REPORT phase 2

Realized By:

BENZARTI Sourour

MISSAOUI Houssem

LAAMIRI Oussama

ALFALAH Taieb

Subject : SMART CITY

Class : 4 Twin 2
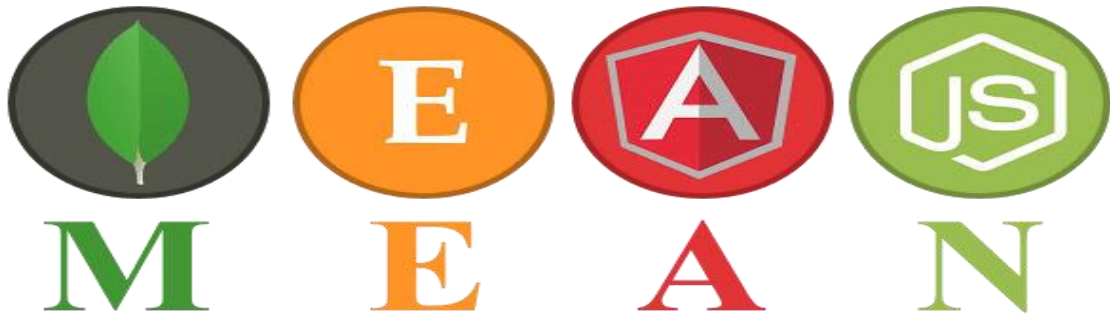
Group : Burial

Table of Contents

I.   Architectural choices

1. MEAN Architecture

The MEAN stack is a popular stack of technologies used to create web applications. However, you should be aware of the fact that many of these technologies are interchangeable. For example, you could use Backbone, Ember, or React on the frontend instead of Angular. You could use Hapi instead of Express, and MySQL of Postgres instead of Mongo. Each of these technologies has certain advantages and drawbacks.
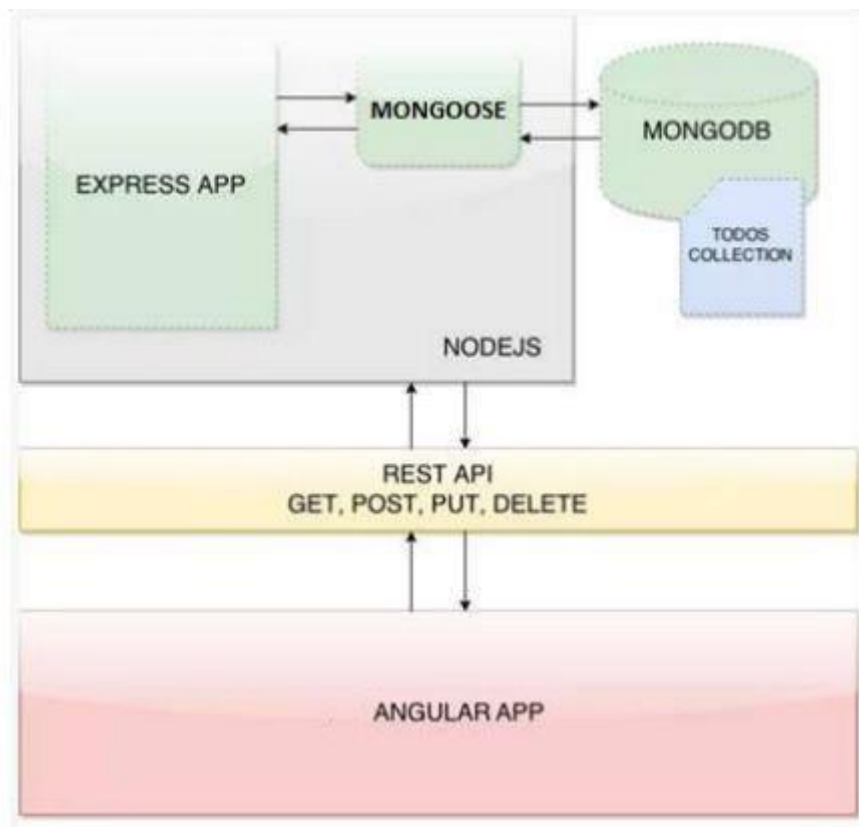
MongoDB **for the database:** MongoDB is the leading cross-platform, document-oriented, NoSQL database that empowered businesses to be more agile and scalable. A number of major websites and services have adopted MongoDB as backend software, including: Craigslist, eBay, and Foursquare.

