

MyTrip

An Android Application for Travel Management and Sharing

CS6365 Spring 2018 Project Proposal

Team Members: Andi Xu and Yi Zhang

1. Motivation and Objective

Travel apps and websites are very important resources for travelers. Currently, travelers can plan and manage their trips with the help of online services such as booking a ticket, reading nearby hotels and restaurants reviews. Travel apps enable everything under users' control and hence bring a relaxing holiday for them. Moreover, nowadays many people like to post their travel experiences online to share their feelings with others. Some of these tips are valuable and help other travellers to organize a trip well.

Travel is tiring as we have to consider too many trivial things. We actually use many web or mobile apps to organize a trip. However, we usually are reluctant to install many mobile apps some of which are only launched few times a year. Thus if all features provided by different apps can be integrated in one app, it will significantly reduce the time and complexity of planning and managing a trip. As a result, we started to think about developing an application which can directly connect users to all services they may need in their trips.

Therefore, we plan to develop an Android app that integrates all features users may need in their trip, such as flight notification, ride requests and nearby travel attractions' reviews . Users can use this app to plan and manage their trip without installing many additional travel apps. Moreover, our app provides a platform for those who like to share travel experiences or to learn how to plan a trip well.

2. Related Work

(1) FlightAware (Android/iOS)

FlightAware is an application that enables users to check flights and airport status. By using this app, users can avoid installing many additional mobile apps for different airports and airlines. This application provides useful features such as flight status lookup, alert and delay notification.

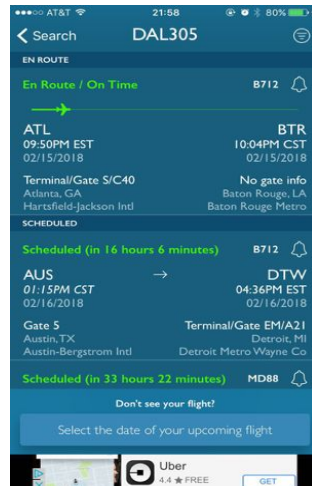


Fig.1.a Checking flight status

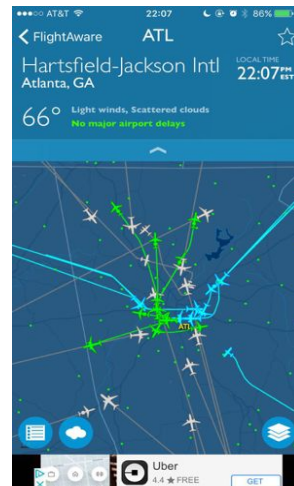


Fig.1.b Seeing airport delays

(2) TripIt (Android/iOS)

TripIt is an application that helps users manage their trips during the whole trip. The most important function of TripIt is that it can automatically import your activities and create a simple plan for you by scanning the confirmation emails in your mailbox. This app also allows users to share their plans with friends or other users.

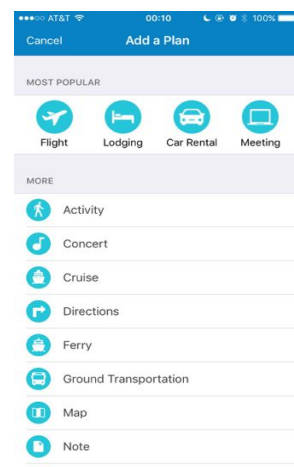
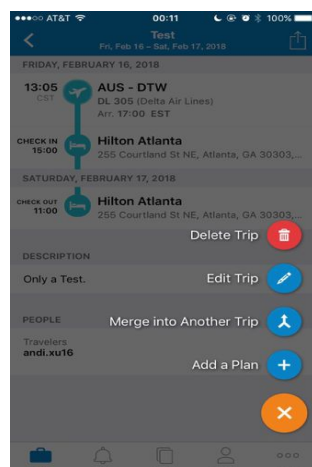


Fig.2. TripIt Features

3. Proposed Work

3.1. Application Features

Our goal is to build an Android^[1] app for users to manage and share their travel plans and memories. The detailed features of the app are as follows.

(1) Travel Management

This feature helps users to plan their travel and preserve their travel memories. First, the app provides one-step transport services. The app automatically tracks the flight status according to flight number and date, and notifies the user promptly if there is a flight delay or cancellation. Also, the app integrates Uber ride request services for the travellers who need a ride.

Second, the app provides a list of nearby restaurants, hotels and travel destinations for entertainment. A user could read the reviews from other users, scan through the photos, and then decides where to go or eat. A user is also encouraged to upload photos and comment on these places if he/she has visited them.

Third, the app tracks travelers' footsteps and preserve their travel memories by creating travel itineraries. Further, the events could be added to Google Calendar automatically.

(2) Travel Sharing

The main feature of the app is to share our travel plans and experiences with others. People who are planning a trip use the app to read others' travel plans and reviews. These plans and tips help people avoid unnecessary trouble and learn to improve their travel plans. Some merchants also learn a lesson from these reviews and improve their services.

3.2. Application Architecture

Our Android app integrates multiple travel features like flight notification, ride requests and reviews of travel destinations and nearby facilities. Thus, we use Yelp^[2], Google Maps^[3], Uber^[4], FlightAware^[5] and Google Calendar^[6] APIs to get data sources. In addition, we plan to build a

RESTful web service using Spring^[7] and deploy it on AWS^[8]. We will use Hibernate^[9] and MySQL^[10] for data storage. The following figure shows our proposed application architecture.

