Electric Scooter Sharing Application

Hamza Díaz

Abstract— In this article, the creation of the Electric Scooter Sharing service is presented from an idea to an industrialized environment with complex mobile applications and a Desktop application to manage users and e-scooters. We will showcase how the IoT or Internet of Things interconnectivity has a significant impact on the API and how it provides a pack of hardware and software fusion that serves the no gas emission program providing sustainable user-friendly mobility transport. We will go through the different stages of the mobile Apps development and discuss the architectures and technologies used to create this service as well as the cross-platform development. Moreover, the real-time technology implementation and the results it produces for the applications will be described. We will also discuss the possible security issues of this service like the man-in-the-middle and how to solve them.

Keywords—Mobile Applications, IoT, e-scooters, real-time technology, sharing service, Ionic, cross platform, Google Maps, Man-In-the-middle.

1 INTRODUCTION

ELECTRIC mobility has become one of the main trends these last years with the ambition to achieve sustainable transport services with zero gas emissions and less air pollution. Electric buses, subways, electric cars, electric motorcycles, e-scooters, etc., are having a great impact in metropolitan cities but still require considerable investments.

During August of 2018, the United States of America spent about $84 Million for the Low or no Emission Program [Forbes sources] which is still a low investment for such a big country. Barcelona TMB *(Transport Metropolità de Barcelona)* also received about $81.8 Million loan from the European Investment Bank (EIB) for the same reason, "*This will enable TMB to accelerate the transition to a zero-emission bus network and to improve service quality and reliability. In addition to improving air quality,... Barcelona's public city buses have seen a 17% increase in use over the last six years – transporting 203 million people in 2018 – a figure that is still growing...* " EIB notes said. Nevertheless, this is just the beginning and we still have a lot to learn and to invest in order to achieve a fully sustainable mobility.

The goal of this project is to prove the fusion between electric mobility and IoT (Internet of Things), which offers countless possibilities to innovate and provide more user-friendly sustainable solutions. With IoT integration, we can analyze data and generate comprehensive solutions alongside Mobile applications and much more. We will focus on electric scooter or e-scooter sharing technology, a very new service that is growing in many different countries and generating significant revenues. It is known as the fastest growing service, with high demand in big cities, not just because of the sustainability but because of the connectivity between districts and simplification of the public transports, transforming the cities into a futuristic environment.

Lime and Bird are one of the first and top companies offering this service. They have raised nearly a billion dollars each within the first 14 months.

We will study the creation of this service from a Software and Security Engineer perspective, the management behind it and we will go further discussing and implementing the service and mobile applications:

- Deep dive of the general Architecture of the electric scooter sharing service and IoT
- Mobile Applications:
  - I. Study of the architecture and Development of the iOS and Android users Application
  - II. Study of the architecture and Development of the iOS and Android Personnel Application
- Desktop Application:
  - I. Study of the architecture and Development of the desktop application to manage the e-scooters and users.
- Security and solutions:
  - I. Prove the Middle man API interception vulnerability and provide possible solutions.
- Testing the communication of the applications with e-scooter and Desktop tool with results.

- *E-mail de contacte: hamza.diaz@e-camous.uab.cat*
- *Menció realitzada: Tecnologies de la Informació.*
- *Treball tutoritzat per: Jose A. Lopez-Salcedo (Telecomunicacions)*
- *Curs 2018/19*

2 State of Art

2.1 Electric Scooter Sharing Service

The Electric Scooter (e-scooter) Sharing Service is a business where users can rent a scooter through a Mobile Application to go to their destination during a specific time. They can view all the available e-scooters and are able to unlock or lock them. The ride is usually charged off by an unlock fee of 1$, plus a charge per minute that is normally about 0.15$.

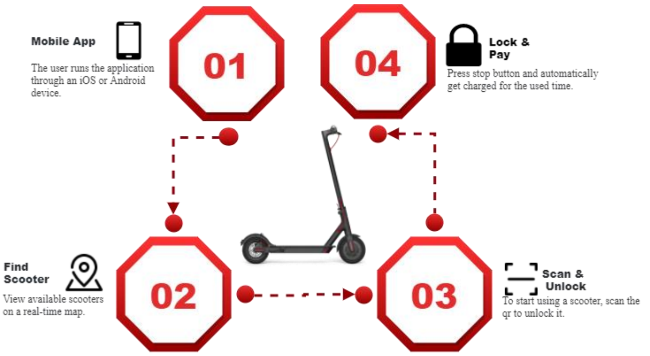


Fig. 1: Diagram of the different stages of electric scooter sharing service

Here is a detailed explanation of each stage:

01. The application is running on iOS and Android devices. Users can install it through AppStore or PlayStore respectively and register an account.

02. When logged in, the Main Page of the App is a real-time map showing the nearby available electric scooters to the user's current location.

03. When the user finds the scooter, they can unlock it and **start riding** by scanning a QR code through the mobile App.

04. After they reach their destination they stop the e-scooter using the App and get charged accordingly.

The simplicity radicates in the capabilities of the IoT device running in the e-scooter. It has different modules implemented that will be studied more in detail in the telecommunications project. These modules send data through some scripts communicating with the server, so the mobile application can retrieve it using an API deployed in the server. The returned data contains different key-values with information of the battery, GPS coordinates of the current e-scooter among others.

To have a prevailing idea of the e-scooter sample used in the project, these are some of the most important features to consider: The sample model is a **Xiaomi Mijia M365** with an autonomy of 30km fully charged, it has a high capacity 7800 mAh lithium battery with charging time up to 4 hours, a maximum load of 120kg, an IoT device, a maximum speed of 25km/h, etc.

