My initial thoughts before starting the project stems from real world development projects with budgets and aesthetics in place but for simplicities sake for this project the main concerns I will have to address are the tools we will be using and the amount of time we have during the development process until we have to push a product out for production. Some points are going to be for both front and back-end development; which approach are we going to go with and what language as well as server are we going to use for our application. When planning to take on this project we will have to consider the workload so organizing tasks to members and helping to solve problems with each other given the amount of time we have. Following up includes testing as when we test, we have to give ourselves sufficient time for error as anything can go wrong. From there we can potentially push the product out and monitor the changes and activity.

As far as methodology goes, we will be using functional Rapid application development as we know the requirements of the application we are developing and we are a small team or pair developing a project that isn’t as big or complex so this method fits our projects development method the best as we are also limited on the amount of time, we have to push something out.

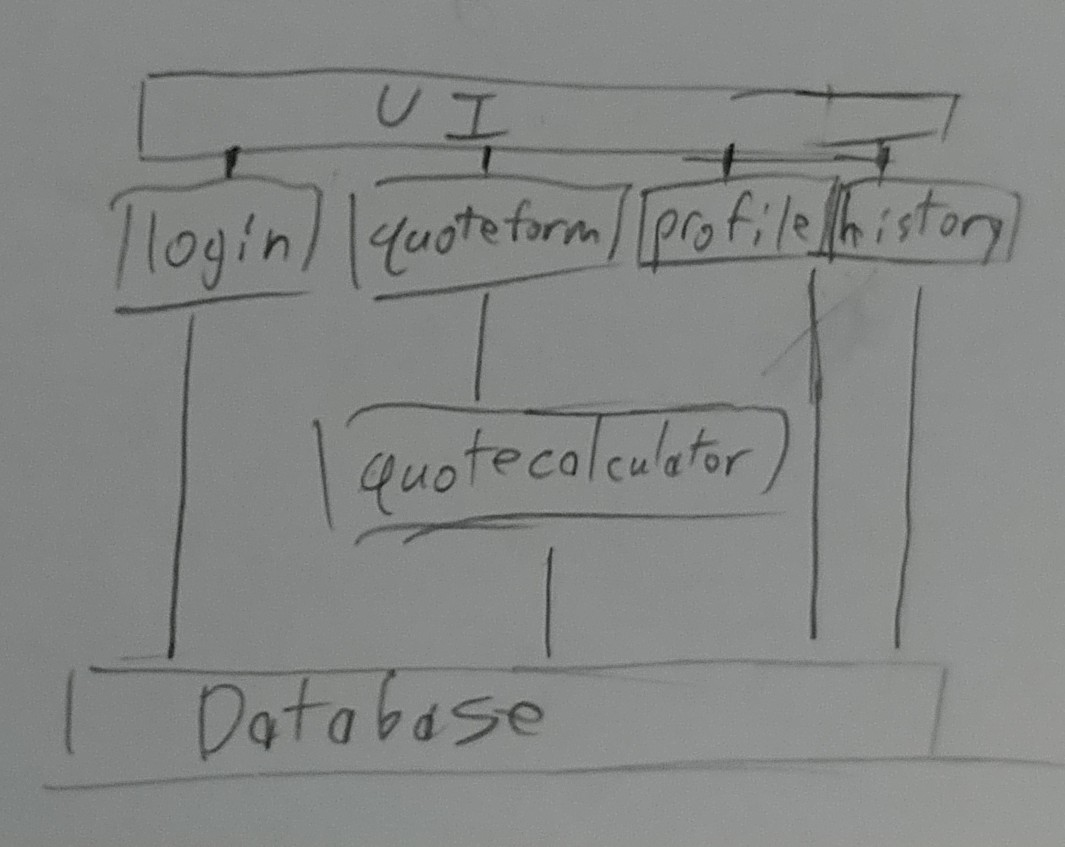
Our high-level design will look something like this:

1. Project Overview

Objective

to build a software application that will predict the rate of the fuel based on client location, history, amount, and company profit margin.

2. Technical Design Diagram



3.1 Servers

* The server we will probably use will be something mainstream along the lines of Azure or something easy to use that we can link with our GitHub.

**3.2 Access Requirements**

* The number of users that can access the app include the dev’s, the clients, the admins, and the employees if any.
* Any user communities can access it as well if they wish to such as departments that the app is tailored to.

3.3 Databases

We will use either a sql or oracle server for simplicity’s sake as the servers are easily manageable.

4. Resource Needs

* The only thing that is needed are the group members as well as a working operating system.

**5. Assumptions**

I will assume that setting a balanced work load for the group members will be difficult as well as being able to push things out on time.

**6. Concerns and Issues**

Things that will affect our work are the time and work load that will need to be under taken given both variables at play.

https://github.com/BryanTieu/SoftwareGroupProject

Workload:  Lyle Costo: designed and created the UML Diagram and the High-Level Architecture

Bryan Tieu: Wrote the documents and designed the high-level design.