**SNACK ORDERING**

**1. INTRODUCTION**

1.1 Overview

1.2 Purpose

**2. PROBLEM DEFINITION & DESIGN THINKING**

2.1 Empathy Map

2.2 Ideation & brainstroming map screenshots.

**3. RESULT**

**4. ADVANTAGES & DISADVANTAGES**

**5. APPLICATION**

**6. CONCLUSION**

**7. FUTURE SCOPE**

**8. APPENDIX**

1. **INTRODUCTION**

The purpose of this thesis was to build a snack server

application. For the customer, this application provides a view of current food

information (category, name, image，price, description etc.) on the website and

Android application. The customer can order food from these two platforms. For

the administrator in restaurant, this application offers a series of operations to

add, update, delete and query the information of food, food order and employees.

With the rapid development of information

technology, web application and Android application have been increasing in

recent years. Compared with the desktop application, the advantages of web

application for users are:

* No need to install and update.
* Easily visit through browsers the advantage of the Android application.
* Easily visit through browsers the advent.

age of the Android application.

* Mobile application is convenient to carry.
* Global partnerships and large install base.
* Powerful development framework.
* Open marketplace for distributing apps Based on the advantages of both applications, I motivated myself to develop a combination project between web and Android application.

A project that demonstrates the use of Android Jetpack Compose to build a UI for a snack squad app. Snack Squad is a sample project built using the Android Compose UI toolkit. It demonstrates how to create 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.

**1.1 TECHNOLOGY OVERVIEW**

The administrator part was developed with struts2

for customers is implemented with servlet/JSP. MySql was chosen as a database

for the project. The Android part was developed with the Android framework.

Following the core frameworks will be generally introduced in this application.

**1.2 PURPOSE**

An snack ordering platform is a place where customers

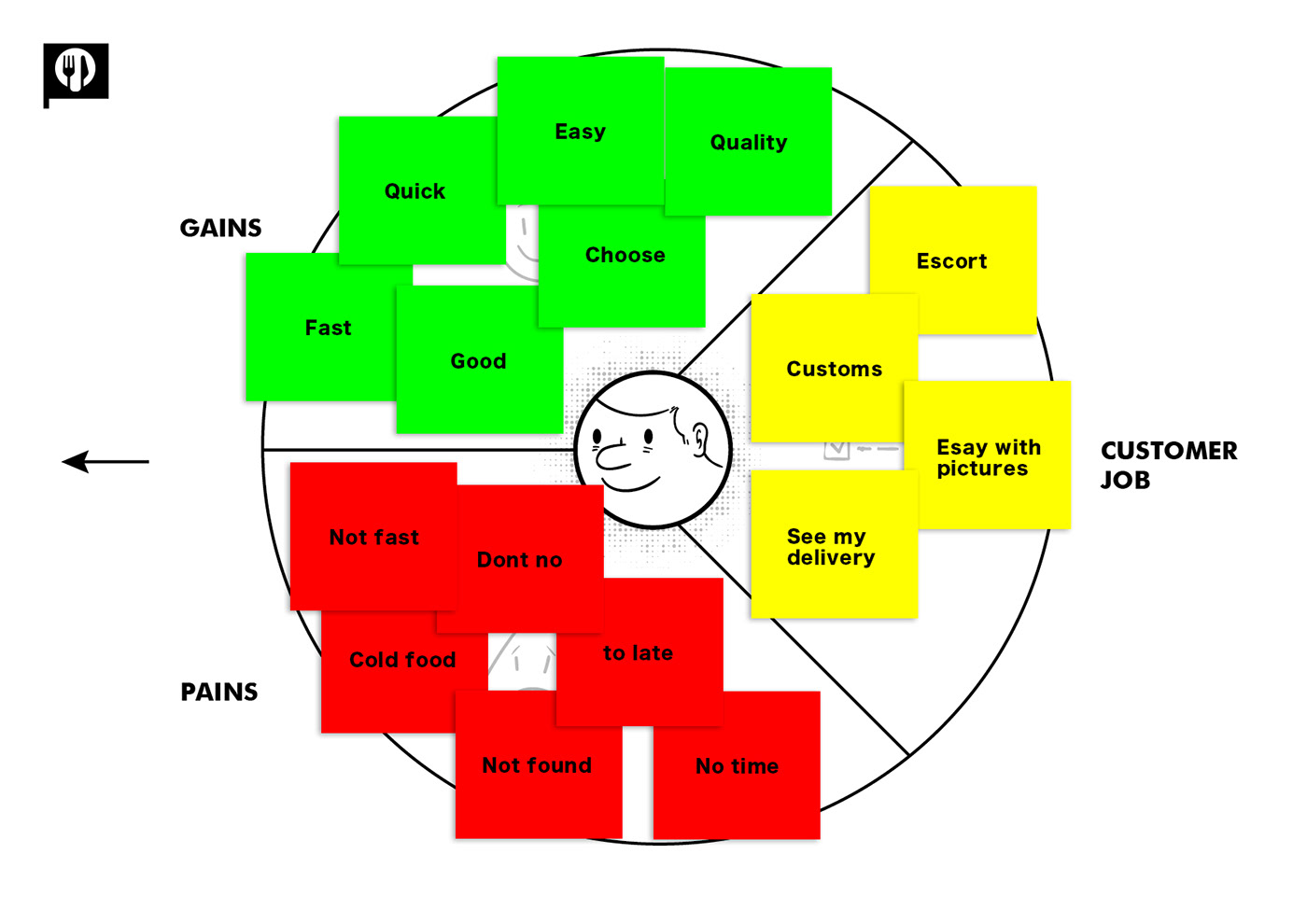
can directly order from the restaurant instead of going through a third-party food

