Travelwise Technical Report
9/30/2020

Motivation

Travelwise is a web app that enables consumers to make ideal travel plans. Through the use of various RESTful APIs, Travelwise is an ideal platform for users to compare destinations by safety, affordability, and more. A feature of Travelwise is providing users with up-to-date statistics on COVID in a given region. Travelwise streamlines the process of making travel plans, where users can search for destinations, corresponding flights, and hotels all in the same place.

User Stories

9/29/2020

User Story 1:

- *“As a customer of Travel Wise I would want to know the number of COVID cases in the country of my destination. This is because my travel plans are now affected by the number of people infected in my destination. The effects of a greater infection rate could mean more restrictions on places I’m allowed to go as well as a higher risk of catching the disease and then having to spend 2 more unplanned weeks at my destination.“*

- Approach:
    - We will have destination city instance pages link to associated COVID stats instance pages of the country/region.
- Estimated Time:
    - 3 days
- Implementation:
    - Not implemented yet
- Actual implementation timet:
    - NA

User Story 2:

- *“As a user of Travelwise who may have a flight with layovers, I would like to know how much time there is in between connections. Information about the layovers for a flight is helpful because I may want to avoid a flight if it has layovers that are too short or too long. This information will help me weigh the pros and cons of the different flights listed on TravelWise and will improve my browsing experience.”*

- Approach:
    - Currently, travelwise displays arrival and departure times, enabling users to manually calculate layover times. However, for superb user experience, we will calculate layover times and clearly display them.
- Estimated Time:
    - 2 weeks. We first must get an operational backend to make this feature.
- Implementation:

    - Not implemented yet
- Actual implementation time:
    - NA

User Story 3:

- *"As someone who has had to travel on a tight budget before, I think in addition to having hotel and flight prices sortable, it would be nice to filter them by range. Other services like Google Flights have this feature and it would make sense to include this functionality on your website. Overall, this would make browsing easier."*

- Approach:
    - We can implement this once we have implemented our database. We will add price range as a searchable attribute, allowing users to specify a lower and upper price.
- Estimated Time:
    - 2 weeks. We first must implement an operational database to make this feature.
- Implementation:
    - Not implemented yet
- Actual implementation time:
    - NA

User Story 4:

- *"Since safety seems to be important to the application, on top of covid stats at the destination, if possible, I would like to see the covid procedures implemented by the flight company that I end up choosing. Companies that have very loose procedures may not be seeing high customer traffic right now. By providing the user this information, it covers one more area of concern regarding COVID."*

- Approach:
    - This seems like a difficult thing to implement. It is a very reasonable concern, so we will attempt to connect users to resources where they can find out this information. The obstacle is that it would be hard to scrape or store this type of information into our database. So our goal will be to be able provide external links to a flight's airline company website, where users can find such information.
- Estimated Time:
    - 3 weeks. We first must implement an operational database to make this feature and then we must find a way to associate company websites to given flights.
- Implementation:
    - Not implemented yet
- Actual implementation time:
    - NA

User Story 5:

- *“As a customer of Travelwise, I am unsure of the functionality of the dropdown under the search bar on each model page. I believe the intention is for the selected field to be the column each table is sorted by, although this is a guess. It would be beneficial to users of this site if the purpose of the dropdown was labeled, or if the default text in the dropdown indicated its purpose, such as "Order columns by:"*

- Approach:
  - We will likely change the default text to be more descriptive, such as “order by safety.”
- Estimated Time:
  - 1 day
- Implementation:
  - Not implemented yet
- Actual implementation time:
  - NA

RESTful API

For destination information (cities), we use GeoDB cities API:

- https://rapidapi.com/wirefreethought/api/geodb-cities?endpoint=5990a0b4e4b075a0d1d6da26

For destination safety scores, we use Amadeus's Safe Place API:

- https://developers.amadeus.com/self-service/category/destination-content/api-doc/safe-place-api/api-reference

For COVID statistics on given regions, we use COVID19 API:

- https://documenter.getpostman.com/view/10808728/SzS8rjbc

For flights, we use Amadeus's Flight Searcher API:

- https://developers.amadeus.com/self-service/category/air/api-doc/flight-offers-search/api-reference

For hotels, we (will) use Amadeus's Hotel Search API:

- https://developers.amadeus.com/self-service/category/hotel/api-doc/hotel-search/api-reference

Models

For this first phase, our three models are Cities, Flights, and COVID regions.

Lower safety scores correspond to a safer environment. Scores are 1-100, 1 being unlikely to suffer from the metric, to 100 being very likely to suffer from the metric.

- **Destination Cities**
  - Name
  - Country
  - Region
  - LGBTQ Safety Score
  - Medical Safety Score
  - Overall Safety Score
  - Physical Harm Safety Score
  - Political Freedom Safety Score
  - Theft Safety Score
  - Women Safety Score

- **Flights**
  - Departure Airport
  - Departure Terminal
  - Destination Airport
  - Destination Terminal
  - One-Way
  - Price
  - Seats Remaining
  - Departure Time
  - Arrival Time
- **COVID Statistics**
  - Country
  - Total Cases
  - Total Deaths
  - Total Recoveries
  - New Cases (daily)
  - New Deaths (daily)
  - New Recoveries (daily)

Tools

We are using a variety of tools:

- **Docker:** We will use docker to create a docker image that packages our tool-chain and dependencies into one convenient container.
- **Postman:** We have used Postman to scrape our initial instances of our models. We have used Postman to create a shared workspace, making it easy to collaborate with team members when working with our RESTful APIs. We also have used Postman to design Travelwise's API. Down the road, we will use Postman to further design and implement our API, once we have a more functional backend.
- **React.js:** We use React.js for our front end.
- **Bootstrap:** We use Bootstrap as our primary CSS framework.
- **Yarn:** We use Yarn, a package manager, for our React App. It makes documenting and installing our dependencies easy and streamlined.
- **Visual Studio Code (LiveShare)**: We utilize Visual Studio Code in developing our React App. We also utilize the LiveShare extension, allowing real-time code collaboration and pair programming.
- **GitLab:** We utilize Gitlab for our version control.
- **Slack:** We've integrated slack to our GitLab repo, for communication purposes and issue tracking.

Hosting

The services we used:

- **Namecheap:** We used namecheap to buy our domains because they support whois protection and have cheap domain names.
- **AWS Amplify:** AWS Amplify allows us to host websites by simply committing and pushing code to our Git Repository. The service allows us to focus on simply writing code and not having to directly interact with our build server/terminal.

- **Create-React-App:** The Create-React-App allows us to have a modern build setup with pre-set configurations that make our react app compatible with most technologies. It is officially supported by Facebook, the creators of React so it is extremely reliable.