

**ANDROID APP DEVELOPMENT**

**MONEY MATTERS: A PERSONAL FINANCE MANAGEMENT APP**

**A PROJECT**

**SUBMITTED BY**

**S. SURIYA SUJI (20201231506241)**

**B. ESAKKIAMMAL (20201231506210)**

**M. KEERTHANA (20201231506220)**

**S. NITHYA STEPHEN (20201231506226)**

**S. VANAPETCHI VANITHA (20201231506245)**

**MENTOR**

**DR . T.ARUL RAJ M.Sc., M.Phil.,Ph.D**

*in partial fulfilment of the requirements for the award of degree of*

**BACHELOR OF SCIENCE**



**SRI PARAMA KALYANI COLLEGE**

**ALWARKURICHI**

**APRIL-2023**

## TABLE OF COTENTS

Chapter No.	Title	Page No.
<b>I</b>	<b>INTRODUCTION</b>	
	<b>1.1 Overview</b>	
	<b>1.2 Purpose</b>	
<b>II</b>	<b>PROBLEM DEFINITION &amp; DESIGN THINKING</b>	
	<b>2.1 Empathy Map</b>	
	<b>2.2 Ideation &amp; Brainstorming Map</b>	
<b>III</b>	<b>RESULT</b>	
<b>IV</b>	<b>ADVANTAGES &amp; DISADVANTAGES</b>	
<b>V</b>	<b>APPLICATIONS</b>	
<b>VI</b>	<b>CONCLUSION</b>	
<b>VII</b>	<b>FUTURE SCOPE</b>	
<b>VIII</b>	<b>APPENDIX</b>	

## **CHAPTER-I**

### **INTRODUCTION:**

#### **1.1 OVERVIEW**

The app allows user to keep track of their expenses and accounts, and provides an overview of their financial status. Users can set a budget for various expenses and view their progress towards it.

Money Matters is a personal finance management app designed to help individuals track their income, expenses, and savings. It allows users to create budgets, set financial goals, and monitor their progress towards achieving those goals.

The app offers a variety of features such as expense tracking, bill reminders, investment tracking, and customized reports. Users can link their bank accounts, credit cards, and other financial accounts to the app for a comprehensive view of their financial status.

Money Matters also offers tools to help users make informed financial decisions such as loan calculators, investment analysis, and retirement planning. It also provides resources and tips on how to improve personal finance management and increase financial literacy.

Overall, Money Matters is a helpful tool for anyone looking to take control of their finances and achieve their financial goals.

Money Matters is a personal finance management app designed to help individuals track their expenses, create budgets, and manage their finance more effectively. The app offers a range of features including:

**Expense Tracking:** The app allows users to track their expenses easily by categorizing them and creating custom tags. This helps users understand their spending patterns and identify areas where they can cut back.

**Budgeting:** Money Matters allows users to create budgets based on their income and expenses. Users can set spending limits for each category and track their progress throughout the month.

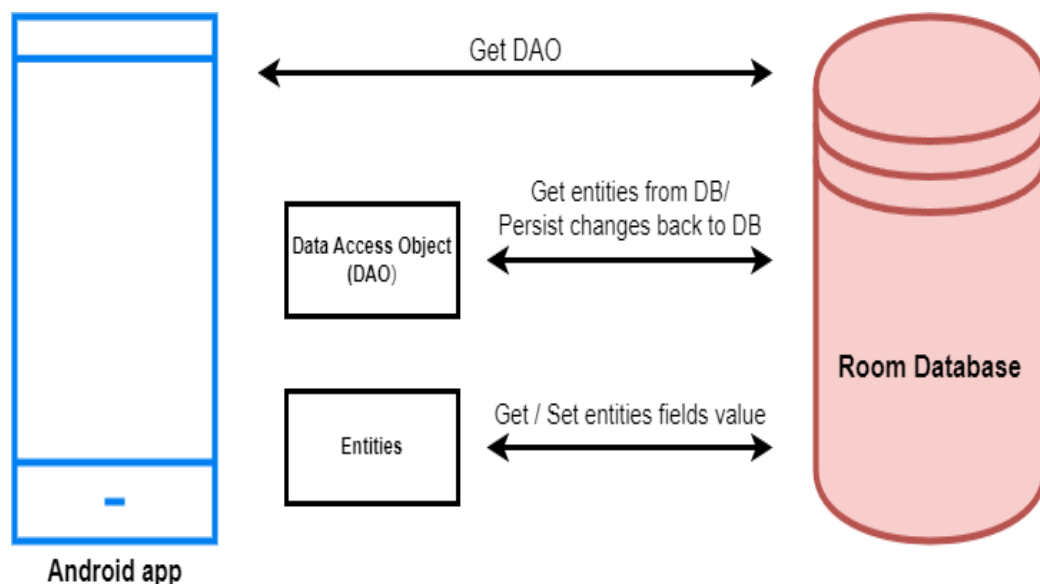
**Bill Reminders:** The app sends reminders when bills are due, ensuring that users never miss a payment and incur late fees.

**Investment Tracking:** Money Matters offers investment tracking to help users keep track of their investments and monitor their portfolio's performance.

**Reports and Analysis:** The app provides reports and analysis of user's Financial data, including income, expenses, and investments. This helps users gain a better understanding of their financial situation and make informed decisions.

Overall, Money Matters is a comprehensive personal finance management app that can help users achieve their financial goals by providing them with the tools they need to manage their money effectively.

## Architecture



## 1.2 PURPOSE

The purpose of a personal finance management app like "*Money Matters*" is to help individuals manage their finances in a more organized and efficient way. The app can provide a variety of features and tools to help users track their spending, create and stick to a budget, set financial goals, and monitor their overall financial health.

