Name : S.Sudalai Vadivu

College code : 9530

College name : St. Mother Theresa Engineering College

Naan Mudhalvan id : au953021104051

Project Name : E Commerce Application for IBM cloud foundry

INTRODUCTION :

Electronic commerce or E Commerce is a platform through which trading of products and services can be done using the internet.Online food ordering is a process that delivers food. This style of food delivery is gaining popularity with more and more people especially the younger generation turning to mobile food ordering apps, thereby changing the way food is delivered and picked up. The customer can generate an order without having to explain it to another human being and have the food delivered at his doorstep.

PROBLEM SOLUTION :

The objective of this project is to provide a convenient platform for users to browse menus, place food orders, and have those orders delivered to their doorstep efficiently and quickly. Key goals include improving customer convenience and streamlining the food ordering and delivery process. The main objective is that it allows customers to order food without the need to visit in person, saving time and effort. It eliminates the need for meal preparation and cooking, making it a time-saving option for busy individuals. It's also convenient for ordering meals for groups or parties, as individual orders can be combined and delivered together. A food delivery app's goals are to offer a practical and effective platform that links customers and enabling food ordering and delivery services.

INNOVATION:

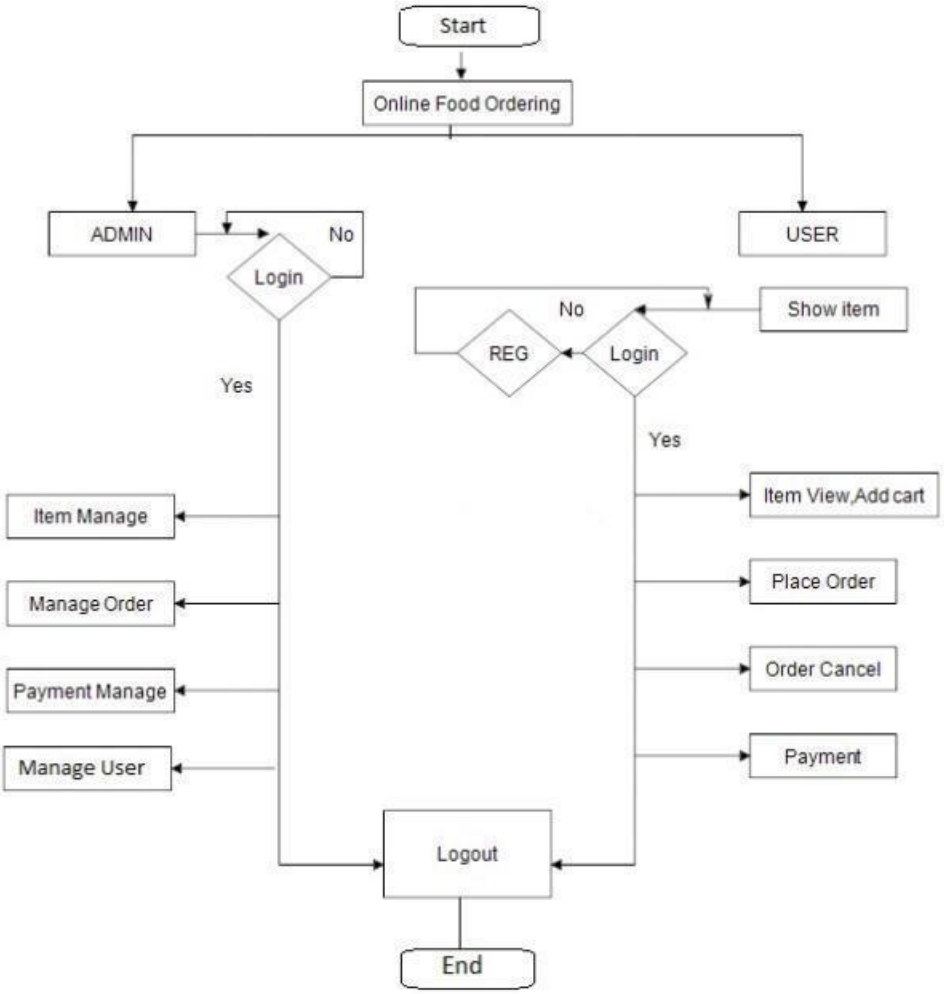
- Using IBM Cloud Functions to implement serverless functions for real-time order tracking, notifications, and automated processes, improving app performance and scalability.
- This E commerce application allows customers to order food without the need to visit in person, saving time and effort which gives the time saving option for busy individuals.