2.2. Market & Business Plan

Among the many benefits of the e-scooter sharing service, sustainability takes the first place. People tend to embrace what's good for the planet, but we are living in a century where we feel safer with high technology surrounding our environment. It can be stated that a considerable percentage of residents are scared of the service, usually because they don't understand its capabilities and how from a technological perspective, the service can be leveraged and maintained. For example, our e-scooter sample satisfies the maximum velocity of 25km/h complying with the regulations of the DGT.

The e-scooter sharing is the new system of urban mobility. It is addicting and it does not compete with public transport but exists to help avoiding the massive crowded scenarios.

The market in business is a group of consumers or organizations interested in the product. In e-scooter sharing service, we will have consumers from students that go from different centers to their school or university, to tourists who explore the city, employees who go to their office, people who want to get closer to a station or a certain location and many more. It is a high demanded and used service with a vast number of users. Lime e-scooters are used from the age 16, and children between the ages of 12 and 16 can ride, but only if they're accompanied by an adult. Every e-scooter company can have its own restrictions. Our applications will require an ID to verify that the user is at least 18 years old to avoid legal issues and law restrictions.

Before going deeper into the earnings, it is important to understand the concept of ride, a **ride** is a trip between two areas from the moment the e-scooter is unlocked until it is locked again, it has an average ride time from a conservative point of view of 12.8 minutes. Since the unlock fee for our application will be 0.50$ and then 0.15$/minute, the average income per ride will be:

Average Income = 12.8 minutes x 0.15 $/minutes + 0.5 $ = **2.42 $**

Lime in the city of Dallas (1.318 million inhabitants) made about 220,000 rides in e-scooters in its first year. And one of the biggest successes was St. Francisco (870.887 inhabitants) where they made more than 300,000 rides in a year. Today, Lime is celebrating 50 million rides across 26 cities. With that, users have avoided about 15 million miles of car travel and prevented 6,220 metric tons of carbon emissions.

2.3 IoT device

Internet of things or IoT is the name given to interconnected devices through the internet, collecting and sharing data. IoT provides a level of digital intelligence to devices, enabling them to communicate real-time data without human involvement.
The role of IoT in the future of mobility is growing each year. We are now able to turn anything from an airplane to self-driving cars into part of the IoT.

Lime, Bird and many new e-scooters sharing startups are using electric scooters with IoT devices to achieve all the mentioned functionalities of this service, even motorcycles sharing companies like eCooltra or Yego use IoT devices to achieve the same goals.
Providers of e-scooters mostly are based in China, e.g. the electronics company Xiaomi. They are responsible for the production of these e-scooters and IoT devices connectivity.

The e-scooter sample used in the project has an IoT device with the following modules:

- Bluetooth Low Energy Communication Module (BLE 4.0)
- 4G Communication Module
- Anti-theft alarm (Activated when scooter is moved prior to unlocking)
- GPS real-time tracking
- Controller to unlock/lock e-scooter
- Battery

2.4 Ionic Cross-Platform Mobile developement

The big question is why build natively for every different platform if you can build it once and deploy it on as many platforms as you want? As iOS and Android become the most used mobile platforms in the world, cross-platform software development allows developers to code once and run it anywhere for these platforms. It is true that some complex native functionalities can't be accessed compared to native development [11] but every year cross-platform frameworks are updating and providing extra functionalities that reduce costs and investments and make it a preferred choice for developers as well as software companies.

We will be using **Ionic**, which is a cross-platform mobile development framework for web developers based on Angular and Cordova. It will be used in this project to create the hybrid Mobile Applications for both iOS and Android. Ionic is one of the best frameworks providing excellent performance and project organization. There are countless successful applications developed under this framework, like *Sworkit Coach Personnel* with more than 5Million downloads.
Cordova is a platform to build Mobile Applications using HTML5, CSS and JavaScript, in which we can access the device functions like Camera, GPS, Accelerometer, Contacts, etc.

2.5 Real-time Technology

The concept of real-time sometimes can be bizarre to understand but also to implement. Facebook, Instagram, Whatsapp, and hundred more top Mobile Applications already come with this technology. It can be implemented in many different ways. For instance, Facebook users can get notifications, messages, new posts and much more without having to update the app or the webpage. This is achieved using WebSockets, comet approaches, and other techniques, which requires considerable programming costs.

All the Applications implemented in this project will be using Firebase, a platform that provides handy tools to achieve real-time functionalities.
Firebase is a Google mobile and web application development platform, with the **real-time database** which will be used to save the data. Every time this data is updated, all the devices connected to this database get automatically updated without the need of manually refreshing. Although this service has high development costs it is a very powerful and fast solution for real-time approaches.

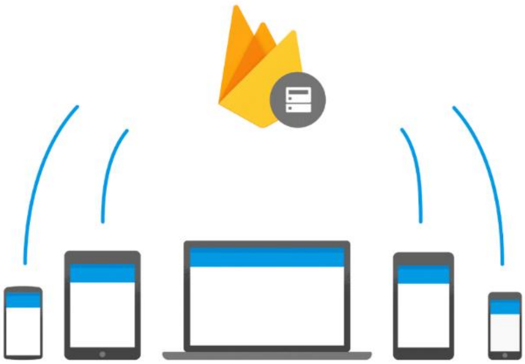


Fig. 2: Diagram of the real-time database returning data in real-time to every device connected to it

2.6 Man In The Middle

The Man-in-the-middle is a hacking strategy in which the attacker has the knowledge to observe and intercept messages between two victims and ensure that none of them knows that the link between them has been violated. They can also read, insert and modify the victim's communications anytime.