Overall, a personal finance management app like "Money Matters" can help users take control of their finances and make better financial decisions, ultimately leading to greater financial stability and security.

The personal finance app development segment has been growing steadily over the last decade, with lockdowns giving it an explainable boost. The need to keep track of earnings and spendings, and shopping online without having to visit a physical bank has made building personal finance apps a highly lucrative investment.

From 2020 through 2021 the number of active users of money management solutions increased to almost 150% by some estimates.

If you are looking to invest into a personal finance app development, this article will give you an understanding of the current state of the market and provide you with some winning strategies for building a successful product

Put simply, it is a piece of software that allows people to control their finances through an online interface or via a mobile device like a smartphone or a tablet.

People use money management apps for a number of purposes:

- Monitoring spendings and earnings,
- saving money and staying on budget,
- regular balance checks,
- online transactions and payments,
- smart deposits and investments, and
- insights and advice about personal finances.

Personal finance apps like Mint have been around for less than a decade, however, they boast an impressive user base and are attracting new users willing to enjoy the obvious benefits:

- a comprehensive view of all their financial transactions,
- the ability to set financial goals and see them through to completion,
- easy budgeting and savings,

- an awareness of one's financial behavior patterns,
- and learning the basic principles of personal moneymanagement

Now that we've covered the benefits of these solutions,let's look closely at the existing types of consumer-oriented money management software.

### **Types of personal finance apps**

Most personal finance apps focus on one or several aspects of customers' needs. Technologically speaking, these applications fall into two distinct categories: simple solutions with manual dataentry, and software capable of automatically loading data and processing it to deliver insights

Both application types have its pros and cons: while basic apps rely too much on the human factor (users may be too lazy to enterdata), they are surely more secure and less expensive to develop compared to software connected to a bank account.

More specifically, the money management software includes the following types

### Types of Personal Finance Apps



Spending  
tracker apps



Investment  
apps



Budget  
planning solutions



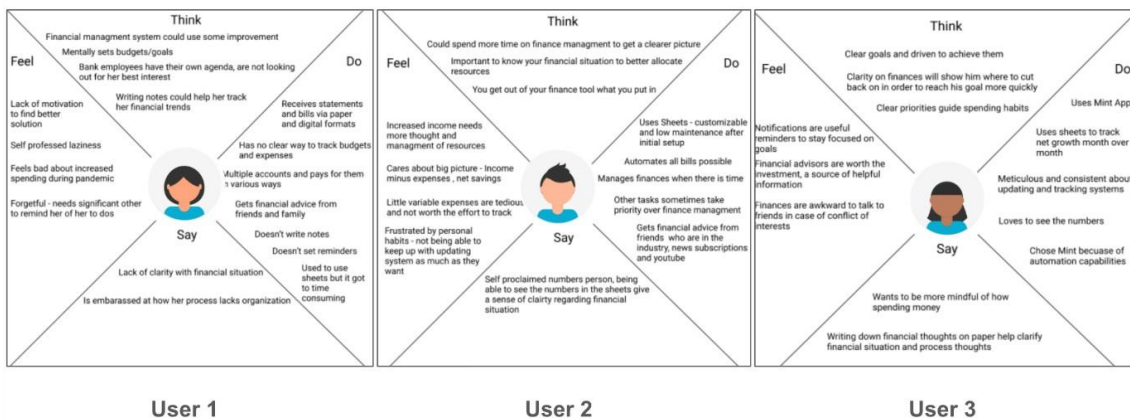


## CHAPTER – II

### PROBLEM DEFINITIONS & DESIGN THINKING

#### 2.1 EMPATHY MAP

EMPATHY MAP For Money matters A personal financemanagement app



Sure, I can help you with that! Here's an empathy map for "Money Matters" personal finance management app:

USER:

Name: Sarah

Age: 28

Occupation: Marketing Manager Annual

Income: \$70,000

Goal: Wants to be more financially stable and save more money

**FEELINGS:**

Anxious about her financial situation Stressed about making ends meet Frustrated with overspending Overwhelmed with the idea of managing her finances

**THOUGHTS:**

"I need to get my finances in order." "I don't know where to start."

"I wish there was an easy way to track my spending." "I want to save money, but it's hard to do."

**PAIN POINTS:**

Difficulty tracking her expenses and income Overspending on unnecessary purchases Not having a clear understanding of her financial situation Feeling like she doesn't have enough money to save

**GAINS:**

Having a better understanding of her financial situation Saving more money each month Feeling more in control of her finances Having a more secure financial future

Based on this empathy map, some potential design solutions for "Money Matters" personal finance management app could include:

A simple and user-friendly interface that makes it easy for Sara to track her spending and income.

Budgeting tools that help Sarah set realistic spending limits and stay on track with her financial goals. Personalized insights and recommendations that help Sarah identify areas where she can cut back on spending and save more money.

Integration with Sarah's existing financial accounts to provide a complete picture of her financial situation in one place.

Notifications and reminders to help Sarah stay on top of her bills and avoid late fees. Sure, here's an ideation and brainstorming map for "Money Matters" personal finance management app:

#### **CORE FEATURES:**

**Accounts:** allows users to link and manage multiple financial accounts in one place

**Budget:** helps users create and track a monthly or weekly budget

**Transactions:** displays all transactions in a categorized and searchable format

**Goals:** allows users to set financial goals and track progress towards them

**Reports:** provides visual charts and graphs to help users understand their spending patterns and financial health

## **ADDITIONAL FEATURES:**

**Bill reminders:** sends notifications for upcoming bills to avoid late fees

**Savings tracker:** allows users to track their savings progress and set savings goals

**Debt payoff planner:** helps users create a plan to pay off debt

**Investment tracker:** allows users to track investments and monitor their portfolio

**Cash flow forecasting:** predicts future cash flow based on current financial data

**USER EXPERIENCE:** Simple and user-friendly interface Customizable dashboards and settings Personalized insights and recommendations Security measures to protect sensitive financial data Integration with third-party financial services

**MARKETING STRATEGY:** Social media campaigns targeting millennials and Gen Z Influencer partnerships with personal finance experts and bloggers SEO optimization to increase app visibility in search results Referral program to incentivize existing users to share the app with their friends and family Educational resources and blog content to help users improve their financial literacy.

