Revised Project Proposal

Group name: Midnight Masterminds
Team member: Yi-Ju Tsai, Ziyao Zhang, Yang Zhou, Naveen Muthukumar, Sanjit Arunkumar

User needs

Most people purchase flight tickets online nowaday, and it's a common situation that people would like to browse multiple tickets and do a comparison (between price, schedule, etc) before making the final purchasing decision. Some people may also want to wait for a certain time until the price drops. On these occasions, popular airline tickets selling platforms, such as Google Flight, Expedia and Hopper, still lack flexibility. For example, as users quit the web page and refresh / reopen it, they need to re-enter the search criterias if they would like to look up for more flight options. This could be annoying and a waste of time. In addition, current platforms provide options for users to track a ticket and receive email notifications whenever there is a price change, but those emails can be easily ignored by the users. Some users might receive hundreds of emails every day and it is not user-friendly if they have to filter from hundreds of advertisement emails and pay extra attention to those flight related messages. Currently platforms also do not support tracking / notifying of multiple trips (not a trip of multiple stops) which is something that can be improved on.

Existing solutions

- Expedia
    - Allows search by specific criteria, but requires page refresh for price updates.
    - Users have to re-enter criteria again once the page is closed / refreshed.
- Google Flights
    - Similarly allows users to search for specific criteria, but need to re-enter search information once the page is closed / refreshed.
    - Users receive email notification for price changes, but are easy to ignore
    - Does not support tickets temporarily saved into a collection.
    - Users can not collect and plan multiple flights.
- Hopper
    - Lets you search for flights and "watch" them.
    - Users receive notifications for price changes.
    - Doesn't let users select many filters.

Incorporation of feedback

After the presentation of our initial project proposal, we went through the feedback and summarized things we need to improve on as follows:

- We did not differentiate our application from the existing flight tracking applications. Google Flight and Hopper already have some of the features we described (such as email notification, track flight change). Need some more unique functionalities.
- Better to be more detailed on how a user could possibly interact with the app. Include more detailed sketches.
- Regarding feasibility: are there any restrictions on the APIs? How much data will be needed to find the optimal price? How much the price changes would the user be notified?

We will be addressing these issues in the next few sections.

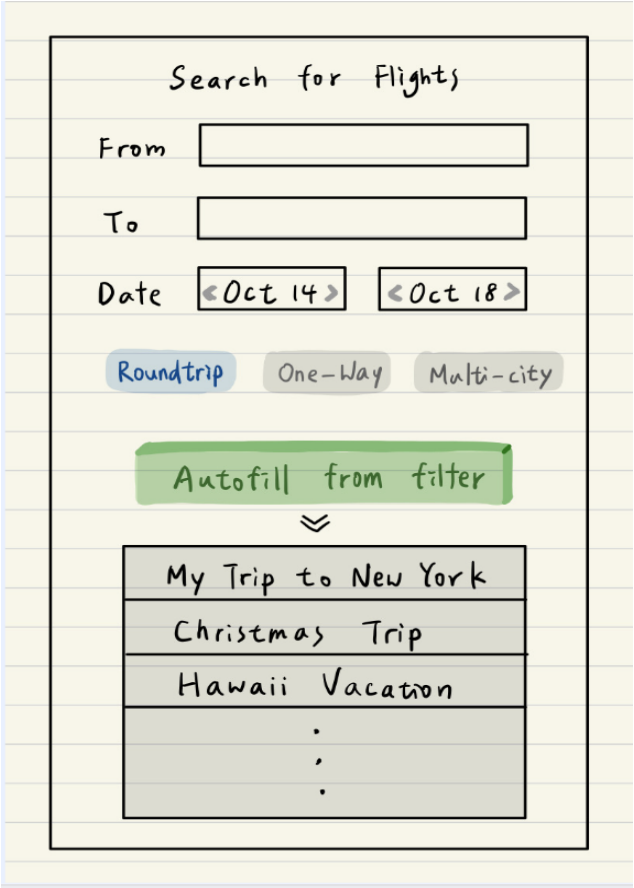
Project idea (revised)