This is very important in the e-scooter sharing service and a vulnerability for every e-scooter sharing company.
If the company doesn’t have the necessary preventions, their business can be affected very severely.

After testing our application API communication, we performed this attack and realized that everyone with the knowledge can be able to start the e-scooter when this API call is intercepted.

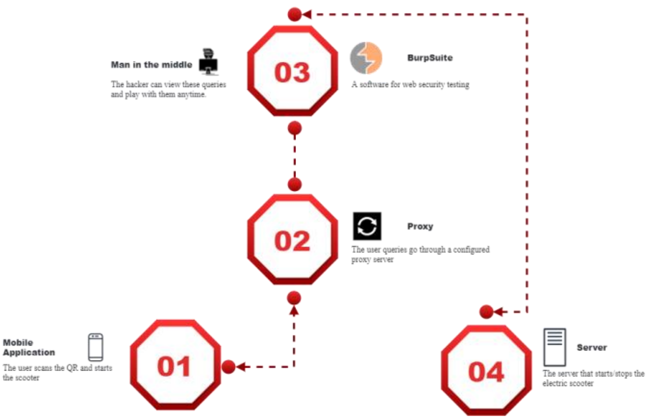


Fig. 3: Diagram of the man-in-the-middle attack in our e-scooter sharing service

Here is a detailed explanation of each stage:

01. The user is logged in and about to start the e-scooter. The moment they scan the QR code an unlock e-scooter query function is sent through the API.

02. This query goes through a configured proxy server connected with burpsuite (03) which is a web security software. From here, the attacker can control these queries, change them and send them to the server. The response from server (04) goes back to the attacker through burpsuite and then to the user.

3 METHODOLOGY

After examining the State of Art in which this project is based, we have decided to design and develop a collection of Mobile Applications for iOS and Android devices for clients and personnel and a Desktop application to manage the e-scooter sharing service.

The idea is to fulfill the objectives and cover all of the important requirements behind the e-scooter sharing business. From one side, we will go through the development of the Mobile application for users, namely clients. This application will meet all the functionalities starting from registering a new account, e-scooter locations in real-time, e-scooter lock/unlock, payments, reporting, preview e-scooter details and much more. On the other side, we can find the Mobile application for personnel in charge of maintaining and charging the e-scooters when needed.
The Desktop application will control the newly registered users to verify and manage scooters. It will have an e-scooter optimization algorithm that will be implemented in the telecommunications projects.
Additionally, we have investigated the possible security issues and man-in-the-middle vulnerabilities in the service to provide some solutions.
To conclude, we will present and analyze the results and screenshots of all the mentioned applications.

4 MOBILE APPLICATIONS

In business, generally, the process of making a Software is extremely long and difficult not just from an Agile perspective but also from a development and engineer perspective. We won't go very deeper into these stages of the application but it is worth mentioning some of the most important ones we have been through.

4.1 Idea

The ideation process of the applications wasn't that challenging since already were many different competitors working with the same concept of e-scooter sharing application, however, we went further by thinking about solutions for both users and the sharing companies. Hence, we thought about two different mobile applications, one for clients and one for personnel.

4.1 Strategy

In this stage, we can find four different procedures:

1. The competition: It is recommended to analyze the number of installs, ratings, and reviews, taking advantage of the negative comments and UI/UX related issues that can be leveraged. Nevertheless, we are aware that companies like Lime or Bird have millions of downloads and provide apps of excellence. Hence, we don't focus on competing from that perspective but we do focus on providing better solutions. We have tested other e-scooters startups and Lime in Stockholm city. Their distribution is very weak and most of the time scooters in the map are taken by other users before reaching them.
One of our proposed solutions is the Personnel Mobile App to maintain broken and low battery e-scooters. We will also work on an e-scooter optimization algorithm as a solution to the high number of low battery e-scooters or badly distributed ones.

2. Monetization: Nowadays, mobile applications, can have in-app purchases, advertisements, app installs, etc. Our App dedicated to users will generate incomes basically from the e-scooters rides, while the other app for personnel won't be generating any income as it will be for company usage only.

3. Marketing: One of the possible marketing strategies is providing discounts after registering or suggesting friends to download the app, social networks ads, in-person conferences, among others.

4. Roadmap: Basically, in roadmap strategy we focus in Minimum Viable Product (MVP) which represents the basic working version of the App. The application subsists because of the hardware and the IoT. The final product is actually a pack of these elements that makes the business grow.
The future of the service is to become one day the first choice for residents to go between different areas and the number one sustainable transport in the world.

4.2 User-Experience Design

This stage has four different procedures: **Information Architecture** where we studied the different functionalities and features the Apps are organized. These are the basic building blocks with which we will build the wireframes. **Wireframes:** This process often takes place on whiteboards or paper, [refer to appendix A1], we basically assign each function or feature to a screen, we could even go further and implement **Workflows** and **Click-through models** by using a design platform like *MarvelApp* for example to create a more interactive mockup for the apps.

4.3 UI Design

For the applications, most of the rendered designs for the UI (User Interface) have been created by Adobe Photoshop CC and Adobe Illustrator CC, however, some advanced designs like the LogoType and color styles have been externalized and discussed with professional freelance designers.

