ANDROID STUDIO – EXPERIENCE BASED PROJECT LEARNING

Snack Box: A Customizable Snack Ordering and Delivery App

Submitted by

ASWIN RAJ	-	711022104009
ALPHIN V T	-	711022104005
JITHU SAARON B	-	711022104021
GANESAN A	-	711022104017
SANGEERTHANA S	-	711022104043

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING

IN

FIFTH SEMESTER

COMPUTER SCIENCE AND ENGINEERING

INFO INSTITUTE OF ENGINEERING, COIMBATORE – 641107

NOVEMBER/DECEMBER - 2024

BONAFIDE CERTIFICATE

Certified that this project "Snack Squad: A Customizable Snack Ordering and Delivery App" is the Bonafide work of ASWIN RAJ (711022104009), ALPHIN V T (711022104005), JITHU SAARON B (711022104021), GANESAN A (711022104017), SANGEERTHANA S (711022104043) who carried out the project work under any supervision.

SIGNATURE

STAFF COORDINATOR

Mrs. A. SARANYA M.E.,

ASSISTANT PROFESSOR

DEPT. COMPUTER SCIENCE AND ENGINEERIN(

INFO INSTITUTE OF ENGINEERING,

KOVILPALAYAM COIMBATORE - 641107

SIGNATURE

HEAD OF THE DEPARTMENT

Dr. G. SELVAVINAYAGAM Ph.D.,

HEAD OF THE DEPARTMENT

DEPT. COMPUTER SCIENCE AND ENGINEERING

INFO INSTITUTE OF ENGINEERING,

KOVILPALAYAM COIMBATORE - 641107

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We sincerely thank to Tamil Nadu Skill Development Corporation (TNSDC), Naan Mudhalvan" Platform and ANDROID STUDIO – EXPERIENCE BASED PROJECT LEARNING (EBPL) for encouragement towards our project work for providing necessary skill training.

We sincerely thank our Principal Dr. N. KOTTISWARAN, M.E., Ph.D., and Head of the Department Dr. G. SELVAVINAYAGAM, M.E., Ph.D., and Staff Coordinator Mrs. A. SARANYA M.E for her encouragement towards our project works.

We also thank our project guide and our parents for the complete and wholehearted support, motivation guidance and help in making our project activities.

ABSTRACT

Simple Food Delivery Application

This project outlines the development of a simple, yet efficient Android-based food delivery application designed to streamline the food ordering and management process. The application features two primary interfaces:

Admin Page: Accessible only through a unique username and password, the admin page empowers administrators to manage the food menu, monitor orders, and track delivery statuses. This interface ensures robust security and efficient control over the operations of the application.

User Page: Designed for an intuitive user experience, this page allows customers to browse the available menu, select items, specify quantities, and provide delivery details such as address. The straightforward layout ensures that users can place orders with minimal effort, enhancing convenience.

The application aims to deliver a seamless food ordering process by bridging communication between customers and administrators. By leveraging the Android platform, this project seeks to demonstrate the integration of secure authentication mechanisms, responsive user interfaces, and essential backend functionalities to meet the needs of a basic food delivery service.