These are just a few ideas for the "Money Matters" personal finance management app. By combining core features with additional features and a user-friendly interface, the app can provide a comprehensive solution to help users manage their finances and achieve their financial goal

## **2.2 Ideation & Brainstorming Map**

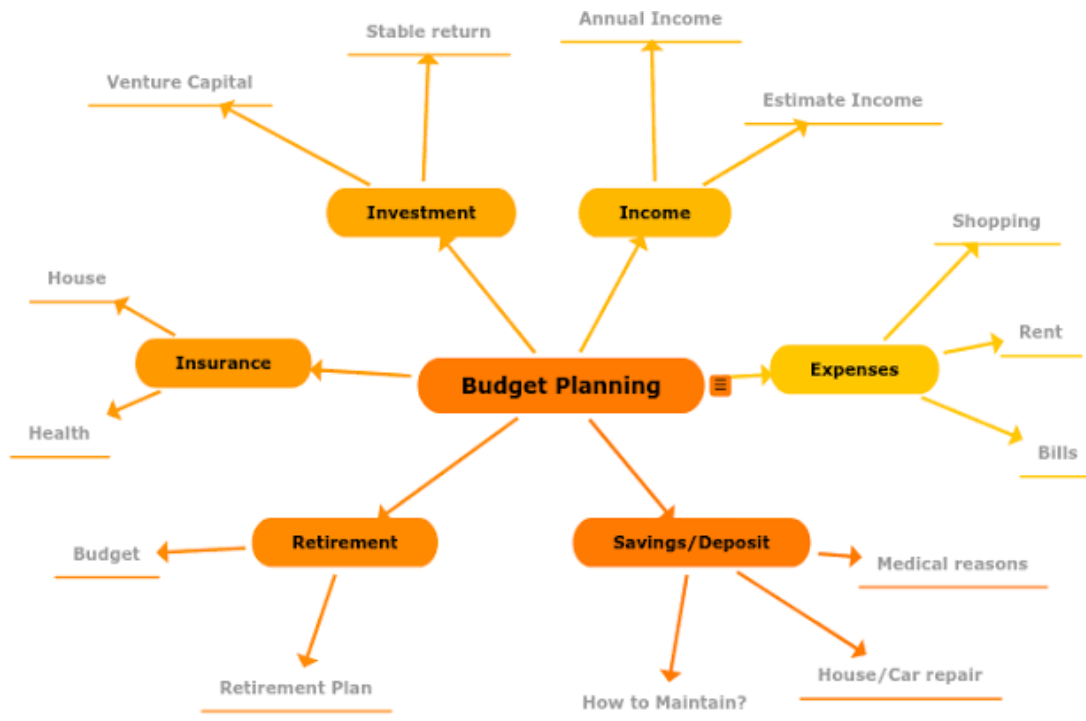
A financial plan can be defined as an exhaustive assessment of a person's, business's or other entity's current and future financial situation.

It is usually composed to evaluate the financial activities and position by following a certain template adapted to the user's needs. There is no strict format of a financial plan that is obligatory to be followed.

Depending on one's needs and goals, a financial plan can be composed and presented in different ways.

A practical way of creating a financial plan is by using the mind mapping technique.

In fact, mind maps are characterized by the ability to support organizing and outlining information, which makes them helpful for making a financial plan.



With a mind map, you could easily see the bigger picture of a specific situation. So if you make your financial plan with a mind map, you would be able to get a better perspective of your personal financial iMindQ financial plan with a mind map, you would be able to get a better perspective of your personal financial state and reflect on how certain aspects could and should be improved.

