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

Abstract **i**

1 Context **1**
1.1 Skyscanner . . . 1
1.2 Marketplace Engine tribe . . . 2
1.3 DeLorean squad . . . 2

2 State-of-the-art **3**
2.1 Fare aggregators and metasearch engines . . . 3
2.1.1 Google Flights . . . 3
2.1.2 Kayak . . . 3
2.1.3 Expedia . . . 3
2.2 Skyscanner services . . . 3
2.2.1 Marketplace Engine . . . 3
2.2.2 Data tribe . . . 4
2.2.3 The gap . . . 4
2.3 Resources and environment . . . 5

3 Skyscanner Heatmap **6**
3.1 Formulation of the problem . . . 6
3.2 Scope . . . 6
3.2.1 Pipeline . . . 7
3.2.2 Service . . . 7
3.2.3 Visual representation . . . 7
3.2.4 Not list . . . 8
3.3 Risks . . . 8
3.3.1 Routes contract . . . 8
3.3.2 Users information . . . 9
3.3.3 Amount of data . . . 9
3.3.4 Web UI . . . 9
3.4 Methodology and rigor . . . 9
3.4.1 Extreme Programming . . . 9
3.4.2 GitLab . . . 10

4 Stakeholders **11**
4.1 DeLorean squad . . . 11
4.1.1 Product Owner . . . 11
4.1.2 DeLorean's squad Lead . . . 11
4.2 Marketing Automation squad . . . 11
4.3 Data tribe . . . 12
4.4 Other Skyscanner developers . . . 12
4.5 OAG . . . 12
4.6 Providers . . . 12

4.7 Traveler 12

5 Project planning **13**
5.1 Tasks 13
5.1.1 Inception 14
5.1.2 Project management 15
5.1.3 Flights offer pipeline 16
5.1.4 User searches pipeline 17
5.1.5 Heatmap server 17
5.1.6 Heatmap Web UI 18
5.1.7 Final presentation 19
5.2 Current plan and alternatives 19
5.2.1 Current plan 19
5.2.2 Alternative: Overlap pipelines 19
5.2.3 Alternative: Service reading from Data Tribe 19
5.3 Gantt 21

6 Budget and Sustainability **22**
6.1 Budget 22
6.2 Sustainability 22
6.2.1 Economical 22
6.2.2 Social 23
6.2.3 Environment 23
6.2.4 Sustainability matrix 23

A Skyscanner structure **24**

Bibliography **25**

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**.

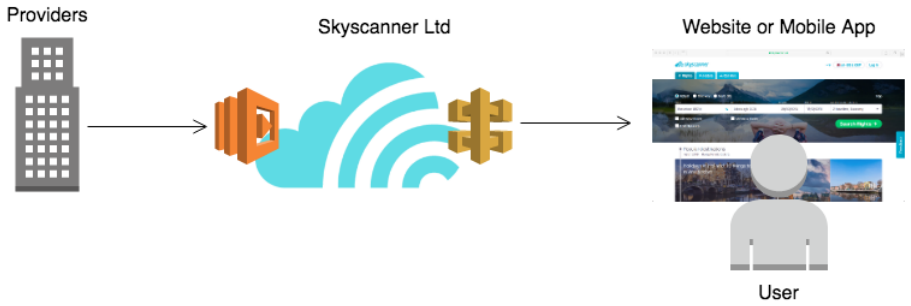


FIGURE 2.1: Simple explanation of Marketplace Engine data flow.

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*.

