SKYSCANNER

IN COLLABORATION WITH

UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC)

FINAL DEGREE PROJECT

Skyscanner Heatmap

Skyscanner's data comparison

Author:
Fèlix Arribas

Director:
Francisco López
University supervisor:
Maria José Casañ

A Project for the Computer Engineering Degree *in the*
Software Engineering and Information Systems department
Facultat d'Informàtica de Barcelona (FIB)
working with
DeLorean squad *from* Marketplace Engine tribe

Wednesday 10th January, 2018

Universitat Politècnica de Catalunya (UPC)

Abstract

Facultat d'Informàtica de Barcelona (FIB)
Software Engineering and Information Systems department

Computer Engineering Degree

Skyscanner Heatmap

by Fèlix Arribas

In the last century, the world has became smaller. Communications are easier and faster than fifty years ago. Back then, you could talk through a fix phone, but you were not able to send any kind of media, like photos, videos, etc. Only the latest technology of that moment was able to do that. Since the smart phone revolution in 2007 almost everyone can text messages, sending images, share live videos or almost whatever you can imagine in less than a second.

But the internet, phones and communications are not the only thing that made the world smaller. Ways of traveling helped to this earth flattering too. In 1918 visiting another place was very difficult. If you wanted to go through the sea, you had to do it by boat. The fastest way to travel very far in a continent was by train, but not all places were connected with rails. Nowadays, all along with the internet revolution, anyone can travel to the other side of the world in less than a day by plane. Even for traveling inside the same country people use planes.

But, is the air industry as efficient enough? Are all airlines users satisfied with their purchases and possibilities? Skyscanner provides an easy to use tool to search cheap flights from any airport to another. Sadly, sometimes is difficult for users to find what they really want.

This project wants to help solving this problem, providing a HeatMap to explore differences and similarities between what users search and what airlines provides. Being able to compare between specific dates to guess user behavior.

Contents

Chapter 1

Context

This is a project developed in *Skyscanner* and evaluated by the *Universitat Politècnica de Catalunya (UPC)* as a Final Degree Project.

The main goal of this project is creating a tool for *Skyscanner* to ease the routes comparison by different parameters, taking into account values like **user demand** and **flights provided** by airlines.

Using this comparison, flights advertisement could be improved according to user demand. The company could also develop complex software using the huge amount of data it will compare through an Application Programming Interface.

Skyscanner have more than 75 million flights information and all its users queries. In order compare all the data available and get significant results, the software should take into account all possible risks working with Big Data frameworks.

1.1 Skyscanner

Skyscanner[1] was formed in 2004 when a group of people was frustrated by the difficulties of finding cheap flights.

In 12 years has evolved from a little office in the suburbs of Edinburgh to a world wide company with ten offices in seven different countries. Having more than 4 million visitors every day and more or less half a million pounds of revenue per day.

Now, is one of the top travel fare aggregate website. In the next 5 years, Skyscanner wants to become the travel experience that people prefer to the myriad confusing and unconnected travel apps.

This growth is possible thanks to the revenue Skyscanner gets from the App and Website, but how does this company make money? Does it get money from its adds as Google and other top tech companies do? Or it sells valuable information to its stakeholders such as user trends like Facebook or Twitter?

Since Skyscanner does not actually sell the flights (or hotel rooms or car hires) it cannot take a percent of the purchase. Skyscanner serves to the user a lot of data from different providers and once the user has selected what he wants to buy, it is redirected to the provider website to finish the acquisition.

The provider knows where the user comes from and they give a percent of the profit to Skyscanner.

1.2 Marketplace Engine tribe

This tribe[2] is one of the most important tribes in Skyscanner[1], its mission is to provide the most comprehensive and accurate flight inventory for Skyscanner and her partners with minimum latency.

Its main goal is to evolve the search, pricing, routes and browse services to be horizontally scalable and set us up to build a lightning fast, super accurate and fully comprehensive flight search engine, enabling the traveler to instantly find the best flight at the best price with minimum effort.