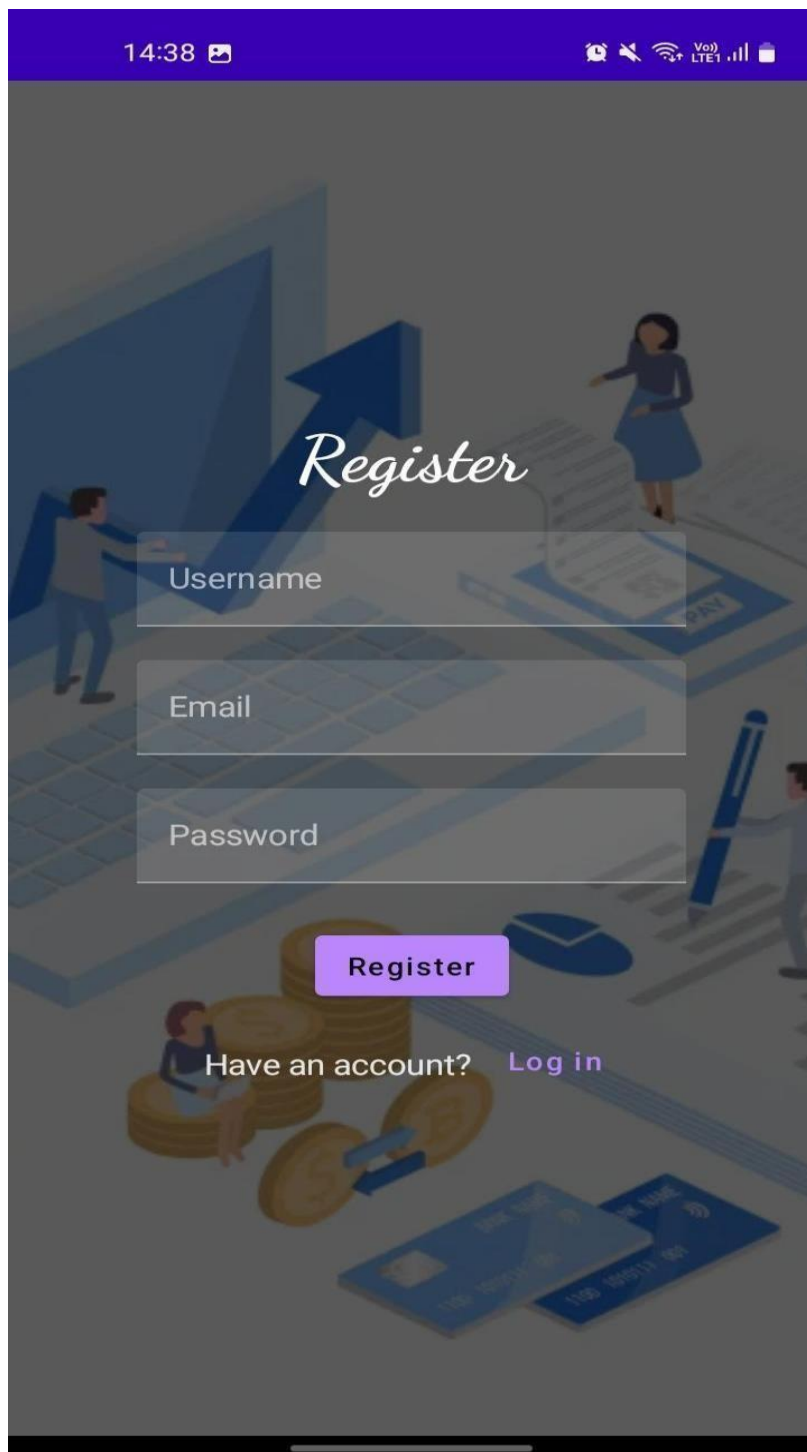
A mind map is a perfect solution for representing a sound financial plan because it's adaptable and flexible to change which is essential when it comes to personal finance.

## CHAPTER – III

### RESULT

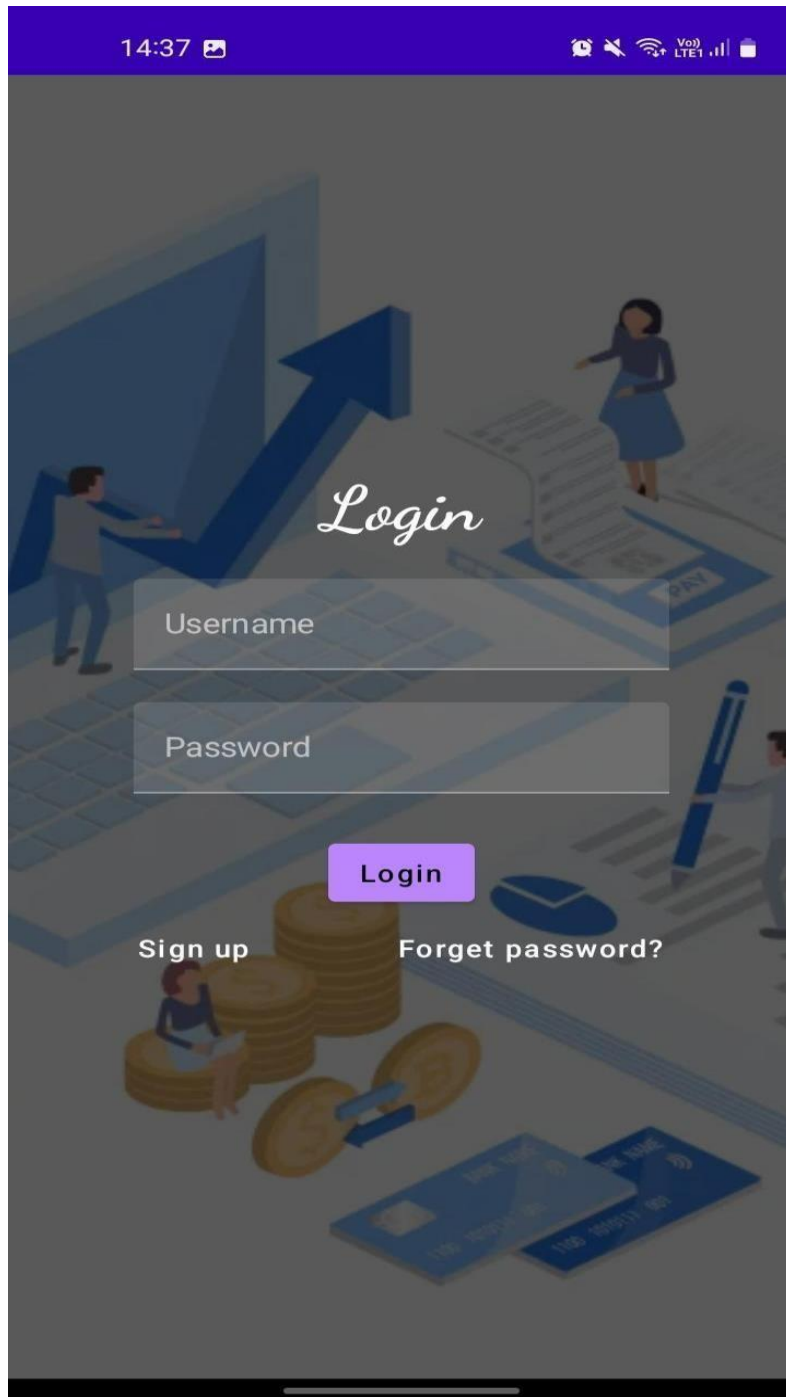
#### 3.1 REGISTER PAGE

- Users register into the application



### 3.2 LOGIN PAGE

- After registration , user logs into the application.













