottomNavigationActivity.xml

```
<androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintBottom_toTopOf="@+id/containerLayout"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />

    <androidx.coordinatorlayout.widget.CoordinatorLayout
        android:id="@+id/containerLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.8"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="1.0">


<com.google.android.material.floatingactionbutton.FloatingActionButton
            android:id="@+id/floating_action_button"
            android:layout_width="68dp"
            android:layout_height="wrap_content"
            android:layout_gravity="bottom|right"
            android:layout_margin="16dp"
            android:src="@drawable/ic_baseline_add_24"
            app:backgroundTint="@color/teal_200"
            app:fabSize="normal"
            android:clickable="true"
            android:focusable="true"
            app:rippleColor="@color/teal_700" />
```

</androidx.coordinatorlayout.widget.CoordinatorLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

6.2.2 BottomNavigationActivity.java

```
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_bottom_navigation);
        recyclerView= findViewById(R.id.recyclerview);
        fab= findViewById(R.id.floating_action_button);
        fab.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new
Intent(getApplicationContext(),AddPasswords.class);
                startActivity(i);
            }
        });
        myDb= new
MyDataHelper(BottomNavigationActivity.this);
        //array_id = new ArrayList<>();
        array_title = new ArrayList<>();
        array_username = new ArrayList<>();
        array_account = new ArrayList<>();
        array_password = new ArrayList<>();
        storeDataInArrays();
        customAdapter = new
CustomAdapter(BottomNavigationActivity.this,
array_title, array_username,
array_account,array_password);
        recyclerView.setAdapter(customAdapter);
        recyclerView.setLayoutManager(new
LinearLayoutManager(BottomNavigationActivity.this));
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.popmenu,menu);
        return super.onCreateOptionsMenu(menu);
```

```
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item)
{
        if(item.getItemId() == R.id.delete_all){
            confirmDialog();
        }
        return super.onOptionsItemSelected(item);
    }
    /*@Override
    protected void onActivityResult(int requestCode, int
resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode,
data);
        if(requestCode == 1){
            recreate();
        }
    }*/
    void storeDataInArrays(){
        Cursor cursor = myDb.readAllData();
        if(cursor.getCount() == 0){
            Toast.makeText(this,"No Data found",
Toast.LENGTH_LONG).show();
        }
        else{
            while(cursor.moveToNext()){
                array_title.add(cursor.getString(0));
                array_username.add(cursor.getString(1));
                array_account.add(cursor.getString(2));
                array_password.add(cursor.getString(3));
            }
        }
    }
```

6.2.3 CustomerAdapter.java

```
CustomAdapter( Context context,ArrayList
arrayList_title, ArrayList arrayList_username, ArrayList
arrayList_account, ArrayList arrayList_password)
    {
        this.context= context;
        //this.activity=activity;
        //this.arrayList_id = arrayList_id;
        this.arrayList_title=arrayList_title;
        this.arrayList_username=arrayList_username;
        this.arrayList_account=arrayList_account;
        this.arrayList_password=arrayList_password;
    }
    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
        LayoutInflater inflater =
LayoutInflater.from(context);
        View
view=inflater.inflate(R.layout.list_layout_data,parent,f
alse);
        return new MyViewHolder(view);
    }
    //@RequiresApi(api = Build.VERSION_CODES.M)
    @Override
    public void onBindViewHolder(@NonNull MyViewHolder
holder, int position) {

holder.title_txt.setText(String.valueOf(arrayList_title.
get(position)));

holder.username_txt.setText(String.valueOf(arrayList_use
rname.get(position)));

holder.account_txt.setText(String.valueOf(arrayList_acco
unt.get(position)));

holder.password_txt.setText(String.valueOf(arrayList_pas
sword.get(position)));
        //Recyclerview onClickListener
        holder.mainLayout.setOnClickListener(new
View.OnClickListener() {
            @Override
```

```
            public void onClick(View view) {
                Intent intent = new Intent(context,
UpdateActivity.class);
                //intent.putExtra("id",
String.valueOf(arrayList_id.get(position)));
                intent.putExtra("title",
String.valueOf(arrayList_title.get(position)));
                intent.putExtra("username",
String.valueOf(arrayList_username.get(position)));
                intent.putExtra("account",
String.valueOf(arrayList_account.get(position)));
                intent.putExtra("password",
String.valueOf(arrayList_password.get(position)));
                context.startActivity(intent);

//activity.startActivityForResult(intent, 1);
            }
        });
    }
    @Override
    public int getItemCount() {
        return arrayList_title.size();
    }
    public class MyViewHolder extends
RecyclerView.ViewHolder {
        LinearLayout mainLayout;
        TextView title_txt, username_txt, account_txt,
password_txt;
        public MyViewHolder(@NonNull View itemView) {
            super(itemView);
            title_txt =
itemView.findViewById(R.id.textViewTitle);
            username_txt =
itemView.findViewById(R.id.textViewUsername);
            account_txt =
itemView.findViewById(R.id.textViewAccountNo);
            password_txt =
itemView.findViewById(R.id.textViewPassword);
            mainLayout =
itemView.findViewById(R.id.mainLayout);
            //Animate Recyclerview
```

```
            Animation translate_anim =
AnimationUtils.loadAnimation(context,
R.anim.translate_anim);
            mainLayout.setAnimation(translate_anim);
        }
    }
```

