# Introduction

* A food delivery app that provides food delivery at your door in very less time and with the best packaging.
* Providing food from every famous food place near you. Order food with the best user experience.
* UX (user experience) research is the systematic investigation of users and their requirements,
* In order to add context and insight into the process of designing the user experience.
* UX research employs a variety of techniques, tools, and methodologies to reach conclusions,
* determine facts, and uncover problems, thereby revealing valuable information that can be fed into the design process.
  1. **Overview:**

What is an online food ordering system? An online food ordering system **allows your business to**

**accept and manage orders placed online for delivery or takeaway**. Customers browse a digital menu,

either on an app or website and place and pay for their order online The actual objective of this business is

**to ensure that good food is delivered to the customers, and on time**.

* 1. **Purpose:**

Food-delivery apps **allow customers to order from a nearby restaurant at their convenience**.

The customers can get their order delivered, they can pick it up themselves or they can dine in.

The restaurants receive the order on the restaurant app and prepare the meal.

Online food ordering **gives customers the freedom and choice to place an order at virtually any time,**

**from anywhere, saving the time and resources typically spent on travelling to pick up a meal**.

It also gives the customers the advantage of reordering the favorite order in the easiest and hassle-free manner.

Your body needs macronutrients (carbs, protein, and fat) to function properly. Food trackers will allow

You to log your food intake and show you exactly how many macros you need — and how many you're

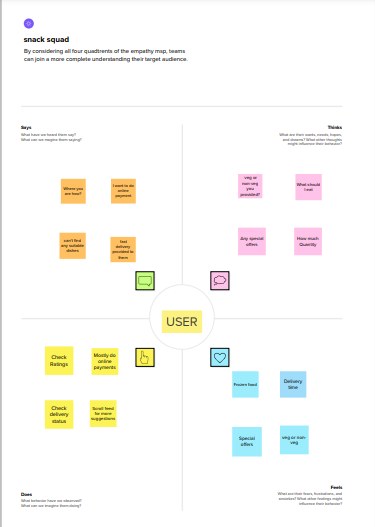
Actually getting. **Customers are more inclined to explore all of their menu options, without the**

**Pressure to close their orders, and spend more than they can spend on ordering over the phone or**

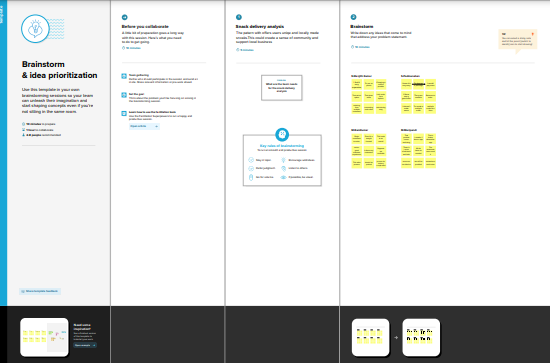
**In person**. With no line behind them, the pressure to order quickly for your guests is gone and they are more inclined to get that extra item.

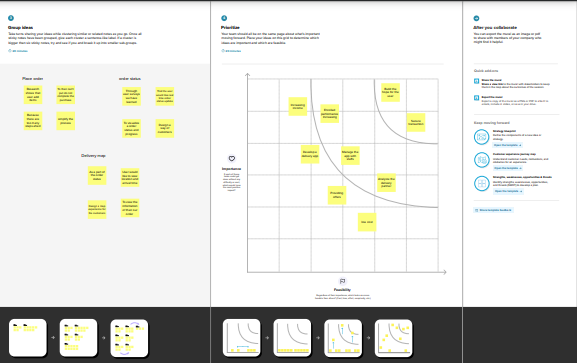
**2. Problem Definition & Design Thinking**