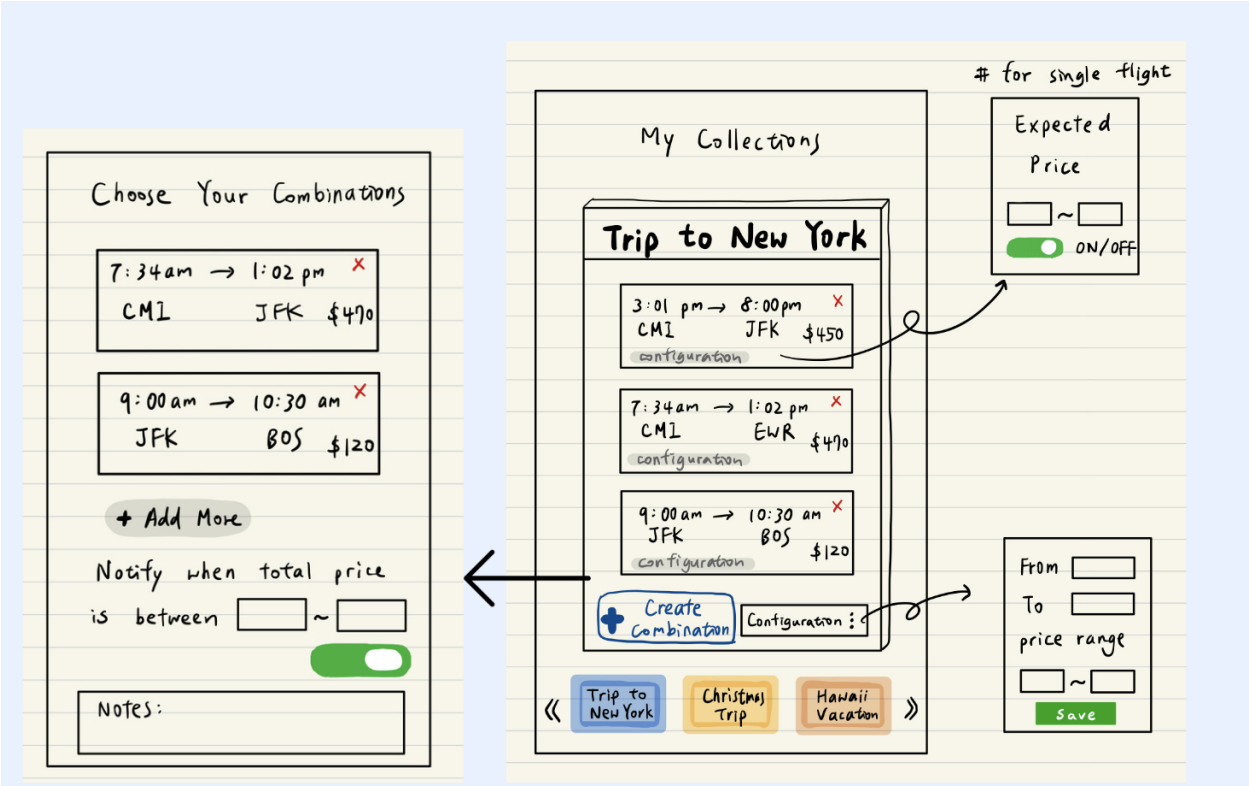
Our goal is to provide a more customized and convenient ticket purchasing experience. Compared with traditional flight tracking applications, our application contains new features that enhance the ticket filtering and management process. As users are planning for trips, they have the option to create folders and collect flights they are interested in into specific folders. For example, if I'm planning to travel to New York and I've started to browse flight tickets from Champaign to New York, I can create a folder called "My trip to New York" and add tickets that I would like to temporarily save for future consideration into this folder. Then, I can revisit this folder to see all the flights I've collected and compare their price, departure time, etc.

Each folder will also contain a pre-saved filter that users can configure. Next time, if they want to look up for more options for a trip, they can simply select the pre-saved filter associated with an existing folder. The system will autofill the searching criteria and let users skip the annoying process of re-entering search information.

As the user clicks into a folder, he/she should be able to see a list of tickets collected for this trip. For each ticket, the user can set a price range themselves, and whenever the price drops or rises beyond this range, the application will ring the alarm to remind (We choose alarm over email notification because email alerts are easy to ignore). An additional functionality we are considering is that if the travel plan contains multiple trips, users are able to select their preferred combination and the range of the total price of the combination beyond which they would like to be notified. Using the previous

example, if after visiting New York I'm planning to go to Boston, I can search for flights from New York to Boston and save them into the same folder. When there is a certain combination of flights (one from Champaign to New York, one from New York to Boston) I'm particularly interested in, I can set a price threshold on this combination so that when the total price of the two flights reaches this threshold I will be notified by the system.