delivery business. It is a web-based ordering system where customers using a

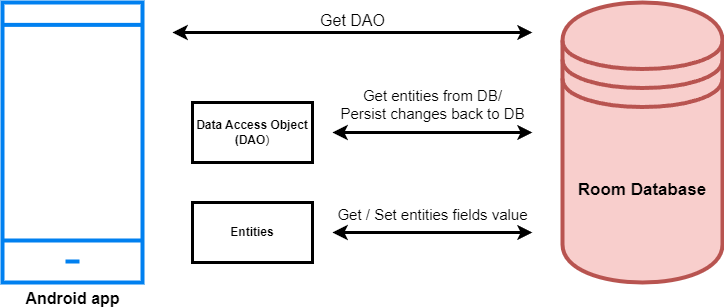
mobile app can use the online user interface to order online.

**2. PROBLEM DEFINTION AND DESIGN THINKING**

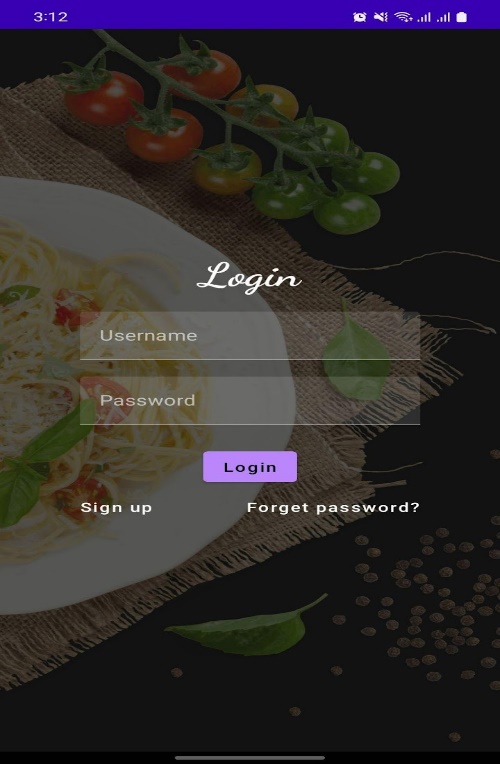
**2.1** **EMPATHY MAP**



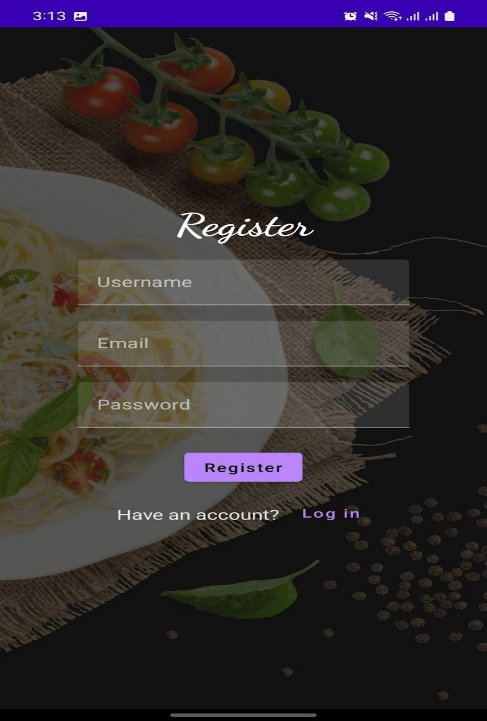
**2.2 IDEATION & BRAINSTROMING MAP**

****

**3.RESULT**

****

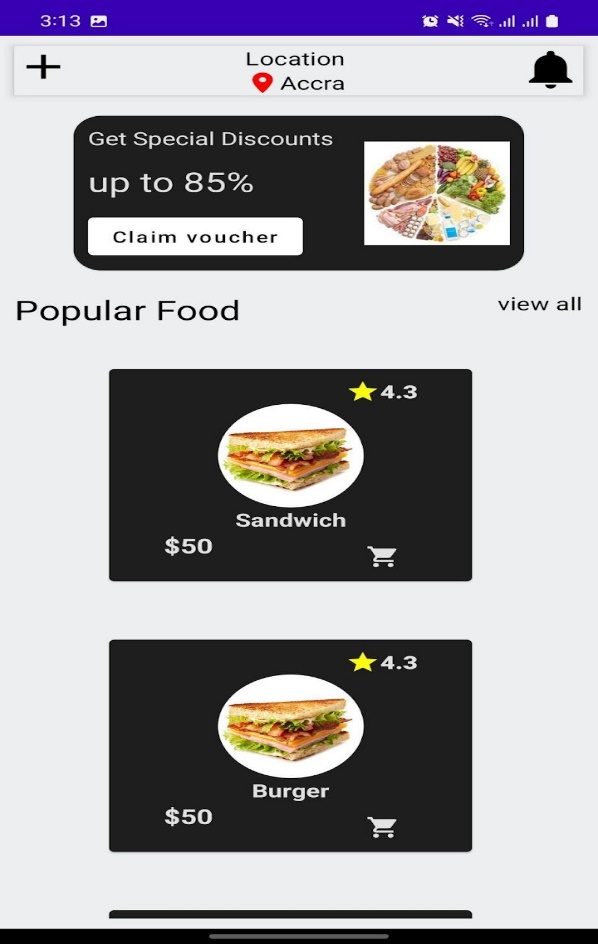
**LOGIN PAGE**

****

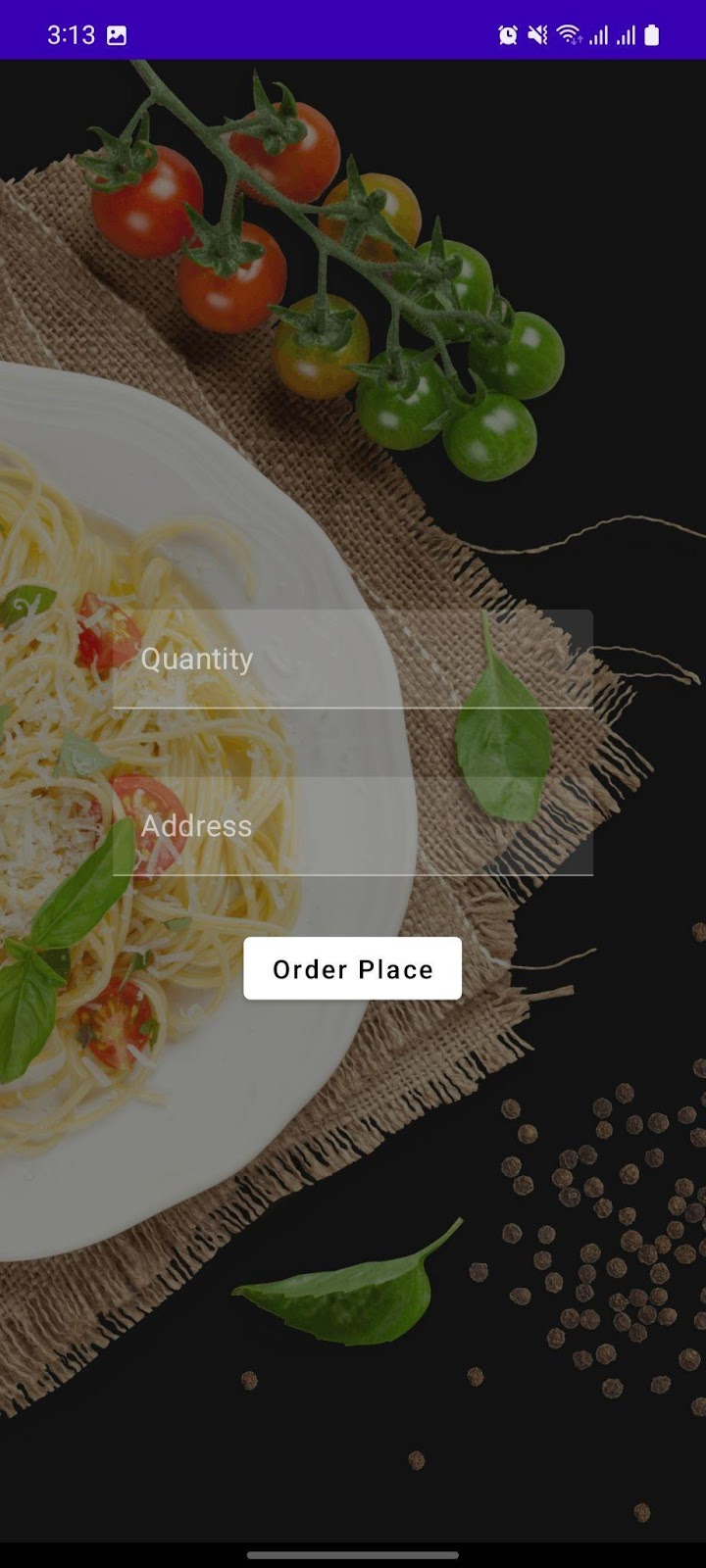
**REGISTER PAGE**

****

**ORDER TRACKING**

****

**MAIN PAGE**

****

**ORDER PAGE**

**4. ADVANTAGES & DISADVANTAGES**

**Advantages of Online Snack Ordering:**

### **Workaholics, Don’t Starve Anymore**

Being a profession-driven person, we afford to miss on our diet but never to miss on our deadlines. But not to worry anymore! You can quickly order your favorite quick munchies from one of the nearest restaurants and grab a bite in no while refilling up your stomach and you are back on the trail.

### **Ladies, You Can Enjoy The Parties Too!**

When you finally have your long-awaited guests coming home; Ladies, you can quickly order delicious food items, single or many, with just a few taps on your mobile screen.

Gone are the days, when you needed to boil yourself in the scorching heat in the kitchen cooking all the time for the guests. With the new online ordering software or online snack delivery apps, ordering food has been easier than before.

### **Urban Restaurants, Reach Out to Remote Foodies**

You captured the foodies of the complete city! Are you sure? Why not extend out the reach to the remote foodies.

With the online snack delivery in place, you can be available to the remote snack-lovers by being visible with your restaurant menu and giving the option to deliver in their area.

