INTRODUCTION

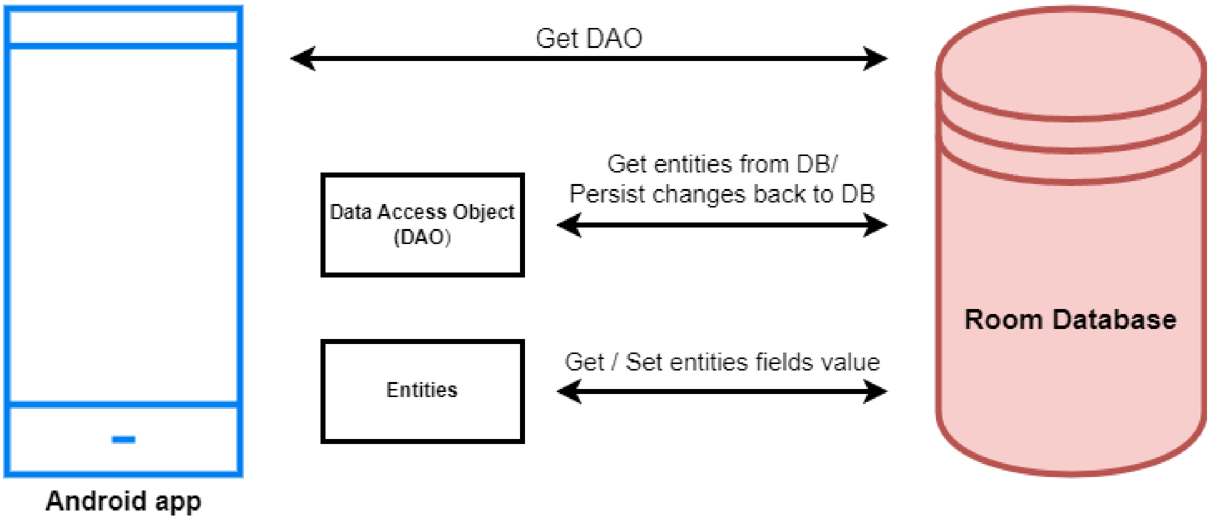
1.1PROJECT OVERVIEW

Money Matters: A Personal Finance Management App

Project Description:

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

ARCHITECHURE


Get DAO
Data Access Object (DAO)
Get entities from DB/ Persist changes back to DB
Entities
Get / Set entities fields value
Room Database
Android app

Learning Outcomes :

By end of this project:

- You’ll be able to work on Android studio and build an app.
- You’ll be able to integrate the database accordingly.

Project Workflow:

- Users register into the application.
- After registration , user logins into the application.

- User enters into the main page
- User can view the subject themes on selecting theme he can read about it.
- **Note:**
- To complete the project you need to finish up the tasks listed below

Tasks:

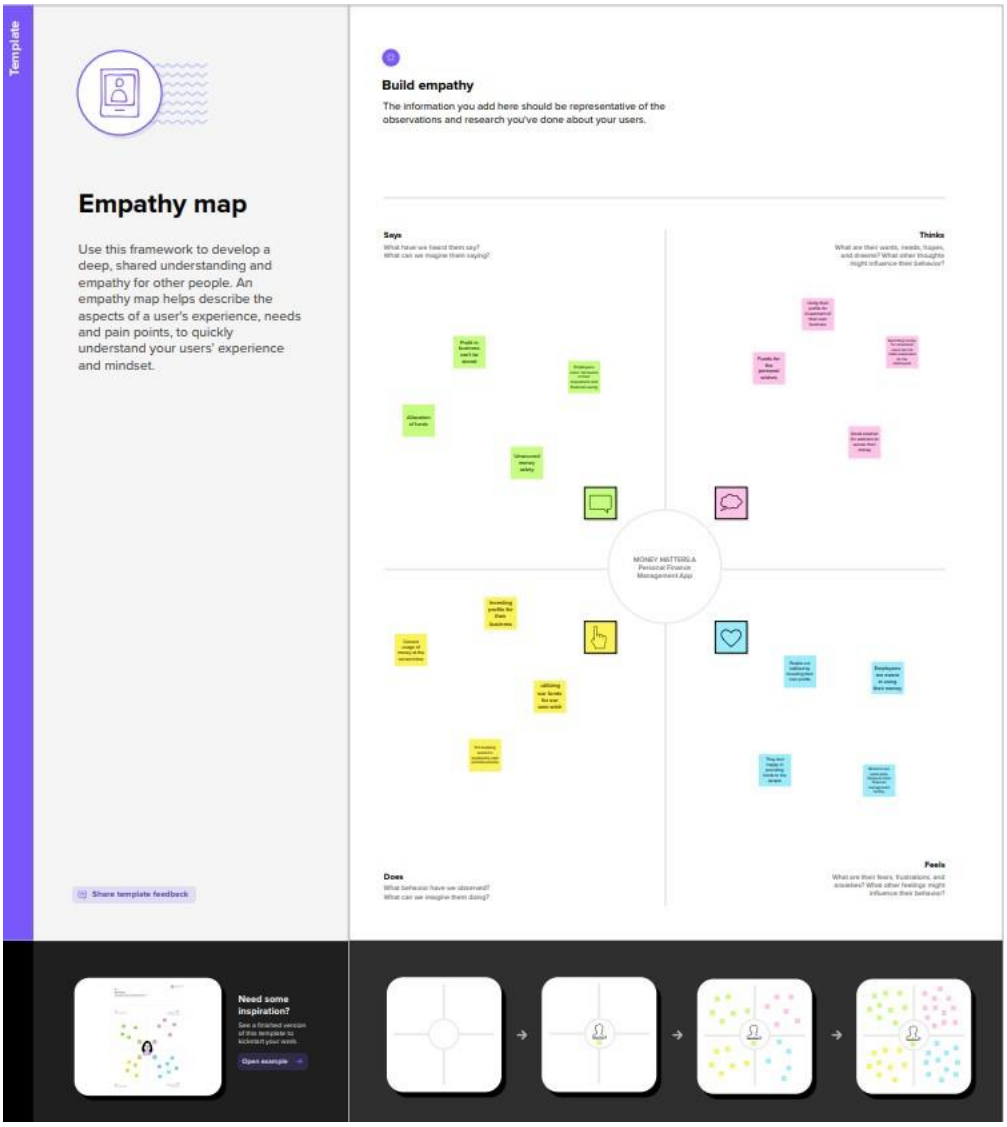
1.Required initial steps
2.Creating a new project.
3.Adding required dependencies.
4.Creating the database classes.
5.Building application UI and connecting to database.
6.Using AndroidManifest.xml
7.Running the application.

1.2 PURPOSE OF PROJECT

A Personal Finance Management App is to help individuals manage their personal finances more effectively. The app can provide users with a clear and easy-to-use platform for tracking income, expenses, and savings, and can offer insights into spending patterns and financial habits.

2 PROBLEM DEFINITION & DESIGN THINKING

2.1 EMPATHY MAP



Template

Empathy map

Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.

Share template feedback

Build empathy

The information you add here should be representative of the observations and research you've done about your users.

Says

What have we heard them say?
What can we imagine them saying?

Thinks

What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

MONEY MATTERS:A Personal Finance Management App

Does

What behavior have we observed?
What can we imagine them doing?

