skyscanner

FIB UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

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

Friday 2nd March, 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.

This project's porpose is to compare routes and airports in **user demand** and **flights provided** by airlines. This comparison could improve flights advertisement 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 to compare all the data available and get significant results, the software should solve **Big Data**.

1.1 Skyscanner

Skyscanner[1] is a travel fare aggregate website. It 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. In the next 5 years, Skyscanner wants to become the travel experience that people prefer to the myriad confusing and unconnected travel apps.

Now, is one of the top travel fare aggregate website. It has more than 4 million visitors every day and. more or less, a revenue of half a million pounds per day.

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 like 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 (neither hotel rooms nor 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 a destination. Some information you can find in a route is the fight number, different carriers, stops, date ranges, etc.

---

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

Chapter 2

State-of-the-art

Since this project is not oriented for Skyscanner users but the company itself, 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 worldwide 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 Google's 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**.
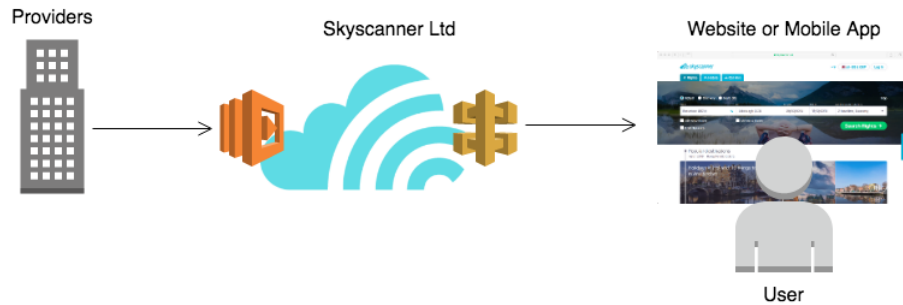
FIGURE 2.1: Simple explanation of Marketplace Engine data flow.

2.2.2 Data tribe

In the other hand, Data tribe has a lot of squads with services used to collect **data from user's activity**. The flow of the information is *from the user to Skyscanner*.
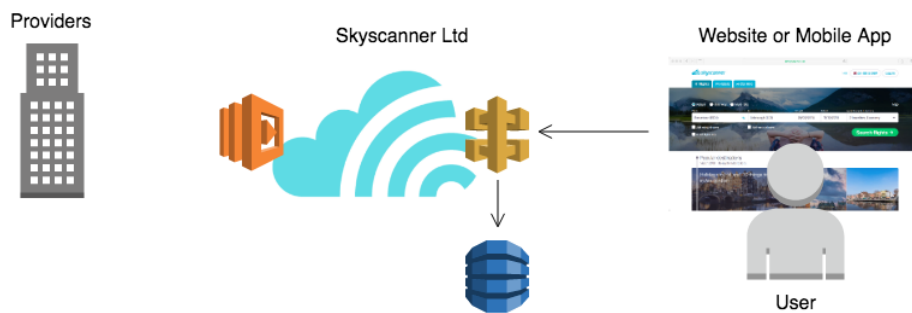
FIGURE 2.2: Simple explanation of Data tribe data flow.

2.2.3 The gap

There is no tribe or 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 Formulation of the problem

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

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

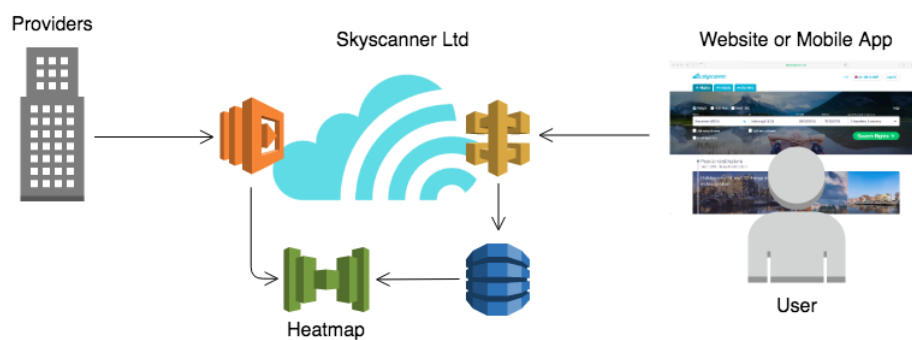Found that gap, a bunch of new ideas appeared. After some talks with product owners of different squads and some senior engineers, a promising idea showed up:

Comparison of **user demand** and **flights offered** by airlines, enabling finding *over-requested* routes or airports. Those routes or airports with more user demand than offers by the providers.