It is simpler to update the entire table in Mongo DB. In Mongo DB, the documents bear a resemblance to objects in an object-oriented programming language.

Express.js **for the web framework:** Express is a minimal, flexible node.js web application framework. Express provides a robust set of features for building single-page, multi-page, and hybrid applications.

AngularJS **for the frontend framework:** AngularJS is an open-source framework (largely supported by Google) that allows you to extend the HTML vocabulary for your application—resulting in an environment that's astonishingly readable, expressive, and quick to develop.

Node.js **for the server platform:** Node.js is a platform built on Chrome's JavaScript runtime to enable building quick scalable network applications. it is more than a traditional web server. Making use of Node.js in your technology stack is obviously a great advantage. Node.js operates non-blocking input-output to deal with new incoming requests proficiently. Moreover, Node.js makes use of web sockets to empower sending data to the client without the client having to request it.

❖ What makes MEAN so great?

- Use one, single language (JavaScript) for both server-side and client-side execution environments.
- JSON is everywhere, and that's a good thing.
- Node.js is really fast and simplifies the server layer.
- Has unrivaled support from industry leaders.
- Open source so the savvy can tweak it to his or her preference.

❖ **JavaScript All along**

StackOverflow Tag Followers (as of February 2016)

Language	Followers
JavaScript	200.1K
Java	191.9K
PHP	125.1K
C#	117.9K
Python	112K
C++	90.3K
C	67.7K
SQL	50.5K
Ruby	31.6K
Swift	20.9K
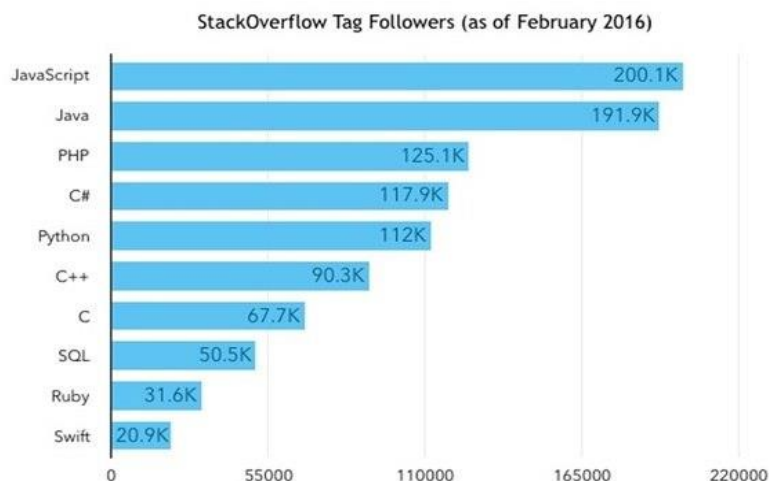
The primary reason for preferring MEAN stack over others is because of the use of a common language for both client-side and server-side. Since, all these technologies are written in JavaScript, your web development process becomes neat. Node.js is a platform that brought JavaScript implementation on the server-side. With Angular.js, using JavaScript on the front-end, it becomes easier to reuse code from back-end to front-end.

Since, all these technologies are written in JavaScript, your web development process becomes neat.

❖ **Cost-Effective**

For the large scale organization MEAN.JS is ratio going technology as they do not require to hire different specialists to work on a single project. MEAN stack has also introduced a new career opportunity to full-stack JavaScript developer.

