Snack Squad: A Customizable Snack Ordering And Delivery App

Introduction

1.1. Overview

Project Description

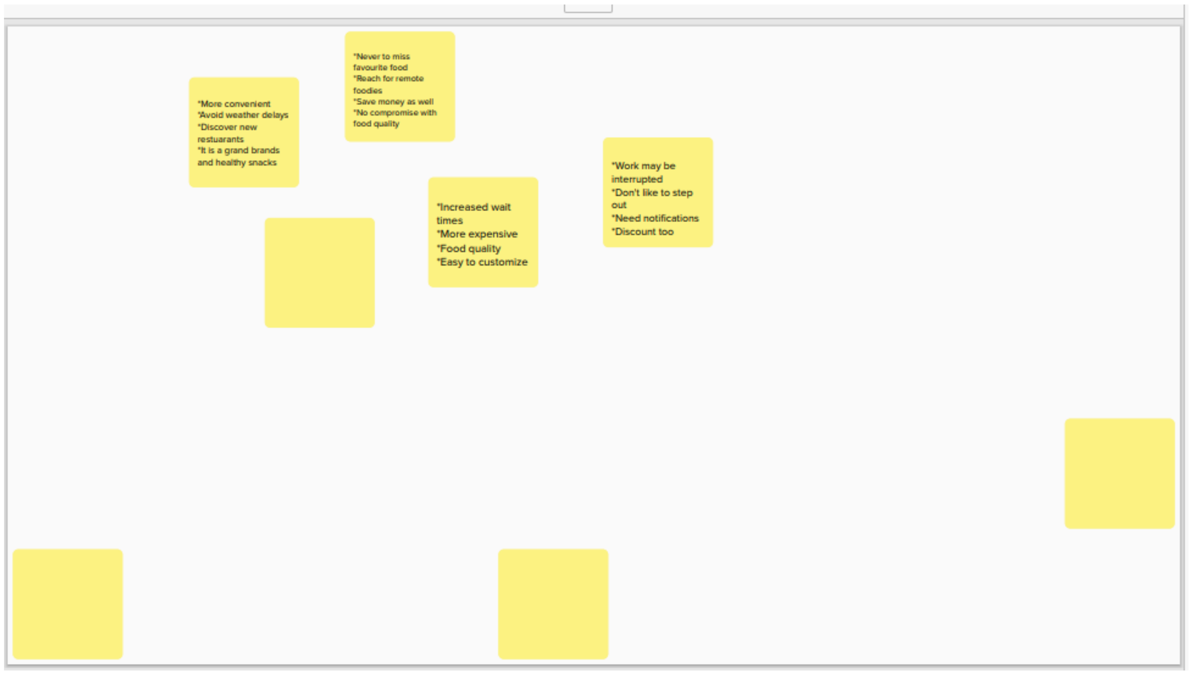
The snack squad project makes use of Android Jetpack Compose to build a UI for creating a simple e-commerce app for snacks using the Compose libraries.The user can see a list of snacks, and by tapping on a snack,and by tapping on the "Add to Cart" button, the snack will be added to the cart. The user can also see the list of items in the cart and can proceed to checkout to make the purchase.

Purpose

The snack squad project is used to order the snacks from the given menu. The app provides provision to order more items or to cancel the order.