DeLorean squad manages 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 processing 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 providers, to complex deep learning systems.

---

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

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:

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, it 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 of them can vary independently without depending on each others' sources and changes may have in the future.

3.2.2 Service

Simple HTTP Service with a basic Application Programming Interface to serve 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 the airport operates.

The Heatmap 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.
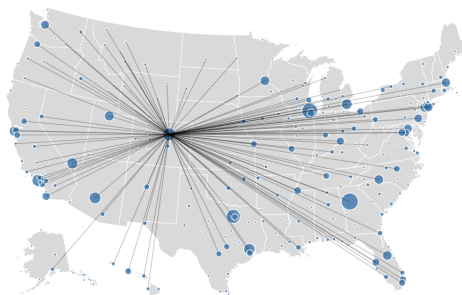

FIGURE 3.1: Example of the world map style. Only displaying the US to make it look clearer.

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 the *Search* button and move to the next page.

Chart visualization

Simple chart with the comparison between providers offer and user demand of the selected entity through time.
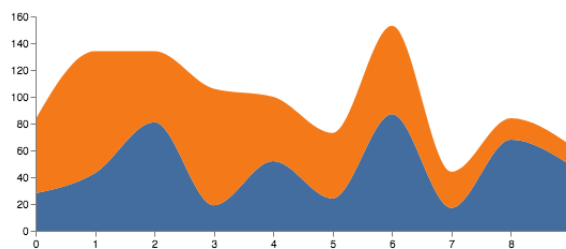
FIGURE 3.2: Chart mock-up. One color goes for Providers Offer and the other one for User Demand.

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 and dependencies with other services. In the other hand, all performance risks of the Pipeline can be 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 the service was giving an *Airport ID*, but now

is giving 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 or by destination. 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.

This way of searching flights may be difficult for comparing with routes and airports offer, because sometimes the route is the actual result and not the query.

3.3.3 Amount of data

As explained before in the Formulation of the problem, there is a very big amount of data that need to be mapped. Luckily, Skyscanner have great cloud machines and *unlimited* space. Anyway, still an issue to be aware of.

3.3.4 Web UI

Creating the interactive map and plots for the proposed website from zero, is a whole project itself. In order to avoid failing in 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 some research and discussions with the rest of the team, Extreme Programming (XP)[7] showed up as the best option.

XP is a style of software development focused in excellent applications, programming techniques and clear communication. 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 for 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.

Planning/Feedback Loops
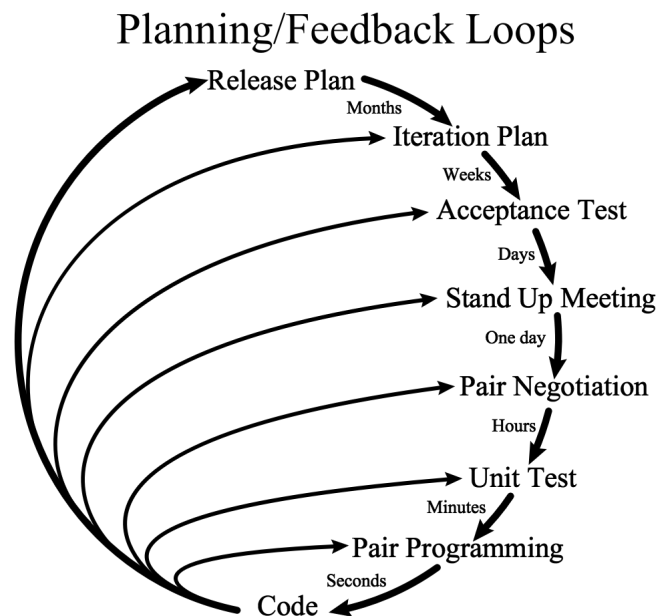
FIGURE 3.3: Extreme programming planning loops.

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. Task 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 latest production version will be.

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, a due date and weight, or just a draft with empty fields.

WIP: Work in Progress. The task is being considered, developed or tested.

Done: Tasks finished, tested and working fine. Ready for production.

Chapter 4

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.

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

4.1   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 historical data because 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 look old versions of the file created by the *Timetable Pipeline* to get older routes. Then, DeLorean squad is interested in the Heatmap because it will be using Pipeline's data.