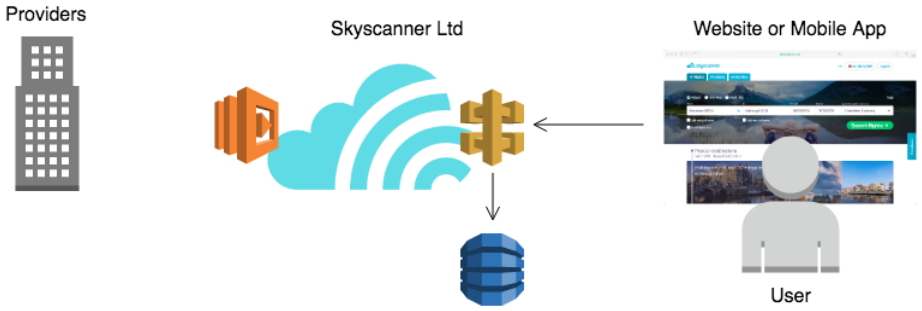


FIGURE 2.2: Simple explanation of Data tribe data flow.

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.

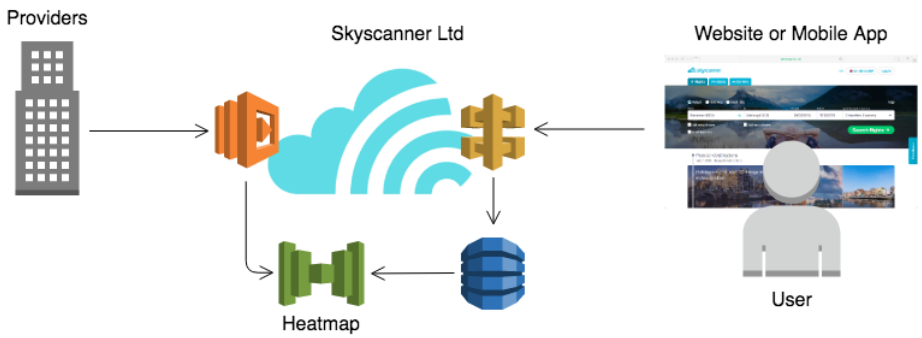


FIGURE 2.3: First approach of the Heatmap's data flow.

2.3 Resources and environment

Skyscanneris already in the air, it is working and has been working for more than ten years. In the past two years it has been migrating all their services to Amazon Web Services[5]. All developers from Skyscanner have access to all Amazon Web Services resources in a protected environment, Sandbox[1].

In this environment, the developer can use computing machines, storage, database, developer tools, management tools, machine learning frameworks, analytics software, etc. This project will mainly use computing machines, storage and databases.

Taking into account that Skyscanner stores all their data in Amazon Web Services databases and the Heatmap will be taking a part of it, we can guarantee that the storage will not run out of memory. Different project have isolated resources.

[1] A sandbox[6] is a testing environment that isolates untested code changes and outright experimentation from the production environment or repository, in the context of software development including Web development and revision control.

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.

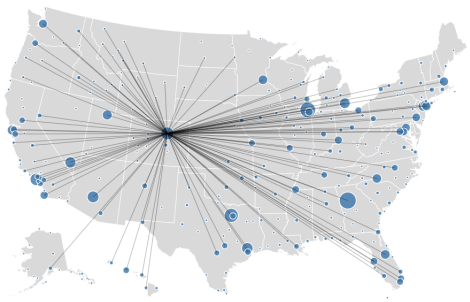


FIGURE 3.1: Example of the world map style. Only displaying the US to make it look clearer.

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.

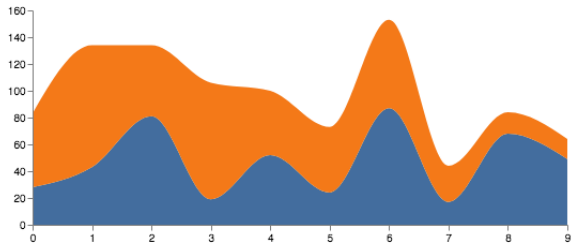


FIGURE 3.2: Chart mock-up. One color goes for Providers Offer and the other one for User Demand.

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[7], 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[2]. 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[8].

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)[9] 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. In order to have some feedback and update the requirements properly with the supervisor, Francisco López, approval, I will be attending all DeLorean squad

[2]Read more in section Resources and environment

stand up meetings and explaining my progress and, if necessary, create meetings with the product owner and the rest of the squad to take the project on the right track.

