My Travel Buddy

Avery, Brogan M

Brogan Avery
March 16, 2021
Final Project Proposal

Project Proposal

Problem Introduction: After graduation, I plan to take a lot of road trips and work from the road often. However, it is difficult to decide where to go and what the most efficient (and most enjoyable) way to go about planning for a road trip is. Additionally, since I plan to camp in my car as much as possible, I need to maximize my space and only bring the things I need. I also know I will need to spend some money on new equipment/ set up, but I need to be mindful of my budget. So the main things I need are way to help me plan road trips with multiple destinations, a way to plan my packing, and a way to monitor / plan out my budget and spending.

Proposed Project:

The app is composed of two main parts.

1. Route Builder
   Here, I will be able to enter a location, destination, etc. that I would like to include on my "bucket list of places to go" and then have that information be written and saved to a CSV file. There will also be a feature to build a map, which will show what the best order to travel to each place is (with the goal of not wasting gas or money). This would be similar to the concept of google maps or some other navigation app/system. I may also implement an API for this part.
   - This part would likely implement some sort of graph structure and potentially a priority queue and easily, a sort method could be beneficial.(TBD)
2. Shopping / Packing List
   This part of the application will let me enter in information for items I potentially would like to pack as well as some accompanying information such as, if I already have the item or if I will need to buy it (and the price of the item). Another piece of accompanying information with each item I would like to add to the packing list is a priority level indicating that the item is essential, important, less important, only a luxury, etc.
   - A priority queue will be used to organize how important each item on my list is and will help me narrow down what to take.

Deadlines

Due for/ complete by:
Module 9, **March 24** : Outline / some pseudocode
Module 10, **March 31**: Pseudocode and some early code
Module 11, **April 07**: Rough working model
Module 12, **April 14**: Nearly finished version
Module 13, **April 21**: Final code
Module 14, **April 28**: Testing and writing actual tests
Module 15, **May 05**: Writing report and slide and presentation and project final touches

Time Log

March 20 – Created a wire frame paper version of a website that this project application could be translated into at a web design/ dev scope.
March 23 – Create a basic full-ish project outline and an outline of the project files and organization structure
March 24- did more research on the data structures for my program
March 25- created an er diagram of all files and the data bases and classes
March 26 – created files for the classes
April 5 – set up a basic Django site
April 7-9 – worked on all the site functionality of connecting parts of the back to the front and setting up the database tables
April 9- worked on setting up the data structures and figuring out how I want to approach that problem from the travel site.

- I will likely make a map/ graph where every node is connected directly to every other node. Then I will use simple coordinates to calculate the distances. Not as practice as actual road distance but will still be a pretty close est. I could not find any distance APIs that were free.

April 10 – Worked on graph and matrix classes to make them compatible with the “nodes” in this program.
April 11 – requested help on the trip planner algorithm
April 13- worked on the form for the trip planner
April 15- Created the priority queue and associated code. Also created a method for the sort function to sort by price or item ID.
April 16- finished the Graph data Structures.
April 18 – worked on Django unit tests
April 20 – worked on Django unit tests

User Manual

For the general user:

This program is designed to be very intuitive. All forms, buttons and page text should direct you what to do. Any errors or Page Not Found errors can simply be resolved by returning to the previous web page in the browser.

For Instructor:

I will assume no prior experience with Django.

When inside of the Django Website file, there are several folder. The only 3 that are super relevant for the purpose of examining my code are the folder titled homePage, tripPlanner, and packingList. These each represent their own application. The homepage is also mostly irrelevant, but it is used as a piece to let the user navigate from one app to the other. Within each of those folders, the files that are relevant will be titled:
-models : a special way to connect classes to a database in Django
-tests: unit tests
-views: the actual programing logic
-template folder: contains HTML files that are mostly written in HTML with some special python Django decorations to display information from the view file
- there will also be several supporting files for each app that have to deal with the data structures such as a class file for the priority queue.

All of the other folders and files in this project are needed for various Django support. Here is a great tutorial for the basics of the framework
https://docs.djangoproject.com/en/3.2/intro/tutorial01/ .

Lessons Learned

I accomplished everything I wanted to do in the scope of this project with a few modifications. One of the biggest challenges I faced in this project turned out to be a problem no one has a solution to. In doing more research on the traveling salesman problem, I realized this is probably true more often than not, where there is no answer to a solution yet and that is why developers and engineers still have jobs. It also made me more aware that there are probably not very many cases in life where we apply an exact data structure. The is usually "custom" data structures that might use elements of various data structures of be derived from a known use. Another big problem is Unit Testing is really difficult to do with Django.

Conclusion

It is very difficult to apply the MERUSE principals to a T or to a ridged degree on most real projects, especially when then get large, require GUIs, or use frameworks. There is a lot of additional files that I as the program have no control over when it comes to using frameworks and required dependencies. However, To accommodate modularity, I am still able to use classes and functions including nested functions. As far as efficiency goes, again, this would require far more advanced knowledge of frameworks than what can be expected in this class. The algorithms I use are not an issue, but I have no ability to speak for all of the dependencies required by GUIs and frameworks. With robustness, again, this is not a practical question to ask in this situation since all of my files are based on local directories and I do not have the ability to use live server for the scope of this project. However, the data structure parts of the program will have no issue in this department. I find my program to be very direct, intuitive, and user friendly. My code is mostly readable, if I had the time to do more research on better ways to organize Django views, I could improve upon the issues it creates, again, this is more complex than is necessary for the scope of the project. And as for elegancy, once again, this is limited by the required GUI/ framework aspect of this project.

This product is currently at a state that lets users add and edit items on a packing list. It also provides a rage of viewing options such as sorting by ID, price, only displaying free items, and only displaying the top ten priority items. The product also lets users keep track of a budget. Additionally, this product gives the user the ability to make a bucket list of locations that they want to travel to as well as edit those locations. It then provides the feature to look up the distance between two locations.

Future additions to this project could include a more advanced route planner with further investigation into the traveling salesman problem. Ideally I would like to integrate it with APIs, however most were too complicated or not free for the scope of this class. I could also add more functionality to both apps to let the user add more customizations. There is of course a lot I could do with the front end design and hooking it to a live server as well.