Snack Squad: A Customizable Snack Ordering And Delivery App

Abstract:

The purpose of this project is to devlop and online snack ordering system. It is a system that enable customer to place their snack order online at any time at any place. The reason to devlop this system is due to the issue facing by snack industry. These issue are such as peak hour-long queue issues, increase of take away than visitors, speed major request of snack management, limited promotion and quality control of snack management. Therefore, this system enhance the speed and standardization of taking order from the cutomer and supply it to the staff in the kitchen accordingly. Beside that it provide user friendly web pages and effective advertising medium to the new product of the online snack ordering restaurant to the customer at reasonable price.

Introduction:

In recent years, the demand for food delivery apps has surged, with users seeking convenience, variety, and instant gratification. A snack ordering and delivery app focuses on providing quick, affordable, and customizable snack options that can be delivered to users at home, work, or other locations. With the rise of fast-paced lifestyles, snacks have become a popular choice for quick bites, making this niche a promising business opportunity.This report outlines the concept, features, design principles, technical architecture, and future scope for a **Snack Ordering and Delivery App**, which aims to create a seamless and enjoyable snack ordering experience.

Project Description:

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.

System Requirements:

Operating System: Android-7 or later

Ram:1Gb (Minimum)

Rom:20Mb (Minimum)

Internet connection

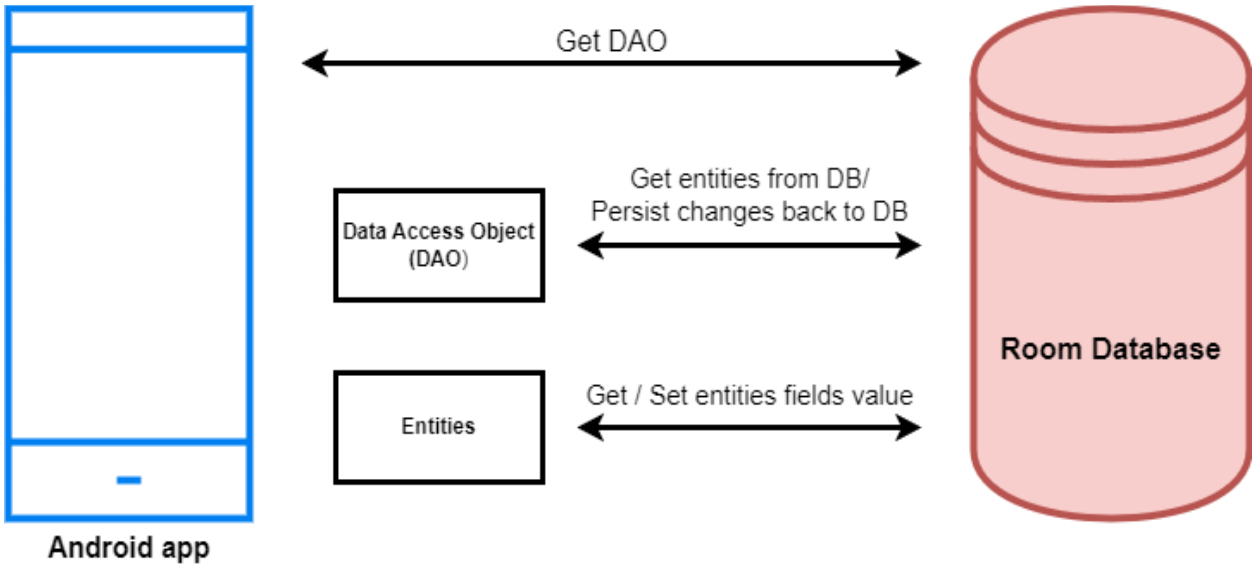
Tools Used:

Android Studio

Firebase

Kotlin

Architecture



Project Workflow:

- Users register into the application.
- After registration , user logins into the application.
- User enters into the main page.
- User can view the items,select and order the items.
- From admin login we can view the orders placed.

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.

Program:

```
package com.example.snackordering

import androidx.room.ColumnInfo

import androidx.room.Entity

import androidx.room.PrimaryKey

@Entity(tableName = "user_table")

data class User(

    @PrimaryKey(autoGenerate = true) val id: Int?,
```