INTRODUCTION

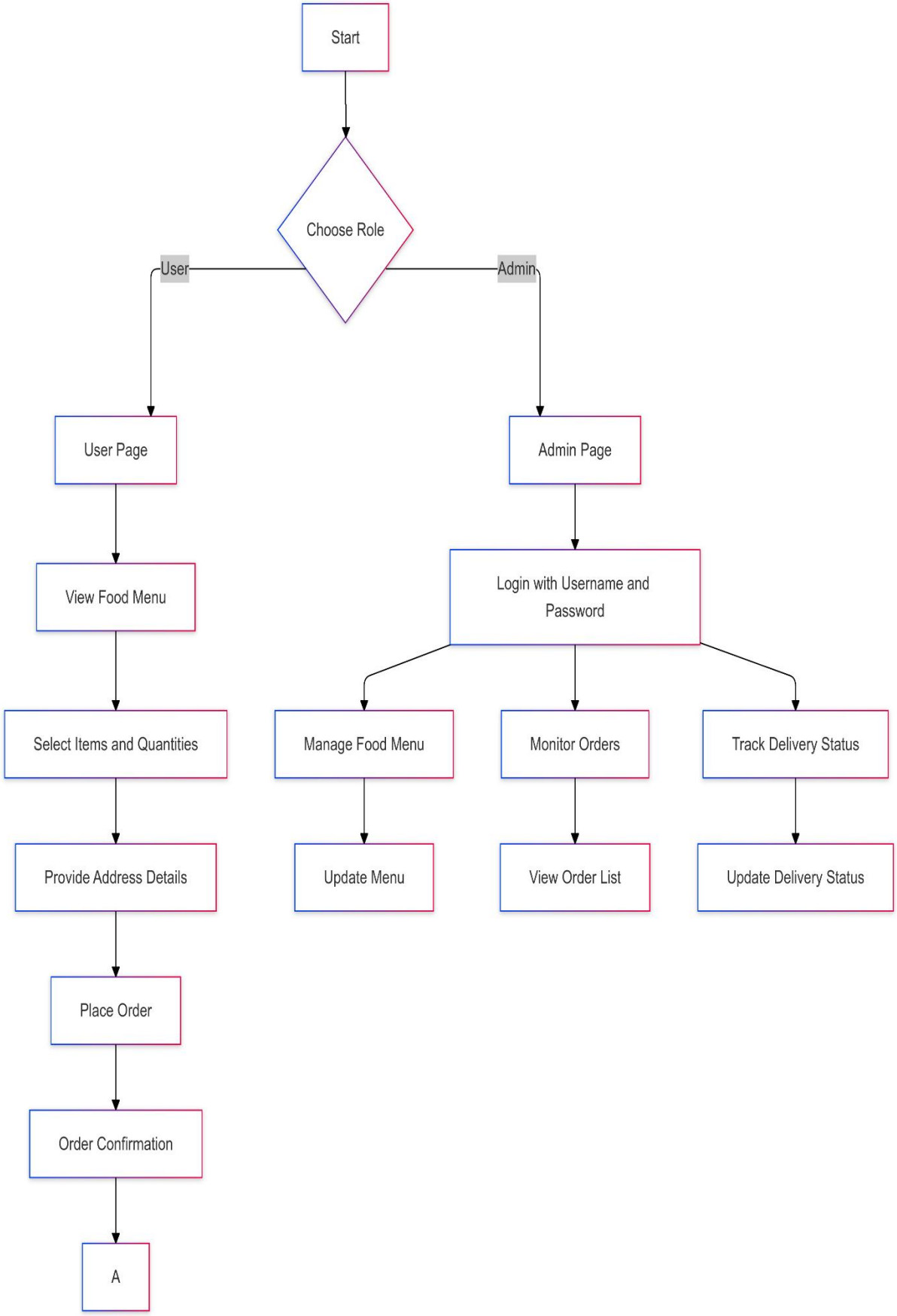
In today's fast-paced world, food delivery services play a vital role in meeting the demands of consumers seeking convenience and quick access to meals. The Simple Food Delivery Application is a lightweight and user-friendly solution designed to facilitate seamless communication between customers and food service providers. By integrating essential features into a simple Android platform, the application aims to address the fundamental needs of both users and administrators.

The application is divided into two primary roles: Admin and User. Administrators can securely log in using unique credentials to manage the food menu, track orders, and oversee delivery statuses. This ensures operational efficiency and secure access to critical management functionalities.