### 3.3 MAIN PAGE

- User enters into the main page



### 3.4 Expenses page:

14:38



**Item Name**

Item Name

pizza

**Quantity of item**

Quantity

2

**Cost of the item**

Cost

400

Submit

Add Expenses

Set Limit

View Records

3.5 Set Limit Page before adding any data in expenses:

14:39

## Monthly Amount Limit

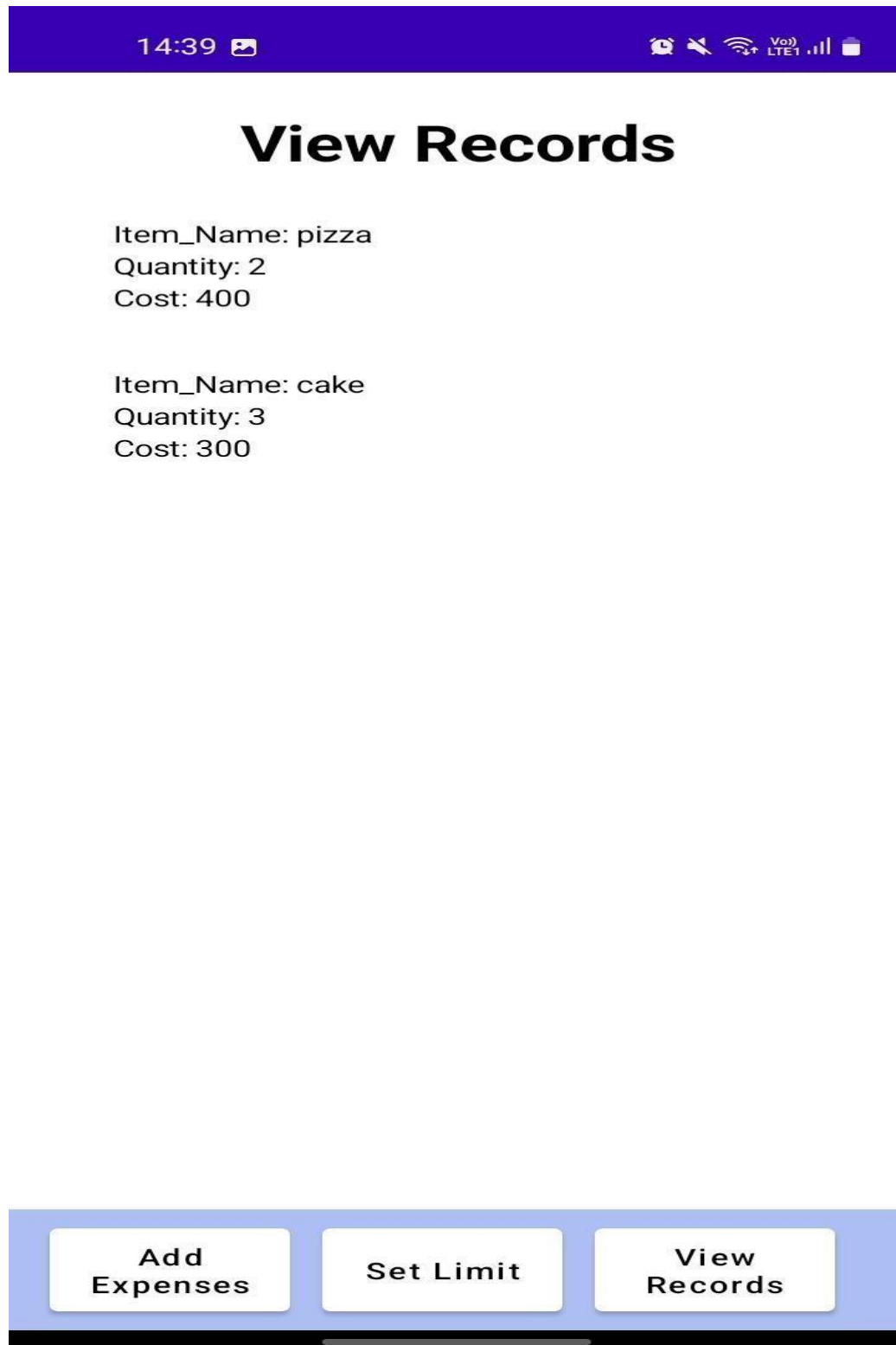
Set Amount Limit

Set Limit

Remaining Amount: 10000







Add Expenses Set Limit View Records

### 3.6 View Records page



### 3.7Set Limit Page After adding expenses in add expense page:

14:39



## Monthly Amount Limit

Set Amount Limit

Set Limit

Remaining Amount: 9300

Add Expenses

Set Limit

View Records

## **CHAPTER – IV**

### **ADVANTAGES & DISADVANTAGES**

#### **4.1 Advantage**

Advantages of Money matters A Personal Finance Management App

There are many advantages to using a personal finance management app like "Money Matters."

Here are some of the key benefits:

**EASY TO TRACK SPENDING:** The app can help users track their spending in a more organized and efficient way, making it easier to monitor expenses and identify areas where they can cut back.

**BETTER BUDGETING:** The app can help users create and stick to a budget, providing tools and insights to help them set realistic spending limits and track their progress towards financial goals.

**FINANCIAL GOAL SETTING:** The app can help users set and track financial goals, such as saving for a down payment on a house or paying off debt, providing motivation and accountability along the way.

**BILL REMINDERS:** The app can provide reminders and alerts for upcoming bills, helping users avoid late fees and stay on top of their payments.