Feels

What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

Allocation of funds

Unsecured money safety

Funds for the personal wishes

Increasing profits for their business

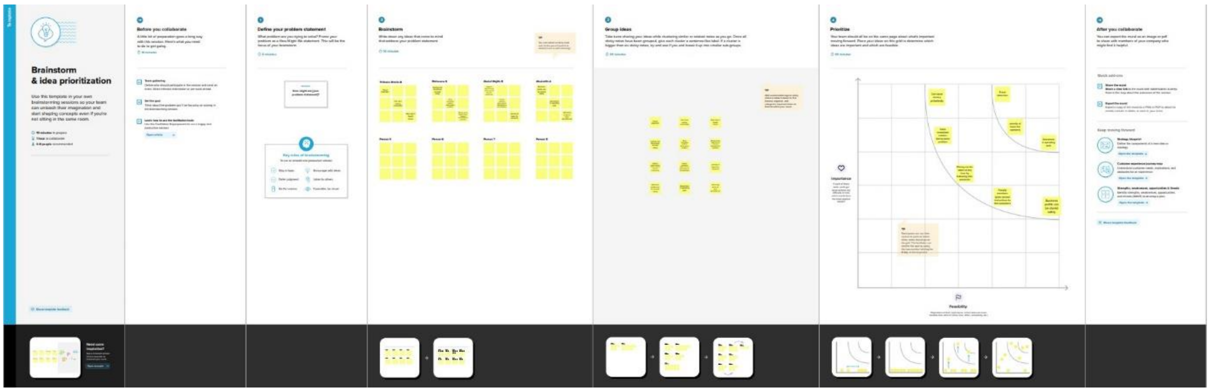
utilizing our funds for our own wish

Employees are stable in using their money

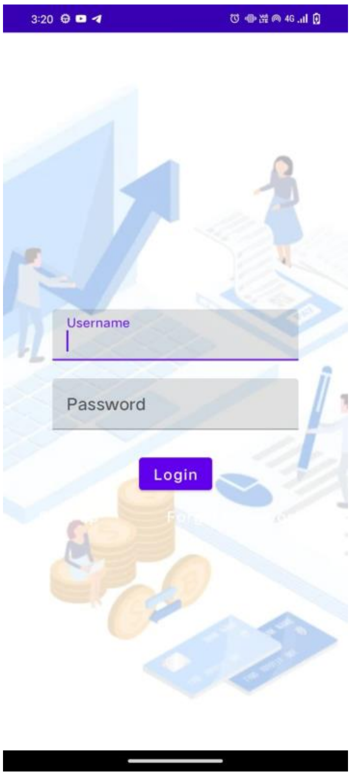
Need some inspiration?

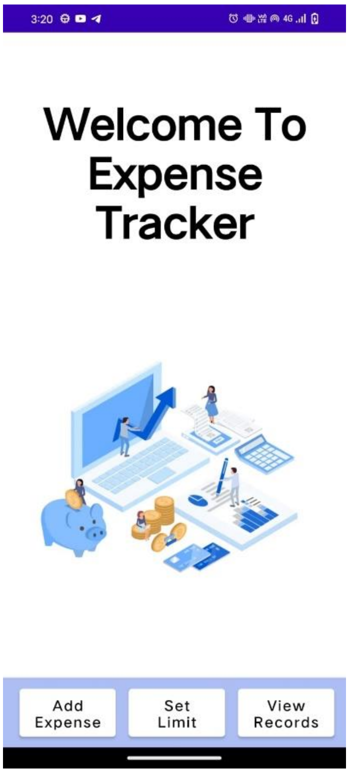
See a finished version of this template to kickstart your work.