SOFTWARE REQUIREMENTS :

Programming Languages:

- Java for Android app development.
- Java or Java-based frameworks for backend development.
- MySQL for database management.

METHODOLOGY :



APPLICATION DESCRIPTION:

There are certain requirements the proposed application must fulfil to meet the objectives of the project.

The requirements to be achieved:

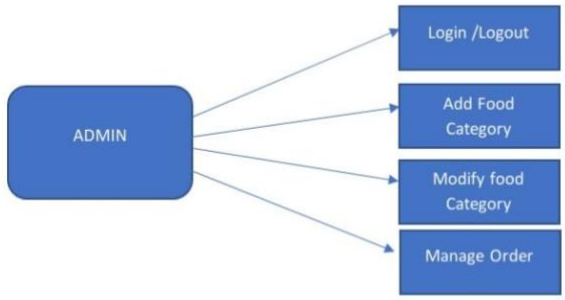
In Background Management Platform:

- Administrator can add and modify food categories.
- Administrator can add, modify and query food information.
- Administrator can add, modify and query employee information.
- Administrator can manage orders produced from the web application and
- Android application.

In the Website Public Page and Android Application:

- Customer can view food information, such as category, name, price, image, description and so on.
- Customer can order food.
- Customer can modify food item, food amount in Shopping Cart.
- Produce food order.

FUNCTIONAL DESCRIPTION :



Background management :

Log In/Out:

Properties:

- Username
- Password

When the administrator connects the background management platform url, a log in interface will be displayed. The administrator needs to input correct username and password to log into the main page of the background management page. One admin username and password is pre-set when the application is initialized.

Add Food Category:

Properties:

- Category Name
- Note

When the administrator clicks “Category Manage”, a list view of dish category will be displayed. On the bottom of the list view page, there is a button “Add Category”.While clicking the “Add Category” button, the page will be linked to a dish-adding interface, the administrator can add a new dish category here.

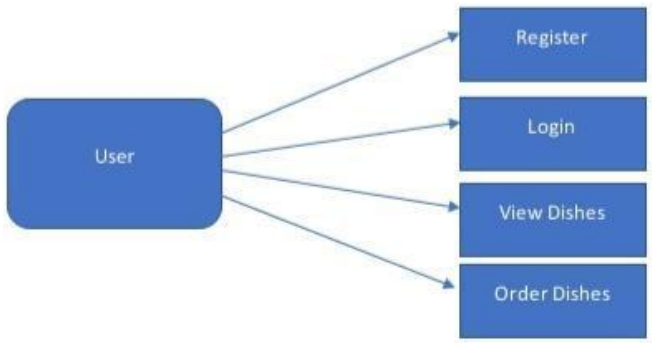
Modify food category:

Properties:

- Category Name
- Note

In addition, in the category list view page, the administrator can modify the existing dish category by clicking “Modify” button.

USER APPLICATION :



Register:

When the customer wants to purchase dishes on the website, first, he needs to register an account by clicking “Register” button to enter the register page, and then just filling in User Name, Password and Repeat Password. If the customer register is successful, a page will display to inform

the customer, also if there is a register fault, such as the User Name already exists, a friendly warning message will be given on the register page.

Log In:

After registering, the customer can log in with the correct User Name and Password to the website to purchase the dishes.

View Dishes:

To click “Online Order” tab, the customer can view the current dishes according to different categories. Every dish item has two buttons: Order and Detail. The customer can click the “Detail” Button to view the detailed information (big image, material of dish, price, etc.) of dishes. By clicking “Order” Button, the dish item you select will be added to the shopping cart.