Product Owner

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

DeLorean's squad Lead

Francisco López is also the supervisor of this project. We both had the initial idea for this project. He saw an oportunity for the future (after project's deliverment) orienting the Heatmap for Machine Learning purpose: The information that the heatmap stores is very useful for constructed routes.

4.2   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

The Heatmap will be very useful for the first goal. The data provided by the Heatmat has high value in marketing decisions. Looking at historical data, Marketing Automation squad could post an advertisement about trend routes in a specific time of the year.

4.3 Data tribe

In Data tribe, State Machine squad captures the user actions, so they know where the user gets stuck or if they finally reach the prodiver of the flight. Other squads like Clan A and Clan B just gets users queries in flights, hotels and car hiring. The second data source of the Heatmap (user requests and queries) will be obtained from these squads.

4.4 User

The user of this project can change a lot in the future. Now it will be only Marketing Automation squad developers and Skyscanner employees. In the future, providers could take profit from Heatmap comparison.

Companies will be able to know which of their routes or airports work better with user tendencies, they will be able to improve the flights service and make it more eficient, reducing number of flights in *non-profitable* routes. They will also know which are the best places to invest looking at *over-requested* airports.

Chapter 5

Project planning

Before defining tasks and time distribution of those, it is important to remember that this project will be developed under an Agile Methodology, Extreme Programming. XP attempts to reduce costs of changes in requirements and long term plans, this means that this *time planning* will provably change during the development. Every week a small iteration plan will be made in order to adjust the tasks with the scope and time of the project.

Even so, here is a general plan. Tasks has been **grouped regardless the technology**, for instance: Task 27, Home Page from the Web UI. It will be developed in HyperText Markup Language and Java Script, but there is only one task for it. As explained in the previous paragraph, it is defined this way for now, in order to be agile in changes of the requirements. Then, tasks are grouped in **Milestones**.

Milestones are used to mark specific points or goals along of a project *timeline*. Milestones are very useful in order to do not losing time perception in long software projects when using Agile Methodologies, in which there are not usually to many deadlines.

Another important thing to take into account is the **tasks overlapping**. Is a good practice[8] not to work to much time in the same thing, task or goal. Swapping between different tasks and milestones can help so the developer **does not obfuscate**.

Last but not least, there are two holiday weeks in March, one is *Holy week* (13th), and the other is extra. That is why the project starts earlier.

5.1   Tasks

Development tasks (from 14 to 30) duration is specified in days. Weeks are composed by five working days, so 5 days equals to a week. In a day, the developer is supposed to work five hours. From week seven to week twenty-three, there are fifteen working weeks (not counting holidays). Doing simple maths, the development will take:

$$15 \text{ weeks} \times 5\ \frac{\text{days}}{\text{week}} \times 5\ \frac{\text{hours}}{\text{day}} = 375 \text{ hours}$$

375 hours of development.

It is important to understand that similar tasks like 16th and 20th or 17th and 21st have different times, that is because the second time it is supposed to be very similar than the first, so there will be some previous knowledge when executing it for second time.

5.1.1 Inception

Regard this an agile project, there has been an small inception part where the project was predefined and explained to Skyscanner product owners to see if it was viable.

Name	Inception
Number	1
Description	Identify the initial scope of the project, stakeholders, context and environment where the project will be developed.
End date	9th of February, 2018

Name	Definition of the problem
Number	2
Description	Defining, at a high level, what the system will do.
Duration	10 days
Milestone	Inception
Dependencies	–

Name	Scope
Number	3
Description	What the software project will do and will not. Not list.
Duration	5 days
Milestone	Inception
Dependencies	2: Definition of the problem

Name	Risks
Number	4
Description	Find possible problems and obstacles may appear in the future development.
Duration	8 days
Milestone	Inception
Dependencies	3: Scope

Name	Scope refinement
Number	5
Description	After finding the risks, review the scope. Make it possible.
Duration	4 days
Milestone	Inception
Dependencies	4: Risks

Name	Environment
Number	6
Description	Find the correct environment in order to build the project.
Duration	2 days
Milestone	Inception
Dependencies	5: Scope refinement

5.1.2 Project management

Name	Project management (GEP)
Number	7
Description	First stage in the TFG. Get thesis started.
End date	20th of April, 2018