❖ **Free and Open-Source**

All the technologies available in the MEAN stack open-source in nature, so it is obviously free. Thus, you are benefited by various developments and version upgrades by a vast community of developers.

❖ **Everything uses JavaScript - even the database!**
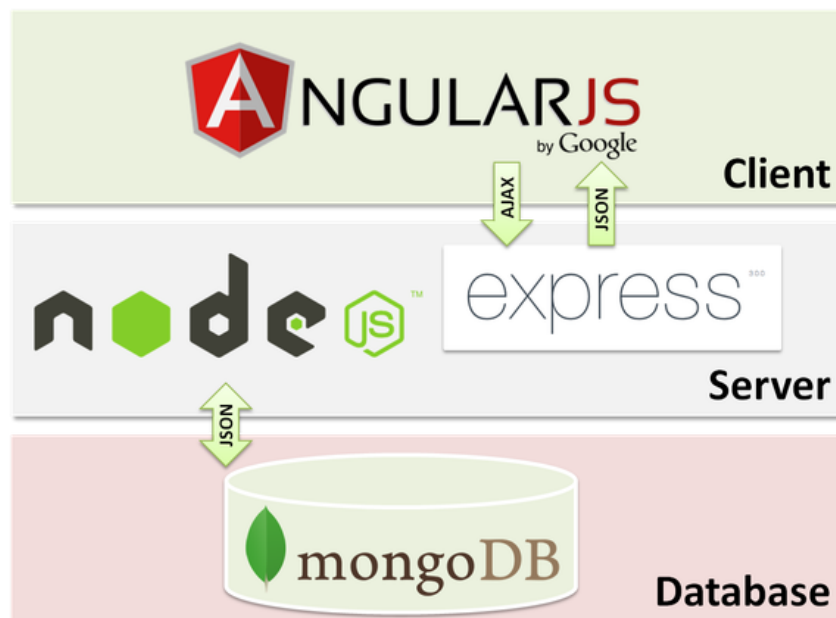
This allows front end developers, back end developers, and even semi-technical designers to collaborate. Being able to read and understand your colleague's code is important.

2. LAMP (Linux, Apache, MySQL, PHP)

It is one of the earliest stacks to get traction and the most traditional stack mode, is made up of all free, open-source software elements that work well for Dynamic web sites and applications. LAMP includes the Linux operating system, PHP application software, MySQL database and the Apache Web server. It's flexible, easy to develop applications with, easy to deploy, customizable, secure, and comes with a huge support community. LAMP's open source nature is also one of its greatest benefits

3. LAMP Vs MEAN

Technology	LAMP	MEAN
Languages	Perl/PHP/Python: Commonly used programming languages.	AngularJS: Extension of Javascript
Database	MySQL (RDBMS): Conventional database system	MongoDB(NoSQL database): A cross-platform document-oriented database system. JSON-style documents with dynamic schemas provide simplicity and power, making the integration of data into certain applications (particularly Javascript-based ones) fast and easy.
Web Server	Apache: Commonly used.	ExpressJS: is a Node.js Web application framework. It creates an MVC-like application on the server side. It also allows users to create routes and templates.
Servers	"LAMP" is derived from [OS]: Linux (L), Macintosh (M), or Windows (W) Apache web server (A) MySQL (M) PHP (often now also Python and Perl) (P) Apache is used to host HTTP files, MySQL for databases, and PHP/Python/Perl for Programming language used for creating dynamic webpages.	"MEAN" is derived from MongoDB (M), ExpressJS (E), AngularJS (A), and Node.js (N). Node.js is the server that runs your application. Node.js is an event-driven I/O server-side JavaScript environment.

summary

MEAN stack offers a modern approach to web development. It also makes use of the power of modern SPAs (single-page applications), which does not require entirely refreshing a web page for every server request like most traditional web applications do.

