Project Report

OWL_M: Material 3 tutorials Application

Team ID: Team - 590951

Team Member Names	College Registration Number and Campus
Vedant Giri	21BCG10057 (VIT - Bhopal)
Saksham Saxena	21BCE11103 (VIT - Bhopal)
Sushant Gupta	21BCG10059 (VIT - Bhopal)

1. INTRODUCTION

1.1 Project Overview

OWL_M is an android application for Google's material design, is an Android-oriented design language, supporting onscreen touch experiences via cue-rich features and natural motions that mimic real-world objects, updated version Material 3. Material 3 includes updated theming, components and Material You personalization features like dynamic color, and is designed to be cohesive with the new visual style and system UI elements.

1.2 Purpose

The Purpose of the OWL_M android application is to provide a basic tutorial to the newcomers in the world of android development to the Material 3 designing, by the integration of which, they can come to understand the possibilities available in the designing process of any android application.

2. LITERATURE SURVEY

2.1 Existing problem

There are so many new android developers who are unaware of the existence of Google's Material Design, by the use of which they can make their android look a cut above the rest. So, if there were a reference guide on this problem, then, they probably can develop much better looking android applications.

2.2 References

–

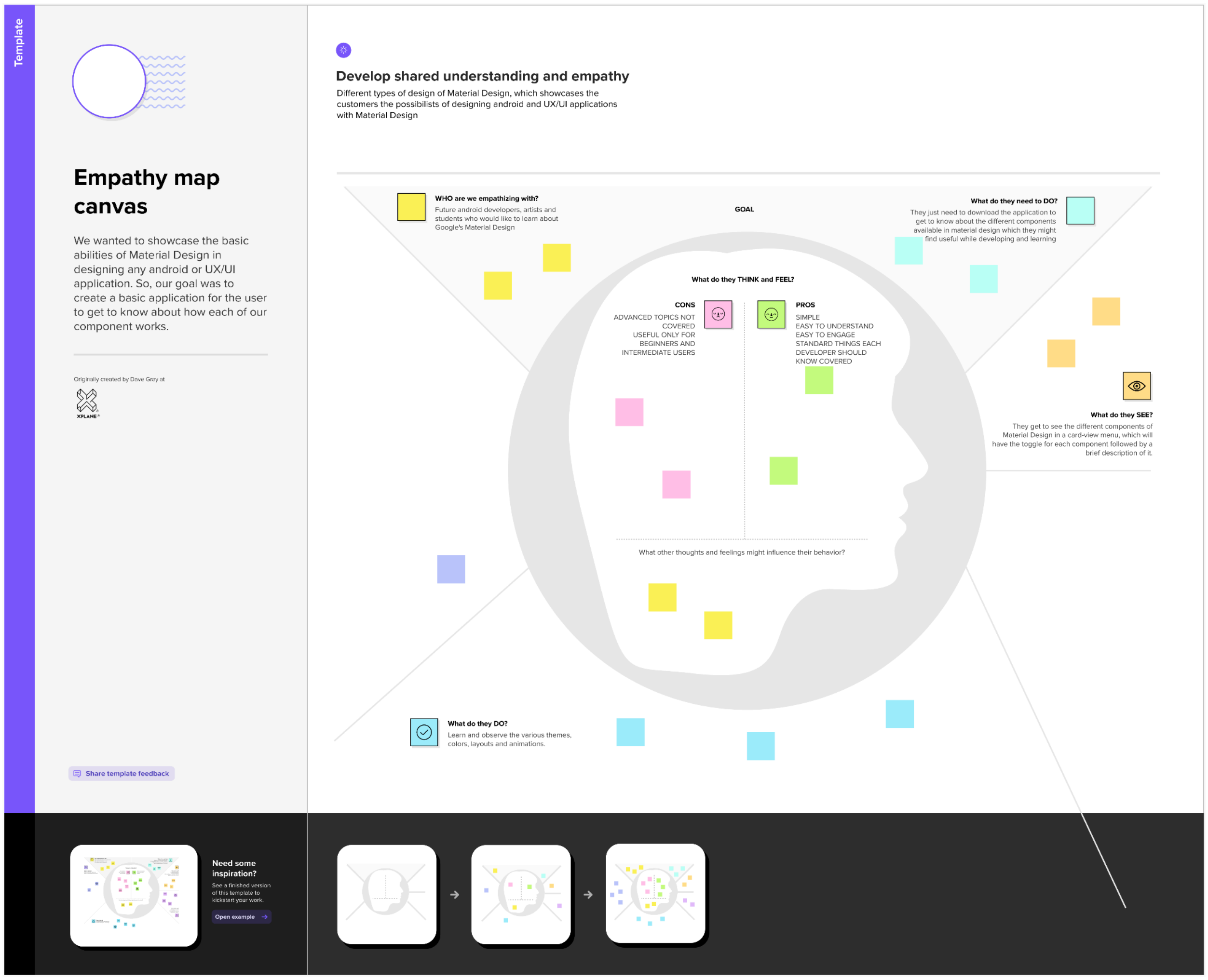
(No References, since this query is not too searched on the internet, but confidential sources at a few institutions told us this)