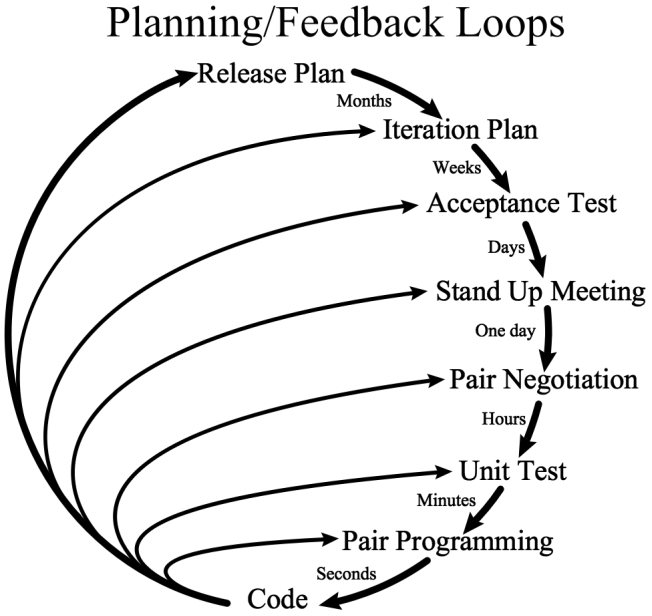


FIGURE 3.3: Extreme programming planning loops.

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.

4.1.1 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).

4.1.2 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 opportunity for the future (after project's delivery) 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 provider 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 Other Skyscanner developers

Last but not least, a new service will appear in Skyscanner, all developers will be able to use it and build software using the Heatmap's compared data. For instance, it can be used as a training for a complex Machine Learning[10].

The server Application Program Interface, used by the Web UI to visualize all data, will be public inside Skyscanner. This and all the documentation will be very helpful for developers.

4.5 OAG

OAG is a company that collects all logistic flights information. DeLorean squadreads data from them, it is the main provider of information regarding routes. They are the world's largest network of air travel data to provide accurate, timely, actionable digital information and applications to the world's airlines, airports, government agencies and travel-related service companies[11].

4.6 Providers

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 efficient, reducing number of flights in *non-profitable* routes. They will also know which are the best places to invest looking at *over-requested* airports.

4.7 Traveler

Skyscannerusers are one of the main sources of information. Without them, the comparison cannot be made. The results of the comparison can also help them, not directly, but if providers somehow manage flights and routes following the Heatmap results, the traveler experience will improve.

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[12]: Task 27, Home Page from the Web UI. It will have an HyperText Markup Language part only for the view and a Java Script part for getting and displaying server's data, 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.

Tasks are grouped in **Milestone**. Used to mark specific points or goals along of a project *timeline*. Milestone 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[13] not to work to much time in the same thing, task or goal. Swapping between different tasks and milestone can help so the developer **does not obfuscate**.

5.1 Tasks

All development tasks (from number 14 to 30), is composed by small cycles where the developer will be designing, coding, testing, refactoring and deploying the software. In every task the developer will be looping through these cycles, as explained in Methodology and rigor section.

Development tasks 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[1]). 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. Adding the small inception of the beginning and the project

[1]There are two holiday weeks in March, one is *Holy week* (13th), and the other is extra. That is why the project starts earlier.

report in the end, 8 weeks, The whole project will take a total of **575 hours**.

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)[14]
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 Current plan and alternatives

5.2.1 Current plan

In the current plan, the whole system is developed from two data layers to a presentation layer, with a domain layer in the middle. Data layer composed by Flights offer pipeline and User searches pipeline, domain layer by Heatmap server and presentation by Heatmap Web UI.

In this project the important part is the source of the data. So it is obvious that first of all, the data want to be obtained correctly, then we can worry about its visual representation.