It is important to note that the technologies in the LAMP architecture were not written to work together to form a full-stack solution. Though, today there is a trend to develop multiple frameworks that correspond to a single programming language and form a full-stack solution. But, as of now, using MEAN stack is a very effective approach to web development.

II.   Other useful tools

Ever since version control systems such as **Git** have become widely-known and well-used, modern development processes have radically changed.

• It allows you to have several versions of a project at the same time

• It allows more than one developer to work on a project

• It allows saving copies of project for backups

Git is used by more and more well-known companies and Open Source projects: Ruby On Rails, jQuery, Perl, Debian, the Linux Kernel and many more.

mLab is a fully managed cloud database service featuring automated provisioning and scaling of MongoDB databases, backup and recovery, 24/7 monitoring and alerting, web-based management tools, and expert support. mLab's Database-as-a-Service platform powers hundreds of thousands of databases across AWS, Azure, and Google and allows developers to focus their attention on product development instead of operations.

III.   Innovation

We are transforming the use of Ibeacon from only shopping area to many other areas like restaurant, museums and airports.

 We are also using Ibeacon not only as a fixed device but also as a mobile one.

Our solution is giving guidance and advices through notifications that allows you to get to know the city and people's way of life better.

The target customers of our project are tourists and citizens that don't know the city well.

So, our Project have for goal to improve the discovery of the city and assure guidance depending on the preferences and keep them safe from being lost.

We want to make a mobile application that works for our clients on which they can receive notifications and visualize pictures and information they need.

It will work on both Android and IOS since they are the most used mobile operating systems and will be connected to the Ibeacons Distributed throughout the city.

We will also develop a web platform that will allow us to read the feedback of our users, see statistics about clients' activities per month, per season …and generate reports.

IV.   IBeacons - proof of concept

Beacons have been generating buzz since 2013, when Apple first introduced iBeacon technology. And while it may have appeared for a time that this new way of connecting with customers might be slow to catch on, today it's catching fire

The possibilities are endless! Beacons can be used to develop apps that feature contactless payments, educational information in exhibits, and more.

Stickers placed on objects allow these objects to communicate contextual information, opening a new world of possibilities.

Inside each Beacon and Sticker is a non-rechargeable lithium battery. A beacon's lifetime depends on its configuration. At default factory settings, Beacons last anywhere from 3-5 years. You can extend the lifetime by utilizing conditional broadcasting, and batteries are easily replaceable. Stickers last up to one year, and batteries are not replaceable.

What is a Beacon's range?

The typical range of Bluetooth low-energy radio modules is anywhere from 1 meter (3 ft) up to 200 meters (650 ft). Of course, it depends on the location and circumstances, because radio signals could be absorbed or diffracted.
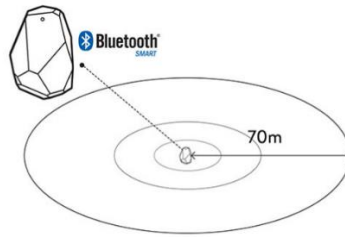
Can beacons be placed outdoors?

Tough Beacon, Designed for Outdoor Use

- Shatter-resistant
- Waterproof up to 10 m
- Works from -20C (-4 F) – +60C (140F)
- Anti-static
- UV-resistant
- 5 – 7 week production lead time

The Tough Beacon is Kontakt.io's newest beacon. Created using the same award-winning bluetooth BLE smart beacon technology as our normal beacon, the Tough Beacon is encased in a special polycarbonate blend that makes it, well, tougher than the average beacon. Virtually shatter-proof (as long as you don't drive a tank over it), completely waterproof, hermetically sealed, anti-static, UV resistant, and operable through a wide range of temperatures, the Tough Beacon is ready for all of your outdoor, extreme environment proximity needs.