1.3 DeLorean squad

DeLorean[3] is a squad of Marketplace Engine tribe, its mission is to provide the best data and services around the routes, timetables and modes of transportation to go from one point on Earth to another.

The squad now provides a very fast service that serves flights logistic information between a given origin and destination. Some information you can find in a route is the fight number, carriers, stops, date ranges, etc.

[1] Learn more about *Skyscanner structure* in section Appendix A

Chapter 2

State-of-the-art

Since this project is not oriented for Skyscanner users but the company, the *State-of-the-art* relates to services inside Skyscanner. Even so, a brief explanation about other metasearch engines would help to find the gap this project is developed:

2.1 Fare aggregators and metasearch engines

2.1.1 Google Flights

In the last years Google Flights has became the main competitor of Skyscanner. The new version is very fast and has a complete new interface, following Android guidelines.

Google is one of the top tech companies and has a lot of different platforms. It is a competitor to be aware of, the integration with Gmail, Google Calendar and Android OS makes Google Flight a part of its ecosystem. The traveler may feel comfortable.

2.1.2 Kayak

Kayak has always been the main competitor, both companies started in 2004. Unlike Skyscanner, Kayak started with Flights, Hotels and Car hiring. Skyscanner added those two extra search engines between 2013 and 2014.

2.1.3 Expedia

Launched in November 1998, is one of the oldest fare aggregator and metasearch engine. Apart of its own website, is also a Skyscanner provider. Some of the prices are taken from Expedia and sometimes the user is redirected to their website to finish their purchase.

2.2 Skyscanner services

In Skyscanner the user has never been a product, in fact, one of the statements of Skyscanner's culture says *Traveler != Product*[4].

There has never been a project getting value from user information because it does not follows the company culture, so the definition of the problem and the scope of the project must be very accurate to ensure it is fulfilling with Skyscanner's strategy[1].

2.2.1 Marketplace Engine

This tribe is formed by five squads, those constantly work to improve the routes and pricing service all along with an efficient search.

Marketplace Engine works with data *from the provider to the user*. In other words, it just serves **information to the user** but does not get any from him/her. All five Squads take all the **data from providers**.

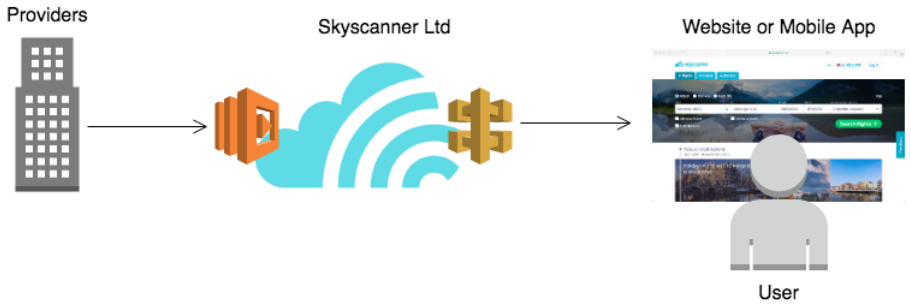

FIGURE 2.1: Simple explanation of Marketplace Engine data flow.

2.2.2 Data Tribe

In the other hand, Data Tribe has a lot of squads with services used collect **data from user activity**. The flow of the information is *from the user to Skyscanner*.

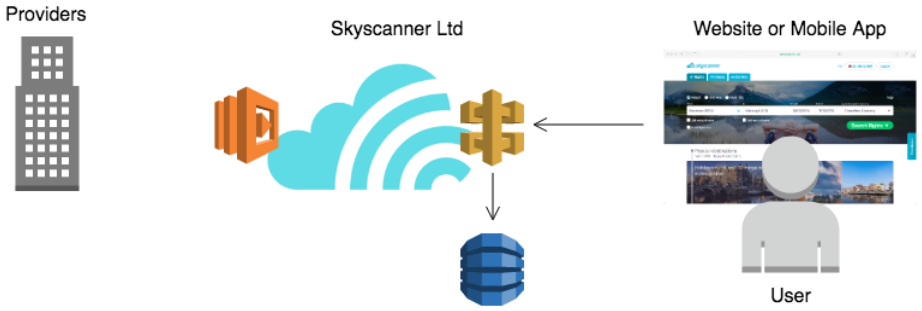

FIGURE 2.2: Simple explanation of Data Tribe data flow.