**2.1 Empathy Map**

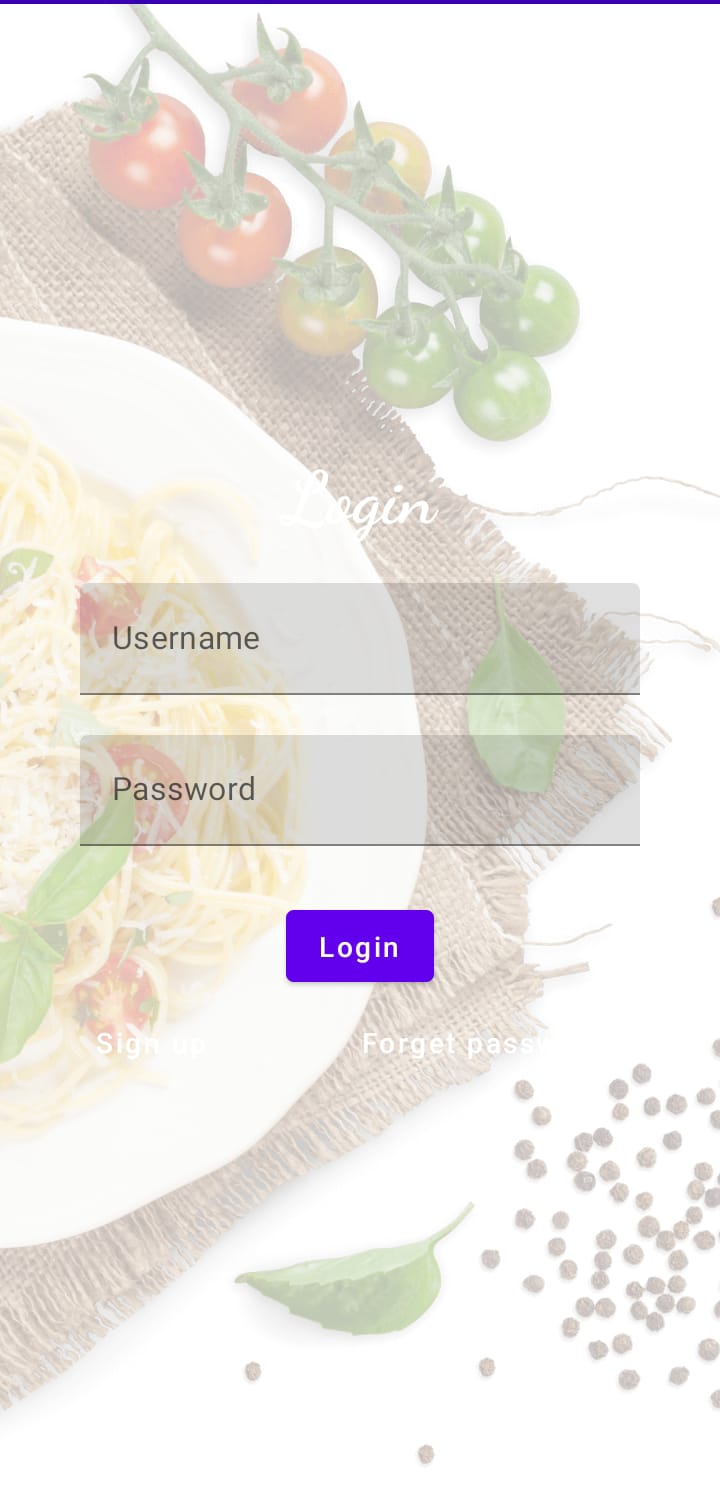
****

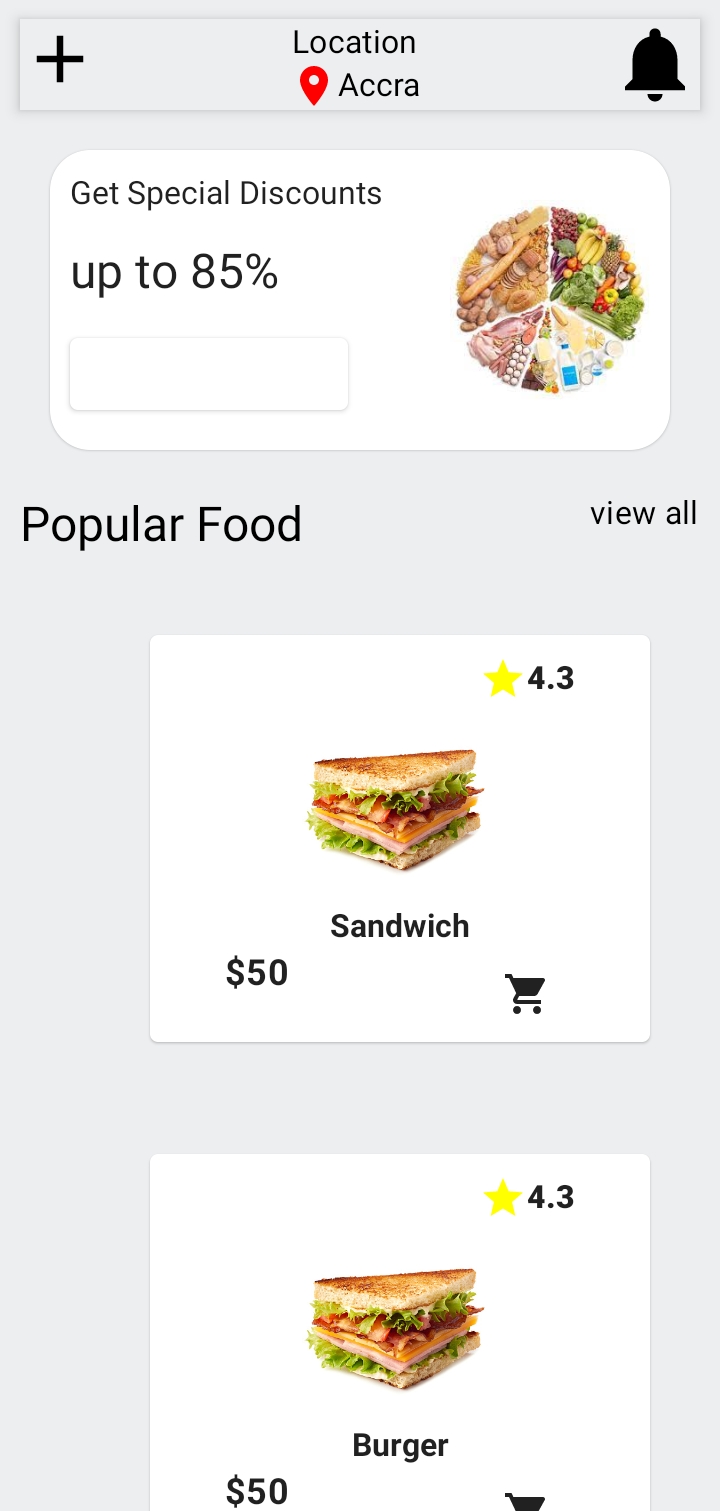
**2.2 Ideation & Brainstorming Map**

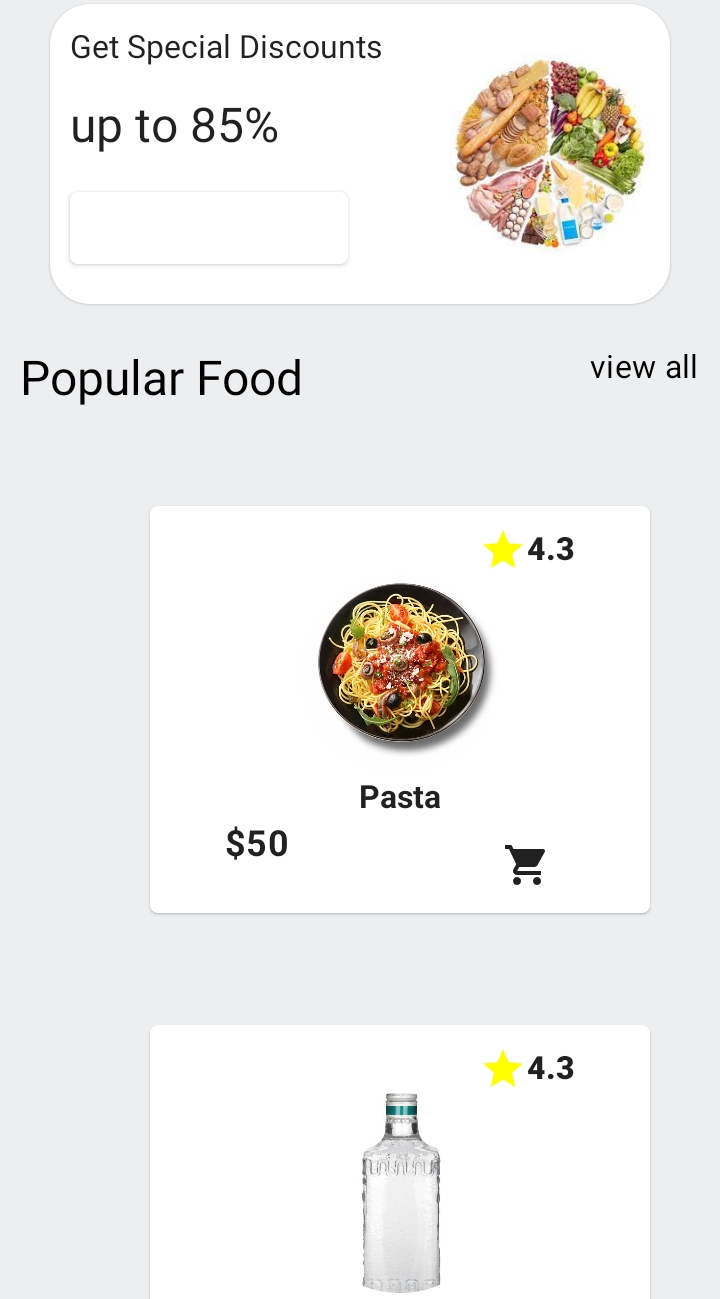
****

****

**3. RESULT:**

****

****

****

**4. ADVANTAGES & DISADVANTAGES**

## The Advantages of Food Delivery

There are many advantages to using a [food delivery system](https://www.thespruceeats.com/best-food-delivery-services-4847285). Here are just a few of them:

### It’s Convenient

One of the biggest advantages of food delivery is that it’s convenient. Customers don’t have to leave their homes or offices to enjoy a delicious meal. They can simply place an order and have it delivered right to their doorsteps.

### It Saves Time

Another advantage of food delivery is that it saves time. Customers don’t have to spend time cooking or traveling to and from a restaurant. Instead, they can use that time to do other things, such as work or spending time with family and friends.