Then, why the Flights offer pipeline comes before the User searches pipeline? The answer is simple: Flights offer pipeline gets data from DeLorean squad, my squad. I know how their system works and, if all initial problems are found in the beginning, the consequences will be softer.

Flights offer and user searches pipelines $\longrightarrow$ Heatmap server $\longrightarrow$ Heatmap Web UI

5.2.2 Alternative: Overlap pipelines

In case the project takes too much time, both pipelines can be totally overlapped. It will be possible because the process of what both pipelines do are the same but with different sources. There are **no dependencies** between them.

The first stage of the pipelines, those are reading from their source. Then, both pipelines map the data to the desired model. Finally the pipelines write in to the service database.

This will reduce the number of weeks from fifteen to **eleven weeks**. All milestone User searches pipeline would be done at the same time as milestone Flights offer pipeline.

5.2.3 Alternative: Service reading from Data Tribe

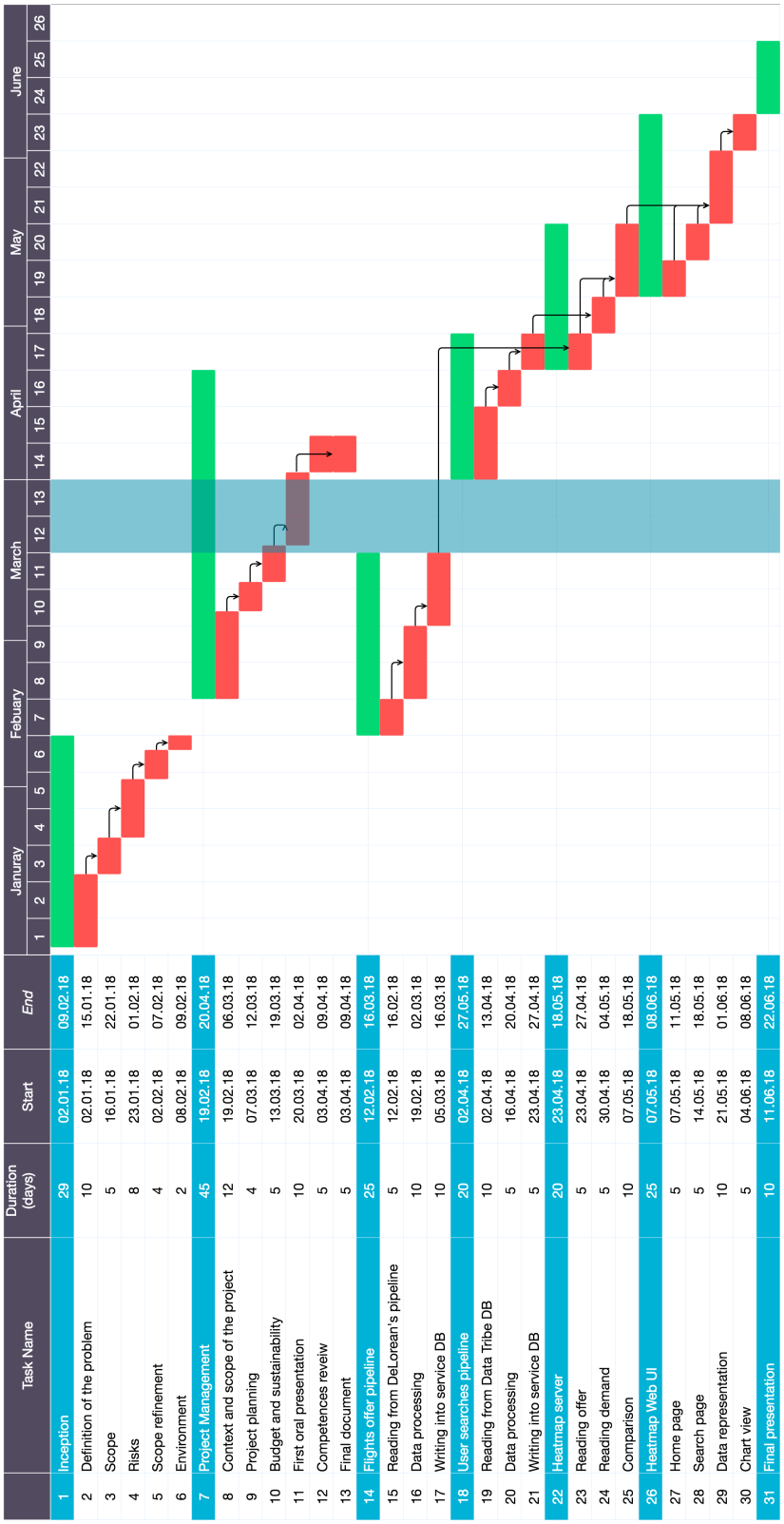
Another alternative, is to remove the second pipeline. It is known that in order to get all flights, data must be flattened and, then, processed. This cannot be done in the service, it would overflow its heap memory. This means that the Flights offer pipeline cannot be removed.