2.3 Problem Statement Definition

We want the new android developers to have an idea about Google's Material Design in its updated form: Material 3. Since there are not any android application options for that purpose, we decided to make it our problem statement.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



Empathy map canvas

We wanted to showcase the basic abilities of Material Design in designing any android or UX/UI application. So, our goal was to create a basic application for the user to get to know about how each of our component works.

Originally created by Dave Gray at XPLANE

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

Develop shared understanding and empathy

Different types of design of Material Design, which showcases the customers the possibilists of designing android and UX/UI applications with Material Design

WHO are we empathizing with?
Future android developers, artists and students who would like to learn about Google's Material Design

What do they need to DO?
They just need to download the application to get to know about the different components available in material design which they might find useful while developing and learning

What do they THINK and FEEL?

CONS
ADVANCED TOPICS NOT COVERED
USEFUL ONLY FOR BEGINNERS AND INTERMEDIATE USERS

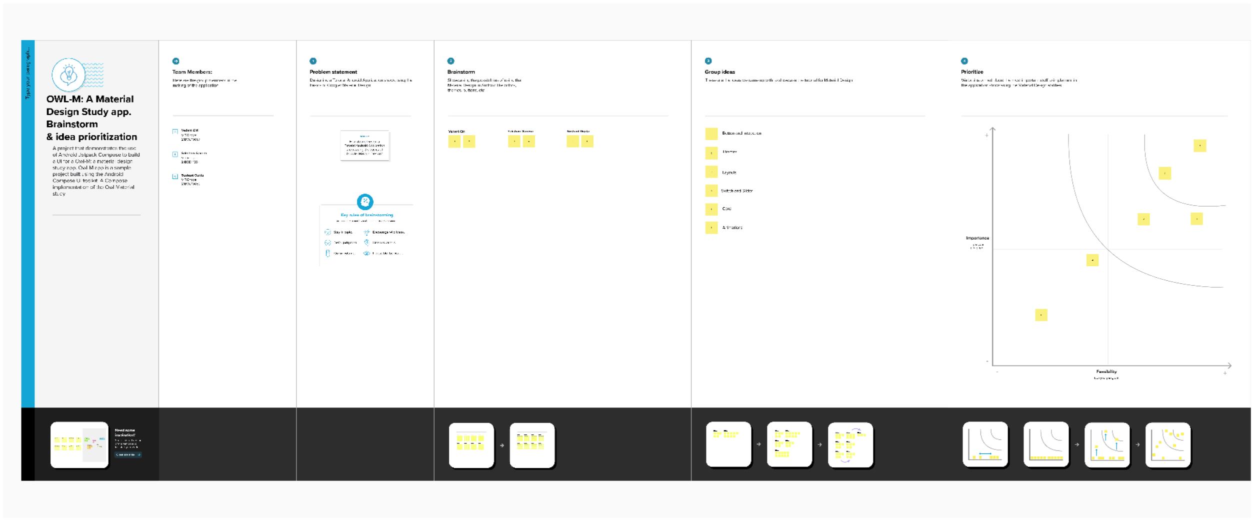
PROS
SIMPLE
EASY TO UNDERSTAND
EASY TO ENGAGE
STANDARD THINGS EACH DEVELOPER SHOULD KNOW COVERED