### It’s Affordable

Food delivery is also affordable. In most cases, the delivery fee is very reasonable. And in some cases, it’s even free. Additionally, customers don’t have to tip the delivery person, which saves even more money.

### It’s Available When You Need It

Another advantage of food delivery is that it’s available when you need it. Most delivery services operate during regular business hours. However, there are some that deliver 24 hours a day, seven days a week. So, whether you’re craving a late-night snack or an early-morning breakfast, you can usually find a food delivery service that will meet your needs.

### It’s Easy to Use

The food delivery system is also easy to use. Customers can order food from their favorite restaurants with just a few clicks of a button. And thanks to mobile apps, it’s even easier to order food on the go.

## The Disadvantages of Food Delivery

While there are many advantages to food delivery, there are also some disadvantages that should be considered. Here are a few of them:

### It’s Not Always Reliable

One of the biggest [disadvantages of food delivery](https://www.frontiersin.org/articles/10.3389/fnut.2020.00014/full) is that it’s not always reliable. In some cases, the food may take longer to arrive than expected. And in other cases, the food may be cold or not as fresh as it would be if you had picked it up yourself.

### You May Not Get What You Want

While most food delivery services are good at delivering what you ordered, there is always the possibility that you may not get exactly what you want. For example, if you order a pizza with pepperoni and mushrooms, but the restaurant is out of mushrooms, you may end up with a pizza that only has pepperoni on it.