4.4 High Level Technical Design & Development

The high level technical design is the process of choosing the right tools for both FrontEnd and BackEnd development. As discussed in the previous sections, the applications will be developed through the cross-platform framework called Ionic. This framework requires Web Technology Programming languages. We will also discuss the development and some BackEnd technologies as well as servers, clouds and payments.
From the FrontEnd perspective, these are the available pages:

- Loading Page
- Login Page.
- Register Page.
- Forgot Password Page
- Home Page (With real-time map showing nearby scooters)
- E-Scooter Details Page
- E-Scooter QR Scanner Page
- E-Scooter Ride Page
- E-Scooter Payment Page
- User Profile Page
- Edit Profile Page
- Others

The application ‘s main functions can be found in the state of art of how the electric scooter sharing service works. These functions are quite simple and are in the following order:

Find -> Scan -> Ride-> Pay

The following picture shows the architecture:

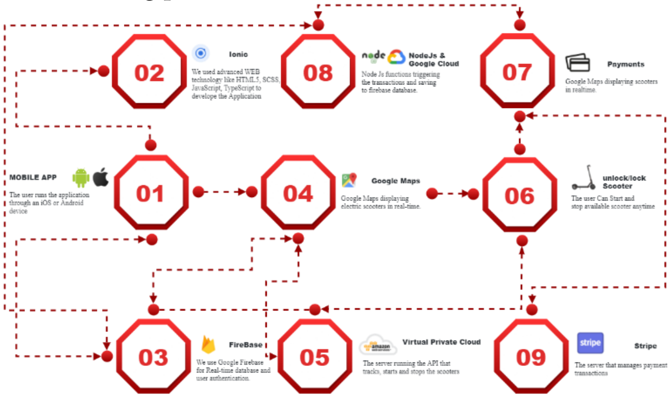


Fig. 4: Architecture of the FrontEnd and BackEnd of the Users Application

The goal of this diagram is to showcase the architecture of the tools and technologies used for the development of the mobile app:

01. The FrontEnd is developed with the Ionic framework using HTML5, CSS3, SCSS, Javascript, Typescript, Angular among other functionalities provided by the framework. We will also have tools to deploy and test the App for both iOS and Android.

02. The application is running on iOS and Android devices. Users can install it through AppStore or PlayStore respectively and register an account. This registration/Login is communicating with Firebase Authentication Service.

03. Firebase is the technology running the **real-time database, firebase authentication** among other services.

Fig. 5: Firebase Real-time database data is stored in JSON

The data is stored in a nested key-values JSON format which makes it easy to read from the FrontEnd side. We can also edit this data from the Firebase database dashboard, which updates all the devices connected to it automatically.

04. Google Maps will be used to position the e-scooters on the homepage of the App in real-time. It will be accomplished by connecting Google Maps with Firebase. In this page, we can see all the available scooters near our current location, and can also preview details of each scooter and unlock it when needed. For a better positioning we will query the location directly from the API running on a Private **amazon web services** and save it to the Firebase real-time database, this is done because the first coordinates saved in the database are the last coordinates when the user locks the scooter. Therefore, it is important to save them directly from the API that consumes IoT GPS module data.

05. The system has a private network analyzed in the telecommunications project running on Amazon Web Service. This server is an EC2 Instance that runs both the e-scooter algorithm developed using R programming language and ideally the API. AWS was the first choice since it provides powerful tools to work with TCP communications, scalability and performances.

06. Next step is the unlock process which basically performs an API call to the server to start the selected scooter by sending the scanned QR code. This QR code is very important because it identifies each scooter uniquely.
When the user has finished the ride, they will lock the scooter by the stop button in the App that also communicates with the API by sending the e-scooter QR code to be stopped.

07. After stopping or locking the e-scooter, the users will be redirected to the payment page to get charged off the ride price. These transactions will be performed through **Stripe (08)**, a software to help us manage these payments securely and to be compliant with PCI Security Standards Council [24].
Since we are not allowed to store the users' credit cards to a normal database, everything will be stored in Stripe, but in order to have information about the payments, we have implemented a backend **NodeJS** trigger function running on Google Cloud that gets called after a payment is done and saves the returned payment transaction details to Firebase real-time database.

It is important to keep in mind that the personnel mobile application will have the same technologies, except for the payment functionalities, which is not needed in this case and lowers the level of complexity in terms of programming costs, however, they will have more visibility of the e-scooters. The goal of this app is to control lost scooters, low battery scooters, and broken scooters. [Refer to appendix A1 to see the diagram].

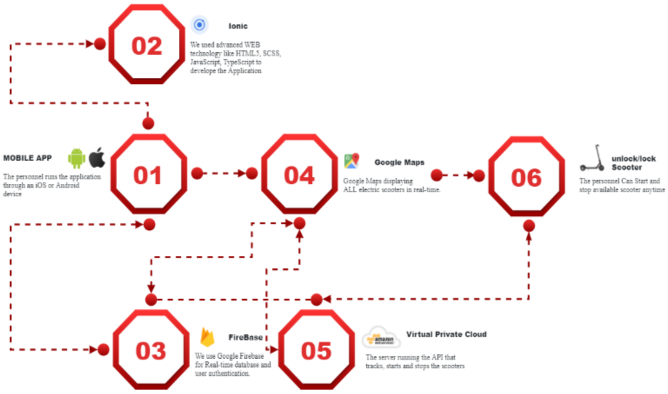


Fig. 6: Architecture of the FrontEnd and BackEnd of the Personnel Application

We added the lock/unlock feature so the personnel can move the e-scooters without the activation of the anti-theft alarm.

4.5 Testing & Deployment

Initially, the testing process was performed alongside the scooter to make sure the App works correctly. This testing was done by non-developer users and developers. As developers, we found many bugs and issues that we solved and also the man-in-the-middle vulnerability.

After the testing was successful, we proceeded with the deployment for the distribution platforms AppStore by Apple and PlayStore by Google.

