Daniel Banuelos

3/2/20

CPSC 362-03

Professor Velasco

Assignment 2 Deliverables

Services App

**Services**

The problem at hand is to create an app that proves services for people whenever they desire. The app must implement an ordering system, and along with that the software must keep track of the users’ history and use that data to predict future services that the user may be interested in.

**Three Features**

The app that I designed provides food and drinks as the main items the customer can order. Three unique features that I added to this app are: health tracking/monitoring. Essentially this feature will log what the user has been ordering from the app, and if there is a consistent amount of “unhealthy” purchases, the app will suggest something “healthier”. This feature can be disabled. This feature could possibly go more in depth and give the customer nutritional details of what they’ve been consuming. The second feature allows the user to customize their search results such as: mile radius (from current location or a select location), price (ordering from least to greatest), type of restaurant (fast food, high-end, etc.), and lastly the type of item (drink, food, desert). This feature will give the customer the ability to narrow down their search to fulfill their desires. The final feature provided to the user is the ability to favorite establishments and or universal items, and they can also narrow their searches via their favorites.

**Flesh out the problem**

From the problem domain I was able to flesh out some of the problems we will encounter, and they are listed as follows:

* A service a user can use to order items
* The app uses predictive technology (by using user activity)
* Log user activity (order history)
* Access to maps
* Access to restaurant information

**Requirement Engineering**

In order to retrieve the necessary requirements for building this software I used the Elaboration technique specified in the textbook to specify what is needed to tackle each problem. I listed every aspect I think we will need as follows:

* User Database: log all necessary information about the user (id, password, name, email)
  + Tables to log Order History
  + Tables to log personal info like weight (for health monitoring), location, and bank info
* Restaurant Database/API: get all the information from the Restaurants like meal items, prices and nutritional facts
* Map API: to utilize map and location services.

**Two Requirement Models**

The two models that I used to define the requirements are:

* Use-Case Model: Overall flow of software
* UML Class Diagram: Ordering flow

**Three Architecture Implementations**

I used the following Architecture patterns/styles

* Data Flow Architecture
  + Pros: easy to follow, process control
  + Cons: vague, not defined in what is needed
* Layered Architecture
  + Pros: scalability, flexibility, and enables distinct segments to work on.
  + Cons: segments can be difficult to associate, segments don’t provide a lot of detail, changes to lower levels alters the rest of the level above it.
* Object-Oriented Architecture
  + Pros: can speed up development, improve software maintainability,
  + Cons: steep learning curve and can be difficult to follow.

**Final Architecture and Justification**

After implementing each of the three architectures and accessing the pros and cons of each, I decided that process control was the better architecture because of readability. I found it easier to create the diagrams and understand where the information is coming from and really going through the steps from the customers’ point of view. I think it will be more beneficial for going forward in the creation of the software, because if there needs to be any changes, we can add another “pipe” and “process” in the mix without disturbing the flow too much.