### **4. Pinchpenny? Get Cashback**

You love the snack food but you want to save money as well, don’t worry. We have got discounts for you. With this best online food ordering system, you get excellent deals for cashback and discount offers while ordering for delivery or eating at a restaurant. This is one of the significant advantages of online food ordering system.

### **5. Reserve that Quiet Side Table for Your Next Gathering**

Tired of facing the problem of last-minute bookings and cancellations of the tables at the time of your gatherings. You can now pre-book the preferred table for your .get a chance to book the table beforehand in another restaurant.

**DISADVANTAGES**

### **Deliverymen Put Themselves in Danger**

Whether it is a heat wave boiling down the city or it is snowing or raining heavily, a Delivery Boy is waiting outside the restaurant to pick and deliver your order. This is one of the disadvantages of ordering food online.

Although we get the joy of our favorite food in any season, they are also humans who forget their human rights putting themselves in danger sometimes.

### **Disguised Increased Expense**

We surely get attracted by yummy-looking food pictures on the app and a small but highlighting banner of cashback offer.

However, we forget that despite cashback, it is costing us higher than the food which we can cook with the groceries available using all our magical cooking skills and spend blindly ordering the food online**.** This can be considered one of the disadvantages of online food ordering for customers.

### **Revenue Conflicts Between the Snack Restaurants**

### Not every restaurant owner can afford to employ ten delivery boys and bear all the transport and remuneration expenditure; so, they choose to contract with the delivery service providers through these apps. This brings the disadvantages of food delivery service.

However, despite automation in place, one can’t control everything through an automated system, and conflicts occur between the restaurant owner and delivery providers regarding the payments.

**5. APPLICATION**

Food delivery is a home delivery service in which a store,

restaurant, or third-party app delivers food to consumers, whenever they ask for

it. These days, the offers are generally placed through a mobile app, website, or phone.

**6.CONCLUSION**

Although efforts have been put forth to examine

the effects of snack foods on satiety and weight status, to our knowledge the

summarized to date. In addition, studies on the topic used various study designs

(ranging from cross-sectional to randomized controlled trials) and often had

methodologic limitations. For the intervention studies, the intervention duration

as well as the timing of the introduction of the snack food varied; specifically,

some studies advised participants to consume the snack food between meals

whereas others advised participants to snack as needed. The intervention studies

included in this review were primarily short term. The type of snack foods that

were used in the various studies also varied; some studies used snack foods that were high in protein and/or fiber, whereas others used snacks that were high in

fat and/or sugar. In addition, the sample sizes of participants in the included

studies were small, and the composition of the control groups was inconsistent

across studies, with some studies lacking a control group altogether. In addition,

the variation in the dietary patterns of participants makes it difficult to interpret

the findings.

**7. FUTURE ENHANCEMENTS**

The global online food delivery services market is expected

to grow at a compound annual growth rate of 18.7% from 2022 to 2030 to reach

USD 253.95 billion by 2030.

**Food Delivery Trends in the Future**

* Online Food Ordering and Pickup in Store.
* Ghost Kitchen and Ghost Groceries.

* Crypto Food Order.
* Personalized Online Food Delivery Services.
* Reliance and Advanced Technologies.

**8. APPENDIX**

<?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>

<?xml version="1.0" encoding="UTF-8"?>

<module type="JAVA\_MODULE" version="4">

<component name="FacetManager">

<facet type="android" name="Android">

<configuration />

</facet>

</component>

<component name="NewModuleRootManager" inherit-compiler-output="true">

<exclude-output />

<content url="file://$MODULE\_DIR$">

<sourceFolder url="file://$MODULE\_DIR$/java" isTestSource="false" />

</content>

<orderEntry type="jdk" jdkName="Android API 33, extension level 4 Platform" jdkType="Android SDK" />

<orderEntry type="sourceFolder" forTests="false" />

</component>

</module>

package com.example.snackordering

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(

@DrawableRes drawable: 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 val FoodList = 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(

@DrawableRes val drawable: Int,

@StringRes val 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)

}

}