2.2.3 The gap

There is no tribe of squad that works with both **data sources**: Providers and Users. And here is where the *Heatmap* will be.

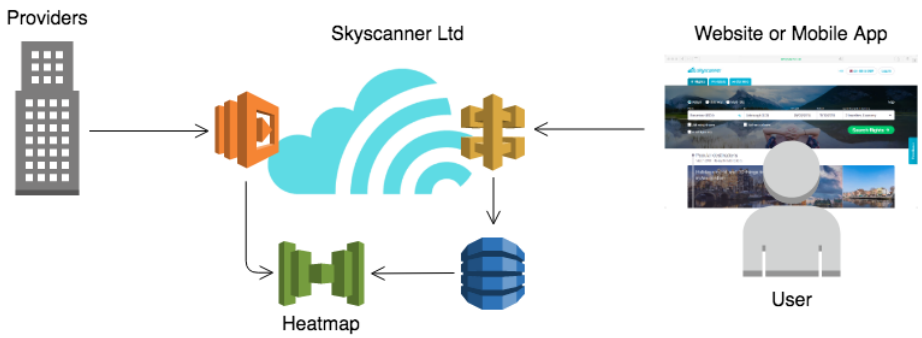

FIGURE 2.3: First approach of the Heatmap's data flow.

Chapter 3

Skyscanner Heatmap

3.1 Definition of the problem

In any team of Skyscanner, the user queries and the providers data is compared in order to guess valuable trends for different Stakeholders.

Found that gap, a bunch of new ideas appeared. After some talks with product owners of different squads and some senior engineers a promising idea showed up:

Comparison of **user demand** and **flights offered** by airlines, enabling finding *over-requested* routes or airports.

DeLorean squadmanages a huge amount of data: All flights planned for the next two years, this are more than 75 million records. The database of all user queries in the website or mobile application is even bigger[1]. Not much more information needed to say that this is **Big Data** problem.

With DeLorean squad's product owner help, we found some use cases for the professing of those 75 million routes and all user session's queries to get some significant results.

Provide a visual tool to find routes and airports with much more demand than offer and be able to observe the evolution of it through time:

- A route or airport with a lot of demand but not enough offer to cover it will be **over-requested**.
- A route or airport with much more offer but not that amount of demand will be **non-profitable**.

3.2 Scope

Merging both data sources (providers and users) generates a lot of new valuable data with a lot of different application: From simply selling it to stakeholders, to complex deep learning systems.

The final goal of this project is displaying the comparison in a simple Web UI for Marketing Squads or Tribes. This can be split in three smaller goals or components:

[1] For instance, if there were only one query per visitor the database would have 4 million new records per day

3.2.1 Pipeline

Distributed application that maps and merge all the data from both sources in its given format, to the required data model.

The pipeline reads from Marketplace Engine and Data Tribe services. Then, the pipeline, maps the provider and user data to the desired data model. The new entities are stored in a database where the service will read from.

The application will be split in two sub applications, one for providers' data and other for users'. So both can vary independently without depending on the each others' sources and changes may have in the future.

3.2.2 Service

Simple HTTP Service with a basic Application Programming Interface to **get** Pipeline's results. The service will have an internal endpoint only available for other Skyscanner applications or developers.

3.2.3 Visual representation

Website with a visual representation of the data. There are plenty of ways to draw charts and maps visualizations.