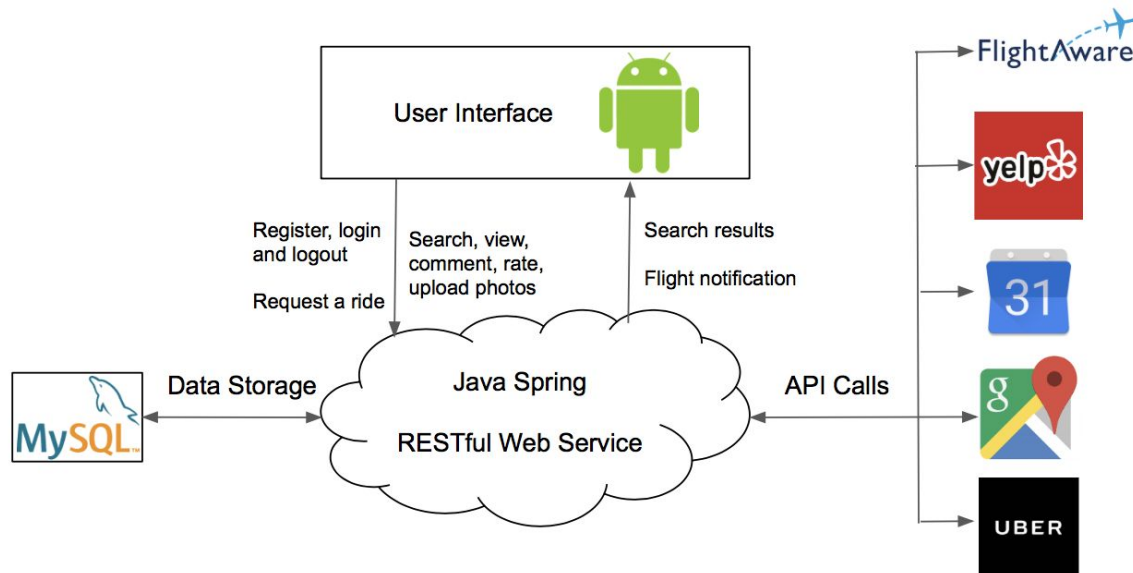


Fig.3. Application Architecture

(1) Data Source

Yelp API: This API provides basic information (location, open hours, reviews, photos) of millions of restaurants, hotels and travel destinations around the United States. To begin with, we select Atlanta, Las Vegas, San Francisco, Chicago, Seattle, Miami, Los Angeles, San Diego, New York, Orlando ten popular tourist cities. Then we run Yelp API to get basic information of top 10 most popular travel destinations, top 50 most popular restaurants and hotels for each city, and store it into MySQL database. Thus, the app firstly queries MySQL database when it gets search request and will call Yelp API if the requested place does not exist in MySQL database.

FlightAware API: This API provides real-time flight status including delays or cancellations according to flight number and date.

Google Maps API: This API allows the app to display nearby restaurants, hotels and travel destinations on Google Maps, which makes it easier for users to interact with the service.

Uber API: This API allows the app to authenticate users with Uber and to request rides on behalf of the users with the permissions from users.

Google Calendar API: This API allows the app to add travel events to Google Calendar.

(2) Backend

We plan to build RESTful web service with Java Spring and deploy it on AWS. Further, we will use Hibernate and MySQL database for data storage. We choose relational database because there are many relationships among user, comments, restaurant, hotel and travel destination models.

(3) Frontend

We plan to focus on Android users and to use Android Studio to implement the Android app. Moreover, we will attempt to use AngularJS^[11] to build web interface if we have enough time.

4. Project Deliverables

- (1) Source code of Android app and backend server with all related documents.
- (2) A final report describing how the whole system works and what features we implement.
- (3) Project presentation slides and a short demo video.

5. Work Schedule and Distribution

Both team members will contribute similar efforts to the project. The future work schedule and distribution are shown in the following table.

Week	Tasks	Person in Charge
Feb.19 ~ Feb.25	Build a basic Android client	Andi Xu
	Collect data from Yelp API	Yi Zhang
Feb.26 ~ Mar.4	Design user interface	Andi Xu
	Clean and analyze data	Yi Zhang
Mar.5 ~ Mar.11	Setup web server environment	Andi Xu
	Build MySQL database and import data	Yi Zhang
Mar.12 ~ Mar.18	Implement web server modules	Andi Xu
	Implement user authentication	Yi Zhang
Mar.19 ~ Mar.25	Implement Google Maps API call	Andi Xu
	Implement FlightAware API call	Yi Zhang
Mar.26 ~ Apr.1	Implement Google Calendar API call	Andi Xu
	Implement Uber API call	Yi Zhang
Apr.2 ~ Apr.8	Implement Yelp API call	Andi Xu
	Implement flight notification feature	Yi Zhang
Apr.9 ~ Apr.15	Implement search feature	Andi Xu
	Implement other features (comment, upload photos)	Yi Zhang
Apr.16 ~ Apr.22	Test	Andi Xu
	Deploy the web service on AWS	Yi Zhang
Apr.23 ~ Apr.29	Test & Final Report & Presentation	Andi Xu, Yi Zhang

Table 1. Work schedule and distribution

6. References

- [1] <https://developer.android.com/guide/index.html>
- [2] https://www.yelp.com/developers/documentation/v3/get_started
- [3] <https://developers.google.com/maps/documentation/android-api/>
- [4] <https://developer.uber.com/docs/riders/introduction>
- [5] <https://flightaware.com/commercial/flightxml/>
- [6] <https://developers.google.com/google-apps/calendar/quickstart/android>
- [7] <https://spring.io>
- [8] <https://aws.amazon.com>
- [9] <http://hibernate.org>
- [10] <https://dev.mysql.com/doc/>
- [11] <https://angular.io/guide/quickstart>