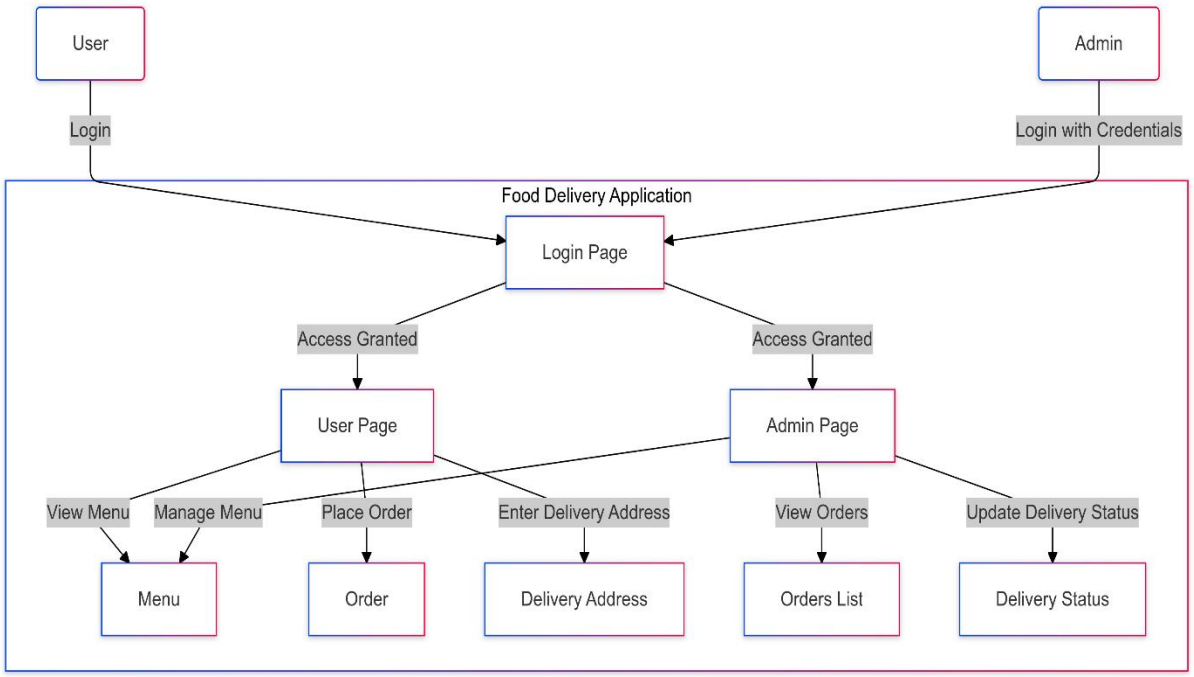
On the other hand, users can interact with the application through an intuitive interface that enables them to browse the food menu, select items with specified quantities, and provide their delivery details. This simplified ordering process ensures a smooth and hassle-free experience for customers.

This project combines key elements of mobile application development, such as authentication, secure data handling, and responsive design, to create a reliable and straightforward food delivery solution. It serves as a foundation for exploring the intersection of user convenience and administrative control in real-world application scenarios.

DATA FLOW DIAGRAM



USER CASE DIAGRAM



SOFTWARE REQUIREMENTS

Here are the software requirements for the Simple Food Delivery Application:

1. Platform Requirements

Operating System: Android OS (API Level 21 or higher for broader compatibility)

Device Compatibility: Smartphones and tablets

2. Frontend Requirements

Language: Kotlin (for Android development with Jetpack Compose)

Framework: Jetpack Compose (for building the UI)

UI Components: Use of Compose components like Image, Text, Button, Lazy Column for list items.

Libraries:

Compose UI Library: androidx.compose.ui:ui

Compose Material Library: androidx.compose.material:material for Material Design components

Navigation Component: androidx.navigation:navigation-compose for screen navigation (e.g., Login, User, Admin pages)

3. Backend Requirements

Database: SQLite database, with Room Database (using androidx.room:room-runtime and androidx.room:room-ktx) for storing user details, menu items, orders, and delivery statuses.