6.2.4 ListView.xml

```
<androidx.cardview.widget.CardView
    android:id="@+id/cardView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="12dp">


  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="12dp"
    android:orientation="vertical">

    <TextView
      android:id="@+id/textViewTitle"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginBottom="5dp"
      android:layout_marginTop="10dp"
      android:text="SBI Bank"

android:textAppearance="@style/Base.TextAppearance.AppCompat.Large"
/>

    <TextView
      android:id="@+id/textViewUsername"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Nirmit Bansal"
```

```
android:textAppearance="@style/Base.TextAppearance.AppCompat.Mediu
m" />

      <TextView
         android:id="@+id/textViewAccountNo"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:text="823482748343"

android:textAppearance="@style/Base.TextAppearance.AppCompat.Mediu
m" />

      <TextView
         android:id="@+id/textViewPassword"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:layout_marginTop="5dp"
         android:text="nir12345" />

   </LinearLayout>


   </androidx.cardview.widget.CardView>
</LinearLayout>
```

6.2.5 UpdateActivity.java

```
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update);
        textUpdateTitle =
findViewById(R.id.text_update_title);
        textUpdatePassword =
findViewById(R.id.text_update_password);
        textUpdateAccount =
findViewById(R.id.text_update_account);
        textUpdateUsername =
findViewById(R.id.text_update_username);
```

```
        update_button = findViewById(R.id.btnUpdate);
        delete_button = findViewById(R.id.btnDelete);
        //First we call this
        getAndSetIntentData();
        //Set actionbar title after getAndSetIntentData
method
        ActionBar ab = getSupportActionBar();
        if (ab != null) {
            ab.setTitle(title);
        }
        update_button.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //And only then we call this
                MyDataHelper myDB=new
MyDataHelper(UpdateActivity.this);
               /* title =
textUpdateTitle.getEditText().getText().toString().trim(
);
                username =
textUpdateUsername.getEditText().getText().toString().tr
im();
                account =
textUpdateAccount.getEditText().getText().toString().tri
m();
                password =
textUpdatePassword.getEditText().getText().toString().tr
im();*/
                myDB.updateData(id, title, username,
account, password);
            }
        });
        delete_button.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                confirmDialog();
            }
        });
    }
    void getAndSetIntentData(){
        if(getIntent().hasExtra("title") &&
```

```
                getIntent().hasExtra("username") &&
getIntent().hasExtra("account") &&
                getIntent().hasExtra("password")){
            //Getting Data from Intent
            //id = getIntent().getStringExtra("id");
            title = getIntent().getStringExtra("title");
            username =
getIntent().getStringExtra("username");
            account =
getIntent().getStringExtra("account");
            password =
getIntent().getStringExtra("password");
            //Setting Intent Data

textUpdateTitle.getEditText().setText(title);

textUpdateUsername.getEditText().setText(username);

textUpdateAccount.getEditText().setText(account);

textUpdatePassword.getEditText().setText(password);
            //Log.d("nirmit", title+" "+username+"
"+account+" "+password);
        }else{
            Toast.makeText(this, "No data.",
Toast.LENGTH_SHORT).show();
        }
    }
```