Due to the fact that the Tough Beacon needs to be built and configured in one factory and then shipped to a special facility to have its case fused around it with microwaves to seal it forever, the lead time for Tough Beacon is 5 – 7 weeks from order to delivery. Also, as the case is sealed, the battery is not user-replaceable.

the battery life of a Tough Beacon is exactly the same as of our regular beacon - which is 2 years with a default settings (350ms interval) and 24-hours daily usage.

What kind of apps can be developed with beacons?

Our Beacons and Stickers just send the Bluetooth signal, it's the custom-built app that will trigger the action on a user's screen, based on proximity to a particular beacon. Theu are provides with the official Software Development Kit (SDK) for building iOS and Android apps and the technical support.

### Which devices are compatible?

All Bluetooth-Low-Energy–enabled devices can pick up the beacon signal.

These include Apple devices from iPhone 4S onwards and Android devices like: Samsung Galaxy S devices, Google Nexus, Google Glass, and many more

V.   Work Plan

- Phase 1: Static view with Angular JS
- Phase II: Implementation of the Data
- Phase III: Rest API/Controllers
- Phase VI: Finalization

BURIAL
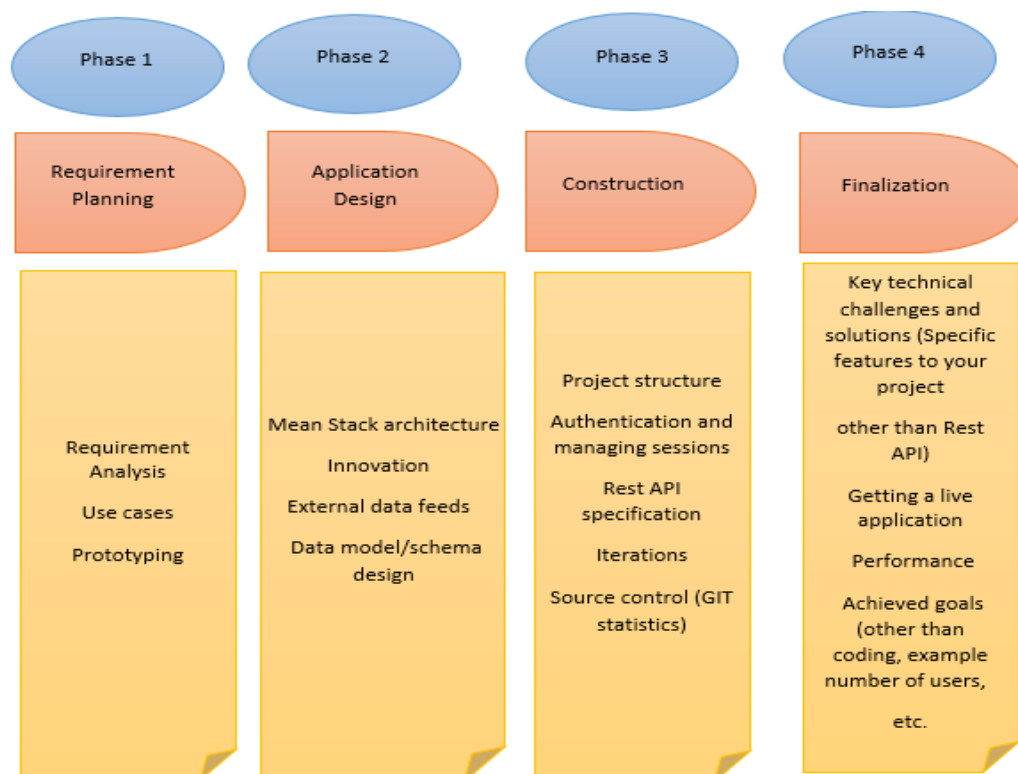
VI.   External data feeds - feasibility study

1. Foursquare

Foursquare is a social location service that allows users to explore the world around them. Users can download the Foursquare application to their iPhone, Blackberry, or Android phone and sign up for free, then connect their Foursquare accounts to their other social media accounts.