What other thoughts and feelings might influence their behavior?

What do they SEE?
They get to see the different components of Material Design in a card-view menu, which will have the toggle for each component followed by a brief description of it.

What do they DO?
Learn and observe the various themes, colors, layouts and animations.

3.2 Ideation & Brainstorming


OWL-M: A Material Design Study app. Brainstorm & idea prioritization
Team Members:
Problem statement
Key rules of brainstorming
Brainstorm
Group ideas
Prioritize
Importance
Feasibility

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

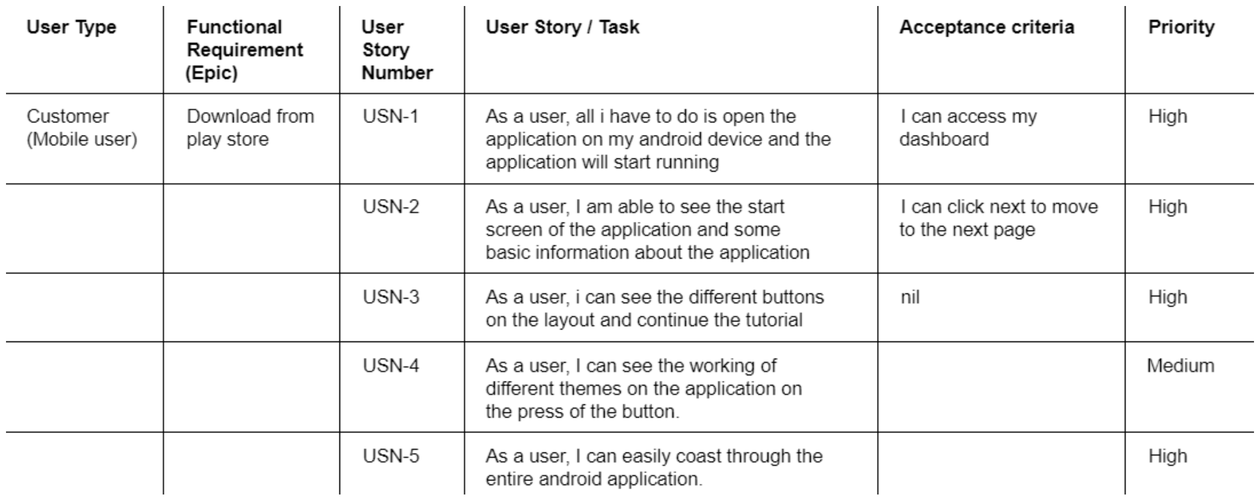
- The user should be able to interact with the different components of the application without any trouble
- The tutorial should be easy to understand, and the learning curve should be minimal.
- The application should run smoothly and not have any runtime issues.

4.2 Non-Functional Requirements

- NA

5. PROJECT DESIGN

5.1 User Stories



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority
Customer (Mobile user)	Download from play store	USN-1	As a user, all i have to do is open the application on my android device and the application will start running	I can access my dashboard	High
		USN-2	As a user, I am able to see the start screen of the application and some basic information about the application	I can click next to move to the next page	High
		USN-3	As a user, i can see the different buttons on the layout and continue the tutorial	nil	High
		USN-4	As a user, I can see the working of different themes on the application on the press of the button.		Medium
		USN-5	As a user, I can easily coast through the entire android application.		High

5.2 Solution Architecture

Initial Page

The initial page is a simple page that contains the following elements:

An app image

App description

A button to navigate to the next page

Second Page

The second page is a page that contains three buttons:

A rounded rectangular color changing button

A shape changing button

A button that raises and depresses on each simultaneous press

Third Page

The third page is a page that contains a button that changes the theme of the app to either light or dark. It also contains a button that navigates to the cards page.