6.2.6 UpdateActivity.xml

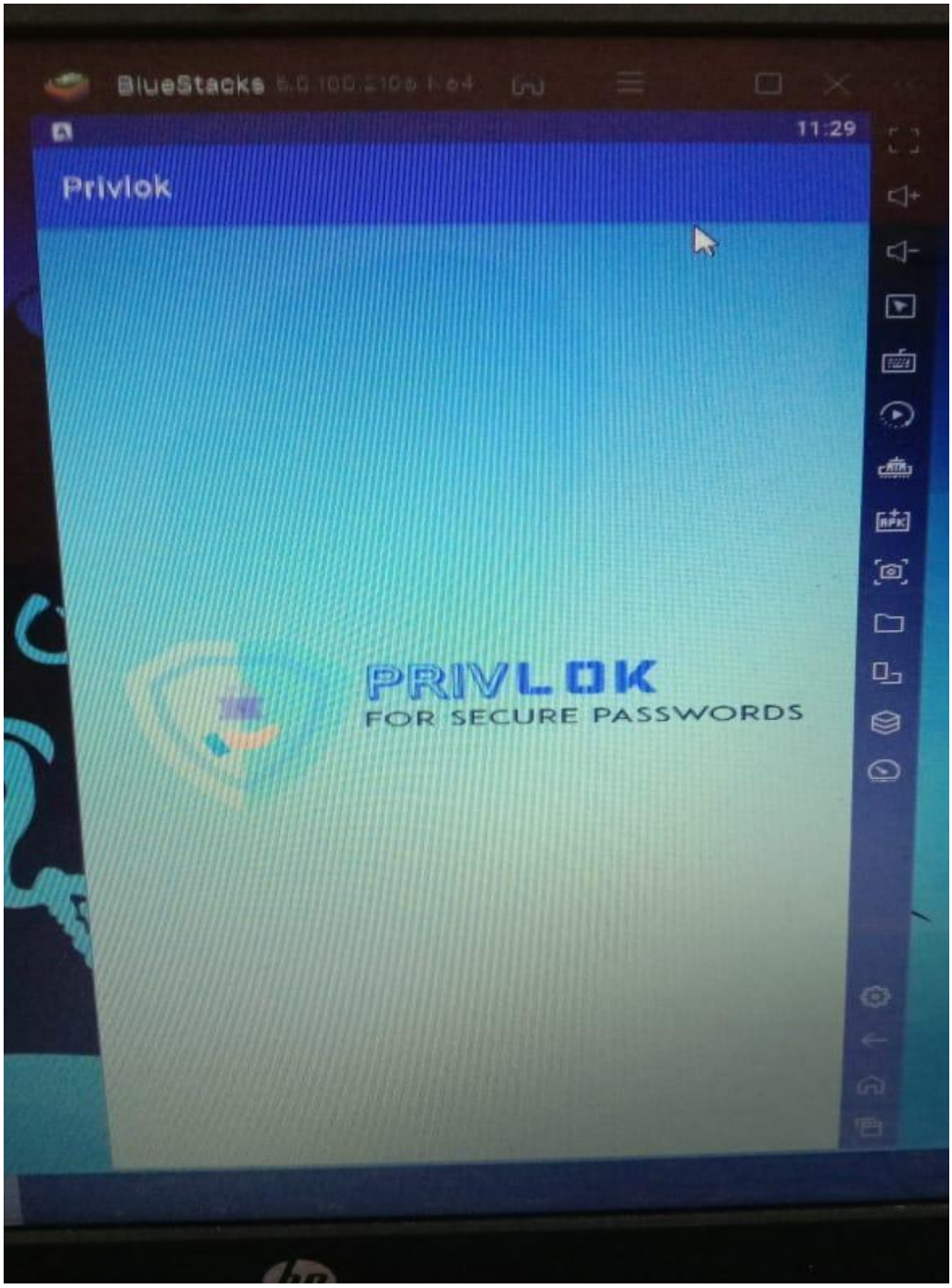
```
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/text_update_title"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    app:errorEnabled="true">

    <com.google.android.material.textfield.TextInputEditText
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
```

```
        android:hint="Title"
        android:textSize="30dp"
        android:background="#00000000"
        android:inputType="text"/>

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/text_update_username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:counterEnabled="true"
    app:counterMaxLength="20"
    android:layout_margin="20dp"
    app:errorEnabled="true">

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username"
        android:textSize="20dp"
        android:background="#00000000"
        android:inputType="text"/>

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/text_update_account"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    app:counterEnabled="true"
    app:counterMaxLength="16"
    app:errorEnabled="true">

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Account Number"
```

