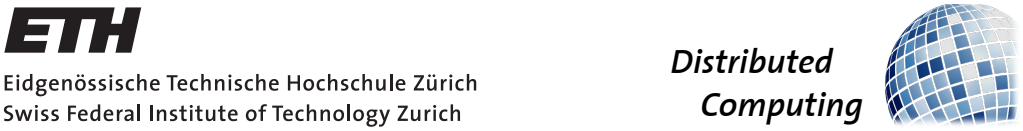

Plan My Vacation

Distributed Systems Lab Project

LastName2, FirstName2
first2.last2@xxxxx.com

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Georg Bachmeier, Gino Brunner
Prof. Dr. Roger Wattenhofer

February 28, 2017

Acknowledgements

We would like to thank Georg and Gino for helping us complete this project and Prof. Dr. Roger Wattenhofer for allowing us to pursue this project. With the weekly meetings, we ensured that a certain portion of the tasks were completed and thanks to Georg and Gino we were able to follow this weekly "sprint" type methodology of software engineering, to have an update for the application every week. Putting this together this entire API and connecting the android application was by no means an easy task and would not have been possible without the help of our supervisors.

Abstract

This project describes a novel application that aggregates data together from the most popular websites/applications on the Internet, that one would use separately, while planning a trip (For eg: Booking.com, AirBnb, TripAdvisor etc.). This project provides a framework to solve the cumbersome problem of manually planning a trip. All current applications focus on an individual aspect of trip planning and require extensive user input. With the Plan My Vacation application, user input is minimized to purely the essentials and the framework automatically provides suggestions to the user while learning from their actions. Ideally the application is aimed at users who have a specific budget or time frame in mind. The application takes into account this budget or time frame and provides the most suitable recommendations for flights, room/board and a thorough list of possible day to day activities that fit to the given time frame. Challenges include processing a large amount of data to compile the most suitable and convenient travel plan, ensuring that this data is valid, gathering this data and minimizing query time. Future improvements would include the addition of learning algorithms that observe user behaviour and optimize future trips based on past behaviour.

Contents

Chapter 1

Understanding the Problem

With the abundance of travel applications these days, end consumers have the option of planning each component of a trip with hundreds of options. For example: There are about 100 different popular applications/websites that allow a user to book flights and hotels, then there are applications that allow a user to book local transport, and mapping applications that allow users to plan routes between places that they would like to visit while travelling, i.e., each component of a trip can be planned individually if desired. This process is cumbersome and there is no application available that unifies all of these individual aspects of trip planning. Not only is there no way of making a comprehensive trip, but neither is there a way for a user to possibly know all the attractions that a new city has to offer. In this project we solve this exact problem by using data from the most popular websites, aggregating this data and then cleaning it and presenting useful and usable information to the user in a neat and concise manner such that while planning a trip, a consumer does not have to spend days researching and reading reviews and can instead plan one in a couple of minutes.

1.1 Available tools

1.1.1 Other travel applications

Skiplagged

Skiplagged is a relatively new player in the flights industry but after some research we found that the flight data that is offered by Skiplagged is easily one of the most comprehensive. Skiplagged was infact sued for undercutting flight prices by exposing the concept of "hidden cities" that airlines do not want passengers to exploit. Naturally, this is something that we would like our users to take advantage of because ideally the target users for this application are those between the ages of 18 and 50, a majority of whom would travel on a budget. Minimizing flight prices was of the highest priority while selecting a source of data for flights and Skiplagged was the best option. The advantages

and disadvantages of this API are outlined in Section 1.1.2 and 2.1.

Booking.com

Booking.com has risen to be the most popular hotel booking website, used by millions of end consumers worldwide. The data about hotels for a particular city available on Booking.com is comprehensive and exhaustive. This was naturally a choice for a data source for hotels for this project. Each hotel presents information about the hotel, location, review, photographs, rating, availability and price.

Airbnb

Airbnb is an application that aggregates home owners to rent out vacant rooms/apartments to travellers for a fee. This application has a user base of over 10 million and offers some excellent accommodation in foreign cities with trusted home owners, for excellent prices. Thus, while aggregating accommodation data, Booking.com and Airbnb were our primary and most important sources.

The highlight of our application deals with fetching and combining sparse data about attractions/popular places in a city, from different sources, into an intuitive form. The following sources are used:

FourSquare

A relatively old platform for travellers to check-in, review and rate places of interest. FourSquare exposes a powerful and exhaustive public API that is exploited here.