Name	Context and scope
Number	8
Description	Indicate general objective of the TFG and context.
Duration	12 days
Milestone	Project management
Dependencies	–

Name	Project planning
Number	9
Description	Planning of the entire execution of the TFG.
Duration	4 days
Milestone	Project management
Dependencies	8: Context and scope

Name	Budget and sustainability
Number	10
Description	Explanation of the sustainability of the project. Economical, social and environmental.
Duration	5 days
Milestone	Project management
Dependencies	9: Project planning

Name	First oral presentation
Number	11
Description	Three minute oral presentation on video.
Duration	10 days
Milestone	Project management
Dependencies	10: Budget and sustainability

Name	Competences review
Number	12
Description	Review of the competences of the bachelor's thesis.
Duration	5 days
Milestone	Project management
Dependencies	–

Name	Final document
Number	13
Description	Project management, Project management and Project management. Reviewed.
Duration	5 days
Milestone	Project management
Dependencies	11: First oral presentation

5.1.3   Flights offer pipeline

Name	Flights offer pipeline
Number	14
Description	Application that maps the provider data to the desired data model.
End date	16th of March, 2018

Name	Reading from DeLorean's pipeline
Number	15
Description	Connect to DeLorean's application that gets all routes and read from it
Duration	5 days
Milestone	Flights offer pipeline
Dependencies	–

Name	Data processing
Number	16
Description	Filter and map all data in DeLorean's model to desired for the
Duration	10 days
Milestone	Flights offer pipeline
Dependencies	15: Reading from DeLorean's pipeline

Name	Writing into service DB
Number	17
Description	Once the data is processed, write into the service DB
Duration	10 days
Milestone	Flights offer pipeline
Dependencies	16: Data processing

5.1.4   User searches pipeline

Name	User searches pipeline
Number	18
Description	Application that maps the user data to the desired data model.
End date	27th of April, 2018

Name	Reading from Data Tribe BD
Number	19
Description	Find correct table and read from Data Tribe DB service.
Duration	10 days
Milestone	User searches pipeline
Dependencies	–

Name	Data processing
Number	20
Description	Filter and map all data in Data Tribe's model to desired for the
Duration	5 days
Milestone	User searches pipeline
Dependencies	19: Reading from Data Tribe BD

Name	Writing into service DB
Number	21
Description	Once the data is processed, write into the service DB
Duration	5 days
Milestone	User searches pipeline
Dependencies	20: Data processing

5.1.5 Heatmap server

Name	Heatmap server
Number	22
Description	Web server that provides all data processed by pipelines
End date	19th of May, 2018

Name	Reading offer
Number	23
Description	Read offer from DB, wrote by Flights offer pipeline
Duration	5 days
Milestone	Heatmap server
Dependencies	17: Writing into service DB

Name	Reading demand
Number	24
Description	Read offer from DB, wrote by User searches pipeline
Duration	5 days
Milestone	Heatmap server
Dependencies	20: Writing into service DB

Name	Comparison
Number	25
Description	Provide comparison of both sources.
Duration	10 days
Milestone	Heatmap server
Dependencies	23: Reading offer, 24: Reading demand

5.1.6 Heatmap Web UI

Name	Heatmap Web UI
Number	26
Description	Website with a visual representation of the data.
End date	8th of June, 2018

Name	Home page
Number	27
Description	World map with all airports, just geographical localization.
Duration	5 days
Milestone	Heatmap Web UI
Dependencies	–

Name	Search page
Number	28
Description	Search inputs for querying the service.
Duration	5 days
Milestone	Heatmap Web UI
Dependencies	–

Name	Data representation
Number	29
Description	Get data from service and display it in the views.
Duration	10 days
Milestone	Heatmap Web UI
Dependencies	25: Comparison, 27: Home page, 28: Search page

Name	Chart view
Number	30
Description	Chart showing the route or airport comparison.
Duration	5 days
Milestone	Heatmap Web UI
Dependencies	29: Data representation

5.1.7 Final presentation

Name	Final presentation
Number	31
Description	TFG whole report document and prepare the final presentation.
End date	22nd of June, 2018

5.2 Alternatives

In case the project takes too much time, both pipelines can be totally overlapped. This will reduce the number of weeks from fifteen to eleven.