The Web UI will be composed by three main pages:

World Map

Interactive world map with all airports represented with a dot. The radius of the dot depends on the amount of flights it operates.

The user will be able to select an airport set a date and go to Chart visualization page. Another option is to select two airports, first the origin, then the destination, set a date and go to the Chart visualization page. If the user does not want to select the entity through the map, he/she can search it using the Browser

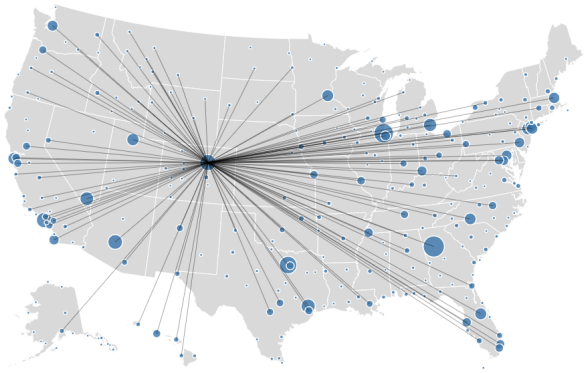

FIGURE 3.1: Example of how the world map style. Only displaying the US make it look clearer.

Browser

Simple browser with two tabs: *Route* and *Airport*. In the route browser will appear three input text fields, one for the origin airport, the second for the destination and the last one for the date. In the airport browser will only appear two input text fields, airport and date.

Once the inputs are set, the user will be able to click a *Search* button and move to the next page, Chart visualization.

Chart visualization

Simple chart with the comparison between providers offer and user demand of the selected entity through time.

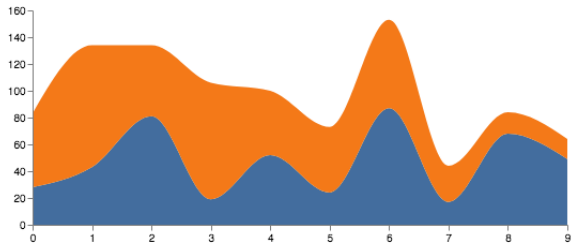

FIGURE 3.2: Chart mock-up. One color goes for Providers Offer and the other one for User Demand.

3.2.4 Not list

It is also important to define what this project will **not** be.

- **Prices** or **quotes**: In any moment will check for flight prices or quotes.
- **Carriers**, **cities** and **countries**: The comparison will be only available between routes and airports, not airlines (carriers), cities nor countries.
- **Create**, **update** or **delete** data through the **Server**: The only input will come from the pipeline. Entities are never deleted or modified in order to keep historical data.
- **Create**, **update** or **delete** data through the **Web UI**: The only input will come from the pipeline. Entities are never deleted or modified in order to keep historical data.

3.3 Risks

There are several risks can appear while developing the project. Most risks appear because of the dependencies with other tribes and squads, dependencies with other services. In the other hand, all performance risks of the Pipeline can ignored because Skyscanner's hardware is enough for big applications like this one.

3.3.1 Routes contract

DeLorean squad's routes service is under development and during the Heatmap development the routes' data model may change a little bit. For example, the origin and destination recently changed, in December 2017 their service was giving an *Airport ID*, but

now are given in an Airport object with more parameters like IATA Code[5], Country ID, City ID, etc.

3.3.2 Users information

In the website and mobile application, the user have plenty of different ways to search the perfect flight. The most common one is by origin, destination and date, but he/she can also search by month, by destination. This way the user search flights may be difficult to compare with routes and airports offer because, sometimes, the route is the actual result.

The user does not search flights for a given route in a given date. It sets the period of time he/she can travel and Skyscanner offers cheap destinations.

3.3.3 Amount of data

As explained before in the Definition of the problem, there is a very big amount of data that need to be mapped. Luckily, Skyscanner have great cloud machines and (almost) unlimited space. But stills an issue to be aware of.

3.3.4 Web UI

Creating the interactive map and graphics of the proposed website from zero is a whole project itself. In order to avoid failing to the Visual representation goal, the best option is to use reliable libraries, like Vega[6].