The Foursquare API allows application developers to interact with the Foursquare platform. The API itself is a RESTful set of addresses

Method	Summary	URL	Parameters	HTTP method(s)	Authentication?
Check-ins	Returns a list of recent check-ins from friends.	http://api.foursquare.com/v1/checkins	geolat, geolong	GET	Yes
History	Returns a list of check-ins for the authenticated user.	http://api.foursquare.com/v1/history	l, sinceid	GET	Yes
User detail	Returns profile information for a given user ID.	http://api.foursquare.com/v1/user	uid, twitter, badges, mayor	GET	Yes
Friends	Returns a list of friends.	http://api.foursquare.com/v1/friends	uid	GET	Yes
Venues	Returns a list of venues near the area specified.	http://api.foursquare.com/v1/venues	geolat, geolong, l, q	GET	No
Venue detail	Returns venue data for a given venue ID.	http://api.foursquare.com/v1/venue	vid	GET	No
Categories	Returns a hierarchical category list.	http://api.foursquare.com/v1/categories	n/a	GET	No

2. Dark Sky

The Dark Sky API allows you to look up the weather anywhere on the globe, returning (where available):

- Current weather conditions
- Minute-by-minute forecasts out to one hour
- Hour-by-hour and day-by-day forecasts out to seven days
- Hour-by-hour and day-by-day observations going back decades

Method	Endpoint	Description
GET	/key	Dark Sky secret key.
GET	/ latitude	The latitude of a location (in decimal degrees)
GET	/longitude	The longitude of a location (in decimal degrees). Positive is east, negative is west.

3. Google Places

The Google Places Web Service is a service that returns information about a "place": an establishment, a geographic location, or prominent point of interest using an HTTP request. Place requests specify locations as latitude/longitude coordinates.

The link below contains the different methods we need to consume the API

https://developers.google.com/places/web-service/search

4. AerisWeather

Supported Actions :

:id	Requesting data by using the *:id* action is used for returning data for a particular item that has an ID associated with it. This is the primary method for requesting general weather information for a single location (observations, forecasts, advisories, etc.) as the location's name or a zip code serves as the id. Other endpoints may expect a certain value for the ID, such as storm cells whose ID value is a combination of the radar site identifier and unique identifier assigned to every storm cell. Refer to an enpoint's detailed documentation for specific information regarding how to use the *:id* action.
closest	The *closest* action will query the API for data that is closest to the requested place and return the results, if any, in order from closest to farthest. If no limit is provided in the request, then only the closest single result will be returned. If a radius is not provided, then the default of 20 miles will be used. If your request does not return results, you may try setting or increasing the radius being used. Note, however, that a maximum of 250 results can be returned in a single request.
within	The *within* action allows for returning data within a variety of different geometrical regions. Currently supported geometries include a circle, square and polygon. Unlike the *closest* action, the results will not be returned in any particular order based on distance.
search	The *search* action is used as a more general query method and expects the **query** to be defined with the custom query for the request. Unlike the *closest* action, results will not be returned in any particular order.

Supported Filters

Airport	Returns all airports, including airbases, airfields and heliports.
Amusement	Returns all theme and adventure parks.
Bridge	Returns all structures erected across an obstacle, such as a stream, road, etc., in order to carry roads, rail and pedestrians.
Camp	Returns all sites occupied by tents, hunts, or other shelters for temporary use.
Church	Returns all buildings used for public christian worship.
Country	Returns French countries (Grenoble in Isère) .
Divisions	Returns smaller US towns, non-incorporation locations such as census divisions.
Feature	Returns all natural geological features.
Fort	Returns all defensive structures or earthworks.
Golf	Returns all golf courses.
Lake	Returns all lakes, including crater lakes, salt lakes, oxbow lakes and underground lakes.
Neighborhood	Returns smaller US non-incorporation locations such as census divisions, local areas of a city etc.
Parish	Alias for the county filter.
Park	Returns all parks and other areas maintained as a place of recreation.
Poi	Returns all categories of places across all filters.