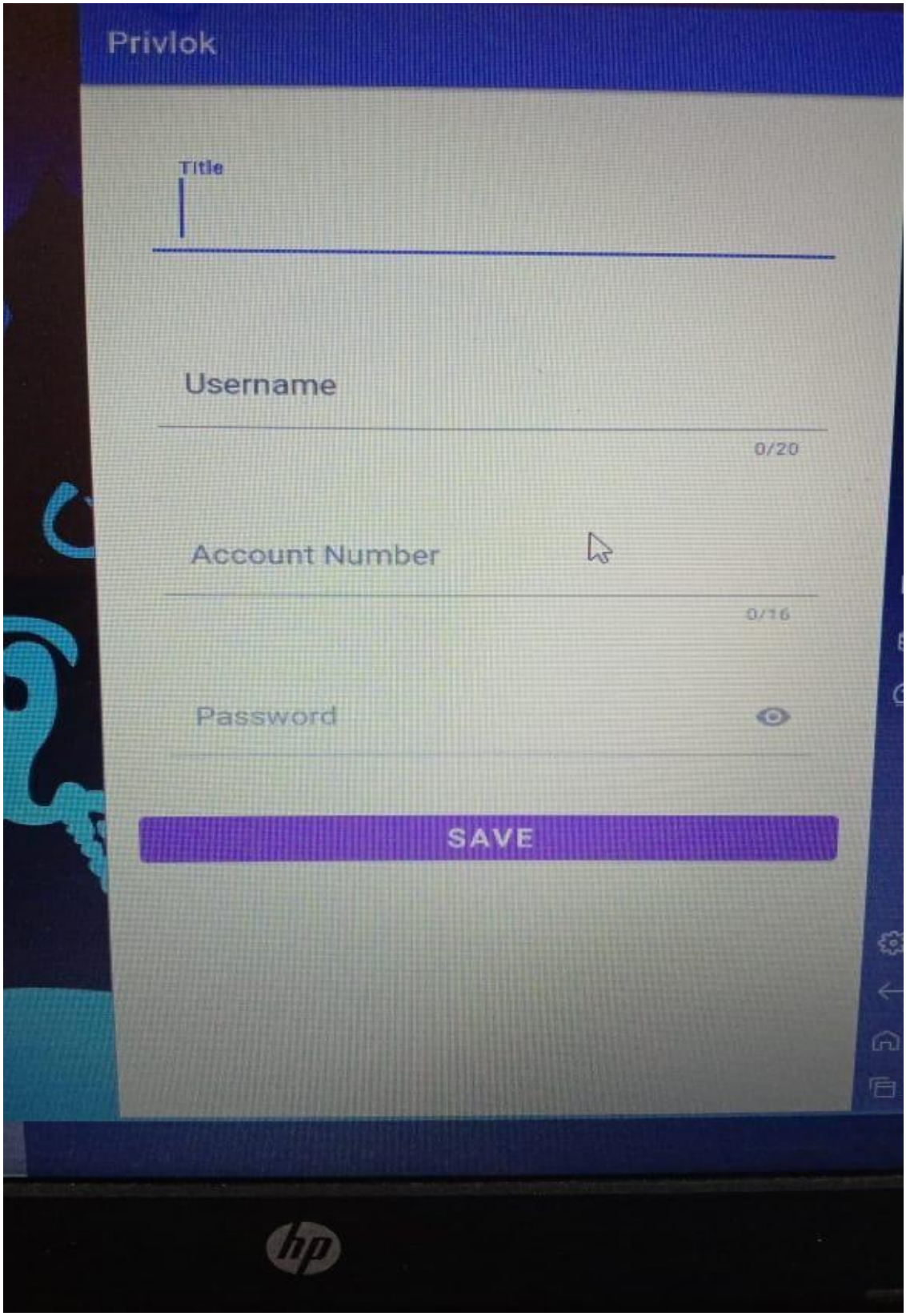
```
            android:textSize="20dp"
            android:background="#00000000"
            android:inputType="number"/>

    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/text_update_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        app:errorEnabled="true"
        app:passwordToggleEnabled="true">

        <com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Password"
            android:background="#00000000"
            android:textSize="20dp"
            android:inputType="textPassword"/>

    </com.google.android.material.textfield.TextInputLayout>

    <Button
        android:id="@+id/btnUpdate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Update"
        android:textSize="20dp"/>

    <Button
        android:id="@+id/btnDelete"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Delete"
        android:textSize="20dp"/>

</LinearLayout>
```

