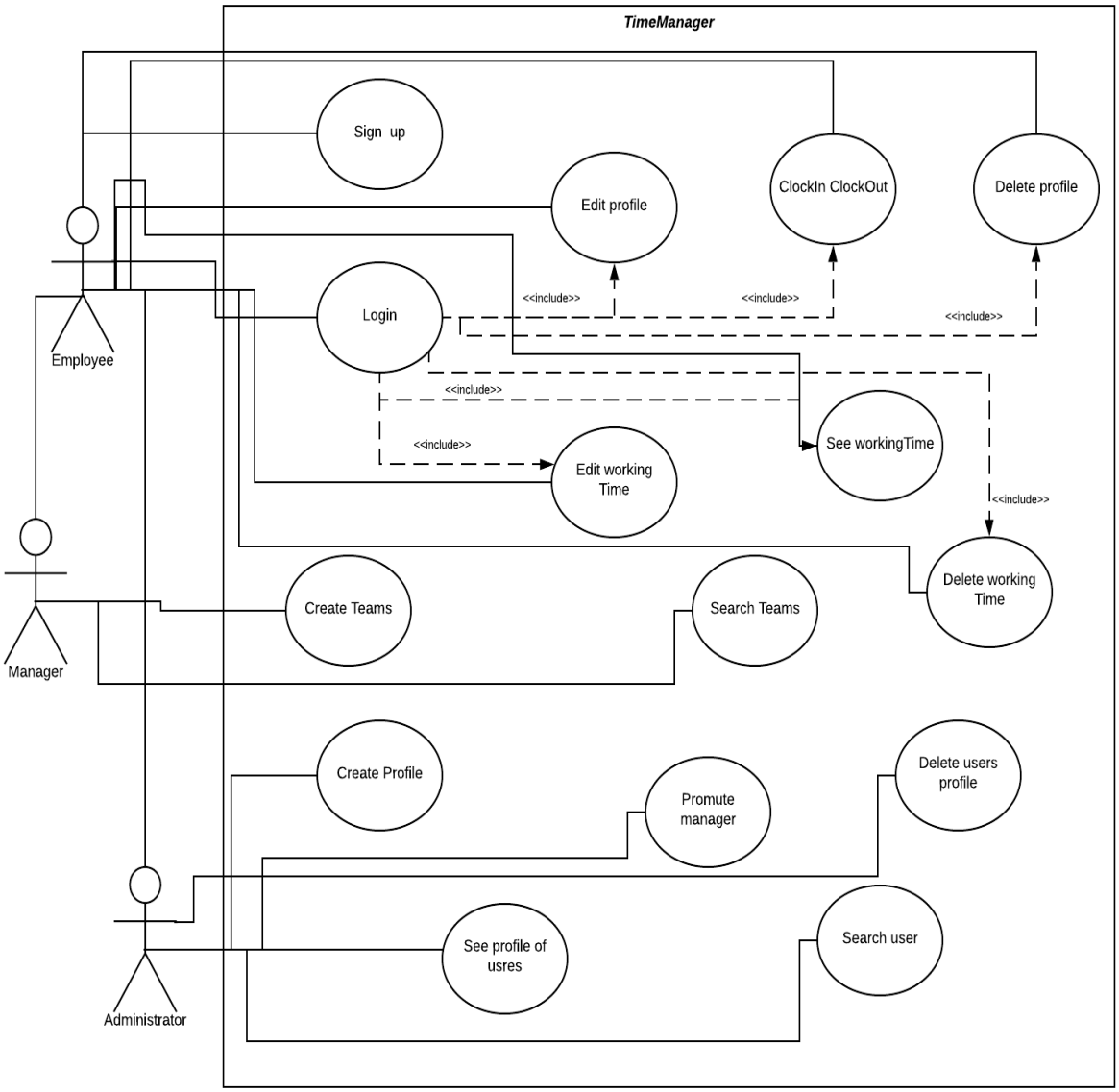
Functional Requirements Document

I. INTRODUCTION

The situation take place in Gotham City, the population is in crisis and protest every day about the working conditions, long hours, the night work to often. To calm the tension the twon hall decides to make a state of the situation and why not to allow the municipal workers to have access to an applications wich inform them of they hours of works. For this we needs to set up a time management application.

