

Assignment 1

Ashutosh Kumar & Neeraj Chowdary Mamillapalli

1) Discuss your initial thoughts in detail on how you will design this application?

The requirements are mentioned quite clearly, however, some clarification about the price calculation will be needed from the partner. While that happens, we can create the front end of the app, because we already know how it needs to be designed. We need a landing page, a login page (which can be integrated with the landing page), a sign up page, client profile page, and a couple pages for the fuel price quotation form/history.

The backend of this webapp should be quite straightforward. First we can implement the logging in part, along with the registration/sign up portion and test it. After that, the profile page can be linked, which will show the information about the client stored in some database—this will require SQL. Fuel quotation will look at various factors and calculate the price when prompted by the client and display. We will also store this information along with the date and the time of when this quotation was requested so that the client can view it.

2) Discuss what development methodology you will use and why? (2 points)

Considering we have all the information required to build this web app, we can use the Waterfall Model by designing this app on paper first, then assigning different team members with different tasks and finally assembling the whole app together for the testing and deployment.

However, it might be more practical for us to choose to go with a spiral or scrum development model as we can implement a few features, assemble the app, test the features implemented, discuss what needs to be changed and then continue with implementing more features.

3) Provide high level design / architecture of your solution that you are proposing? (6 points)

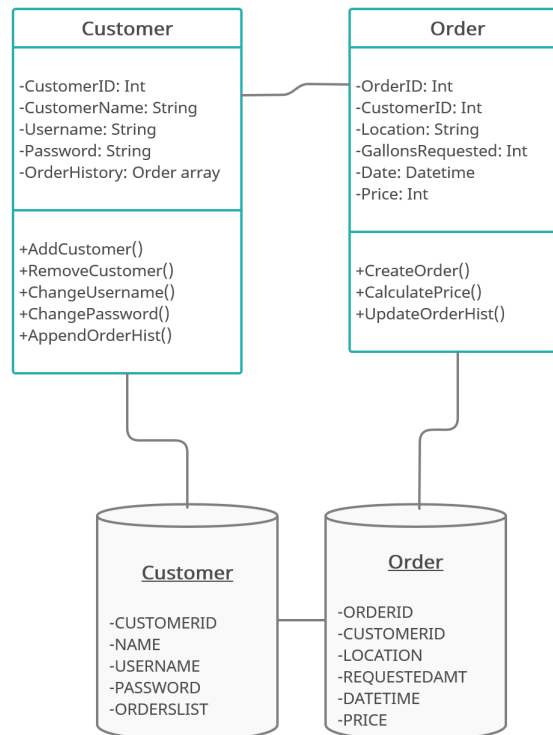
We plan on first creating the frontend of the app; We will need to create a landing page, where people can login/register. Additionally, profile pages where the users can add their information will be created. The frontend of this app is straightforward, requiring only a few pages—most of the work is in the backend.

For the backend, we will be creating a few classes—one for the customer, another for the order being placed, and any more as and when the requirement arises. The customer class will hold a unique customer identification, name, username, password, and their order history. There will be methods to add/update/remove information from the customer objects. The order class will hold a unique order ID, along with the customer ID of the customer who has ordered, the location of where the order was

placed, date and time, price, and the amount of fuel requested.

All the data will need to be stored in a database using SQL. The customer ID and the order ID will act as the primary keys of the two tables in the database, with the customer ID playing the role of the secondary key in the order table, in order to link the two tables. To make the relationship between the two tables in both directions, we will need to maintain a list of order ids in the customer table as well.

The UML diagram below (and also attached as a separate file) show the aforementioned details.



Group Member Name	What is your contribution?	Discussion Notes
1. Ashutosh Kumar	Created UML diagram, discussed how classes would need to be implemented, various variables required, as well as the database tables required and the frontend.	We need at least two classes—one for the customer and another for the order. Corresponding tables in the database are required. The front end needs only a few pages.
2. Neeraj Chowdary Mamillapalli	Discussed the implementation of classes, details about the	The front end pages required are: one each for logging

	database tables, and the frontend pages needed.	in/registration, profile page, quotation form, order history, and result page. MySQL or PostgreSQL are good contenders.
--	---	---