3.4 Methodology and rigor

3.4.1 Extreme Programming

This project will be developed along with DeLorean squad's work. This squad is following Scrum, an agile methodology. After come research and some discussions with the rest of the team, Extreme Programming[7] showed up as the best option.

XP is a style of software development focused in excellent applications, programming techniques, clear communication, etc. To accomplish that, XP includes a philosophy of software development, body of practices, complementary principles and a community that shares these values.

This methodology works with short development cycles, resulting in early, concrete and continuing feedback. Has an incremental approach, making way to a flexible schedule of the implementation. It relies in oral communication and tests to reach the goal of the project.

The original Extreme Programming methodology is for teams of developers, but this project will be only developed by myself, so the original idea has been modified a little bit. The pair programming and pair negotiation has been removed because I have nobody to pair with.

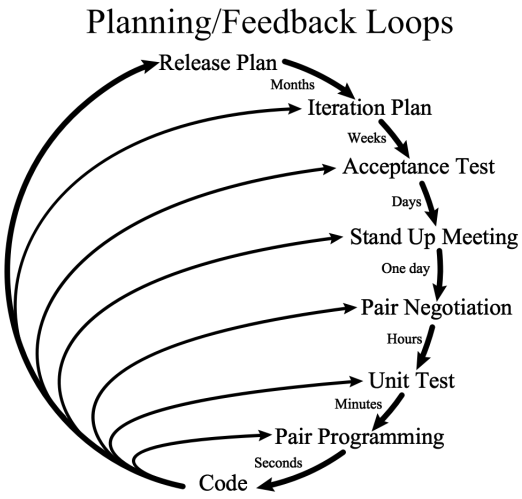

FIGURE 3.3: Extreme programming planning loops.

3.4.2 GitLab

This platform will be the main tool for version control of the source code and issue tracking of different tasks.

`git`

All code projects (pipelines, service and website) will be stored in my personal projects space in Skyscanner's GitLab domain. Tasks and issue projects will be related to each project.

Using `git`, the versions will be forked in branches, each branch will stand for an specific issue. *Master* will be the main branch where the production version will.

Tasks and issues

The **issue tracking** will be very helpful in order to monitor the evolution of the project. Issues will be composed by a title, description, milestone, labels (if needed), due date and weight, and represents a new functionality. In order to know the status, issues will be listed in three columns:

Backlog: Known tasks that haven been started yet. Could be a well defined task, with a very clear description, due date and weight, or just a draft.

W.I.P.: Work in Progress. The task is being considered, developed or tested.

Done: Tasks finished, tested and done. Ready for production.

Chapter 4

Requirements analysis

4.1 Stakeholders

Initially it seemed difficult to find stakeholders and actors in these project apart from the providers. It is not a tool for the user of Skyscanner so, as explained before, one risk of these project was not finding enough support.

After walking with the Squad Lead and then the Product Own of DeLorean squad a lot of stakeholders appeared: DeLorean Squad, Marketing Automation Squad, Fuel RaTS Squad, etc. Each of these stakeholders has different use cases and the project became very interesting for a considerable part of Skyscanner .

4.1.1 DeLorean squad

DeLorean's Single Flight Number service, also known as *Timetable SFN Service*, provides all the **current** flights. This is a little bit of a problem when trying to get past routes: Timetable SFN Service does not provide past flights information, it is always **up-to-date**. In order to get this data it is needed to go one step back in the whole DeLorean data processing: *Timetable Pipeline*.

The heatmap must reference old versions of the file created by the Timetable Pipeline to get older routes.

Timetable SFN Service

The *timetable SFN* endpoint returns details for time tabled Single Flight Number itineraries series. Note that SFNs are not ticket-able, so they do not include itineraries which cannot be bought on their own, neither the price nor restrictions.