Problem Definition and thinking

2.1. Empathy Map



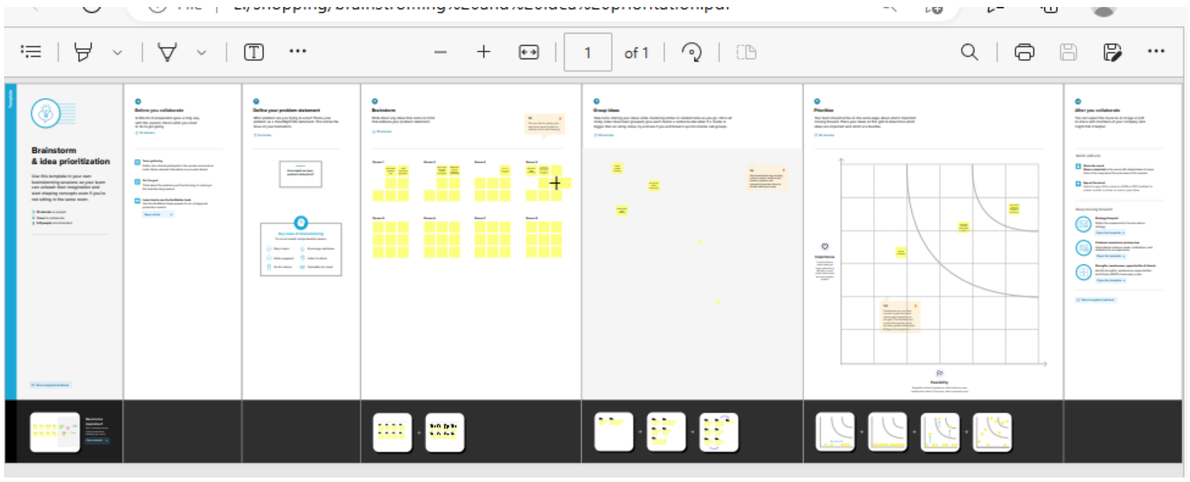
*More convenient
*Avoid weather delays
*Discover new restuarants
*It is a grand brands and healthy snacks

*Never to miss favourite food
*Reach for remote foodies
*Save money as well
*No compromise with food quality

*Increased wait times
*More expensive
*Food quality
*Easy to customize

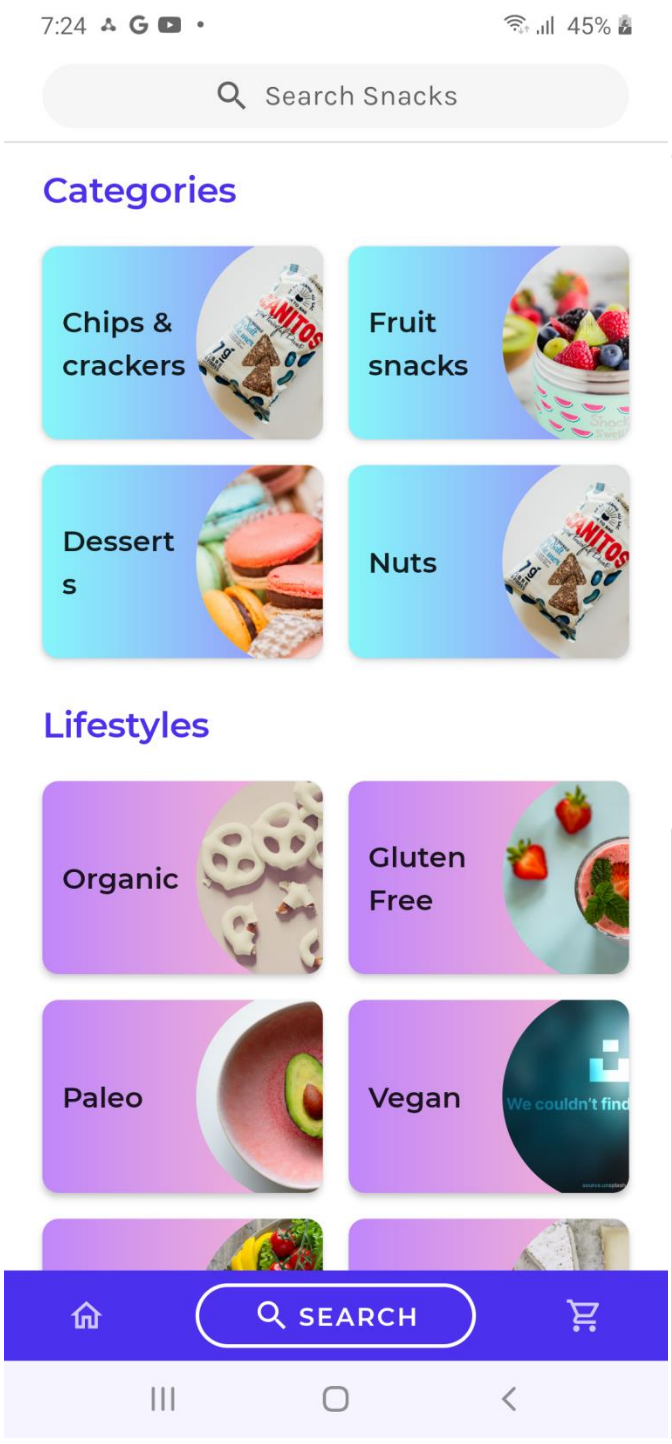
*Work may be interrupted
*Don't like to step out
*Need notifications
*Discount too