User searches pipeline will read data from a database that has the data already flattened. It has a lot of not necessary information, but filtering from a single record is faster than exploding and consumes much **less memory**. The service could do the users search data processing.

One problem is that the server latency will increase a lot. It will be reading from a very slow database for single queries (Data Tribe's database work very well when querying big amounts of data, but not small ones). Also, Data Tribe people will provably complain, because their service is not created for to much requests per second and is exactly what will happen if the services maps their data without a pipeline in the middle.

This option reduces the development time to **ten weeks** instead of fifteen.

5.3 Gantt



	Task Name	Duration (days)	Start	End
1	Inception	29	02.01.18	09.02.18
2	Definition of the problem	10	02.01.18	15.01.18
3	Scope	5	16.01.18	22.01.18
4	Risks	8	23.01.18	01.02.18
5	Scope refinement	4	02.02.18	07.02.18
6	Environment	2	08.02.18	09.02.18
7	Project Management	45	19.02.18	20.04.18
8	Context and scope of the project	12	19.02.18	06.03.18
9	Project planning	4	07.03.18	12.03.18
10	Budget and sustainability	5	13.03.18	19.03.18
11	First oral presentation	10	20.03.18	02.04.18
12	Competences reveiw	5	03.04.18	09.04.18
13	Final document	5	03.04.18	09.04.18
14	Flights offer pipeline	25	12.02.18	16.03.18
15	Reading from DeLorean's pipeline	5	12.02.18	16.02.18
16	Data processing	10	19.02.18	02.03.18
17	Writing into service DB	10	05.03.18	16.03.18
18	User searches pipeline	20	02.04.18	27.05.18
19	Reading from Data Tribe DB	10	02.04.18	13.04.18
20	Data processing	5	16.04.18	20.04.18
21	Writing into service DB	5	23.04.18	27.04.18
22	Heatmap server	20	23.04.18	18.05.18
23	Reading offer	5	23.04.18	27.04.18
24	Reading demand	5	30.04.18	04.05.18
25	Comparison	10	07.05.18	18.05.18
26	Heatmap Web UI	25	07.05.18	08.06.18
27	Home page	5	07.05.18	11.05.18
28	Search page	5	14.05.18	18.05.18
29	Data representation	10	21.05.18	01.06.18
30	Chart view	5	04.06.18	08.06.18
31	Final presentation	10	11.06.18	22.06.18

FIGURE 5.1: Gantt diagram.

Chapter 6

Budget and Sustainability

6.1 Budget

Once the project is planned in time and its technologies are drafted, we can calculate the project's budget. Skyscanner is not economically transparent, even for the employees. So, this whole calculation will be approximated.

First of all, we have to take into account the employees, which is only one and in Intern position, and also count the taxes.

Apart from that, all the hardware material and software licenses. AWS costs will continue but since it is going to be used during the development for testing it is counted in the total budget.