Port	Returns all places that transfer waterborne cargo or passengers, usually in a harbor.
Ppl	Returns all populated places, including larger cities, towns and smaller locations and neighborhoods.
Reserve	Returns all tracts of public land reserved for future use or restricted use, including agricultural, forest, hunting, nature, reservation and wildlife.
School	Returns all schools, including colleges, military, maritime and technical.
Stadium	Returns all facilities used for athletic games and spectators.
Temple	Returns all places used for religious worship.
Trail	Returns all paths, tracks or routes used by pedestrians, animals, or off-road vehicles.
Tunnel	Returns all subterranean passageways used for transportation, including natural caves, road and rail passages.
University	Returns all institutions for higher learning, including prep schools.

5. Facebook

The Graph API is the primary way to get data in and out of Facebook's social graph. It's a low-level HTTP-based API that is used to query data, post new stories, upload photos and a variety of other tasks that an app might need to do.

Facebook Graph API

This guide explains how to accomplish all these things in the Graph API.

6. Paris API

The Paris Connect API allows read access to the data made available by Digital Paris. Access to these data requires:

RESPONSES

The API response is in the JSON format with the following schema:

- Status: the status of the response, can be set to "success" or "error".

- Data: query data, returns an object on success or "null" on error.

- Message: The detail of the error or "null" on success.

The schema contains two additional entries if successful:

- RequestTime: the response time of the query

- Api-version: the version of the API used.

Proposed requests

Request	Method	Parameters
https://api.paris.fr/api/data/2.1/ QueFaire/get_categories	GET	Token
https://api.paris.fr/api/data/2.1/ QueFaire/get_tags/	GET	Token
https://api.paris.fr/api/data/2.2/ QueFaire/get_events/	GET	Token Tags Categories

7. GuideMEAPI

 allows users to access the data, the site provides all sorts of information on Paris including information on public transportation, cultural events, museums, wifi hotspots, education, urban planning, parks, and more.
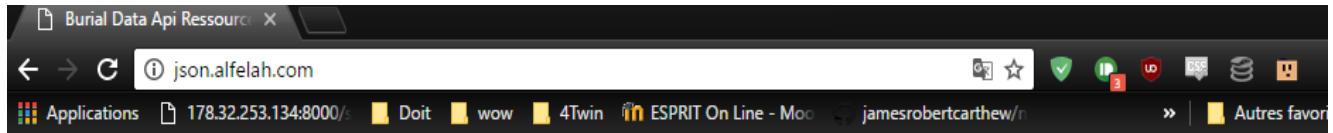
About