TripAdvisor

One of the biggest players in the attractions industry, a platform that aggregates user reviews about cities all over the world and rates destinations and trips according to popularity. The data that TripAdvisor has is extremely valuable and they do not expose this through a public API anymore and it is not possible to access their data without manually crawling the website. However, for this project we were able to gain access to their API through unconventional means.

Google Places

The largest and the most useful database of popular locations and reviews. We combine the Google Places API along with Google queries that a consumer would normally use to find places of interest that are not listed in the aforementioned sources.

Thus, by combining the 4 largest sources of attraction information, we were able to create a unique list of recommendations for the most relevant attractions.

1.1.2 Shortcomings

All of the above listed sources of information, each have their own shortcomings that are listed below. However, it is already seen that out of all the biggest and most popular travel applications, each of these applications provide a method of planning a single, individual component of a trip but not an entire comprehensive trip.

Skiplagged

SkipLagged does not have a publicly accessible API, however, they do not restrict developers from writing applications that access their website to grab information. This is exactly what has been done here. Another shortcoming is the natural User Interface of SkipLagged. Without a publicly accessible API, to actually book a flight, a web view has to be rendered inside the application, disrupting the natural flow. However, out of all available options for flight data, which was already scarce in itself, SkipLagged proved to be the best option, especially considering the ability of SkipLagged to include the "hidden cities" exploit.

Booking.com

Accessing data from Booking.com is not an easy task as they protect their valuable datasets and make it difficult for developers to gain access to this data, without paying for it in the means of a commission model. Since we do not aim to make any monetary gains out of this application, we went through the rout of SkipLagged, i.e., manually scrape the webpage and extract useful information to display inside the android application.

Airbnb

There is an unofficial wrapper for the airbnb website which works just as the Booking.com and SkipLagged wrappers, i.e., scraping the airbnb platform. However this wrapper presents data in a JSON format that is relatively easy to manipulate. The downside of this JSON data is that it is massive and unstructured.

FourSquare

The FourSquare API enforces rate limits that would pose problems when scaling. Also, the data available from FourSquare is not as comprehensive and updated as that from Google. However, it does contain some lesser known attractions that may not be visible in the Google API. Thus, it plays a crucial role in using this data in combination with the other API's.

Google Places

The Places API does provide a comprehensive list of popular attractions, however the rate limit is a major restriction and in this project, we use multiple API keys to extend the rate limit as best as possible.

1.1.3 Overview of the Plan My Vacation server side application

All of the above described API endpoints are running on a reverse proxy NGINX linux server, serving a node application for each API endpoint on different ports. The following are the categories and sample REST API endpoints for Plan My Vacation :

Four Square

- http://82.130.102.16:8991/getFourSquareCategories?lat=47.3769&lon=8.5417
- http://82.130.102.16:8991/getFourSquareTrending?lat=47.3769lon=8.5417& radius=2000
- http://82.130.102.16:8991/getFourSquareExplore?lat=47.3769&lon=8.5417& radius=2000&category=casino
- http://82.130.102.16:8991/getFourSquareExplore?lat=47.3769&lon=8.5417 &category=casino
- http://82.130.102.16:8991/getFourSquareExplore?lat=47.3769&lon=8.5417& radius=2000&category=Movie Theater
- http://82.130.102.16:8991/getFourSquareExplore?lat=47.3769&lon=8.5417
- http://82.130.102.16:8991/fourSquareSearch?near=Hannover,DE&radius=5000 &qTerm=donut

Trip Advisor

- http://82.130.102.16:4121/getAttractions?lat=47.3769&lon=8.5417

Booking.com

- http://82.130.102.16:4015/getBookingcomListings?destination=Berlin&checkinday=21 &checkinyearmonth=2016-12&checkoutday=23&checkoutyearmonth=2016-12&numberrooms=1&adults=2&children=0

Airbnb

- http://82.130.102.16:4015/getListings?Tanzania,TZ&checkin=07/12/2016 &checkout=09/12/2016&guests=2&page=1
- http://82.130.102.16:4015/getInfo?id=15200584

Skiplagged

- http://82.130.102.16:4012/getCheapestFlight?from=ZRH&to=BLR &year=2016&month=12&day=15
- http://82.130.102.16:4012/getLeastLayovers?from=ZRH&to=BLR &year=2016&month=12&day=15&returnMonth=12&returnDay=20&returnYear=2016
- http://82.130.102.16:4012/getShortestFlight?from=ZRH&to=BLR &year=2016&month=12&day=15&returnMonth=12&returnDay=20&returnYear=2016