Fig. 7: Mobile App distribution platforms Appstore and Playsotre respectively.

4.6 Man-in-the-Middle Solutions

To prevent these type of attacks in the applications development, there are many security solutions. First of all, it is a good practice to upgrade from REST HTTP API calls to HTTPS through SSL/TLS Certificates. A TLS Certificate will activate the HTTPS protocol, which is a safer version of HTTP. This provides an encrypted, secure connection between the server and the users' devices. For example, some apps use their own certificate trust store, and some implement certificate pinning to only trust specific server-side certificates. In this situation, breaking the SSL tunnel is non-trivial.
Secondly, in our situation there is a tokenized API implemented in HTTP, which is a good practice but the

token lives for hours and gets updated after some time. One solution to avoid extra unlocks after interception attacks were successful, is using tokens that change in the least possible time. Finally, upgrading from Restful API to WebSocket protocol API which defines 2 new URI schemes: WS and WSS for TLS encryption, providing a much higher level of security compared to Restful HTTP requests.

5 Desktop Application

In the e-scooter sharing business, we will need an administration tool for the managers and specialists to control the service and manage the users and e-scooters. The development of this tool will also go through different Agile stages from an idea to a working solution. We won't be going deeper to these stages, but we will discuss the high-level technical details regarding the FrontEnd and BackEnd implementation.

Most of the heavy components of this tool will be implemented in the **telecommunications project** like the optimization algorithm. [Refer to appendix A2 to see the diagram].

Fig. 8: Architecture of the FrontEnd and BackEnd of the Desktop Application

Before going deeper into some of the important elements in the architecture of the application, we will discuss the main pages of the tool:

- Page to Login as an Administrator.
- Page of the Visualization of analytical KPIs and charts of the E-Scooter Sharing Service.
- Page to view registered users and validate their Identity, paymnets can also be viewed.
- Page to view the e-scooters on Google Maps in real-time and the algorithm results.

The architecture of the application is the following:

01. The resulting application will be a WebApp which can be deployed through a private server or locally.
02. The application was coded with Angular 7, a platform to build mobile and Web Applications by using advanced Web technologies like HTML5, SCSS, JavaScript, TypeScript, etc.
03. The analytical mockup page showing the current day statistical KPIs and charts.
04. Firebase Service for real-time database to query registered users and changes.
05. Real-time map page showing the scooters worldwide. We will also show the result of the implemented algorithm in R
06. R programming language to perform data management analysis and create the algorithm for the e-scooters optimization.
07. The admin can start or stop e-scooters anytime through the tool.
08. In order to allow administrators to perform operations on the scooters, they will need to be connected to the API mentioned in the mobile applications and deployed in the virtual private server.

6 Results

6.1 Applications communication with Real-time database

First of all, we have added the only available e-scooter sample to the **Firebase real-time database,** we opened the mobile application for users in two devices.

(a) User A (b) User B

Fig. 9: Two users viewing the HomePage

User A discovers an available scooter, the moment they scan it automatically it gets removed from everyone else's HomePage. That is why user B can't see the scooter anymore.

This was achieved by connecting to the Firebase Real-time database and by setting the flag **booked** to true when used and to false when available:

```
"-LPbjawLlwpkqB9G8HRj" : {
  "battery" : "100%",
  "booked" : true,
  "id" : "1",
  "lat" : 41.5290944,
  "lng" : 1.6877891,
  "name" : "xxxx",
  "status" : "free"
},
```


Fig. 10: JSON data of the e-scooter

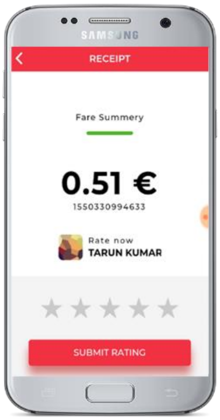
Firebase generates a unique key, *LPbjawL1wpkqB9G8HRj* where the information about the e-scooter is saved. This information holds the battery, an id, the location with latitude and longitude, the name of the e-scooter and importantly the status, which can be *free* if available for usage, *low* and *very_low* for bad battery status, ergo not available, and *fix* if the scooter is broken, aslo not available. These key-values were implemented manually in the BackEnd, however more data can be added in the future making the correct changes in this JSON

6.2 Applications communication with E-Scooter & Payments

In this section, we will see how the e-scooter is unlocked through the mobile app and how the payment is handled.

(a) Scanning the QR

(b) Riding the e-Scooter



(c) Receipt Page after Lock

(d) Payments Page

Fig. 11: The e-scooter sharing application flow

As a result the e-scooter gets unlocked and locked successfully, the IoT receives the API calls correctly from the server. The payment is managed by **Stripe**. We are using a test mode where the payment is done by a Stripe VISA or MasterCard credit card for testing purposes so we don't get charged real money. This test mode can be changed through Stripe portal to production mode anytime. When the payment is successfully executed, it gets stored to Firebase, where we could query it to our Desktop application:


```
payments + ×
  M3w3WSEnmvaegoOetplJcGjEz263
    -Li5fJGOFxWSxlcceZAd
      amount: 52.249999999999
      token
        card
          brand: "Visa"
          country: "US"
          cvc_check: "pass"
          exp_month: 1
          exp_year: 2022
          funding: "credit
          id: "card_1Eof8hHnkJbWdIq54oS8Dq
          last4: "4242"
          name: "hamza.diaz.house@gmail.cc
          object: "card"
        client_ip: "95.19.99.240
        created: 156133262
        email: "hamza.diaz.house@gmail.cc
        id: "tok_1Eof8hHnkJbWdIq5AQXs1V
        livemode: false
        object: "token"
        type: "card"
        used: false
```


Fig. 7: Payments JSON data

We save the payment information returned by Stripe the moment the user gets charged, but at any moment the full credit card information is saved to comply with CPI.
In the Desktop tool, we can see this payment in real-time. As we are only reading the amount of the payment and the corresponding email address.