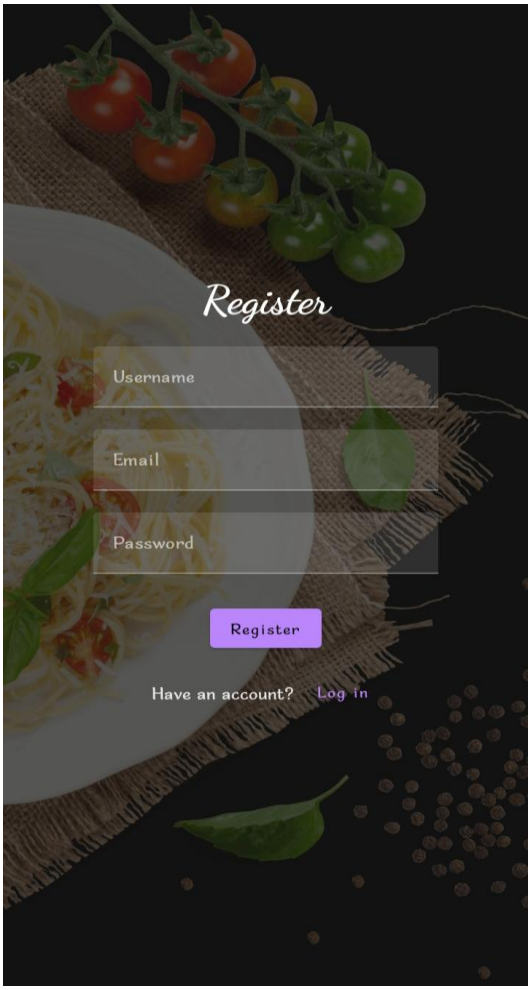
```
    @ColumnInfo(name = "first_name") val firstName: String?,
   @ColumnInfo(name = "last_name") val lastName: String?,
    @ColumnInfo(name = "email") val email: String?,
    @ColumnInfo(name = "password") val password: String?,
 )
package com.example.snackordering
import androidx.room.ColumnInfo
import androidx.room.Entity
import androidx.room.PrimaryKey
@Entity(tableName = "user_table")
data class User(
    @PrimaryKey(autoGenerate = true) val id: Int?,
    @ColumnInfo(name = "first_name") val firstName: String?,
    @ColumnInfo(name = "last_name") val lastName: String?,
    @ColumnInfo(name = "email") val email: String?,
    @ColumnInfo(name = "password") val password: String?,
 )
```

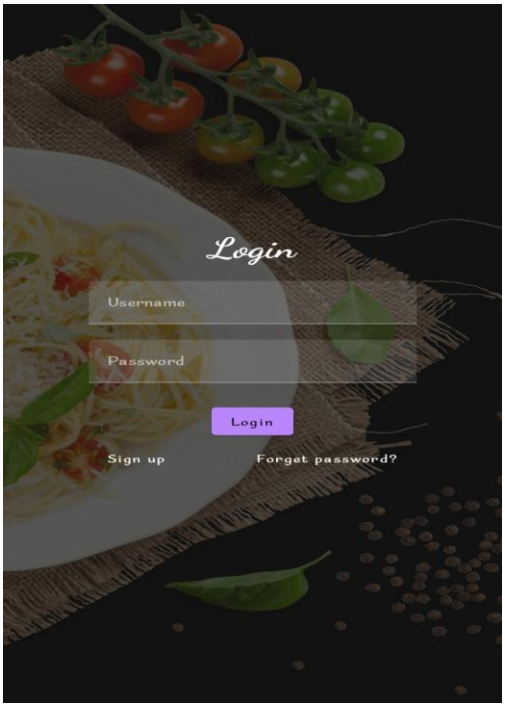
package com.example.snackordering

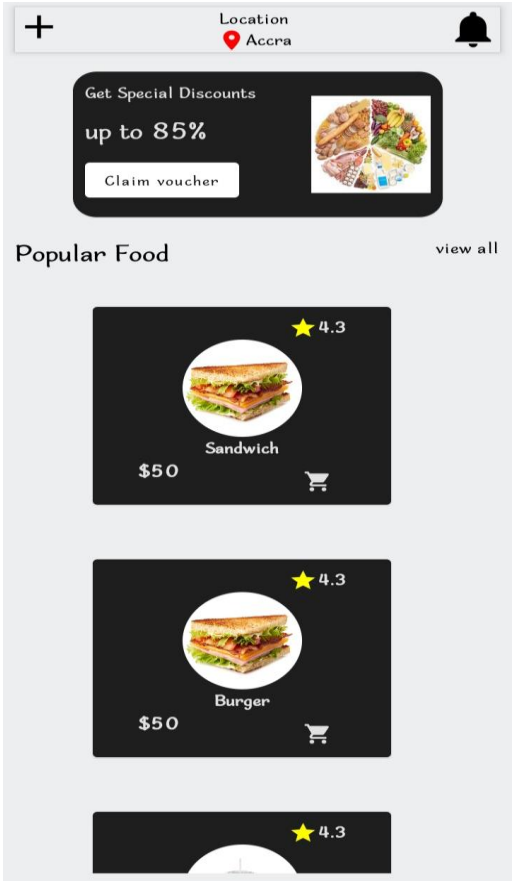
```
import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase
@Database(entities = [User::class], version = 1)
abstract class UserDatabase : RoomDatabase() {
 abstract fun userDao(): UserDao
 companion object {
@Volatile
        private var instance: UserDatabase? = null
     fun getDatabase(context: Context): UserDatabase {
            return instance ?: synchronized(this) {
                val newInstance = Room.databaseBuilder(
                    context.applicationContext,
                    UserDatabase::class.java,
                    "user_database"
                ).build()
                instance = newInstance
```