PROGRAM CODE

Main activity

```
package com.example.SnackBox

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.SnackBox.ui.theme.SnackBoxTheme

import android.content.Intent as Intent1


class MainPage : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            SnackBoxTheme {
                // 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.burger 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) {
    SnackBoxTheme {
        Scaffold() {
            val context = LocalContext.current
            App(context)
        }
    }
}
```

Admin Activity

```
package com.example.SnackBox

import android.icu.text.SimpleDateFormat
import android.os.Bundle
import android.util.Log
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.LazyRow
import androidx.compose.foundation.lazy.items
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
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.unit.dp
import androidx.compose.ui.unit.sp
import com.example.SnackBox.ui.theme.SnackBoxTheme
import java.util.*

class AdminActivity : ComponentActivity() {
    private lateinit var orderDatabaseHelper: OrderDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        orderDatabaseHelper = OrderDatabaseHelper(this)
        setContent {
            SnackBoxTheme {
                // A surface container using the 'background' color from
the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    val data=orderDatabaseHelper.getAllOrders();
                    Log.d("swathi" ,data.toString())
                    val order = orderDatabaseHelper.getAllOrders()
                    ListListScopeSample(order)
                }
            }
        }
    }
}

@Composable
fun ListListScopeSample(order: List<Order>) {
    Image(
        painterResource(id = R.drawable.order), contentDescription = "",
        alpha =0.5F,
        contentScale = ContentScale.FillHeight)
    Text(text = "Order Tracking", modifier = Modifier.padding(top = 24.dp,
start = 106.dp, bottom = 24.dp ), color = Color.White, fontSize = 30.sp)
    Spacer(modifier = Modifier.height(30.dp))
    LazyRow(
        modifier = Modifier
```

```
            .fillMaxSize()
            .padding(top = 80.dp),

        horizontalArrangement = Arrangement.SpaceBetween
    ){
        item {

            LazyColumn {
                items(order) { order ->
                    Column(modifier = Modifier.padding(top = 16.dp, start =
48.dp, bottom = 20.dp)) {
                        Text("Quantity: ${order.quantity}")
                        Text("Address: ${order.address}")
                    }
                }
            }
        }

    }
}
```