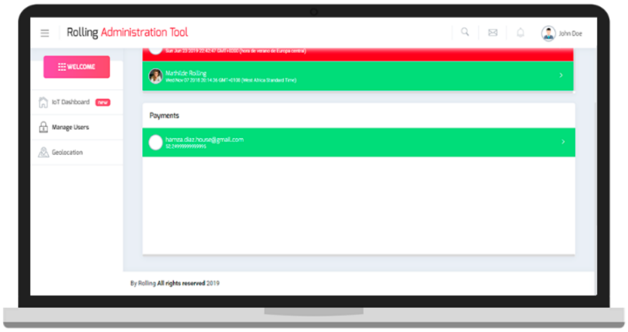


Fig.12: Payments from Manage Users Page of the Desktop Application

6.3 New users and verification via Desktop application

In the mobile application, users can register and upload an ID to be verified. The verification procedure is done through the Desktop application in the Manage Users page, where the admins have to view the users' ID and hit a verify or dismiss button, depending on the provided information. No verified users are highlighted in red color and verified users are highlighted in green.

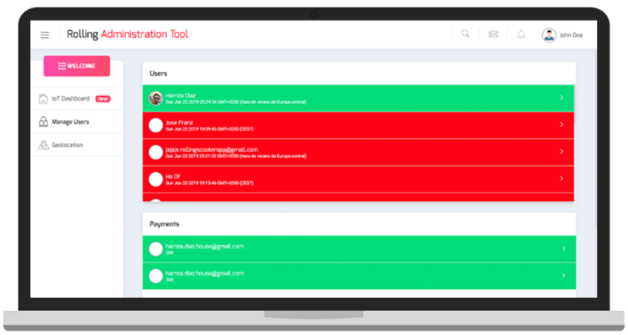


Fig. 13: Manage Users Page of the Desktop Application

6.4 Personnel Mobile Application

We added 3 extra imaginary scooters with the status of low and very low battery, and to be fixed e-scooters on random locations. The homepage of the personnel mobile application was able to show all these e-scooters providing full visibility of the e-scooter sharing service from the management side.

Fig. 14: HomePage of Personnel Mobile Application

7 Project conclusions

IoT has proven its importance in the electric mobility, particularly focusing on the electric scooters business. This fusion allowed us to create a user-friendly sharing service operated by mobile applications and a Desktop application, leveraging the usage of e-scooters not just from an analytical perspective but also from a software perspective with the possibility to innovate in different sectors. Most of the proposed objectives have been accomplished successfully, both the mobile applications and the Desktop application have been developed in an industrialized approach. Also, other important components were implemented in the telecommunications project which should be connected with this paper to have an all-in-one pack of the Electric Scooter Sharing Service.

This project had more than 1500€ of expenses among them the sample and IoT acquired through the Chinese providers. There were many blockers throughout the development of the applications, since we are working on a brand-new service, which is still unknown for many developers and users and still growing and changing by big startups and government regulations. We had a lot of obstacles investigating solutions for the errors we were having, not just from the internet but also from IEEE articles, books, and other resources. However, these blockers allowed us to boost the ability to work under pressure and provide genuine decision-making strategies. Also, many days have been spent debugging and changing the app after Google's new updates to both Android and Firebase, this is one of the negative things I personally feel about the mobile App development, these updates are constant and require a lot of time to solve.

Unfortunately, we didn't manage to deploy the API in our own AWS server, but we managed to use the Chinese provider's server to start and stop the e-scooter through their API, basically, because the API **Knowledge Transfer (KT)** had a cost of 10.000$ plus other fees, and we didn't have such investment yet. We understood that it's the core module for their sharing business, and proceeded with having access to their API so we can have a running solution as soon as possible. Even though, as an experienced engineer, I think the procedure of making an API from scratch and deploy the IoT to run it, will require months of work and constant meetings to get the full KT from Chinese suppliers' engineers.

Finally, I want to say that thanks to this project, I was able to learn many different concepts and techniques. I had the pleasure to put into practice my knowledge and all my programming skills to create something of value.

AKNOWLEDGEMENTS

First of all, I want to give special thanks to my project tutor Jose A. Lopez-Salcedo for his constant support and dedication, for providing helpful suggestions and guidance whenever I needed it. I want to thank Jordi Pons Aróztegui for making this project possible. Also, the Chinese providers and Leila for supplying us the right tools: the e-scooter sample and the API; for having faith and trust on us.

Finally, I would like to thank my friends, Rolling Startup partners, and my family, who have helped me keep a more positive attitude during all this year.