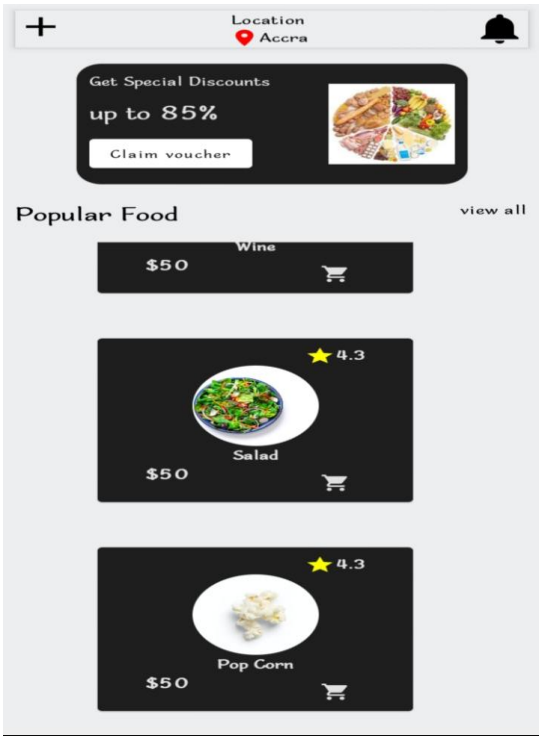
```
            newInstance
        }
    }
  }
}
```

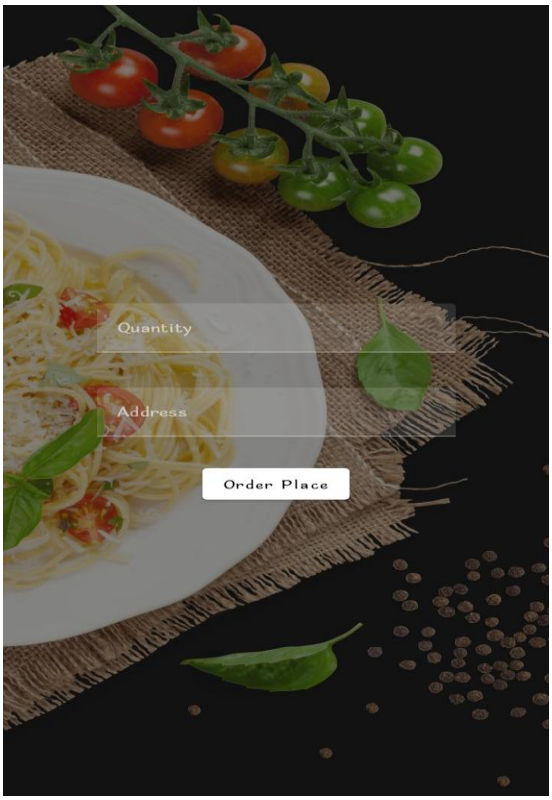
Output




Login
Username
Password
Login
Sign up
Forget password?


Location
Accra
Get Special Discounts
up to 85%
Claim voucher
Popular Food
view all
4.3
Sandwich
$50
4.3
Burger
$50
4.3


Location
Accra
Get Special Discounts
up to 85%
Claim voucher
Popular Food
view all
Wine
$50
4.3
Salad
$50
4.3
Pop Corn
$50


Quantity
Address
Order Place

Conclusion:

Snackify is a user-friendly snack ordering and delivery app designed to offer a seamless and enjoyable experience for snack lovers. With a clean UI, real-time tracking, multiple payment options, and an easy-to-use interface, the app aims to make snack ordering a hassle-free, enjoyable experience. Through continuous updates and new features, Snackify aims to become the go-to app for snack enthusiasts.

Future Scope for Snack Ordering and Delivery App:

The **Snack Ordering and Delivery App** has a great potential for growth, expansion, and innovation. As the food delivery industry continues to evolve and consumer preferences change, the app can evolve to meet new demands, integrate emerging technologies, and offer more personalized, engaging experiences. Below are some key areas where the future scope of the app can be expanded.