Database

- http://82.130.102.16:9771/registerUser?uid=2&name=PrashanthB &email=x@x.com&photoURL=abcd
- http://82.130.102.16:9771/getAllUsers
- http://82.130.102.16:9771/updateLikes?uid=2 &place=Empire State Building
- http://82.130.102.16:9771/deleteUser?uid=2
- http://82.130.102.16:9771/updateRejects?uid=1place=Broadway Musical &reason=Too Expensive&reasonCategoryNumber=3
- http://82.130.102.16:9771/updateCategoryRejects?uid=1&category=casino &reason=no money&reasonCategoryNumber=3
- http://82.130.102.16:9771/addTrip?uid=1&tripName=kosovoTrip &tripData=firstStop:beach,secondStop:hotel
- http://82.130.102.16:9771/getTrip?uid=1&tripId=2
- http://82.130.102.16:9771/addReview?uid=211&place=NYC&review=tall building
- http://82.130.102.16:9771/removeLikes?uid=211 &place=Empire State Building
- http://82.130.102.16:9771/removeRejects?uid=211&place=Broadway Musical
- http://82.130.102.16:9771/removeCategoryRejects?uid=211&category=casino
- http://82.130.102.16:9771/getUserById?uid=211
- http://82.130.102.16:9771/generateMatrix?uid=211

- http://82.130.102.16:9771/renameTrip?uid=666&tripId=2 &newTripName=testTrip
- http://82.130.102.16:9771/deleteTrip?uid=666&tripId=1

Category Tree

- http://82.130.102.16:1947/getTopAttractions?location=london&radius=5000
- http://82.130.102.16:1947/getAnnotatedTree
- http://82.130.102.16:1947/queryAnnotatedTree?lat=51.50195785&long=-0.130199 &length=10
- http://82.130.102.16:1947/clearAnnotatedTree

1.1.4 Overview of the Plan My Vacation android application

The basic idea when making the android app was to minimize the user interaction necessary for generating a trip. However, offering the possibility to customize the trip and general preferences for future trips. Users are authenticated using external authentication providers (Google and Facebook) to support user profiles. Users will first select a flight and a hotel/room and then a complete itinerary is generated. It can be customized be rearranging attractions, removing them, or adding new ones. For generating a trip a number of parameters are taken into account e.g. trip duration, daily tour time interval, location of the attractions, user preferences etc.

Chapter 2

Flight Data

2.1 Using the skiplagged API

2.2 Flight selection and ordering

Chapter 3

Hotel Data

3.1 The official un-official Airbnb API

3.1.1 Using the Airbnb API

3.2 Booking.com

3.2.1 Overview of API

3.2.2 Scraping Booking.com

3.3 Hotel selection and ordering

Chapter 4

Attractions

4.1 FourSquare

4.1.1 Categories

4.1.2 Time of day and popular activities

4.1.3 Limitations

4.2 TripAdvisor

4.2.1 Querying Data

4.2.2 Limitations

4.3 Google Queries

4.3.1 Places API

4.3.2 Querying google.com

4.3.3 Top Attractions

4.3.4 Limitations

4.4 Creating an Annotated R-Tree and Trip Generation

4.4.1 Trip Ordering

Chapter 5

System architecture

5.1 Architecture overview

5.2 Server implementation

5.3 Client implementation

5.4 Server-client communication

5.4.1 REST API

Chapter 6

User Database

6.1 Saving data to mongodb

6.1.1 Saving User Data

6.1.2 Saving Trips

6.1.3 Saving User Preferences

Chapter 7

Future Work

Theorem 7.1 (First Theorem)**.** *This is our first theorem.*

Proof. And this is the proof of the first theorem with a complicated formula and a reference to Theorem 7.1. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

$$\frac{\mathrm{d}}{\mathrm{d}x} \arctan(\sin(x^2)) = -2 \cdot \frac{\cos(x^2)x}{-2 + (\cos(x^2))^2} \tag{7.1}$$

□

And here we cite an external document [1].

Bibliography

[1] One, A., Two, A.: A theoretical work on computer science. In: 30th Symposium on Comparative Irrelevance, Somewhere, Some Country. (June 1999)

Appendix A

Appendix Chapter