Chapter 7
Application Format

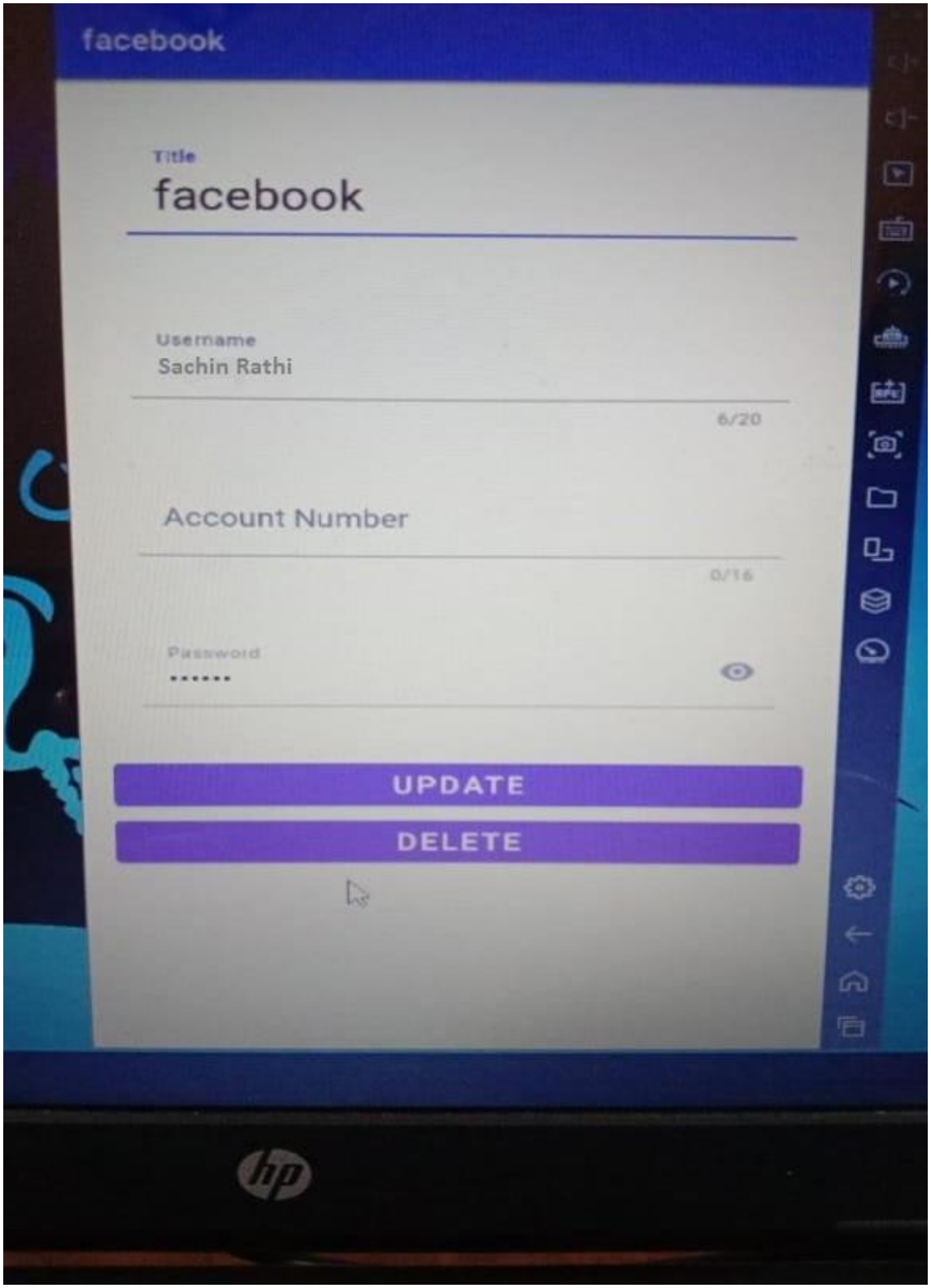
7.1.1 Starting:



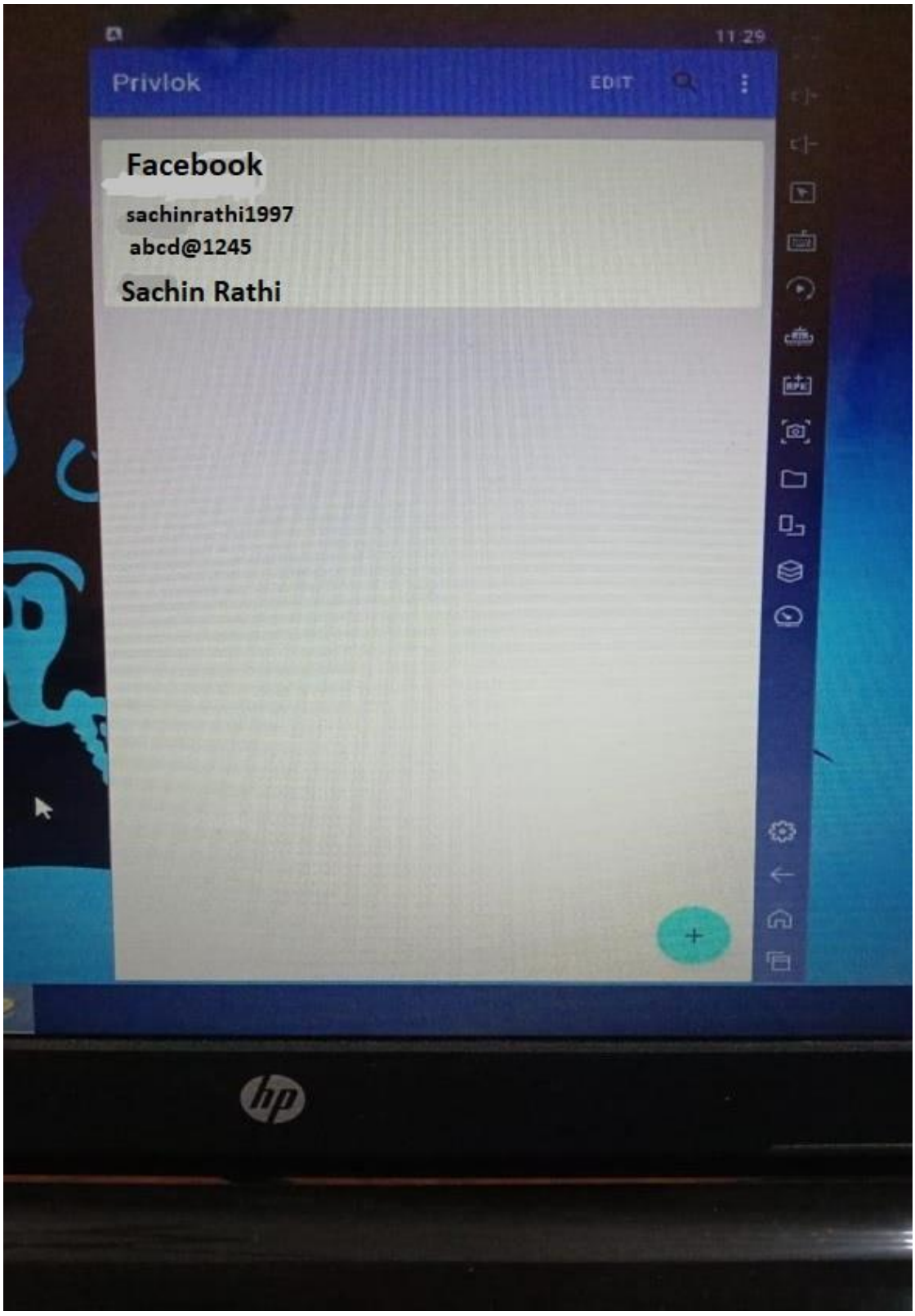
7.1.2 Preview:


Privlok
Title
Username
0/20
Account Number
0/16
Password
SAVE

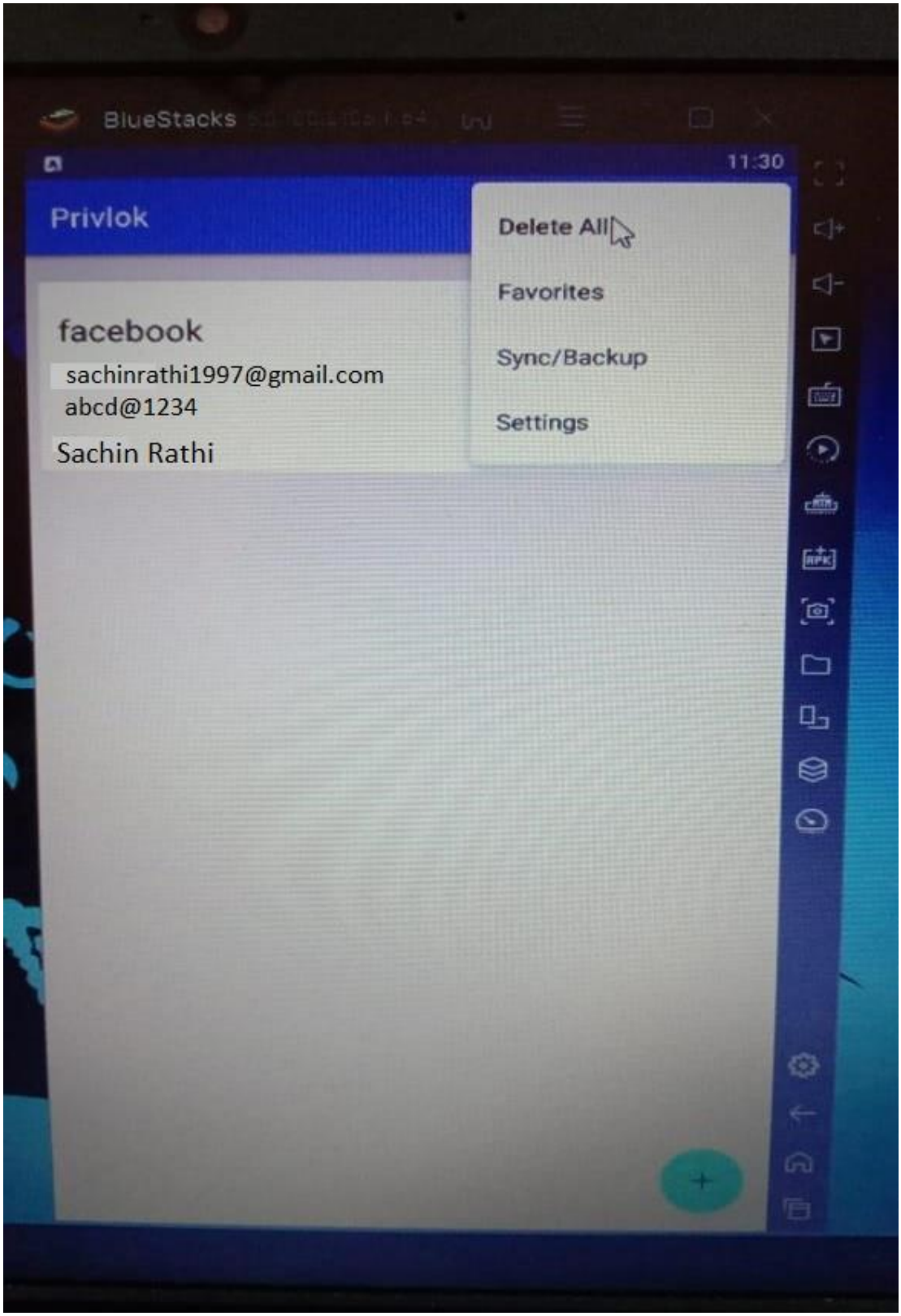
7.1.3 Preview:


facebook
Title
facebook
Username
Sachin Rathi
6/20
Account Number
0/16
Password
UPDATE
DELETE

7.1.4 Preview:


11:29
Privlok
EDIT
Facebook
sachinrathi1997
abcd@1245
Sachin Rathi

7.1.5 Preview:


BlueStacks
11:30
Privlok
Delete All
Favorites
Sync/Backup
Settings
facebook
sachinrathi1997@gmail.com
abcd@1234
Sachin Rathi

Chapter 8
Summary:

Problem Statement: Security is the first thing that comes to our mind before using it. In day-to-day life, we have a lot of different accounts on different platforms. And the challenging task is to remember all the passwords. That is the biggest problem in today's digital world for everyone. So, it is required to make a solution for it.

Solution: For this problem, we must decide to make an application which is meet the requirement of what we need. After all the research we decided to make an application (Privlok+) which stored all the credential information in encrypted form for security purpose through which the new user makes there an account on privlok+ and existing user can directly login. Users can store their passwords with their account handle. User information will store in encrypted form. Our project is based on android technology. The technology which we are using in our application is as follows i.e., Android Studio and SQL. The benefit of our application in the real world is that users do need to remember all the passwords all the time. Users can access their data whenever is required.