Shopping Cart:

The dish items purchased by the customer will be added to the shopping cart. The customer can view the dish items he/she purchases, unit price, dish amount and total price. The default amount for each dish item is one, the customer can update the amount according to his own requirement. After modifying the amount,

customer needs to click “Update” button to update the dish amount. Also, if the customer needs to delete some dish item, it is easily implemented by clicking “Delete” Button following each dish item.

CREATING APPLICATION IN IBM CLOUD FOUNDRY:

Step 1:

To start developing your application locally, begin by logging in to the IBM Cloud Platform directly from the command line, as shown in the example. You can specify optional parameters, such as your organization with option -o and the space with option -s. If you’re using a federated account use –sso.

Ibm cloud login

After logging in, when you are asked if you want to install any extensions, you may see an announcement regarding the Cloud Foundry plugin. Type the command as shown in order to download and install the CLI extension used in this tutorial.

Ibmcloud cf install

When you log in you might be asked to choose a region. For this exercise, select us-south as the region, as that same option is used to build a CD Toolchain later in this tutorial.

Next, set the endpoint (if it isn't set already). Other endpoints are possible, and might be preferable for production use. For now, use the code as shown, if appropriate for your account.

Ibmcloud api cloud.ibm.com

Target the Cloud Foundry (cf) aspect of IBM Cloud Platform by using the target command and the –cf option. The cf API is no longer embedded within the CLI Developer Tools and will have to be downloaded separately.

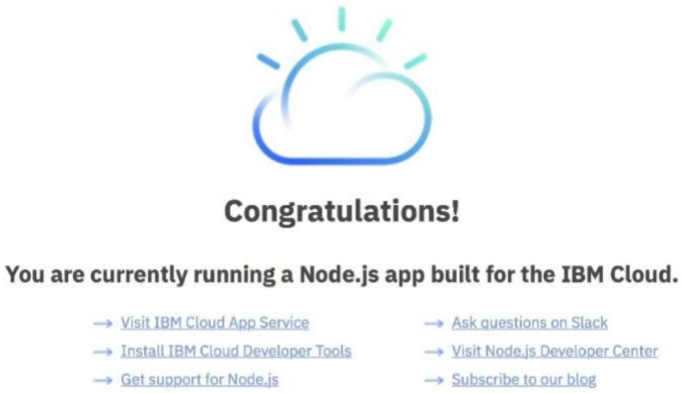
Ibmcloud target –cf

And now, time to create a web application. The dev space is a default option for your organization, but you might prefer to create others for isolating different efforts. For example, keeping 'finance' separate from 'development'.

Ibmcloud dev create

With that command, you're asked a series of questions. You can go back at many points in the process, so if you feel lost you can start over by deleting the existing directory and creating a new directory. Even when you create your application on the command line, you'll still see the results in your IBM Cloud console.

Now there is a option for choosing the application type web or mobile application and have to select the language for our application such as Java or python and then we have to choose the app service after all of this application is created.



Step 2:

Creating food delivery app in IBM cloud foundry. There is a complete solution and methodology about our food delivery application.

Some of the method functions we are using is given below:

Order Module:

execute():

Implement this method

cancleOrder():

Cancel the dish order.

confirmOrder():

Change order state from “Unexamined” to “Wait Deliver”

Product Module:

execute():

Implement this method

add():

Add dishes.

edit():

Edit current existing dish.

visible():

Start to sell a dish corresponding to stop to sale some dishes

Main methods for admin:

Login():

Customer logs in the website when purchasing dishes.

addUser():

Customer registers in to the website.

getUserByName():

Query the user by name in order to validate whether the user name exists or not

getAllFood():

Get all dishes information for displaying on the website public page.

addCartInformation():

Add dish amount and dish id to shopping cart.

findOrderByOrderId():

Find order by order id, return type is list.

getFoodById():

Get Food Entity according to food id.

getFoodOrderByOrderId():

Get food order according to order id.

Main methods for customers:

login():

Customer logs in the Android application platform to purchase dishes.

addUser():

Customer registers in the Android application before purchasing dishes.

addOrder():

Insert customers deliver information to the database.

getAllFood2():

Return all dishes information from website.

findOrderByOrderId():

Get some order according to order id.