2.2. Ideation and Brainstorming Map

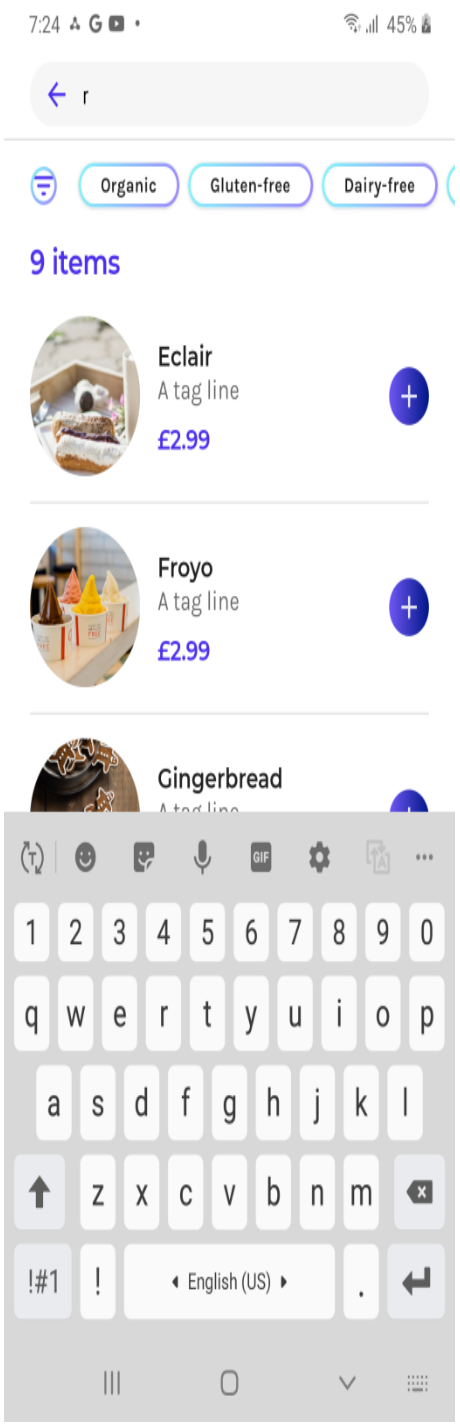


Result

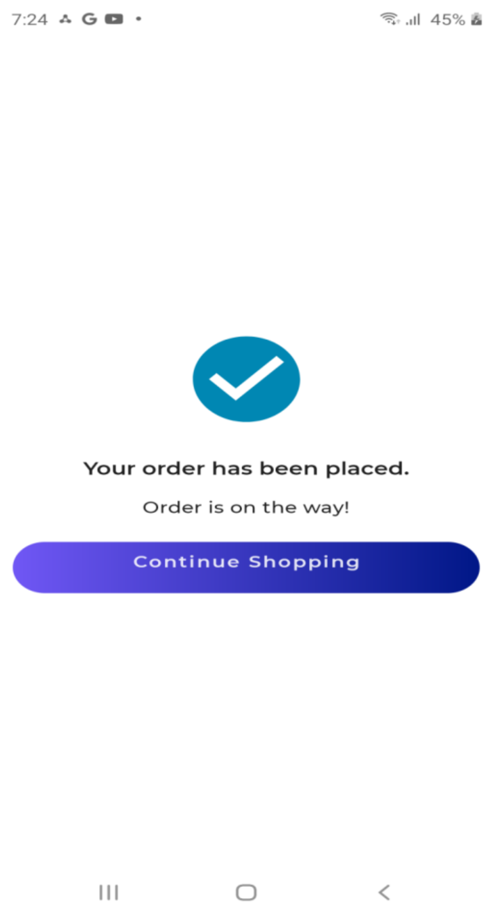
Home

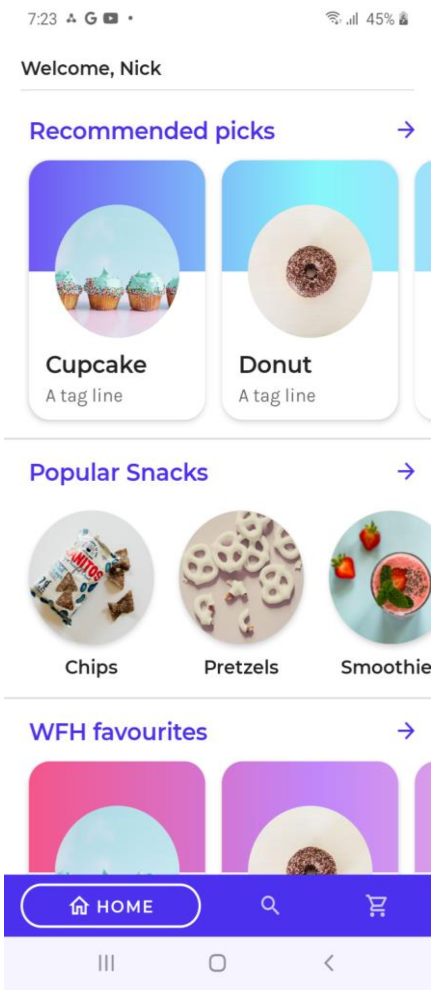


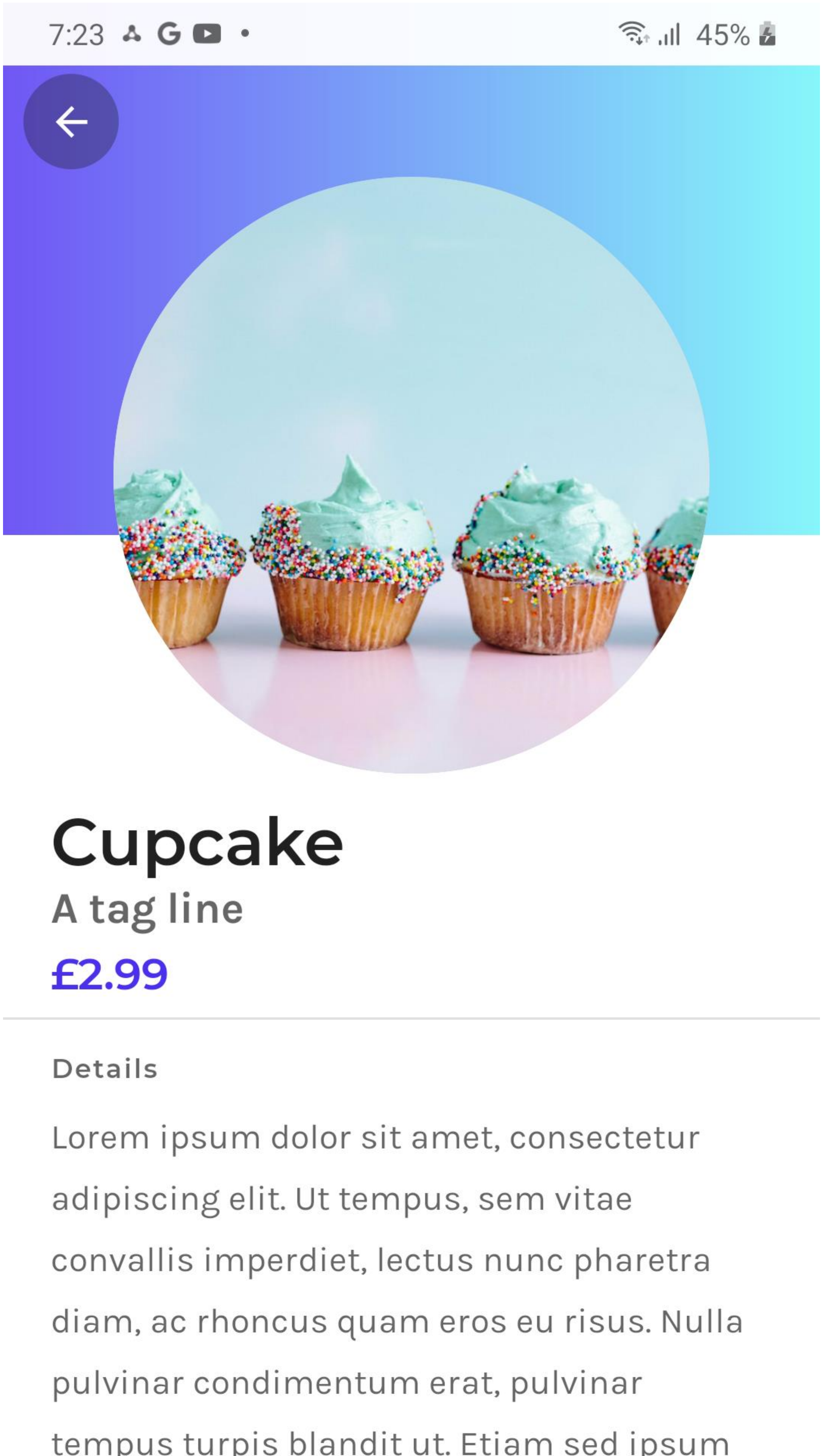
Details



Order Confirmed







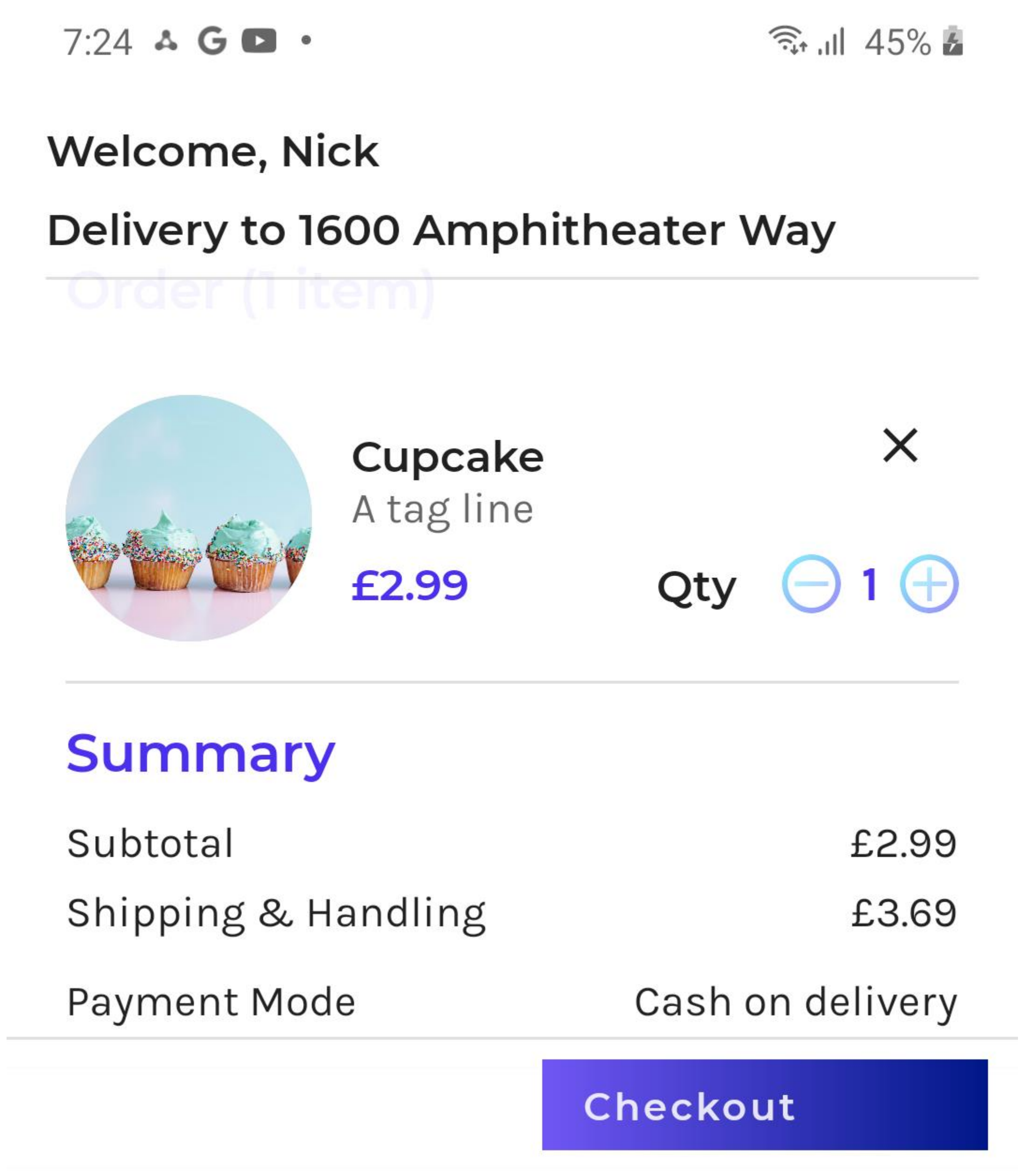
Cupcake

A tag line

£2.99

Details

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut tempus, sem vitae convallis imperdiet, lectus nunc pharetra diam, ac rhoncus quam eros eu risus. Nulla pulvinar condimentum erat, pulvinar tempus turpis blandit ut. Etiam sed ipsum



7:24 45%

Welcome, Nick

Delivery to 1600 Amphitheater Way

Order (1 item)

Cupcake

A tag line

£2.99

Qty 1

Summary

Subtotal	£2.99
Shipping & Handling	£3.69
Payment Mode	Cash on delivery

Checkout

Advantages

The app is helpful for the profession – driven persons to have their favourite munchies at the time they need. The users can quickly order the snacks even from remote foodies.

Disadvantages

Besides the usually talked junk food health issues, there may be chance for profit conflict between the restuarants and deliver app. Sometimes the user has to compromise with food quality.

Conclusion

The number of online ordering from various surveys shows the active contribution of the working youth who are highly dependent on these online restaurant apps to feed their brains through tummies. This surely comes as a blessing to the **Modern-Gen** who believe in doing what they are good at and in experimenting to develop and invent something new every day.

Future Scope

The app could be modified to collect the feedback from the user and to deliver more food items besides the snacks items.

