Final report

1. The project accomplished

- Look up nearby gas prices in gas stations according to user's location
- Find the best driving car routine from user's location to the gas station
- Given user's location and his/her destination, calculate and schedule the optimal routine for user to go and fuel the gas on the way. The optimal routine is the minimal cost.
- Collect historical gas prices and predict the tendency of gas prices in each gas stations

2. The usefulness of this project

We did some research and found that most gas price website just show the price of a certain area on map. And based on our own experience, we need a website to help us find a best gas station depending on current location of users and gas price.

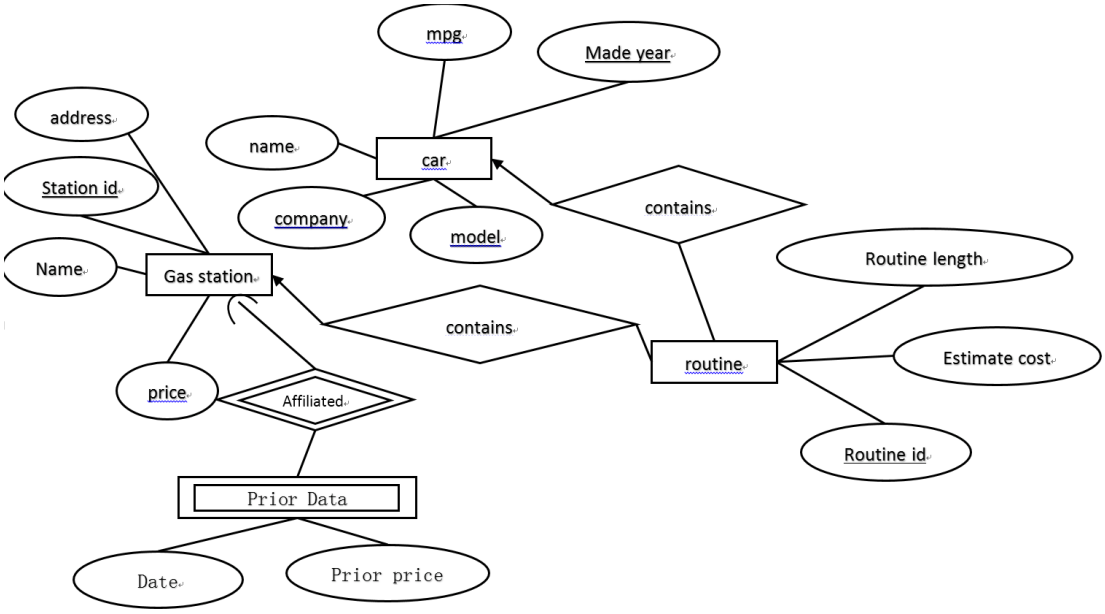
Sometimes, we want to find out the cheapest gas prices near our location.

Sometimes, we want to go to somewhere and fuel the gas on the way. How to minimize the cost?

Sometimes, we may not need to fuel the gas immediately, we can wait some days to find out the cheapest gas prices. An application is needed to predict the rises and falls of gas prices.

Our application can meet all the requirement above.

3. Data in our database



Gas station (Station_id, address, name, price)

Car (company, model, Made year, mpg, name)

Routine (Routine_id, Estimate cost, Routine length, Gas Station id (foreign key), CarCompany (foreign key), CarModel (foreign key), CarMadeYear (foreign key))

PriorData (Gas Station id (foreign key), Date, Prior Price)

Collecting data:

Use PythonScrapy to scrawl real data from websites.

4. Functionality of your application

- Users add, edit or delete car information
  - MySQL database operation
- Look up nearby gas prices in gas stations according to user's location
  - Locate user's location on Google Map.
  - Search database and show nearby gas stations on Google Map.
- Find the best gas station (with minimal cost) and show route on Google Map
  - Calculate the cost to fuel gas for each gas station
  - Find the best gas station and schedule the routine from current location to there
- Given destination, find out a routine from current location to gas station and then finally to the destination with minimal cost
  - Design algorithm to find optimal routine
  - Calculate the cost of each routine and render routine on Google Map
- Predict falls and rises of gas prices

- Design algorithm to process historical gas price data and predict the tendency of gas prices
- Scrawl real gas price data from websites
  - Set up scheduler to update database everyday

5. Explain one basic function

User select a car information record to delete:

- User chooses one car information record to delete

Car Information Form

Choice	Company	Model	MadeYear	MPG	Location
○	BMW	X5	2004	38	61801
◉	123	123	123	123	61801

Add Edit Delete Search Trip Planner Predict

- Backend function receives the id of record that user wants to delete
- Execute delete sentence in MySQL

```
if 'delete' in request.POST:
    if 'choice' in request.POST:
        #Car.objects.get(pk=request.POST['choice']).delete()
        cursor = connection.cursor()
        cursor.execute ("""DELETE FROM gm_car WHERE id=%s""", (request.POST['choice']))
```

- The website automatically refresh the interface, and the record has been deleted

Car Information Form

Choice	Company	Model	MadeYear	MPG	Location
○	BMW	X5	2004	38	61801

Add Edit Delete Search Trip Planner Predict

6. Advanced functions:

1)

Find the best gas station and find out the routine

Users want to fuel gas but they don't know which gas station has lowest gas prices and they don't know how to go to the gas station. What's more, they want the gas station is not too far from them because a long trip not only cost gas but also make them boring.