Timetable Pipeline

This phase, basically collects all the OAG[1] from a provider and maps it into routes in JSON[9] format. For each different version of the OAG file, the pipeline creates a new file with all the routes.

Product Owner

Jen Agerton is the Product Owner of DeLorean Squad. She found that the Heatmap is very useful for other squads like Marketing Automation squad and providers (air carrier companies).

[1]OAG file (also know as WTF file), is a CSV[8] file which each row represents a timetable for a Single Flight Number.

DeLorean's Squad Lead

Francisco López is also the supervisor of this project. Me and had the initial idea for this project. He oriented it for a Machine Learning purpose: The information that the heatmap stores is very useful for constructed routes.

4.1.2 Fuel RaTS squad

Routes and Timetable Servies Squad provides the best data and services around the routes, timetables and modes of transportation to go from one point on Earth to another. Fuel RaTS has the same mission as DeLorean Squad, but develop different services. Since Fuel RaTS provides basic routes data, pricing, live update information and multi-destination combinationcs, DeLorean squad provides a very fast service for only routes.

4.1.3 Marketing Automation squad

Marketing Automation squad enables scalable growth by automating workflows, and the collection of insightful data. They have three main goals:

- Provide data to support decision making
- Automated, data driven campaign management
- Budget process automation

4.1.4 User

The user of this project can change a lot in the future. Now it will be only Marketing Automation Squad developers and Skyscanner employeers. But it is oriented for

4.2 Functional requirements

4.3 Non functional requirements

4.4 Use cases

Name	Routes offer and demand comparison **heatmap**
ID	UC0
Description	Heatmap of the comparison between providers offer and user demand. The heat is represented by the *over requests* of a route.
Actors	User
Triggers	Loading home page
Precondition	
Postcondition	Wolrd heatmap with most relevant routes and their heat.
Basic Flow	
Alternate Flow	
Exceptions	

TABLE 4.1: *Routes offer and demand comparison **heatmap*** use case

Name	Offer and demand plot of route
ID	UC1
Description	Compare the user demand and the providers offer of a specific route from city A to city B in a given date in a plot with two data sets, offer and demand.
Actors	User
Triggers	Request to get comparison of route from city A to city B in a specific date.
Precondition	City A and city B exists and there is some connection (SFN or Constructed) in the date.
Postcondition	Plot with the evolution through time of the user demand and air carrier offer. Time limit goes from fist offer apperance to arrival date or current date, depending which comes first.
Basic Flow	1. System provides a list of cities under *origin* tag. 2. User selects an origin city. 3. System provides another list of cities. Now with *destination* tag. 4. User selects destination (See exception 1). 5. System provides an interactive calendar. 6. User selects a date of the calendar (See exception 2). 7. System provides the plot of the demand and offer evolution of the route.
Alternate Flow	Alternate course 1 1. User **changes** destination city (See exception 1). 2. Return to basic flow step 6. Alternate course 2 1. User **changes** date (See exception 2). 2. Return to basic flow step 7.
Exceptions	1. There are no connections between to given cities. 2. There are connections between to given cities, but not in the given date.

TABLE 4.2: *Offer and demand plot of route* use case

Name	Offer and demand data set of route
ID	UC2
Description	Data set of the evolution of the user demand and providers offer in order to create metrics, alerts, etc.
Actors	Marketing Automation squad, DeLorean squad
Triggers	Request to get data set of route from city A to city B in a specific date.
Precondition	City A and city B exists and there is some connection (SFN or Constructed) in the date
Postcondition	Plot with the evolution through time of the user demand and air carrier offer. Time limit goes from fist offer apperance to arrival date or current date, depending which comes first.
Basic Flow	1. System provides an HTTP endpoint to request data. 2. The developer does a GET request to the endpoint with an origin, destination and a date (See exception 1). 3. System provides a data set in JSON format with all the demand and offers of the entity.
Alternate Flow	
Exceptions	1. There no connections between city A and city B in the given date.

