Text

Description automatically generated

Mini project report on

**Bank Application**

Submitted by

**Namith Babu E : 18030141CSE031**

**Amartya Singh : 18030141CSE072**

**Sajid Ahmad : 18030141CSE044**

Under the Guidance of

Radha R

Assistant Professor

Department of Computer Science and Engineering

Alliance college of Engineering and Design

Course name: Mobile Application Development Using Android

Course code: CSE703

**TABLE OF CONTENT**

**SL No. Title .**

1 Abstract

2 Introduction

3 Introduction to Android (Versions & Architecture)

4 Anatomy of android application with figure

5 Introduction to your project

6 Explain modules implemented

7 Project Code

8 Output Screenshot

**Abstract**

An android application and for banking system has been developed where the transaction can be made very easily. The system we have developed which is based on android application. Android is an operating system for cellular phone which is based on Linux kernel that is at present development by Google. We use this application in bank sector. Normally our most of the bank provides manual registration for creating bank account. Customer must be presented in bank to open bank account. These systems sometimes difficult for the people and time wasting. Our application will save time and miseries. To increase accessibility we develop an android version of our application. By using our system, User can transfer money from one account to another. User can check bank statement, user details and transfer money through android apps.

**Introduction**

Mobile banking is a system that allows customers of a financial institution to conduct a number of financial transactions through a mobile device such as a mobile phone or tablet. Mobile banking offers many advantages, easy access and plentiful applications for smart phones, provide a good security. The biggest benefit is that you have more control of your money, Mobile banking takes us back to the days when we used passwords, and many services are pretty basic. People use it to check balances and monitor finances.

The earliest mobile banking services used SMS, services known as SMS banking. With the introduction of smart phones with WAP support enabling the use of the mobile web in 1999, the first European

banks started to offer mobile banking on this platform to their customers. It has until recently most often been performed via SMS or the mobile web. Advancements in web technologies such as HTML, CSS3 and

JavaScript have seen more banks launching mobile web based services to complement native application.

Information system/technology can be any organized combination of people, hardware, software, communications networks, and data resources that collect, transforms, and disseminate information in an

organization, Banks urgently need to improve the ability to think strategically about information technology investments. Only banks that use their

technology resources effectively have the opportunity to secure real competitive advantage in this fast changing industry through real product or service differentiation. Since 1980, banking in the UK has undergone

tremendous structural changes, this has been primarily the result of new regulation and new technology, which itself precipitated the change in regulation. One of the most important regulatory changes is the abolition of

credit controls on sterling lending, which was followed by the expansion of the bank’s direct consumer lend ing,

credit card and montage business.

**Introduction to Android**

Android is an operating system and programming platform developed by Google for mobile phones and other mobile devices, such as tablets. It can run on many different devices from many different manufacturers. Android includes a software development kit (SDK) that helps you write original code and assemble software modules to create apps for Android users. Android also provides a marketplace to distribute apps. All together, Android represents an ecosystem for mobile apps.



**Android Architecture**

Figure 1 outlines the current (layered) Android Architecture. The modified Linux kernel operates as the HAL, and provides device driver, memory management, process management, as well as networking functionalities, respectively. The library layer is interfaced through Java (which deviates from the traditional Linux design). It is in this layer that the Android specific libc (Bionic) is located. The surface manager handles the user interface (UI) windows. The Android runtime layer holds the Dalvik Virtual Machine (DVM) and the core libraries (such as Java or IO). Most of the functionalities available in Android are provided via the core libraries.

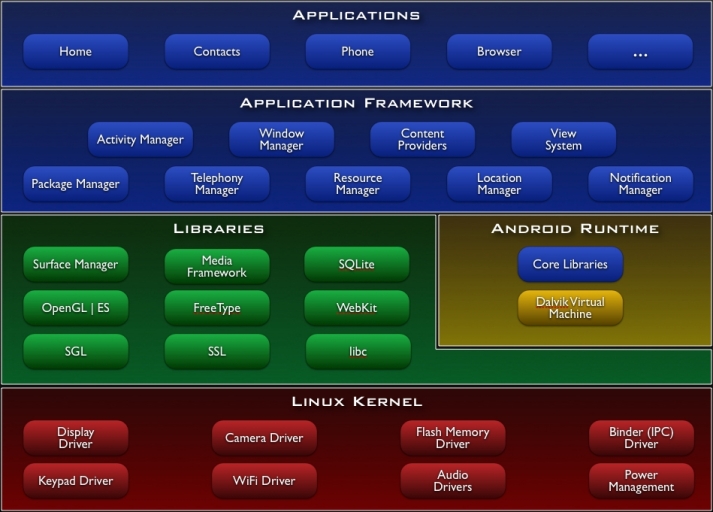


Figure 1 :- Android system architecture. Green items are written in C/C++, blue items are written in Java and run in the Dalvik VM. Image taken from [06, What is Android?]