Our application implements the functionality to find out the best gas station considering the factor include: gas prices, distance between gas station and users current location, the MPG (miles per gallon) of users' cars. Our application will show users the cost to fuel

the gas and provide a fastest routine from user location to gas station.

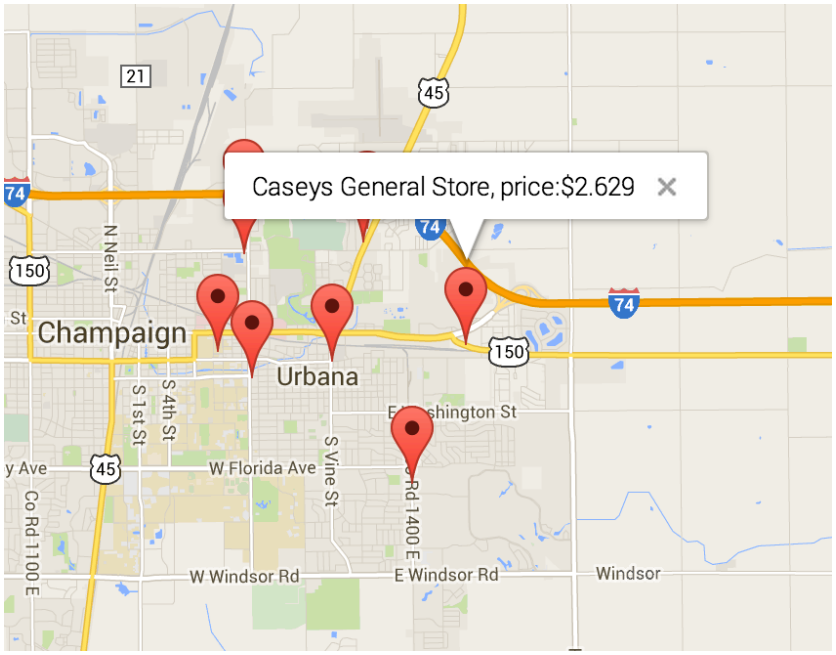


Figure 1 Show nearby gas stations

Set Pi represents the gas prices for gas station i.

Set Di represents the distance between user's current location and gas station i

Set MPG represents the miles per gallon for users car

Set Ci represents the cost to fuel gas at station i

Our goal is to find out: $Gas\ station\ i, that\ C_i = \min(C_i) = \min(\frac{P_i * D_i}{MPG})$

Di can be accessed by calling Google API: convert each point into corresponding longitude and latitude and then we can calculate the distance.

Also calling Google API to find out the best routine once we determine the best gas station.

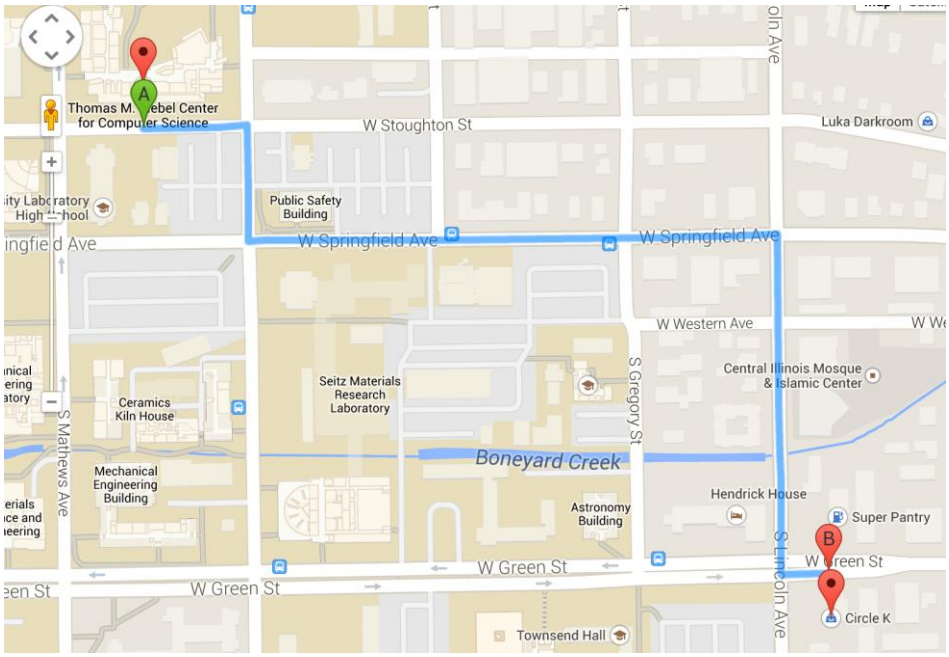


Figure 2 Find out the best gas station and routine to get there

2)

Scrawl real data and predict rises and falls of gas prices

Maybe users don't want to fuel gas immediately, they want to fuel the gas when the price falls. So, they care more about the tendency of gas prices.
We not only scrawl current gas price to find out best gas station, but also collect historical gas prices data to analyze the tendency of gas prices.