TABLE 4.3: *Offer and demand data set of route* use case

Name	name
ID	id
Description	description
Actors	actors
Organzational Benefits	benefits
Frequency of Use	frequency
Triggers	trigger
Precondition	pre
Postcondition	post
Basic Flow	main
Alternate Flow	alt
Exceptions	exc

TABLE 4.4: *title* use case

Appendix A

Skyscanner structure

Skyscanner has a very horizontal structure, based on Spotify's[10].

In the top of the company's hierarchy there is Gareth Williams (CEO and Co-founder). Just bellow the rest of CxOs: CCO, CTO, CPO, CFO, CLO and the Senior Executive Assistant. Then vice presidents, senior managers, managers and then developers and interns[11].

Apart from this hierarchy structure, the whole team, except the CEO, CxOs and the Senior Executive Assistant, is mainly split in **Squads**, each of those belong to a **Tribes**. Apart from Squads and Tribes there are also Chapters, Guilds ans XBT'S[12].

Squad

Are independent teams of no more than 8 people that are focused on delivering a core mission. Each squad has the freedom to act and be accountable to its mission.

Tribe

Squads belong to a Tribe. The tribe will have an aligning mission linking to each squad's mission and is only achievable depending on the success of each squad. The Tribe lead is responsible for providing the right environment to deliver and providing direction.

Chapters

Are people who do similar work. This is a secondary home, and how people are line managed. Chapter leads are responsible for developing people and in tribe practices.

Guilds

Are communities of interest of people who do not necessarily do similar work. It is people from across the business that want to share knowledge, tools, and word practices.

XBT'S (cross business teams)

XBTS' provide a platform to help solve business problems or opportunities with no natural home while giving all employees the ability to make an impact across any area of Skyscanner.

Bibliography

[1] Mark Logan. *Skyscanner's Strategy*. 2015. URL: https://skyspace.sharepoint.com/sites/CxOGMblogs/Marksblog/Lists/Posts/Post.aspx?ID=2 (visited on 02/27/2018).

[2] Francisco Lopez. *Marketplace Engine Tribe Home*. 2017. URL: https://confluence.skyscannertools.net/display/MET/Marketplace+Engine+Tribe+Home (visited on 02/27/2018).

[3] Francisco Lopez. *DeLorean Home*. 2017. URL: https://confluence.skyscannertools.net/display/DEL (visited on 01/28/2018).

[4] Skyscanner Ltd. *The Road Ahead*. 2016. URL: https://skyspace.sharepoint.com/docs/Internal%20Communications%20and%20Events%20Squad/The%20Road%20Ahead.pdf (visited on 02/28/2018).

[5] Multiple authors. *IATA airport code*. 2018. URL: https://en.wikipedia.org/wiki/IATA_airport_code (visited on 03/01/2018).

[6] Interactive Data Lab. *Vega: A visualization grammars*. 2013. URL: https://vega.github.io/vega/ (visited on 03/01/2018).

[7] Kent Beck. *Extreme Programming Explained. Embrace Change*. John Wait, 2005.

[8] Y. Shafranovich. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. 2005. URL: https://tools.ietf.org/html/rfc4180 (visited on 01/30/2018).

[9] JSON.org. *The JavaScript Object Notation (JSON) Data Interchange Format*. 2017. URL: https://tools.ietf.org/html/rfc8259 (visited on 01/30/2018).

[10] Skyscanner Engineering. *The Culture of Growth Squads in Skyscanner*. 2016. URL: https://www.youtube.com/watch?v=5_x-_EJNWPs (visited on 03/01/2018).

[11] Skyscanner Ltd. *Crew Chart hierarchy*. 2017. URL: https://flightdeck.skyscannertools.net/crewchart.html (visited on 06/26/2017).

[12] Skyscanner Ltd. *How Skyscanner Works*. 2017. URL: https://skyspace.sharepoint.com/sites/Information/Pages/How-Skyscanner-Works.aspx (visited on 03/01/2018).