GuideMEAPI allows users to access the data, the site provides all sorts of information on Paris including information on public transportation, cultural events, museums, wifi hotspots, education, urban planning, parks, and more.
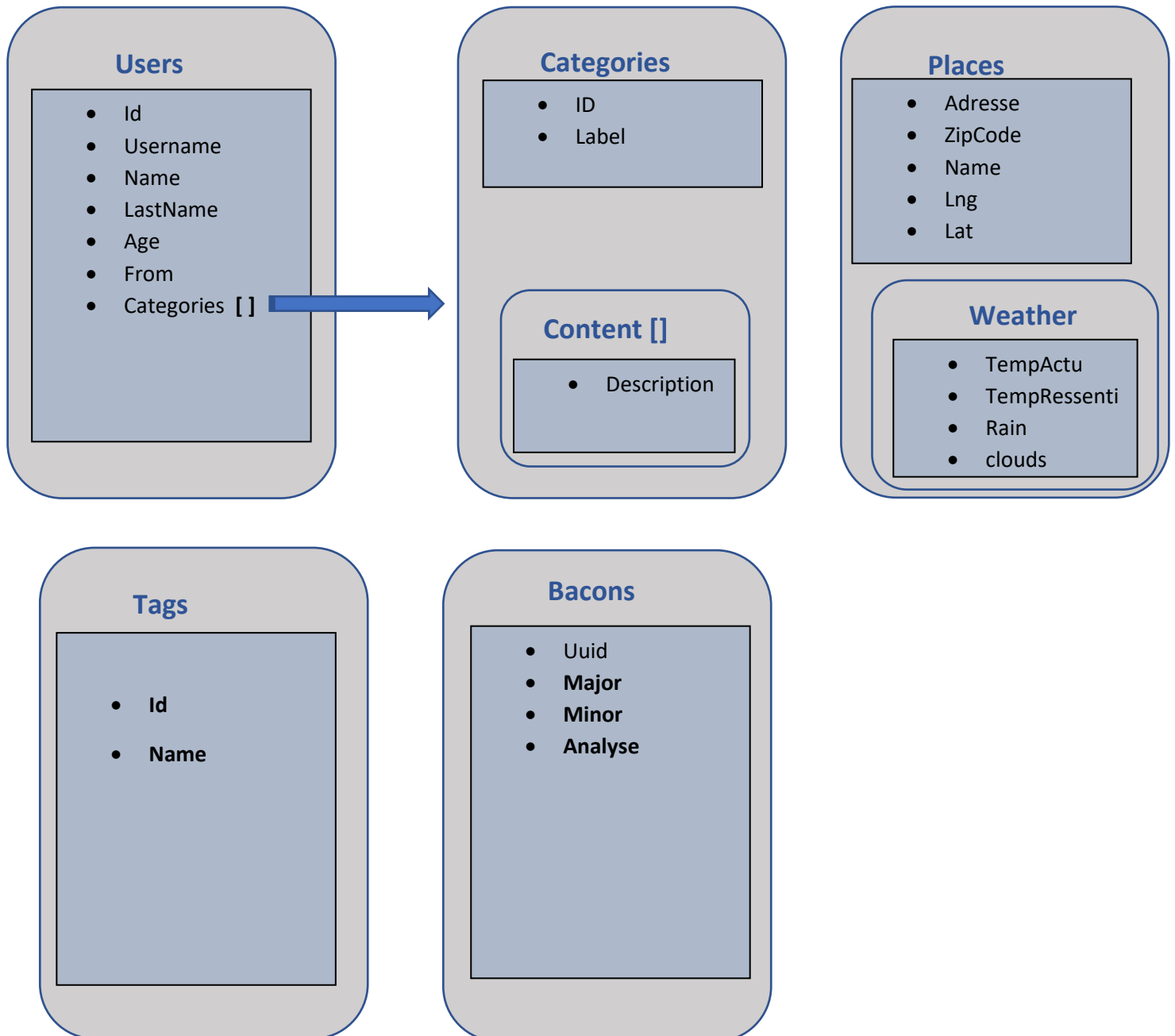
Github

Data

[JSON] All Paris Museums
[JSON] All Paris Cinema
[JSON] All Paris Monument
[JSON] All Paris Bar
[JSON] All Paris Coffee
[JSON] All Paris Cultural Events..

BURIAL✈

VII.   Data model

As a team we created a mongo Database named "GuideMe", that will contains many collections specific to each feature of our application.

Users
- Id
- Username
- Name
- LastName
- Age
- From
- Categories **[ ]**

Categories
- ID
- Label

Content []
- Description

Places
- Adresse
- ZipCode
- Name
- Lng
- Lat

Weather
- TempActu
- TempRessenti
- Rain
- clouds

Tags
- **Id**
- **Name**

Bacons
- Uuid
- **Major**
- **Minor**
- **Analyse**

Conclusion

This phase is essentially refining and validate organizational models: flow, treatments and data. Using and filtering all gathered data, and developing most of the specific features we laid the ground for the next phase which is the implementation of our app server side and client side combined.