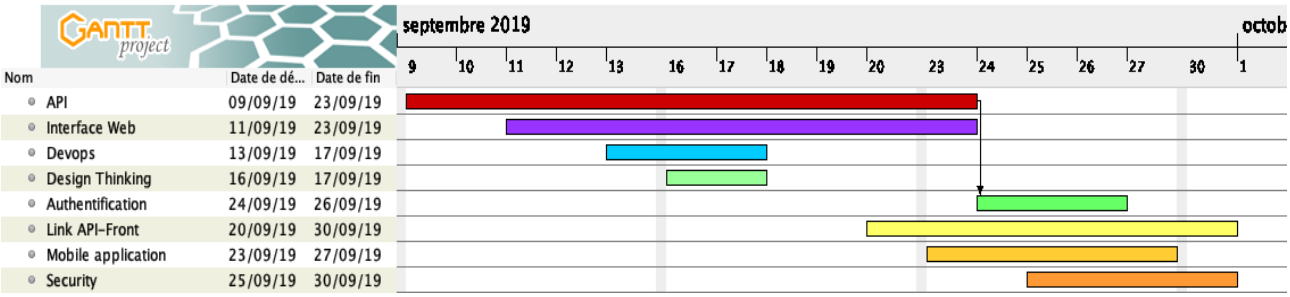
II. DIAGRAMS

a. Use case diagram



Use case diagram Time Manager

b. Gantt diagram



Gantt diagram Time Manager

III. FONCTIONAL REQUIREMENTS

In the application Time Manager we have three categories oif users :

- *Employees :* municipal worker who work for Gotham City
- *Managers :* who create and manage teams
- *Administrator :* who manage users profile

All users can :

- Sign up to create a new account
- Login to their account
- Logout
- Edit thier account
- Delete their account
- Report their arrival and departure times
- View their working times in tabs and graphs
- Edit or delete his working times

The manager can :

- Create teams
- Search teams
- Edit teams
- View their teams

The administrator can :

- Create users profile
- Search users profile
- Promote employee to managers
- View users profile

IV. INTERFACE REQUIREMENTS

We have six templates in Time Manager application :

- ***Home :*** This is the dashbord of users, he can see his profile, if he is an employee he can see his informations about the teams he is in, he also can see graphs of his working times. If he is a manager he can see the same things as

an employee but he can also see his teams and manage his teams, and if he is an administrator he can see the same things as an employee but he can also see users profile and create, edit or delete them.

- ***Clocker :*** We can clock in when we arrived and clock out when we leave. This templates display the hours of clock in and clock out.

- ***WorkingTimes :*** We can see a tabs of our working times rank by months, which contains clock in times , clock out times and dates. The User can aslo edit his working or delete them.

- ***Login :*** The users can login with his email and password.

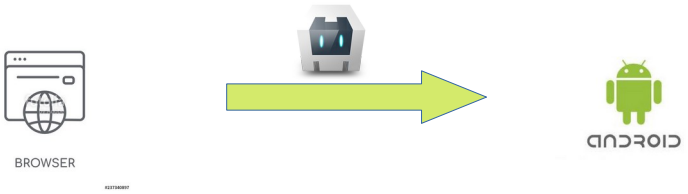
- ***Settings :*** The users can edit his profile to change his firstname, lastname, email , password or deleted his profile.

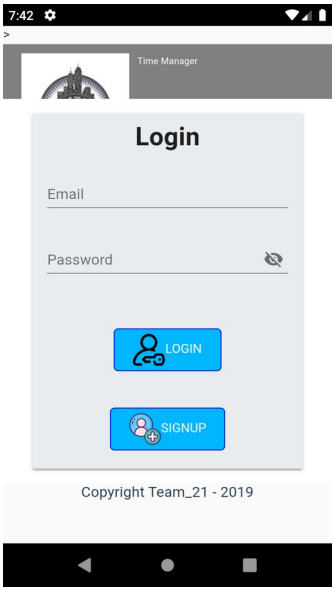
- ***Logout :*** The users can logout and he'll be redirecting to login page.

V. MOBILE

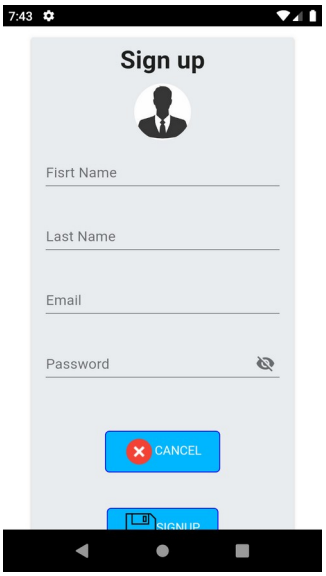
During our project we adapt the website on android platform for smartphone to allow person without computer to save their working times. We had to think about the design of a smartphone and adapt our site :

- Smaller screen
- Adapt the menu bars
- Adapt location of information
- Make it convenient to use