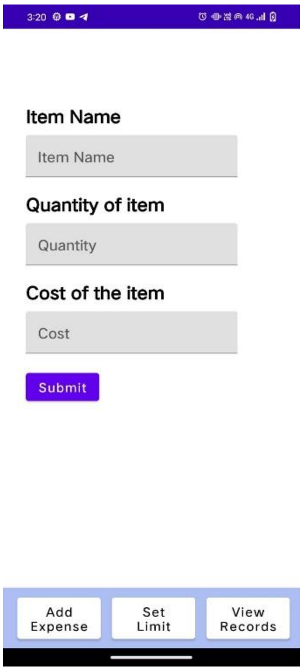
2.2 IDEATION & BRAINSTROMING

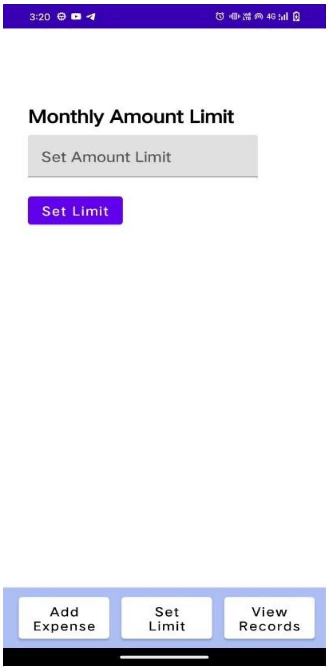

Brainstorm
& idea prioritization
Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.
Before you collaborate
Define your problem statement
Brainstorm
Group ideas
Prioritize
Importance
Feasibility
After you collaborate

3.RESULT









3:20
View Records
Add Expense
Set Limit
View Records

4.ADVANTAGES

convenience: Money Matters provides users with a convenient way to manage their personal finances on-the-go from their mobile device, without having to manually track expenses or keep paper records.

Organization: The app helps users keep their finances organized by categorizing expenses and providing detailed financial reports.

Budgeting: Money Matters allows users to create a budget and track expenses, helping them stay on top of their finances and make more informed financial decisions.

DIS-ADVANTAGES

Cost: Some personal finance apps charge a fee for premium features or access to advanced tools, which may not be cost-effective for all users.

Data Privacy: Users need to be careful about sharing personal and financial information with third-party apps, as there is a risk of data breaches and identity theft.

User Error: While the app is designed to be user-friendly, there is still a possibility of user error when manually inputting financial data, which could result in inaccurate reports or miscalculations.

5.APPLICATION

Money Matters: A Personal Finance Management App can be applied in various ways to manage personal finances. Here are some of the most common applications:

Budgeting: The app can be used to create and track a budget, helping users manage their expenses, save money, and achieve financial goals.

Expense Tracking: Users can track their daily expenses using the app, which can help them identify areas where they can reduce their spending and make more informed purchasing decisions.

Debt Management: Money Matters can help users manage their debt by tracking payments, calculating interest rates, and suggesting strategies for paying off debt more quickly.

6.CONCLUSION

In conclusion, Money Matters: A Personal Finance Management App is a valuable tool for individuals looking to take control of their personal finances. The app offers a range of features and benefits, including budgeting, expense tracking, debt management, investment monitoring, retirement planning, tax planning, and financial education.

While there are potential disadvantages to using a personal finance app, such as cost and data privacy concerns, the advantages of using Money Matters outweigh the potential risks. The app provides users with a convenient and user-friendly platform for managing their finances, helping them make more informed financial decisions, and achieving their financial goals.

Overall, Money Matters is a highly useful app for anyone looking to improve their financial literacy, manage their finances more effectively, and build a more secure financial future.

7.SCOPE AND FUTURE:

The scope of Money Matters: A Personal Finance Management App is vast and continues to expand as more people turn to technology to manage their finances. As the app evolves and new features are added, its scope is likely to grow even further. Some potential areas of growth for the app include:

Integration with other financial services: The app may be integrated with other financial services, such as banking and investment platforms, to offer users a more comprehensive financial management experience.

Machine Learning and Artificial Intelligence: The app may use machine learning and artificial intelligence to provide users with personalized financial advice and insights.

Cryptocurrency Management: The app may incorporate features for managing cryptocurrencies and other digital assets, as these types of assets become more mainstream.

8.APPENDIX

A. SOURCE CODE

MainActivity.kt

```
            package com.example.expensestracker

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.expensestracker.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(0xFFadbef4),
                        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,AddExpensesActivi
ty::class.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))

    }
}
```

LoginActivity.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)
}
```