**INVESTMENT TRACKING:** The app can make it easier to monitor investments and track progress towards long-term financial goals, such as retirement savings.

**SAVINGS TRACKING:** The app can help users track their savings progress and set savings goals, making it easier to achieve financial milestones.

**BETTER FINANCIAL DECISIONS:** The app can provide users with a clearer understanding of their financial situation, helping them make more informed decisions about how to manage their money.

**INCREASED FINANCIAL SECURITY:** By using the app to monitor their finances and make better financial decisions, users can increase their overall financial stability and security.

Overall, a personal finance management app like "Money Matters" can provide users with a range of benefits that can help them improve their financial health and achieve their financial goals.

## 4.2. Disadvantages

### Disadvantage of Money matters A Personal Finance Management App

While there are many advantages to using a personal finance management app like "Money Matters," there are also some potential disadvantages to consider. Here are a few:

**SECURITY CONCERNS:** Personal finance management apps often require users to input sensitive financial information, which could be at risk of being compromised if the app's security measures are not sufficient.

**TECHNICAL ISSUES:** Like any software, personal finance management apps can experience technical glitches or bugs, which could potentially cause problems with tracking or managing finances.

**COST:** Some personal finance management apps may require a subscription or charge fees for certain features, which could be a disadvantage for users who are on a tight budget.

**RELIANCE ON TECHNOLOGY:** Using a personal finance management app can make it easy to rely too heavily on technology for financial management, which could be a disadvantage if the app experiences technical problems or if the user loses access to their account.

**LACK OF HUMAN INTERACTION:** Personal finance management apps are often designed to be automated and self-serve, which means users may not have access to human support or guidance if they have questions or need assistance.

While these potential disadvantages should be taken into account, they are generally outweighed by the benefits that personal finance management apps can provide for users.



## **CHAPTER – V**

### **APPLICATIONS**

#### Application of Money matters A Personal Finance Management App

The "Money Matters" personal finance management app can be used in a variety of ways to help users manage their finances more effectively.

Here are some of the key applications of the app:

**TRACKING SPENDING:** One of the primary uses of the app is to track spending across multiple accounts and categories, making it easier for users to stay on top of their expenses and identify areas where they can cut back.

**BUDGETING:** The app can be used to create and track a budget, providing users with tools and insights to help them set realistic spending limits and achieve their financial goals.

**GOAL SETTING:** The app can be used to set and track financial goals, such as saving for a down payment on a house or paying off debt, providing motivation and accountability along the way.

**REMINDERS AND ALERTS:** The app can provide reminders and alerts for upcoming bills and payments, helping users avoid late fees and stay on top of their financial obligations.

**INVESTMENT TRACKING:** The app can be used to monitor investments and track progress towards long-term financial goals, such as retirement savings.

**SAVINGS TRACKING:** The app can help users track their savings progress and set savings goals, making it easier to achieve financial milestones.

**BETTER FINANCIAL DECISIONS:** The app can provide users with a clearer understanding of their financial situation, helping them make more informed decisions about how to manage their money.

Overall, the "Money Matters" personal finance management app can be used in a variety of ways to help users achieve greater financial stability and security. By providing tools and insights to help users track and manage their finances more effectively, the app can be a valuable tool for anyone looking to improve their financial health.

## **CHAPTER – VI**

### **CONCLUSIONS**

In conclusion, "Money Matters" is a personal finance management app that offers a range of tools and features to help users manage their finances more effectively.

By providing an easy-to-use interface for tracking spending, creating and sticking to a budget, setting and tracking financial goals, and monitoring investments and savings, the app can help users achieve greater financial stability and security.

While there are potential disadvantages to using a personal finance management app, such as security concerns, technical issues, and the cost of subscriptions or fees, these are generally outweighed by the benefits that the app can provide.

With its range of features and intuitive interface, "Money Matters" is a valuable tool for anyone looking to improve their financial health and achieve their financial goals

## **CHAPTER – VII**

### **FUTURE SCOPE**

The future scope of "Money Matters" personal finance management app is vast, as there are always new and innovative ways to improve the user experience and provide even more tools and insights to help users manage their finances more effectively. Here are a few potential areas for future development and expansion:

**AI AND MACHINE LEARNING:** As artificial intelligence and machine learning continue to advance, "Money Matters" could incorporate these technologies to provide even more personalized insights and recommendations based on each user's unique financial situation and goals.

**CRYPTOCURRENCY INTEGRATION:** As cryptocurrencies become more mainstream, "Money Matters" could incorporate integration with popular cryptocurrencies and blockchain technology to provide users with a more comprehensive view of their overall financial portfolio.

**COMMUNITY AND SOCIAL FEATURES:** "Money Matters" could expand its community and social features to enable users to connect with others who share their financial goals and challenges, providing support and encouragement along the way.

**GAMIFICATION:** The app could incorporate features to make financial management more engaging and fun, encouraging users to stick to their budget and savings goals.

**FINANCIAL EDUCATION:** The app could expand its educational resources to provide users with more comprehensive financial education and guidance, helping them make informed decisions about their money and achieve greater financial literacy.

Overall, the future scope of "Money Matters" is bright, as there are many potential areas for development and expansion that could help users achieve greater financial stability and security. By continuing to innovate and provide valuable tools and insights, "Money Matters" has the potential to become an even more essential tool for managing personal finances in the years to come.