**Anatomy of android application**

There are four building blocks to an Android application:

* Activity
* Intent Receiver
* Service
* Content Provider

Not every application needs to have all four, but your application will be written with some combination of these.

Once you have decided what components you need for your application, you should list them in a file called AndroidManifest.xml. This is an XML file where you declare the components of your application and what their capabilities and requirements are. We will discuss soon, what the AndroidManifest.xml is responsible for.

**Activity** :

Activities are the most common of the four Android building blocks. An activity is usually a single screen in your application. Each activity is implemented as a single class that extends the Activity base class. Your class will display a user interface composed of Views and respond to events. Most applications consist of multiple screens. For example, a text messaging application might have one screen that shows a list of contacts to send messages to, a second screen to write the message to the chosen contact, and other screens to review old messages or change settings. Each of these screens would be implemented as an activity.

**Intent and Intent Filters :**

Android uses a special class called Intent to move from screen to screen. Intent describe what an application wants done. The two most important parts of the intent data structure are the action and the data to act upon. Typical values for action are MAIN (the front door of the application), VIEW, PICK, EDIT, etc. The data is expressed as a Uniform Resource Indicator (URI). For example, to view a website in the browser, you would create an Intent with the VIEW action and the data set to a Website-URI.

**Intent Receiver :**

You can use an IntentReceiver when you want code in your application to execute in reaction to an external event, for example, when the phone rings, or when the data network is available, or when it's midnight. Intent receivers do not display a UI, although they may display Notifications to alert the user if something interesting has happened. Intent receivers are also registered in AndroidManifest.xml, but you can also register them from code using Context.registerReceiver().

**Service :**

A Service is code that is long-lived and runs without a UI. A good example of this is a media player playing songs from a play list. In a media player application, there would probably be one or more activities that allow the user to choose songs and start playing them. However, the music playback itself should not be handled by an activity because the user will expect the music to keep playing even after navigating to a new screen. In this case, the media player activity could start a service using

**Content Provider :**

Applications can store their data in files, a SQLite database, preferences or any other mechanism that makes sense. A content provider, however, is useful if you want your application's data to be shared with other applications. A content provider is a class that implements a standard set of methods to let other applications store and retrieve the type of data that is handled by that content provider

.

Explain Modules Implemented

import android.graphics.Color;

import android.os.Bundle;

import android.view.View;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.android.volley.Request;

import com.android.volley.RequestQueue;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.StringRequest;

import com.android.volley.toolbox.Volley;

import com.google.android.gms.ads.AdRequest;

import com.google.android.gms.ads.AdView;

import com.google.android.gms.ads.MobileAds;

import com.google.android.gms.ads.initialization.InitializationStatus;

import com.google.android.gms.ads.initialization.OnInitializationCompleteListener;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.text.DecimalFormat;

Project Code

**HomeScreen.java**

A screenshot of a computer

Description automatically generated

**SendToUserList.java**

A screenshot of a computer

Description automatically generated with medium confidence

**TransactionHistory.java**

A screenshot of a computer

Description automatically generated

**UserData.java**

A screenshot of a computer

Description automatically generated

**Transaction.java**

A screenshot of a computer

Description automatically generated

**User.java**

Text

Description automatically generated

**activity\_home\_screen.xml**

Graphical user interface

Description automatically generated

**activity\_send\_to\_user.xml**

A screenshot of a computer

Description automatically generated with medium confidence

**activity\_transaction\_history.xml**

A screenshot of a computer

Description automatically generated with medium confidence

**activity\_user\_data.xml**

A screenshot of a computer

Description automatically generated with medium confidence

**activity\_users\_list.xml**

A screenshot of a computer

Description automatically generated with medium confidence

**dialog\_box.xml**

A screenshot of a computer

Description automatically generated with medium confidence

**transaction\_history\_item.xml**

A screenshot of a computer

Description automatically generated with medium confidence

**user\_list\_item.xml**

A screenshot of a computer

Description automatically generated with medium confidence

**OUTPUT SCREENSHOT**

**Home**

Logo, company name

Description automatically generated

**All Users**

Graphical user interface

Description automatically generated

**User Details**

Graphical user interface, application

Description automatically generated

**Transfer money**

Graphical user interface, text, application

Description automatically generated

**Choose Account to tranfer ( gagan tripathi )**

Graphical user interface

Description automatically generated

**Transaction Successful**

Logo, company name

Description automatically generated

**All transaction history**

Graphical user interface, text, application

Description automatically generated