Fourth Page

The Fourth page is a page that contains a horizontal list of cards. Each card has the following elements:

A title

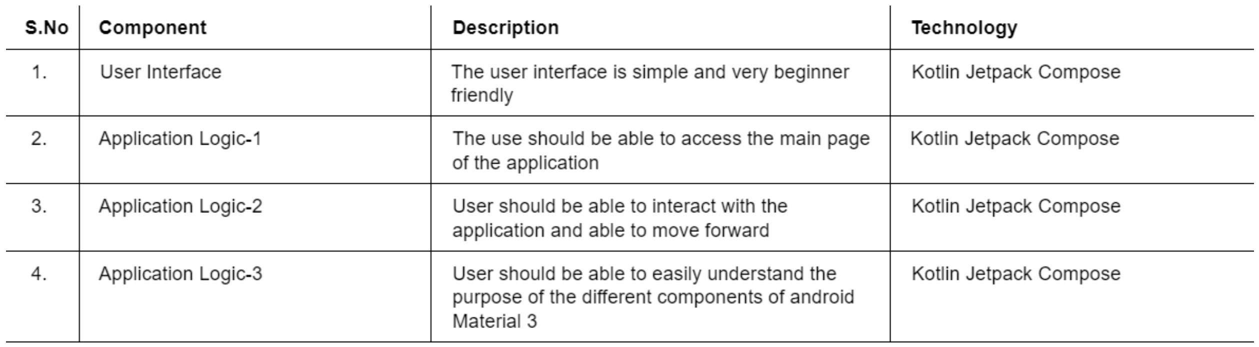
An image

A description

A button to navigate to the details page for the card

6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture



S.No	Component	Description	Technology
1.	User Interface	The user interface is simple and very beginner friendly	Kotlin Jetpack Compose
2.	Application Logic-1	The use should be able to access the main page of the application	Kotlin Jetpack Compose
3.	Application Logic-2	User should be able to interact with the application and able to move forward	Kotlin Jetpack Compose
4.	Application Logic-3	User should be able to easily understand the purpose of the different components of android Material 3	Kotlin Jetpack Compose

6.2 Sprint Planning , Schedule and Deadlines

S.No.	Process	Deadlin e	Description	Team Member	Remarks	

1	Introduction page	22rd October	The initial page the user will see upon opening of the application on the device	Sushant Gupta	Done within the deadline	
2	Material 3 buttons	26th October	1st implementation of material 3 tutorial android application has to be buttons	Sushant Gupta	Done a date late	
3	Material 3 themes	28th October	2nd implementation of that has to be the visual theme of the application	Vedant Giri	Done within the deadline	
4	Navigation	29th October	Without any navigation, user will not be able to progress with the tutorial	Vedant Giri	Done within the deadline	
5	Material 3 cards view	3rd November	The cards view is important application of material 3 since cards are now-a-days used on a large scale in android applications	Saksham Saxena	Done within the deadline	
6	Testing and integration	4th November	Testing of our android application and integration of it on emulators like Blue-stacks	All	Done within the deadline	
7	Play Store upload	7th November	Uploading of android application onto the Playstore	-	Was not done due to technical error	

7. CODING & SOLUTIONING

7.1 Feature 1

The first feature of the application was a buttons UI, where the user could see the different components of Material 3 at work in coordination with buttons.