The gas prices can be affected by season changes (like Atlantic hurricane) and incidents (like Mideast war). And the gas price for a certain gas station can not change a lot in a short time.

We collect data about monthly average gas prices in US and the previous data of gas stations.

To predict the gas prices in December, 2014:

Because the gas prices influenced by season changes, so price in same season has similar value. We weight the average prices in December, 2013, Jan. 2014, Feb, 2014 equals to 1.

Because the gas prices influenced by some incidents. So we weight the average price in recent month equals to 1.

We weight the average prices of rest months in this year equals 0.5. Then we normalize the weight so that $\sum_{i=1}^{k} w_i = 1$

We get the estimated price as (di is the price for ith month)

$$A = \sum_{i=1}^{k} w_i d_i$$

We also collect the gas price of gas stations in recent days. Set B as average price of recent days

So the final estimate price is $P = \alpha A + \beta B \; where \; \alpha = 0.8, \beta = 0.2$

If the current price is larger than P, then the price is likely to fall and vise versa.

We put our prediction result at the location of gas station on Google Map so user can easily find out the rises and falls of gas prices. ('+' means that the price will arise and '- ' means that the price will decrease)

Figure 3 Predict the rises and falls of gas prices

3)

User provides current location and his/her destination, the gasmanager will find optimal routine so that the user can fuel the gas on the way and the cost is minimized.

If we take each city as a node and each pair of cities in neighbor will have an edge to connect each other. So the process of user going from current location to destination is

like departing start node, and going to the destination node along the edge. Since we most likely go to gas station only once, we enumerate each gas station belonging to the nodes that we will pass by. And find out the optimize routine which has minimal cost.

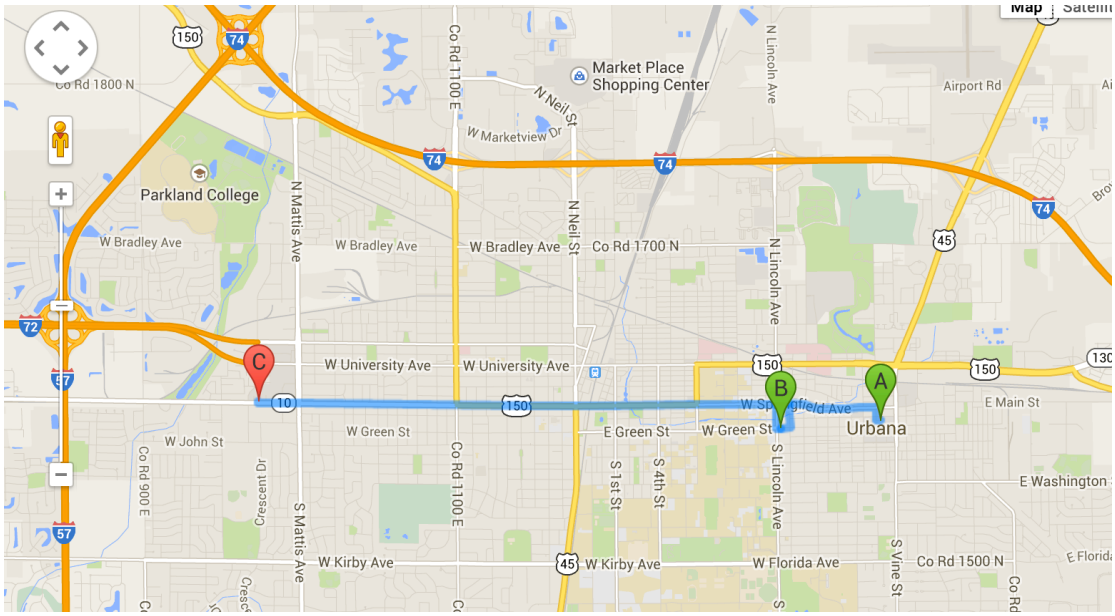


Figure 4 Find out optimize routine to fuel gas and go to destination

7. Technical challenge

At first, we try to set up the django framework on CPanel, but it seems that the CPanel doesn't support django framework so well. And the space is limited on the server.
We solve this problem by set up the environment on pythonanywhere.com, which is a website support django framework well and we can build our own project at this website.

8. Work division

Back end: Ruichao Qiu
Algorithm design and implementation, webserver setup and scripting, database management

Front end: Tong Zhang

Webpage design, data parsing and demonstration in webpage, application of Google Map API