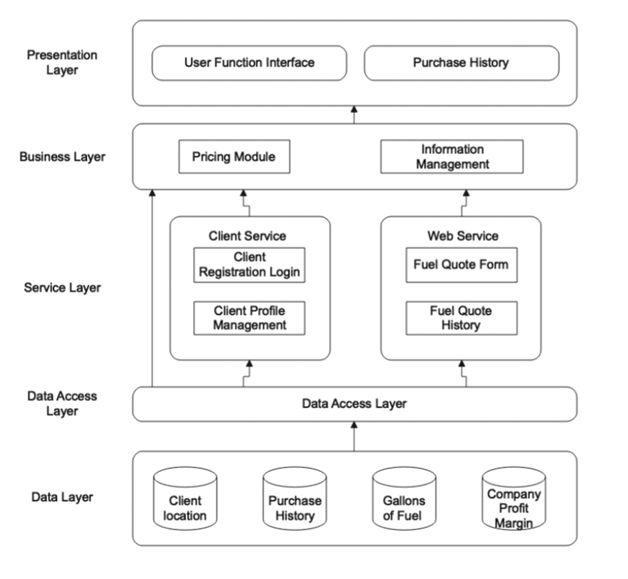
1. Discuss your initial thoughts in detail on how you will design this application? (2 points)

* Application should have a user login interface and profile, 2 pages (one page for the login and second page for the data)
  + The login page should have methods for logging in, logging out and registering for an account
    - Login(user\_name, password) - Should take 2 arguments, Username and Password. If Username and Password matches an existing Username and Password in the database, grants access to the account of that client.
    - Logout() - No arguments. Only accessible if the client is already logged in. Logs out of the current client’s account, removing access to that user’s account until logged back in.
    - Register(user\_name, password) - Should take 2 arguments, Username and Password. First check if the username that the user entered exists, if true: display “Username already exists. Please try a different username.” Else, Create a new Client account with the credentials of (user\_name, password).
  + The data page will have the rest of the functionality that the client has requested of us. This includes: Client Profile Management, Fuel Quote Form, Pricing Module, Fuel Quote History.
    - ProfileManagement() - No arguments. This method will display the description of the client’s information. It will display username, (hidden) password, location, history, gallons requested, and profit margin
    - FuelQuoteForm() - No arguments. A report version of the Pricing module, which displays the prospective price for the client with their current information. This will also save a version of this form to the database so that it can later be accessed by FuelQuoteHistory()
    - PricingModule(location, history, gallonsRequested, profitMargin) - 4 arguments each representing the information of the client. This module will then calculate the price for a client with those arguments. This should be a module, meaning that the values can be input and would not be saved.
    - FuelQuoteHistory() - No arguments. Accesses the database and retrieves all reports that have been made by FuelQuoteForm().
* Frontend: HTML, JavaScript, CSS
  + These are very popular and reliable languages for designing the frontend of the website. JavaScript is for creating the scripts for running the page. CSS is for the styling of the page and HTML will be for the actual contents of the page and utilizing the CSS and JavaScript.
* Backend: Python, MongoDB (to store information from client profile)

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

* We will use agile methodology to allow us to build the application in parts and assign smaller tasks to each of our team members.
* Agile methodology with TDD will be most optimal to use because it gives the opportunity to clean our code with multiple test cases. It is a modern model that has been proven effective. The iterative components of the methodology can let us have multiple versions as we progress as well.

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



4.

| Name | Contribution |  |
| --- | --- | --- |
| Leena Alafifi | Question 3, Question 2 and discussions deciding what to use for design/methodology |  |
| Younus Mustafa | Brainstorm and discussions regarding methodology and technology to use |  |
| Nicholas Hoang | Question 1, Question 2, and discussions for design/methodology |  |