REFERENCES

[1] T. Huddleston Jr.: Uber and Alphabet just invested $335 million in Lime — here's why scooter start-ups are suddenly worth billions. https://www.cnbc.com/2018/07/11/lime-bird-spin-why-scooter-start-ups-are-suddenly-worth-billions.html , Jul 13, 2018

[2] M. Suster: Lime and Bird worth $10B+ each or 5x to 10x more than their last valuations. https://www.futureengine.org/articles/scooters-are-worth-10b , Dec 3, 2018

[3] R. Campbell: Six Months With My Electric Scooter: A Review of the Xiaomi M365. https://medium.com/@RosieCampbell/six-months-with-my-electric-scooter-a-review-of-the-xiaomi-m365-42fbc16c5b73 , Dec 15, 2018

[4] S. Charkaoui, Z. Adraoui, E. H. Benlahmar: Cross-platform mobile development approaches. https://ieeexplore.ieee.org/document/7016616 , January 22, 2015

[5] S. Schulz, C. Engel, U. Dombrowski: Service Science, Management, and Engineering. *Chapter 15 - Influence of Electric Vehicles on After-Sales Service,* Pages 305-323. 2012

[6] S. Ranger: What is the IoT? Everything you need to know about the Internet of Things right now. https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/ , August 21, 2018

[7] Alibaba: New solution 2G 3G 4G GPS lock for renting sharing Electric Scooters 500w electric scooter 40km iot controller. https://www.alibaba.com/product-detail/New-solution-2G-3G-4G-GPS_60824327378.html?spm=a2700.7724857.normalList.86.79282e34C71HUM , 2018

[8] A. Manchanda: Where Do Cross-Platform App Frameworks Stand in 2019? https://www.netsolutions.com/insights/cross-platform-app-frameworks-in-2019/ , January 21, 2019

[9] Appsee: Getting Started with Cross-Platform App Development in 2019. https://hackernoon.com/getting-started-with-cross-platform-app-development-in-2019-dd2bf7f6161b ,March 2018

[10] J. Morgan: A Simple Explanation Of 'The Internet Of Things' https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#6c3ef6991d09 , May 13, 2014

[11] M. Zorzi, A. Gluhak, S. Lange, A. Bassi: From today's INTRAnet of things to a future INTERnet of things: a wireless- and mobility-related view. https://ieeexplore.ieee.org/abstract/document/5675777 , Dec 23, 2010

[12] Mobile Hackers: Best apps built with Ionic Framework and downloaded more than 10M times. https://medium.com/@mobilehackersio/top-8-apps-built-with-ionic-framework-downloaded-more-than-10m-times-601cedd75886 Jul 22, 2018

[13] Sustainable Bus: TMB to buy 250 new buses (116 electric) thanks to a EIB programme. https://www.sustainable-bus.com/electric-bus/tmb-to-buy-116-electric-buses-thanks-to-a-eib-programme/ June 7, 2019

[14] S. Kounis: How To Authenticate with Firebase and Ionic 3 — Email/Password and Google Sign-In. https://medium.com/appseed-io/integrating-firebase-password-and-google-authentication-into-your-ionic-3-app-2421cee32db9 , Feb 5, 2018

[15] K. Dechant: Angular 5: Making API calls with the HttpClient service. https://www.metaltoad.com/blog/angular-5-making-api-calls-httpclient-service , November 27, 2017

[16] Didin J.: Ionic 3, Angular 5, Firebase and Google Maps Location Tracking. https://www.djamware.com/post/5a48517280aca7059c142972/ionic-3-angular-5-firebase-and-google-maps-location-tracking ,Dec 31, 2017

[17] J. Delany: Angular Stripe Payments Series. https://angularfirebase.com/lessons/angular-stripe-payments-part-2-firebase-cloud-functions-backend/, Jul 11, 2017

[18] J. Delany: Location Queries With Firebase GeoFire and Angular Google Maps (AGM). https://angularfirebase.com/lessons/geofire-location-queries-with-google-maps/ , Aug 24, 2017

[19] Rapid7: Man-in-the-Middle (MITM) Attacks. https://www.rapid7.com/fundamentals/man-in-the-middle-attacks/

[20] Virinchy P.: What is Cordova and how does it work ? https://blogs.sap.com/2014/07/27/what-is-cordova-and-how-does-it-work/ July 27, 2014

[21] S. Tzur-David: The Ultimate Guide to Man in the Middle (MITM) Attacks and How to Prevent them. https://doubleoctopus.com/blog/the-ultimate-guide-to-man-in-the-middle-mitm-attacks-and-how-to-prevent-them/ Oct 8, 2018

[22] N. Handy: Burp Suite: Webpage Enumeration and Vulnerability Testing. https://medium.com/cyberdefenders/burp-suite-webpage-enumeration-and-vulnerability-testing-cfd0b140570d , Jul31, 2018

[23] T. Gubbins: New feature from Lime lets Dallas riders rate bad scooter parking jobs. http://dallas.culturemap.com/news/city-life/07-25-18-lime-bike-share-scooter-parking/ Jul 25, 2018