Code:

```
 @Composable
     fun ColorChangingButton(modifier: Modifier, text: String) {
     }

     @Composable
     fun ShapeChangingButton(modifier: Modifier, text: String) {
         TODO("Not yet implemented")
     }

     @Composable
     fun RaisingAndDepressingButton(modifier: Modifier, text: String) {

     }

     @Composable
     fun ThemeChangingButton(modifier: Modifier, text: String) {
         TODO("Not yet implemented")
     }
     @Composable
                 fun SecondPage(onClickNext: () -> Unit) {
                     Column(
                         modifier = Modifier.fillMaxSize(),
                         verticalArrangement = Arrangement.Center,
                         horizontalAlignment = Alignment.CenterHorizontally
                     ) {
                         // Rounded rectangular color changing button
                         ColorChangingButton(
                             modifier = Modifier.padding(bottom = 16.dp),
                             text = "Color changing button"
                         )

                         // Text about color changing button
                         Text(
                             "This button changes color when you press it.",
                             modifier = Modifier.padding(bottom = 16.dp),
         style = MaterialTheme.typography.bodyMedium
     )

     // Shape changing button
     ShapeChangingButton(
         modifier = Modifier.padding(bottom = 16.dp),
         text = "Shape changing button"
     )

     // Text about shape changing button
     Text(
         "This button changes shape when you press it.",
         modifier = Modifier.padding(bottom = 16.dp),
         style = MaterialTheme.typography.bodyMedium
     )

     // Button that raises and depresses on each simultaneous press
     RaisingAndDepressingButton(
         modifier = Modifier.padding(bottom = 16.dp),
         text = "Raising and depressing button"
     )

     // Text about raising and depressing button
     Text(
         "This button raises and depresses when you press it.",
         modifier = Modifier.padding(bottom = 16.dp),
         style = MaterialTheme.typography.bodyMedium
     )

     // Next button
     Column(
         modifier = Modifier
             .fillMaxSize()
             .padding(horizontal = 16.dp)
```

```
                ) {
                    // Your other UI elements here

                    Spacer(modifier = Modifier.weight(1f))

                    Button(
                        onClick = onClickNext,
                        modifier = Modifier
                            .align(Alignment.End)
                            .padding(bottom = 16.dp)
                    ) {
                        Text("Next")
                    }
                }
            }
        }

    @Composable
    fun ColorChangingButton(
        modifier: Modifier = Modifier,
        text: String,
        colors: ButtonColors = ButtonDefaults.buttonColors(backgroundColor =
MaterialTheme.colorScheme.primary)
    ) {
        var color by remember { mutableStateOf(colors.backgroundColor) }

        Button(
            onClick = { color = colors.disabledBackgroundColor },
            modifier = modifier,
            colors = colors.copy(backgroundColor = color),
            shape = RoundedCornerShape(16.dp)
        ) {
            Text(text)
        }
```

```
}

@Composable
fun ShapeChangingButton(
    modifier: Modifier = Modifier,
    text: String,
    colors: ButtonColors = ButtonDefaults.buttonColors(backgroundColor =
MaterialTheme.colorScheme.primary)
) {
    var shape by remember { mutableStateOf(RoundedCornerShape(16.dp)) }

    Button(
        onClick = { shape = CircleShape },
        modifier = modifier,
        colors = colors,
        shape = shape
    ) {
        Text(text)
    }
}

@Composable
fun RaisingAndDepressingButton(
    modifier: Modifier = Modifier,
    text: String,
    colors: ButtonColors = ButtonDefaults.buttonColors(backgroundColor =
MaterialTheme.colorScheme.primary)
) {
    var isRaised by remember { mutableStateOf(false) }

    Button(
        onClick = { isRaised = !isRaised },
        modifier = modifier,
        colors = colors,
        elevation = if (isRaised) 8.dp else 0.dp
    ) {
        Text(text)
    }
}
@Composable
fun MySecondPage() {
    SecondPage(onClickNext = { /* Navigate to the next page */ })
}
```

7.2 Feature 2

Another Key feature of the application is the use of cards to demonstrate how well can we integrate cards in an android application with Material 3 components.