Appendix

A. Source Code
   AndroidManifest.xml

```
<?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="@drawable/fast_food"
                 android:label="@string/app_name"
                 android:supportsRtl="true"
```

```
android:theme="@style/Theme.SnackOrdering"
tools:targetApi="31">
<activity
android:name=".AdminActivity"
android:exported="false"
android:label="@string/title_activity_admin"
android:theme="@style/Theme.SnackOrdering" />
<activity
android:name=".LoginActivity"
android:exported="true"
android:label="SnackSquad"
android:theme="@style/Theme.SnackOrdering">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<activity
android:name=".TargetActivity"
android:exported="false"
android:label="@string/title_activity_target"
android:theme="@style/Theme.SnackOrdering" />
<activity
android:name=".MainPage"
android:exported="false"
android:label="@string/title_activity_main_page"
android:theme="@style/Theme.SnackOrdering" />
<activity
android:name=".MainActivity"
android:exported="false"
android:label="MainActivity"
android:theme="@style/Theme.SnackOrdering" />
</application>
</manifest>
```

```
package
com.example.snackord
ering
              import android.annotation.SuppressLint
              import android.content.Context
              import android.os.Bundle
              import android.widget.Toast
              import androidx.activity.ComponentActivity
              import androidx.activity.compose.setContent
              import androidx.annotation.DrawableRes
              import androidx.annotation.StringRes
              import androidx.compose.foundation.Image
              import androidx.compose.foundation.background
              import androidx.compose.foundation.layout.*
              import androidx.compose.foundation.shape.CircleShape
              import androidx.compose.foundation.shape.RoundedCornerShape
              import androidx.compose.material.*
              import androidx.compose.material.icons.Icons
              import androidx.compose.material.icons.filled.*
              import androidx.compose.runtime.Composable
              import androidx.compose.ui.Alignment
              import androidx.compose.ui.Modifier
              import androidx.compose.ui.draw.clip
              import androidx.compose.ui.graphics.Color
              import androidx.compose.foundation.lazy.LazyColumn
              import androidx.compose.foundation.lazy.items
              import androidx.compose.material.Text
              import androidx.compose.ui.unit.dp
              import androidx.compose.ui.graphics.RectangleShape
              import androidx.compose.ui.layout.ContentScale
              import androidx.compose.ui.platform.LocalContext
              import androidx.compose.ui.res.painterResource
              import androidx.compose.ui.res.stringResource
              import androidx.compose.ui.text.font.FontWeight
              import androidx.compose.ui.unit.sp
              import androidx.core.content.ContextCompat.startActivity
              import com.example.snackordering.ui.theme.SnackOrderingTheme
              import android.content.Intent as Intent1
              class MainPage : ComponentActivity() {
              override fun onCreate(savedInstanceState: Bundle?) {
              super.onCreate(savedInstanceState)
              setContent {
              SnackOrderingTheme {
                              // A surface container using the 'background' color
              from the theme
                              Surface(
                                  modifier = Modifier.fillMaxSize(),
                                  color = MaterialTheme.colors.background
                              ) {
              FinalView(this)
              val context = LocalContext.current
                                  //PopularFoodColumn(context)
                              }
                          }
                      }
                  }
              }
              @Composable
              fun TopPart() {
                  Row(
                      modifier = Modifier
                          .fillMaxWidth()
                          .background(Color(0xffeceef0)), Arrangement.SpaceBetween
                  ) {
                      Icon(
              imageVector = Icons.Default.Add, contentDescription = "Menu Icon",
                          Modifier
                              .clip(CircleShape)
                              .size(40.dp),
                          tint = Color.Black,
                      )
                      Column(horizontalAlignment = Alignment.CenterHorizontally) {
                          Text(text = "Location", style =
              MaterialTheme.typography.subtitle1, color = Color.Black)
                          Row {
                              Icon(
              imageVector = Icons.Default.LocationOn,
              contentDescription = "Location",
                                  tint = Color.Red,
                              )
                              Text(text = "Accra" , color = Color.Black)
                          }
                      }
                      Icon(
              imageVector = Icons.Default.Notifications, contentDescription =
              "Notification Icon",
                          Modifier
                              .size(45.dp),
                          tint = Color.Black,
                      )
                  }
              }
              @Composable
```

```
fun CardPart() {
    Card(modifier = Modifier.size(width = 310.dp, height = 150.dp),
RoundedCornerShape(20.dp)) {
        Row(modifier = Modifier.padding(10.dp),
Arrangement.SpaceBetween) {
            Column(verticalArrangement = Arrangement.spacedBy(12.dp))
{
                Text(text = "Get Special Discounts")