## CHAPTER -VIII

### APPENDIX

#### SOURCE CODE :

##### LOGIN ACTIVITY.KT

```
package  
com.example.expensestracker
```

```
import android.content.Context  
import android.content.Intent  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.Image  
import androidx.compose.foundation.layout.*  
import androidx.compose.material.*  
import androidx.compose.runtime.*  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.graphics.Color  
import androidx.compose.ui.layout.ContentScale  
import androidx.compose.ui.res.painterResource  
import androidx.compose.ui.text.font.FontFamily  
import androidx.compose.ui.text.font.FontWeight  
import  
androidx.compose.ui.text.input.PasswordVisualTransformation  
import androidx.compose.ui.text.input.VisualTransformation  
import androidx.compose.ui.tooling.preview.Preview  
import androidx.compose.ui.unit.dp  
import androidx.compose.ui.unit.sp  
import androidx.core.content.ContextCompat  
import  
com.example.expensestracker.ui.theme.ExpensesTrackerTheme  
  
class LoginActivity : ComponentActivity() {  
    private lateinit var databaseHelper: UserDatabaseHelper  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        databaseHelper = UserDatabaseHelper(this)  
        setContent {
```

```

ExpensesTrackerTheme {
// A surface container using the 'background' color from the
theme
Surface(
modifier = Modifier.fillMaxSize(),

color = MaterialTheme.colors.background
) {
LoginScreen(this, databaseHelper)
}
}
}
}
}
@Composable
fun LoginScreen(context: Context, databaseHelper:
UserDatabaseHelper) {

Image(
painterResource(id = R.drawable.img_1), contentDescription
= "",
alpha = 0.3F,
contentScale = ContentScale.FillHeight,

)

var username by remember { mutableStateOf("") }
var password by remember { mutableStateOf("") }
var error by remember { mutableStateOf("") }

Column(
modifier = Modifier.fillMaxSize(),
horizontalAlignment = Alignment.CenterHorizontally,
verticalArrangement = Arrangement.Center
) {

Text(
fontSize = 36.sp,
fontWeight = FontWeight.ExtraBold,
fontFamily = FontFamily.Cursive,
color = Color.White,
text = "Login"
)
Spacer(modifier = Modifier.height(10.dp))

TextField(
value = username,
onValueChange = { username = it },

```

```

label = { Text("Username") },
modifier = Modifier.padding(10.dp)
.width(280.dp)
)

TextField(
value = password,
onValueChange = { password = it },
label = { Text("Password") },
modifier = Modifier.padding(10.dp)
.width(280.dp),
visualTransformation = PasswordVisualTransformation()

)

if (error.isNotEmpty()) {
Text(
text = error,
color = MaterialTheme.colors.error,
modifier = Modifier.padding(vertical = 16.dp)
)
}

Button(
onClick = {
if (username.isNotEmpty() && password.isNotEmpty()) {
val user = databaseHelper.getUserByUsername(username)
if (user != null && user.password == password) {
error = "Successfully log in"
context.startActivity(
Intent(
context,
MainActivity::class.java
)
)
//onLoginSuccess()
}
else {
error = "Invalid username or password"
}

} else {
error = "Please fill all fields"
}
},

modifier = Modifier.padding(top = 16.dp)
)

```

```

        {
            Text(text = "Login")
        }
        Row {
            TextButton(onClick = {context.startActivity(
            Intent(

                context,
                RegisterActivity::class.java
            )
            })
            )
            { Text(color = Color.White,text = "Sign up") }
            TextButton(onClick = {
            })

        }

        {
            Spacer(modifier = Modifier.width(60.dp))
            Text(color = Color.White,text = "Forget password?")
        }
        }
        }
        }
        private fun startMainPage(context: Context) {
            val intent = Intent(context, MainActivity::class.java)
            ContextCompat.startActivity(context, intent, null)
        }
    }

```

## REGISTER ACTIVITY.KT

```

package
com.example.expensetracker
r

```

```

import android.content.Context
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color

```



```

import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
import
androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat
import
com.example.expensetracker.ui.theme.ExpensesTrackerTheme

```

```

class RegisterActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(this)
        setContent {
            ExpensesTrackerTheme {
                // A surface container using the 'background' color from the
                theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {

```

```

                    RegistrationScreen(this, databaseHelper)
                }
            }
        }
    }
}

```

```

@Composable
fun RegistrationScreen(context: Context, databaseHelper:
UserDatabaseHelper) {

    Image(
        painterResource(id = R.drawable.img_1), contentDescription
        = "",
        alpha = 0.3F,
        contentScale = ContentScale.FillHeight,

    )

```

```
var username by remember { mutableStateOf("") }
var password by remember { mutableStateOf("") }
var email by remember { mutableStateOf("") }
var error by remember { mutableStateOf("") }
```

```
Column(
    modifier = Modifier.fillMaxSize(),
    horizontalAlignment = Alignment.CenterHorizontally,
    verticalArrangement = Arrangement.Center
) {
```

```
    Text(
        fontSize = 36.sp,
        fontWeight = FontWeight.ExtraBold,
        fontFamily = FontFamily.Cursive,
        color = Color.White,
        text = "Register"
    )
```

```
    Spacer(modifier = Modifier.height(10.dp))
    TextField(
        value = username,
        onChange = { username = it },
        label = { Text("Username") },
        modifier = Modifier
            .padding(10.dp)
            .width(280.dp)
    )
```

```
    TextField(
        value = email,
        onChange = { email = it },
        label = { Text("Email") },
        modifier = Modifier
            .padding(10.dp)
            .width(280.dp)
    )
```

```
    TextField(
        value = password,
        onChange = { password = it },
        label = { Text("Password") },
```

```
modifier = Modifier
.padding(10.dp)
.width(280.dp),
visualTransformation = PasswordVisualTransformation()
)
```

```
if (error.isNotEmpty()) {
    Text(
        text = error,
        color = MaterialTheme.colors.error,
        modifier = Modifier.padding(vertical = 16.dp)
    )
}
```

```
Button(
    onClick = {
        if (username.isNotEmpty() && password.isNotEmpty() &&
            email.isNotEmpty()) {
            val user = User(
                id = null,
                firstName = username,
                lastName = null,
                email = email,
                password = password
            )
            databaseHelper.insertUser(user)
            error = "User registered successfully"
            // Start LoginActivity using the current context
            context.startActivity(
                Intent(
                    context,
                    LoginActivity::class.java
                )
            )
        }
    }
)
```

```
} else {
    error = "Please fill all fields"
}
},
modifier = Modifier.padding(top = 16.dp)
) {
    Text(text = "Register")
}
```

```
Spacer(modifier = Modifier.width(10.dp))
Spacer(modifier = Modifier.height(10.dp))
```

```

Row() {
    Text(
        modifier = Modifier.padding(top = 14.dp), text = "Have an
        account?"
    )
    TextButton(onClick = {
        context.startActivity(
            Intent(
                context,
                LoginActivity::class.java
            )
        )
    })
}

{
    Spacer(modifier = Modifier.width(10.dp))
    Text(text = "Log in")
}
}
}
}

private fun startLoginActivity(context: Context) {
    val intent = Intent(context, LoginActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}