### It’s Can Sometimes Be Dangerous for Delivery Drivers

This is especially true in large cities, where traffic can be a nightmare. Delivery drivers often have to contend with traffic, construction, and other hazards while trying to get your food to you on time. And in some cases, these hazards can lead to accidents. It’s always better to be safe than sorry, so if you’re ever in an accident while working for a food delivery service, be sure to[contact a bicycle accident lawyer.](https://www.feldmanlee.com/practice-areas/personal-injury-law/bicycle-accidents/)

### It’s Not Always Cheaper Than Eating Out

While delivery can be more convenient than going out to eat, it’s not always cheaper. In some cases, restaurants will charge a delivery fee. And in other cases, they may add a “convenience fee” to your order. So, if you’re looking to save money, it may be better to eat out instead of ordering in.

As you can see, there are both advantages and disadvantages to using a food delivery system. Before deciding whether or not to use one, be sure to weigh the pros and cons carefully.

**5. APPLICATIONS:**

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 Application Delivery is a combination of services that work in tandem to

deploy an application (typically a web application) to end-users with fast, efficient, and reliable data processing

and computing within data centers or cloud environments.

**6. CONCLUSION:**

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses. At the end it is concluded that we have made effort on following points...• A description of the background and context of the project and its relation to work already done in the area.• Made statement of the aims and objectives of the project. • The description of Purpose. Scope, and applicability• We define the problem on which we are working in the project.• We describe the requirement Specifications of the system and the actions that can be done on these things.• We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.• We included features and operations in detail, including screen layouts• We designed user interface and security issues related to system . Finally the system is implemented and tested according to test cases

**7. FUTURE SCOPE:**

According to Market Research Future (MRFR), the global snack food packaging market is predicted to grow at a compound annual growth rate of 5.5% over the next five years. The Indian snacks market size reached INR 38,603

Crore in 2022. Looking forward, **IMARC Group expects the market to reach INR 70,731 Crore by 2028**, exhibiting

a growth rate (CAGR) of 10.4% during 2023-2028.

**8. APPENDIX**:

|  |
| --- |
|  |
| package com.example.snackordering |
|  |  |
|  | 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.SnackOrderingTheme |
|  |  |
|  | class MainActivity : ComponentActivity() { |
|  | private lateinit var databaseHelper: 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 |
|  | ) { |
|  |  |
|  | RegistrationScreen(this,databaseHelper) |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |
|  |  |
|  |  |
|  | @Composable |
|  | fun RegistrationScreen(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 email 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 = "Register" |
|  | ) |
|  |  |
|  | 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 = email, |
|  | onValueChange = { email = it }, |
|  | label = { Text("Email") }, |
|  | 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() && email.isNotEmpty()) { |
|  | val user = User( |
|  | id = null, |
|  | firstName = username, |
|  | lastName = null, |
|  | email = email, |
|  | password = password |
|  | ) |
|  | databaseHelper.insertUser(user) |
|  | error = "User registered successfully" |
|  | // Start LoginActivity using the current context |
|  | context.startActivity( |
|  | Intent( |
|  | context, |
|  | LoginActivity::class.java |
|  | ) |
|  | ) |
|  |  |
|  | } else { |
|  | error = "Please fill all fields" |
|  | } |
|  | }, |
|  | modifier = Modifier.padding(top = 16.dp) |
|  | ) { |
|  | Text(text = "Register") |
|  | } |
|  | Spacer(modifier = Modifier.width(10.dp)) |
|  | Spacer(modifier = Modifier.height(10.dp)) |
|  |  |
|  | Row() { |
|  | Text( |
|  | modifier = Modifier.padding(top = 14.dp), text = "Have an account?" |
|  | ) |
|  | TextButton(onClick = { |
|  | context.startActivity( |
|  | Intent( |
|  | context, |
|  | LoginActivity::class.java |
|  | ) |
|  | ) |
|  | }) |
|  |  |
|  | { |
|  | Spacer(modifier = Modifier.width(10.dp)) |
|  | Text(text = "Log in") |
|  | } |
|  | } |
|  | } |
|  | } |
|  | private fun startLoginActivity(context: Context) { |
|  | val intent = Intent(context, LoginActivity::class.java) |
|  | ContextCompat.startActivity(context, intent, null) |
|  | } |
|  |  |