TABLE 6.1: Budget calculation

Concept	**Price per unit**	**Units**	**Amortization**	**Total £**
Salary	28 (taxes included)	575 h		16100
MacBook Pro	1700	1	life cycle 8 years	210
JetBrains License	230	1	1 year per developer	130
AWS S3	0.023	50 TB		1150
AWS	EC2	575 h		213.325
Screen	200	2	life cycle 4 years	100
Office	Uknown	1		Uknown
TOTAL				**17903.325**

6.2 Sustainability

6.2.1 Economical

In economic terms, this project is initially unsustainable. It uses resources from Skyscanner for a comparison that might be useful in the future for other projects, improving some services or advertisement.

But, if this product is sell to providers, Skyscanner can take a lot of profit from them. It is a very valuable application for providers, since they could compare airlines offer with actual user demand. Letting them improve their flights distribution and make more money.

6.2.2 Social

The Heatmap is not directly involving society, but, as explained before, if providers have access to the comparison, flights will improve in terms of traveler experience. Travelers will have accurate routes depending on what they really want.

For example, imagine that X carrier have several flights from BCN to ORY, Paris, and a few from BCN to FCO, Rome. The Heatmap shows that the demand, compared with the offer is bigger in Rome than in Paris. Then, X airline could schedule more flights to FCO instead of ORY.

6.2.3 Environment

The environmental impact of the Heatmap is directly related with the social impact.

Right now, some airlines may have half full flights. This means that the airplane is not taking its most advantage of the fuel. It could be carrying more people.

If carriers know where flights are really needed those flight will be full of people, which means that the fuel a flight uses is profited at its most.

Otherwise, if an offer is under requested, the flight is not giving all the profit it could. In other words, fuel per person will decrease.

6.2.4 Sustainability matrix

In order to understand the general impact of the project, the following general rating and evaluation is provided:

The economical impact will be rated in 7/10. It could be a 10/10 it is sell to providers. Air companies could pay a lot of money for the Heatmap because of the information it provides.

The social impact will get a 4/10. It does nothing good nor bad to the society, only if the application is sell to providers and they use it properly, it could make some good to the people. In the other hand, the software will not be free, it will be property of Skyscanner.

The environmental impact is a 4/10 as well. The environmental impact could be good if the application is sell to providers and they use it properly, but for now will not be sell to anyone. It gets a 4 because it will be using Amazon Web Service, and those machines are powered mainly by non-renewable energy[15].

TABLE 6.2: Sustainability matrix

Economical	Social	Environmental
7/10	4/10	4/10
15/30		

Appendix A

Skyscanner structure

Skyscanner has a very horizontal structure, based on Spotify's[16].

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[17].

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[18].

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] Inc. Amazon Web Services. *Amazon Web Services (AWS) Documentation*. 2018. URL: https://aws.amazon.com/documentation/ (visited on 03/12/2018).

[6] Multiple authors. *Sandbox (software development)*. 2018. URL: https://en.wikipedia.org/wiki/Sandbox_(software_development) (visited on 03/12/2018).

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

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

[9] Kent Beck. *Extreme Programming Explained. Embrace Change*. Addison-Wesley Professional, 2005.

[10] Stanford University. *Machine Learning*. 2018. URL: https://www.coursera.org/learn/machine-learning (visited on 03/12/2018).

[11] OAG Aviation Worldwide Limited. *OAG: Connecting the World of Travel*. 2018. URL: https://www.oag.com/about-oag (visited on 03/12/2018).

[12] W3.CSS Website Templates. *Some responsive W3.CSS website templates to use.* 2018. URL: https://www.w3schools.com/w3css/w3css_templates.asp (visited on 03/12/2018).

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

[14] Rubrics in English. *GEP rubrics in English*. 2018. URL: http://atenea.upc.edu/mod/folder/view.php?id=1605178 (visited on 03/12/2018).

[15] Greenpeace. *Clickclean 2016*. 2016. URL: http://www.clickclean.org/international/en/ (visited on 04/09/2018).

[16] 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).

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

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