Another option is to remove the some pipeline, the server will provably have a higher latency, but it will reduce to ten or eleven weeks.
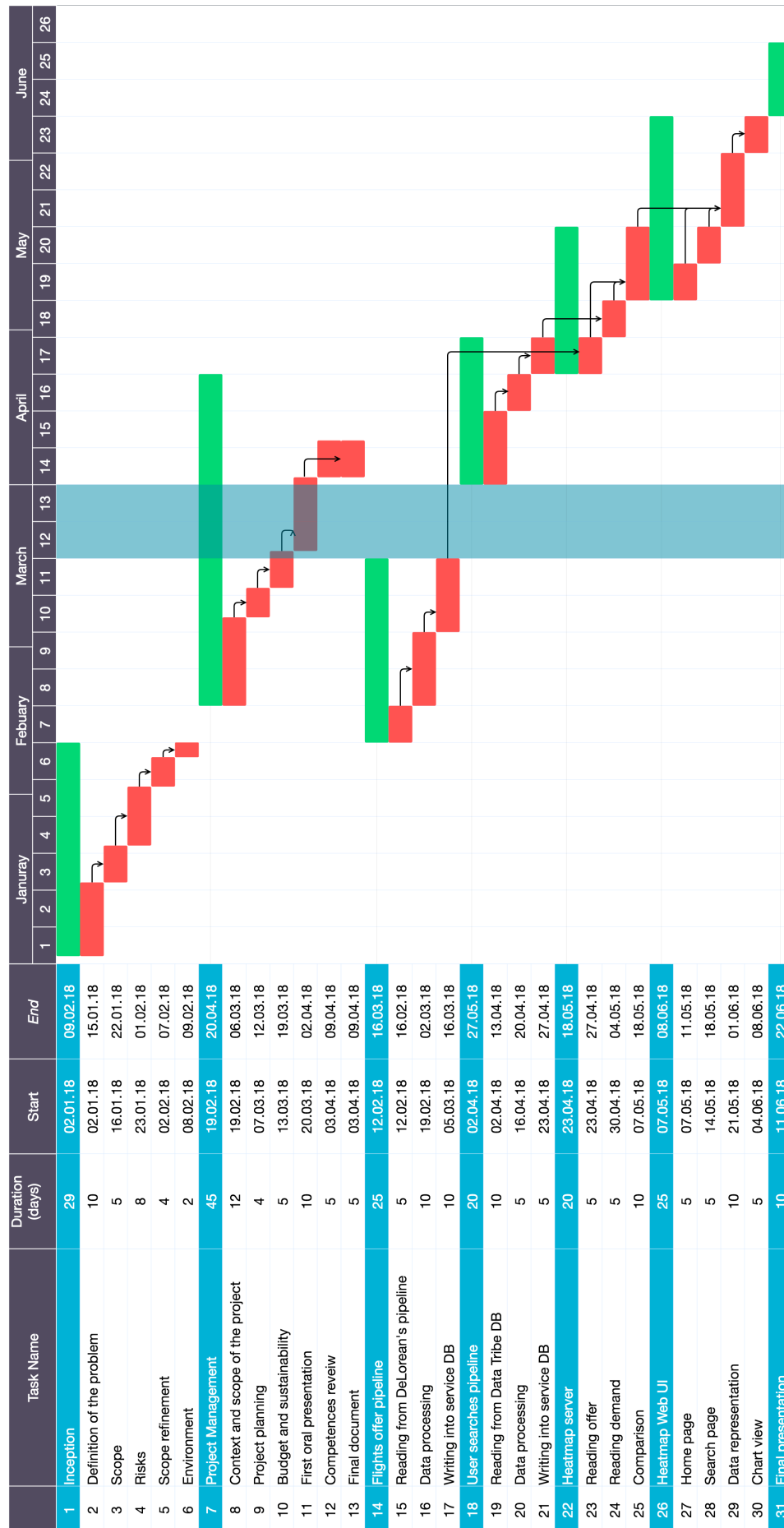
5.3 Gantt

FIGURE 5.1: Gantt diagram.

Appendix A

Skyscanner structure

Skyscanner has a very horizontal structure, based on Spotify's[9].

In the top of the company hierarchy there is Gareth Williams (CEO and Co-founder). 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[10].

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[11].

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*. Addison-Wesley Professional, 2005.

[8] Robert C. Martin. *The Clean Coder. A Code of Conduct for Professional Programmers*. Prentice Hall, 2011.

[9] 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).

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

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