                Text(text = "up to 85%", style =
MaterialTheme.typography.h5)
                Button(onClick = {}, colors =
ButtonDefaults.buttonColors(Color.White)) {
                    Text(text = "Claim voucher", color =
MaterialTheme.colors.surface)
                }
            }
            Image(
                painter = painterResource(id =
R.drawable.food_tip_im),
contentDescription = "Food Image", Modifier.size(width = 100.dp,
height = 200.dp)
            )
        }
    }
}
@Composable
fun PopularFood(
    @DrawableResdrawable: Int,
    @StringRes text1: Int,
    context: Context
) {
    Card(
        modifier = Modifier
            .padding(top=20.dp, bottom = 20.dp, start = 65.dp)
            .width(250.dp)
    ) {
        Column(
verticalArrangement = Arrangement.Top,
horizontalAlignment = Alignment.CenterHorizontally
        ) {
            Spacer(modifier = Modifier.padding(vertical = 5.dp))
            Row(
                modifier = Modifier
```

```
                        .fillMaxWidth(0.7f), Arrangement.End
                ) {
                    Icon(
imageVector = Icons.Default.Star,
contentDescription = "Star Icon",
                        tint = Color.Yellow
                    )
                    Text(text = "4.3", fontWeight = FontWeight.Black)
                }
                Image(
                    painter = painterResource(id = drawable),
contentDescription = "Food Image",
contentScale = ContentScale.Crop,
                    modifier = Modifier
                        .size(100.dp)
                        .clip(CircleShape)
                )
                Text(text = stringResource(id = text1), fontWeight =
FontWeight.Bold)
                Row(modifier = Modifier.fillMaxWidth(0.7f),
Arrangement.SpaceBetween) {
                    /*TODO Implement Prices for each card*/
                    Text(
                        text = "$50",
                        style = MaterialTheme.typography.h6,
fontWeight = FontWeight.Bold,
fontSize = 18.sp
                    )
IconButton(onClick = {
                        //var no=FoodList.lastIndex;
                        //Toast.
val intent = Intent1(context, TargetActivity::class.java)
context.startActivity(intent)
                    }) {
                        Icon(
imageVector = Icons.Default.ShoppingCart,
contentDescription = "shopping cart",
                        )
                    }
                }
            }
        }
}
```

```
private valFoodList = listOf(
R.drawable.sandwish to R.string.sandwich,
R.drawable.sandwish to R.string.burgers,
R.drawable.pack to R.string.pack,
R.drawable.pasta to R.string.pasta,
R.drawable.tequila to R.string.tequila,
R.drawable.wine to R.string.wine,
R.drawable.salad to R.string.salad,
R.drawable.pop to R.string.popcorn
).map { DrawableStringPair(it.first, it.second) }
private data class DrawableStringPair(
    @DrawableResvaldrawable: Int,
    @StringResval text1: Int
)
@Composable
fun App(context: Context) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .background(Color(0xffeceef0))
            .padding(10.dp),
verticalArrangement = Arrangement.Top,
horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Surface(modifier = Modifier, elevation = 5.dp) {
TopPart()
        }
        Spacer(modifier = Modifier.padding(10.dp))
CardPart()
        Spacer(modifier = Modifier.padding(10.dp))
        Row(modifier = Modifier.fillMaxWidth(),
Arrangement.SpaceBetween) {
            Text(text = "Popular Food", style =
MaterialTheme.typography.h5, color = Color.Black)
            Text(text = "view all", style =
MaterialTheme.typography.subtitle1,  color = Color.Black)
        }
        Spacer(modifier = Modifier.padding(10.dp))
PopularFoodColumn(context) // <- call the function with parentheses
    }
}
@Composable
fun PopularFoodColumn(context: Context) {
```

```
LazyColumn(
        modifier = Modifier.fillMaxSize(),
        content = {
            items(FoodList) { item ->
PopularFood(context = context,drawable = item.drawable, text1 =
item.text1)
                abstract class Context
            }
        },
verticalArrangement = Arrangement.spacedBy(16.dp))
}
@SuppressLint("UnusedMaterialScaffoldPaddingParameter")
@Composable
fun FinalView(mainPage: MainPage) {
SnackOrderingTheme {
        Scaffold() {
val context = LocalContext.current
            App(context)
        }
    }
}
  package
  com.example.snackord
  ering
                         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.unit.dp
import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat
import
com.example.snackordering.ui.theme.SnackOrd
eringTheme
class LoginActivity : ComponentActivity() {
    private lateinitvardatabaseHelper:
UserDatabaseHelper
override fun onCreate(savedInstanceState:
Bundle?) {
super.onCreate(savedInstanceState)
databaseHelper = UserDatabaseHelper(this)
setContent {
SnackOrderingTheme {
                // 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.order), 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)
        )
        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,
MainPage::class.java
                            )
                        )
                        //onLoginSuccess()
                    }
                        if (user != null
&&user.password == "admin") {
                            error =
"Successfully log in"
context.startActivity(
                                Intent(

context,
AdminActivity::class.java
                                )
                            )
                        }
                        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,
MainActivity::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,
MainPage::class.java)
ContextCompat.startActivity(context,
intent, null)
```