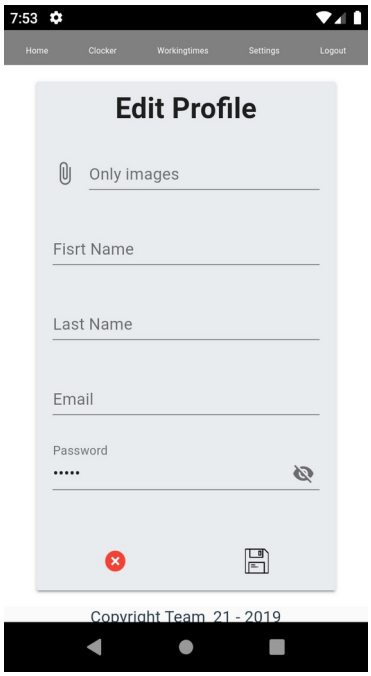




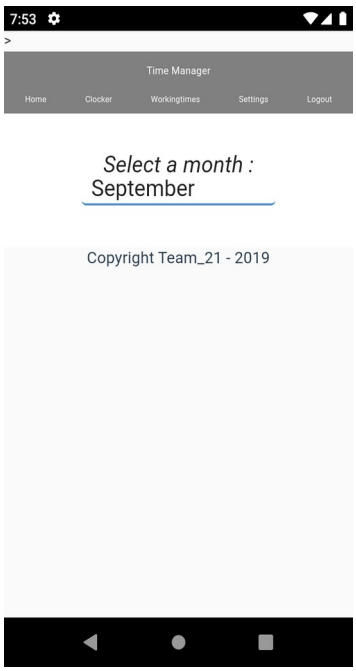
Login page



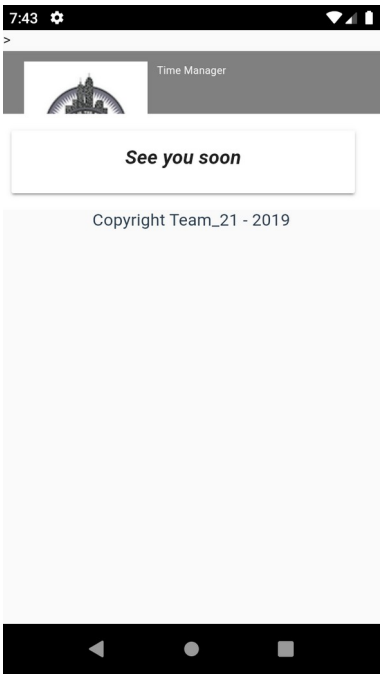
Signup page



Edit page



Workingtimes page



Logout page

VI. SECURITY

a. DIRB

DIRB is a scanner which scan all the files and tag keywork to exploit them and find the flaws. The results allow to exploit the flaws. Indeed, we can spot the username and password.

The goal is to spot these words and hide them to avoid any vulnerabilities.

```
ger# dirb http://54.198.162.120:8080/#/login

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Mon Sep 30 12:47:57 2019
URL_BASE: http://54.198.162.120:8080/#/login/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://54.198.162.120:8080/#/login/ ----

-----------------
END_TIME: Mon Sep 30 12:56:58 2019
DOWNLOADED: 4612 - FOUND: 0
```

There is no words to spot, the site is protected.

b. *NAMP*

It's a tool that scans all ports used by the website and obtains information about the OS used by the computer. This tools test if the ports are open or close and if we can acces datas. This allows to check if the ports of the websites are secure.

The goal is to spot open ports to secure them and avoid data leakage.

```
Starting Nmap 7.60 ( https://nmap.org ) at 2019-09-30 13:54 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000028s latency).
Not shown: 1993 closed ports
PORT     STATE         SERVICE
631/tcp  open          ipp
3001/tcp open          nessus
5432/tcp open          postgresql
8080/tcp open          http-proxy
68/udp   open|filtered dhcpc
631/udp  open|filtered ipp
5353/udp open|filtered zeroconf

Nmap done: 1 IP address (1 host up) scanned in 5.92 seconds
```

c. *SQLMAP*

It's a tool that does automatic sql injections to collect data from the database. Indeed, the injection of certain specific characters into url can cause database leaks.

The goal is to protect its sql queries to avoid any injections.

```
miguel@ubuntu:~/Downloads/sqlmapproject-sqlmap-5168daf$ python sqlmap.py -u 54.
198.162.120:8080/#/login=id1
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.3.9.20#dev}
|_ -| . [.]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutua
l consent is illegal. It is the end user's responsibility to obey all applicabl
e local, state and federal laws. Developers assume no liability and are not res
ponsible for any misuse or damage caused by this program

[*] starting @ 14:43:51 /2019-09-30/

[14:43:51] [INFO] testing connection to the target URL
[14:43:51] [CRITICAL] previous heuristics detected that the target is protected
 by some kind of WAF/IPS
[14:43:51] [INFO] testing if the target URL content is stable
[14:43:52] [INFO] target URL content is stable
[14:43:52] [CRITICAL] no parameter(s) found for testing in the provided data (e
.g. GET parameter 'id' in 'www.site.com/index.php?id=1')

[*] ending @ 14:43:52 /2019-09-30/
```

VII. TECHNOLOGIES REQUIREMENTS

We use differents technologies for develop our application Time Manager and for our organisation.
For our orgnaisation we use *teams* for exchange within our group, we also use *trello* to plan the diffrent steps of pur project and *github* to push the different versions.
For the developement we use the langage *node js* and *mysql* for the database.
The web interface is coded with *vue js* and *vuex*. For the authentification we decides to use *JWT token*.
To link our api with our front we use *axios* and to make our mobil application we use *Andriod studio* with the langage *cordova*.