Here is another example for our combination feature:

Suppose a businessman, Jerry, wants to travel to Los Angeles and New York for business purposes. He has a strict time schedule, so he has to confirm the 2 flights (from Champaign to Los Angeles and from Los Angeles to New York) at the same time. There are 3 tickets from Champaign to LA and 2 tickets from LA to New York which is suitable for his schedule. He wants to buy 1 ticket from each trip and the total price should be under 1000 dollars.

Now, Jerry could use our collection! He could add the 5 tickets to a new collection called “Business Trip”. In this “Business Trip” collection, Jerry could plan his flights freely. He could combine any ticket from Champaign to LA with any ticket from LA to New York and set a threshold for his combination. For instance, when the total price of 2 tickets in this combination is under 1000 dollars, Jerry will get an alarm from our app and buy the 2 tickets at the same time to ensure a smooth trip.

User Audiences

Any passenger who has a demand for flight tickets, whether it is for travel, or business travel..., etc. For the general audiences, it doesn't make sense to keep an eye on the price change and info of a flight for a long time. However, for passengers who often need to buy flight tickets, allowing saving the filter options of flights can save their time searching for tickets. For example, if a senior executive of a company needs to visit Japan frequently, he can set Japan as the destination and save this condition in collection 1.

One of the well-known ticketing sites, Expedia, has 30% of its audience between the ages of 25 and 54 and earning more than $100,000/year. Therefore, in addition to the user's occupation, we speculate that the user's income will also affect the frequency they use this website/app, since it is obvious that flight tickets are mostly more expensive than trains or other kinds of transportation.

To be more specific, our target users can be out-of-state students who buy tickets frequently, company managers who need to travel to other branches frequently, or backpackers who really love to travel around the world.

Context

Context of the app can be for different booking scenarios. It can be for users who are planning to book a particular ticket for a specific flight and time, and they want to keep track of prices and book it when the prices reach a favorable level. The scope of our project will be limited to tracking prices for different flights based on filters as specified by the user. Locations of use could naturally be anywhere as not much attention is required when setting the filters. You could do it at home, when on a walk, while in the car, in the office discussing with colleagues while planning a trip together, etc.. Frequency of use would be based on when you get a notification, and will involve a short burst of attention until the point when the user decides to actually purchase the ticket. The app will not compete for too much attention of the user as it just serves as a reminder for ticket details.

The app is mainly useful for users who are planning to book a particular ticket for a specific flight and time, and they want to keep track of prices and book it when the prices go below a threshold level, and for frequent travelers, their previous flights can be saved, and with a click of a button, the app can keep track of those flights(very little work required)

Feasibility

The most important module in this project is to gain the information of flights. In 2018, Google ended access to the public-facing API and now only offers access for enterprise products. Fortunately, we still have many substitutes for it.

Duffel's Flights API enables everyone to access the flights' information by offering search, book, seat reservation, and paid extra additions from 150+ airlines seamlessly through a developer-friendly platform. Duffel provides the detailed official API document which is used for reference. "Built on a RESTful architecture, providing JSON responses, and using OAuth 2.0, Duffel gives developers the security and flexibility needed to make and manage booking requests with a few lines of code." Duffel's Flights API also provides client libraries in JavaScript, Python, Ruby and C# which makes it easy to build Duffel integration. Even if you do not use the programming language list above, Duffel's Flights API provides the API demo in an HTTP client like Postman which allows developers to extend these functions to the programming language they use.

Based on the Duffel's Flight API and Database (like SQL or MongoDB), we can save the filter information for different users in the database and get the corresponding flights' information through the Duffel's Flight API. Whenever users want to check their specific flights' information, we can retrieve their preferences from the database and send a request to the API to get the flights' information easily.

We also prepared a plan B to get the flights' information – aviationstack API. This API also allows users to get free, real-time flight status and global aviation data. In some specific situations, the Duffel's Flights API does not work, we can transfer our data source to aviationstack API which ensures the continuous work of our app. What's more, aviationstack API could be used as a backup which could ensure the consistency of flights' data.