```

## MAIN ACTIVITY.KT

```

package
com.example.expensest
racker

```

```

import android.annotation.SuppressLint
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontWeight

```

```

import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import
com.example.expensetracker.ui.theme.ExpensesTrackerTheme

class MainActivity : ComponentActivity() {
    @SuppressLint("UnusedMaterialScaffoldPaddingParameter")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Scaffold(
                // in scaffold we are specifying top bar.
                bottomBar = {
                    // inside top bar we are specifying
                    // background color.
                    BottomAppBar(backgroundColor = Color(0xFFadbf4),
                        modifier = Modifier.height(80.dp),

                    // along with that we are specifying
                    // title for our top bar.

                    content = {

                        Spacer(modifier = Modifier.width(15.dp))

                        Button(
                            onClick =
                                { startActivity(Intent(applicationContext, AddExpensesActivity::class
                                    s.java)) },
                            colors = ButtonDefaults.buttonColors(backgroundColor =
                                Color.White),
                            modifier = Modifier.size(height = 55.dp, width = 110.dp)
                        )
                        {
                            Text(
                                text = "Add Expenses", color = Color.Black, fontSize = 14.sp,
                                textAlign = TextAlign.Center
                            )
                        }
                    }
                }
            )
        }
    }
}

```

```
Spacer(modifier = Modifier.width(15.dp))
Button(
    onClick = {
        startActivity(
            Intent(
                applicationContext,
                SetLimitActivity::class.java
            )
        )
    },
    colors = ButtonDefaults.buttonColors(backgroundColor =
        Color.White),
    modifier = Modifier.size(height = 55.dp, width = 110.dp)
)
{
    Text(
        text = "Set Limit", color = Color.Black, fontSize = 14.sp,
        textAlign = TextAlign.Center
    )
}
```

```
Spacer(modifier = Modifier.width(15.dp))
```

```
Button(
    onClick = {
        startActivity(
            Intent(
                applicationContext,
                ViewRecordsActivity::class.java
            )
        )
    },
    colors = ButtonDefaults.buttonColors(backgroundColor =
        Color.White),
    modifier = Modifier.size(height = 55.dp, width = 110.dp)
)
{
    Text(
        text = "View Records", color = Color.Black, fontSize = 14.sp,
        textAlign = TextAlign.Center
    )
}

}
```

```

    )
    }
    )

    {
MainPage()
    }
    }
    }
    }

@Composable
fun MainPage() {
Column(
    modifier = Modifier.padding(20.dp).fillMaxSize(),
    verticalArrangement = Arrangement.Center,
    horizontalAlignment = Alignment.CenterHorizontally
) {

    Text(text = "Welcome To Expense Tracker", fontSize = 42.sp,
fontWeight = FontWeight.Bold, textAlign = TextAlign.Center)

    Image(painterResource(id = R.drawable.img_1),
contentDescription = "", modifier = Modifier.size(height = 500.dp,
width = 500.dp))
    }
    }
    }
    }

```

#### **ANDROIDMANIFEST.XML CODE**

```

<?xml
version="1.
0"
encoding="
utf-8"?>

    <manifest
xmlns:android="http://schemas.android.com/apk/
res/android"
xmlns:tools="http://schemas.android.com/tools"
>

    <application
android:allowBackup="true"

```

```
android:dataExtractionRules="@xml/data_extraction_rules"
android:fullBackupContent="@xml/backup_rules"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
```

```
android:supportRtl="true"
android:theme="@style/Theme.TravelApp"
tools:targetApi="31">
<activity
android:name=".RegisterActivity"
android:exported="false"
android:label="RegisterActivity"
android:theme="@style/Theme.TravelApp" />
<activity
android:name=".SingaporeActivity"
android:exported="false"
android:label="@string/title_activity_singapore"
android:theme="@style/Theme.TravelApp" />
<activity
android:name=".ParisActivity"
android:exported="false"
android:label="@string/title_activity_paris"
android:theme="@style/Theme.TravelApp" />
<activity
android:name=".BaliActivity"
android:exported="false"
android:label="@string/title_activity_bali"
android:theme="@style/Theme.TravelApp" />
<activity
android:name=".MainActivity"
android:exported="true"
android:label="@string/app_name"
android:theme="@style/Theme.TravelApp"/>
<activity
android:name=".LoginActivity"
android:exported="true"
android:label="@string/app_name"
android:theme="@style/Theme.TravelApp">
<intent-filter>
<action
android:name="android.intent.action.MAIN" />
```



```
<category  
  android:name="android.intent.category.LAUNCHER" />  
</intent-filter>  
</activity>  
</application>  
  
</manifest>
```