Phase Zero - Product features

Group Members: Arman Eghlimi, Fatemeh Azami, Ali Behrad

April 15, 2023

In this section of project , we introduce some suggested elements to build certain app features such as algorithms , maps as theoretical ones and classes , methods , fields as programming elements.

As the first part of any application , the identity validation , he/she should be able verify his identity and the role. roles are admin , costumer , delivery person , shop owner etc.
Authorization class : this class makes us able to verify the identity of the person logging in. this class is connected to the database to authorize the user. Sign up class : if the person could not be authorized successfully in the previous section , he will be prompted with the sign up section which is the service of this class. this class has some usual fields like username , password , phone number . it then , records these into the database and also gives the person a random number as its ID. also its suggested to save the current state of user in the database (logged in or not). then the user should be able to specify his role in the app. this also will be recorded in the database.
note that these two classes will record everything in the database and other classes will read the data from it. so as we make the information providing section stronger , it'll make the future processes much easier.

Before diving into the classes section , let's discuss about the development hierarchy of the application. the preferred development method is making the classes as separate as possible in order to reduce the coupling in the app. for example the costumer class and the owner class will not be related by inheritance but the relation will be controlled with some methods and object creation maximum. it helps the maintenance of the app much easier. for this purpose , using different data types like json , maps etc can be useful.
also everything here is done using database literally and data storing elements.

Here we get to the classes part :

1) **Admin class** : this class everything about every person using the app overlooking his role. however this class breaks into some branches that give away some of the required data to specific classes to use. like it gives the location of the costumer saved in the database to the owner (restaurant) class and delivery person or serves the amount of coupon every costumer possesses to the restaurant class. every class in this project will inherit from it.

2) **Owner (restaurant) class** : the owner should be able to specify his restaurant's food , beverages , deserts etc menu. for this purpose it's possible to save all these info in a map with the serving types as keys. if wanted to save all the info about a kind of food in one place , using a json file can be very useful. the meal can be the main key and all the attributes be the values. this will result in possessing a info tree about the restaurant's servings. the owner has to provide the restaurant's precise location on a map that will be prompted to. another field should be the geographical x-y position of the restaurant used in path-finder classes.
one of the most important things that should be provided is the *orders per person*. it means that the owner should be able to control the orders and sanitize their delivery processes. *per person* means that the owner should know who ordered what which gets accomplished by their id.
this class can use another json file called **costumer data**. in this json file , we store the orders , whether he wants the delivery to be express or not , whether he has a coupon or not , his location and so on. to get some of these costumer-related info , like coupons , we can ask the admin class that literally knows everything about the user! also his rating can be stored in this file.
it's also recommended that the app possesses a food recommendation system that recommends food based on several properties like the rating of restaurant , the rating of food , price etc. its also possible that the owner class inherits from the costumer class to override some methods and change some values in it. not to mention that this class inherits from the Admin class.

3) **Costumer class** : this class uses the data of delivery class and owner class and has some info taken from the admin class about the currently logged in person. as example the owner class will represent the menu so that the costumer can choose from and the delivery class will give info about the position of the delivery guy and remaining time. class also shares some info with other classes like the rating , if he cancels the order and so on. this class as usual will inherit from the admin class but for furthermore capabilities , its

possible to inherit from the delivery class and also be the parent of suggestion system.

4) **Delivery class** : the delivery class should know about the location of the costumer which will be given from the owner , the type of delivery (express or not) etc. this class also should be able to estimate the time and also should represent the remaining time. this is the only class uses the mapping algorithms. will be connected to admin, owner and possibly costumer.

5) **End Notes of Classes** : As stated before , there should be a suggestion system. this system also should have access to the costumers recent purchases to suggest related food types to user. if the system be mobilized with multiple optimization algorithms , such as lowest-price recommendation , fastest route finding etc , and uses their results to suggest the best option , users will be much happier!

Mapping algorithms : for this purpose , modeling the city as a graph can be very useful like the buildings as nodes and roads as branches. then using some graph-related mathematical theories can help us optimize the performance of the app. using APIs can be useful too. using something like Google Maps API can make the life easier as you don't need to implement many pre-implemented things.

If wanted to state the whole app in words , we can say that every single class that offers a service , is connected to the root class of the app which is admin class and admin class itself controls everything the has direct access to the entire database and is deep down the product to not to be accessed easily which is a security feature. other classes do have some connections with each other but admin also controls them and saves necessary info.

Author : Ali Behrad