OUTPUT

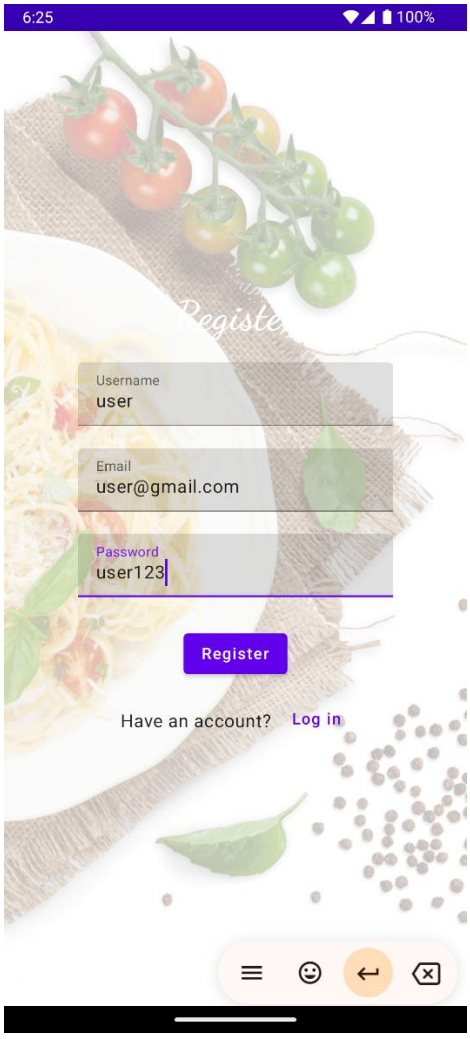


Fig: Registration Page

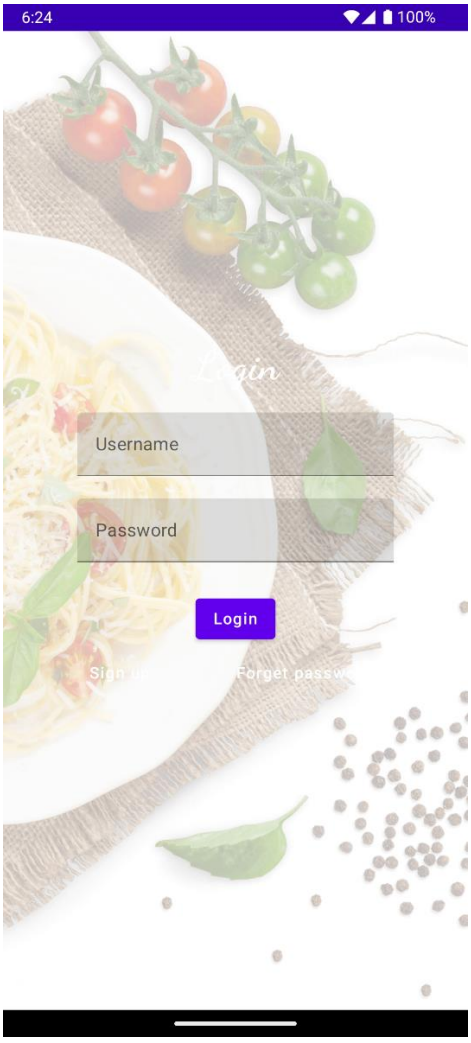


Fig: Login Page

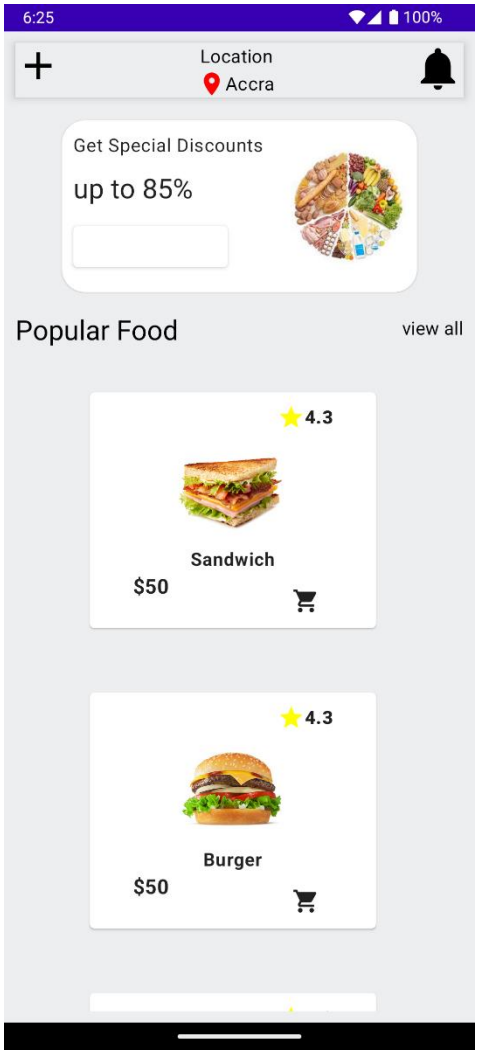


Fig: Home Page

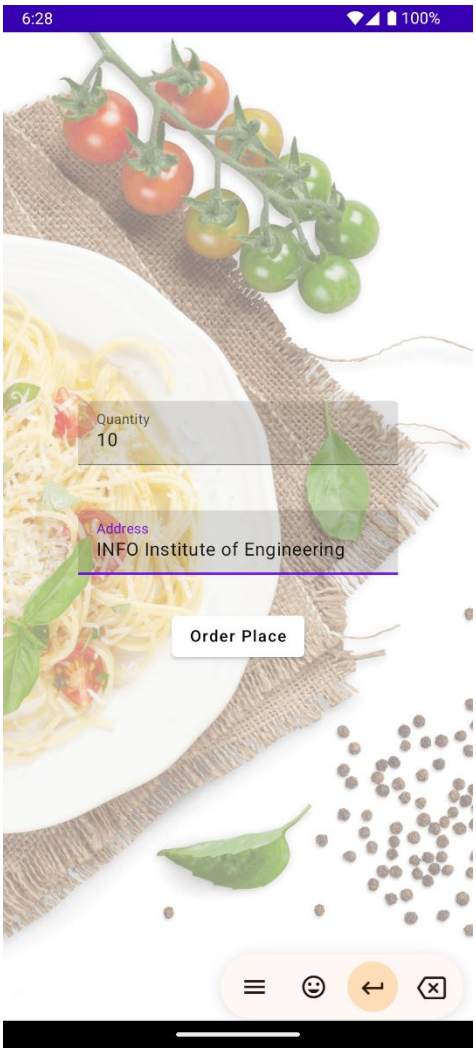


Fig: Cart Page

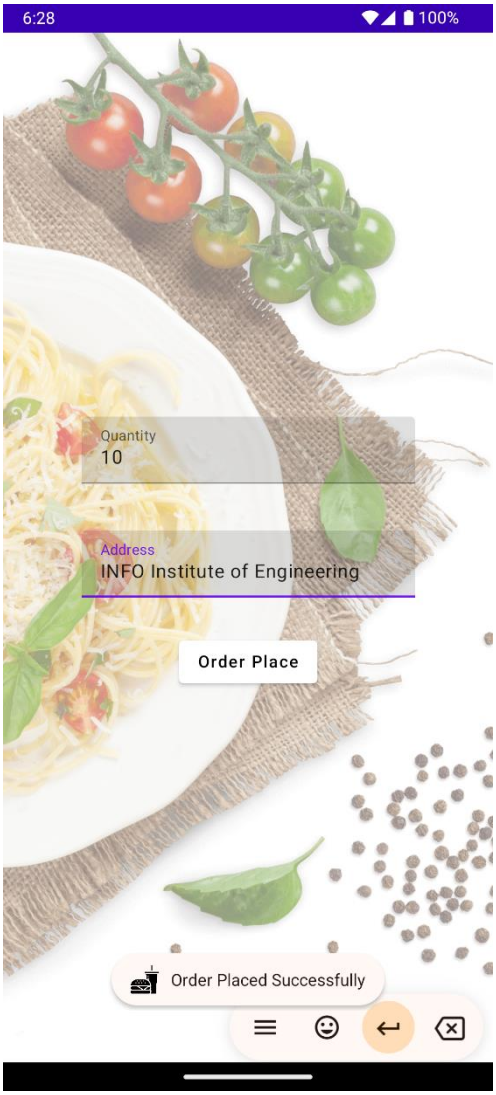


Fig: Order Placed Successful Message

FUTURE ENHANCEMENT

1. Real-Time Order Tracking:

Integrate real-time GPS tracking so users can see the location of their order and estimated delivery time.

This would require additional permissions for location access and possibly a third-party mapping service like Google Maps API.

2. Payment Integration:

Add secure online payment options, including credit/debit cards, UPI, and popular payment gateways like PayPal, Stripe, or Razorpay.

This would enhance the app's functionality, making it a complete end-to-end ordering system.

3. Push Notifications:

Implement push notifications to keep users updated on the status of their order (e.g., order confirmed, preparing, out for delivery).

Use Firebase Cloud Messaging (FCM) or a similar service for notification delivery. And many more.

CONCLUSION

The Simple Food Delivery Application successfully demonstrates a streamlined approach to managing and fulfilling food orders through an Android platform. By integrating two distinct interfaces—an Admin Page for secure management and a User Page for effortless food ordering—the application effectively bridges the gap between service providers and customers.

The project highlights essential features such as secure authentication, menu management, and an intuitive user experience, ensuring both operational efficiency and customer satisfaction. It's simple yet functional design showcases the practical application of core mobile development concepts, offering a foundation for scaling and enhancing food delivery services in the future.

Through this application, the project achieves its objective of providing a reliable, user-friendly solution that simplifies the food delivery process while maintaining control and security for administrators. This project lays the groundwork for future enhancements, including advanced features such as real-time order tracking, payment integration, and analytics, to further elevate its utility and functionality.