Code:

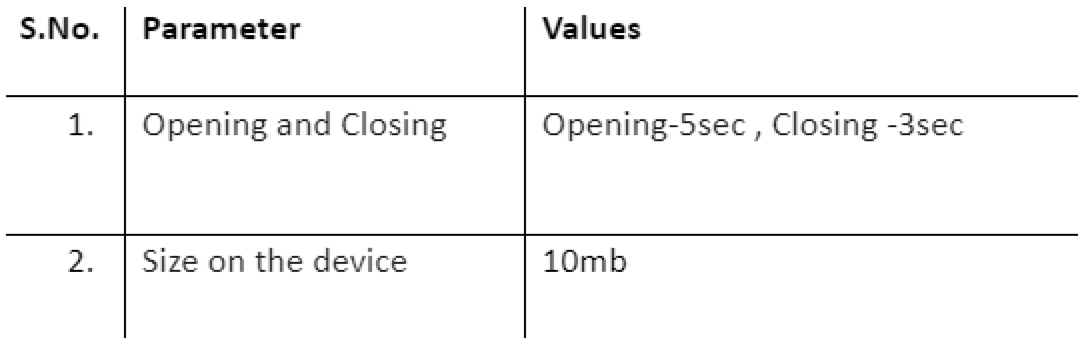
```
@Composable
    fun MyCustomCard(
        modifier: Modifier = Modifier,
        @DrawableRes image:Int,
        title:String,
        text:String,

        ) {

        var showFullText by remember {
            mutableStateOf(false)
        }

        Card(
            modifier = Modifier.animateContentSize(),
            shape = MaterialTheme.shapes.medium,
            colors = CardDefaults.cardColors(
                containerColor = Color.LightGray
            )
        ) {
            Column {
                Image(
                    modifier = Modifier
                        .fillMaxWidth()
                        .height(200.dp),
                    painter = painterResource(id = image),
                    contentDescription = null,
                    contentScale = ContentScale.Crop
                )
                Column (
                    modifier = Modifier.padding(vertical = 5.dp, horizontal = 10.dp)
                ) {
                    Text(
                        text = title,
                        color = Color.Black,
                        fontSize = 25.sp,
                        fontWeight = FontWeight.ExtraBold
                    )

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

                    Text(
                        modifier = Modifier.clickable {
                            showFullText = !showFullText
                        },
                        text = text,
                        color = Color.Black.copy(alpha = 0.5f),
                        fontSize = 30.sp,
                        fontWeight = FontWeight.Bold,
                        maxLines = if (showFullText) 100 else 2
                    )
                }
            }
        }
```

7.3 Database Schema (if Applicable)

NA

8. PERFORMANCE TESTING

8.1 Performance Metrics



S.No.	Parameter	Values
1.	Opening and Closing	Opening-5sec , Closing -3sec
2.	Size on the device	10mb

9.. ADVANTAGES & DISADVANTAGES

9.1 Advantages

- The application takes up very less space on the local device's storage
- The UI is very simple and easy to navigate
- The Tutorial is very interactive and quite easy to understand
- The application is highly scalable and can be expanded in future to include other topics as well.

10.2 Disadvantages

- The topics covered are basic, and an experienced programmer might not find much use of the application.
- The application may not be supported by APIs below level 24.

11. CONCLUSION

In Conclusion, OWL_M is not just another android application; it's a gateway for budding developers to explore the vast realm of Material 3 design. With a user-friendly interface and interactive tutorials, it aims to bridge the gap for those unfamiliar with Google's Material Design. The simplicity, scalability, and efficiency of OWL_M make it a promising tool for anyone looking to enhance their Android development skills.

12. FUTURE SCOPE

Since the application is quite simple, and no revenue generating yet, it has the potential to become a recommended tutorial application for the budding android developers who are keen on learning the vast possibilities of designing aspects of any android application by the use of the vast libraries of Google's Material 3 designing language.

13. APPENDIX

GitHub Link

https://github.com/Jiren69/Android-OWL_M

Demo Link

Owl_M Demo link