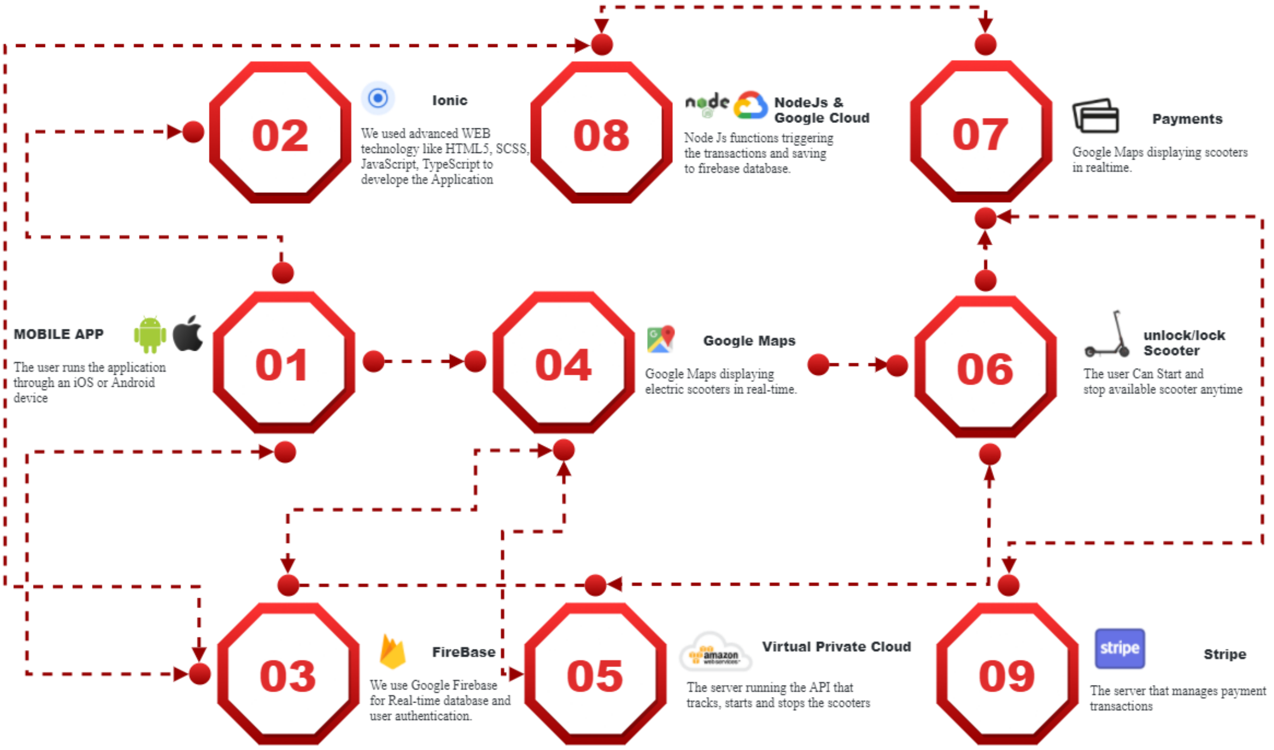
[24] The PCI Security Standards Council: https://www.pcisecuritystandards.org/

APPENDIX

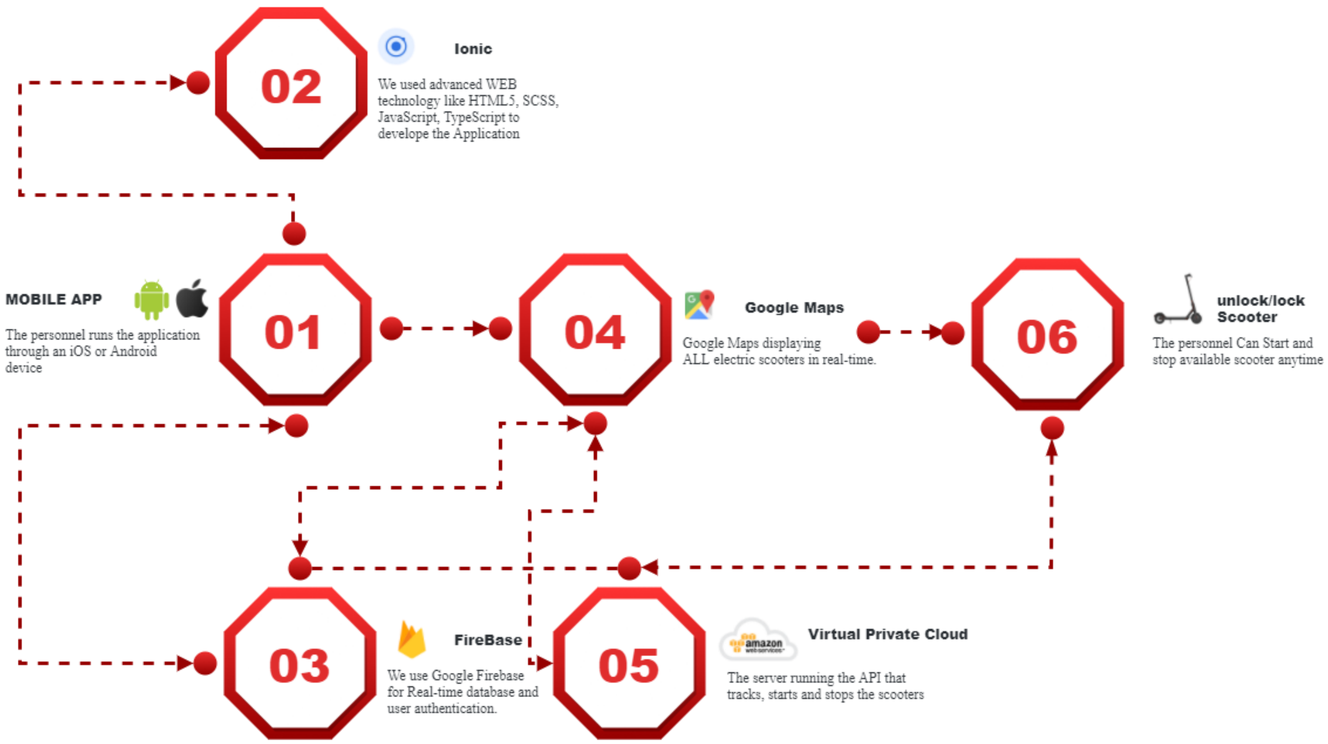
A1. MOBILE APPLICATIONS

In this section, we can find some of the diagrams and screenshots associated with Mobile Applications.

Fig. 15: Wireframe of the Users' Mobile Application



(a) Users Mobile Application



(b) Personnel Mobile Application

Fig. 16: Architecture of the FrontEnd and BackEnd

Fig. 17: SplachScreen, Register Page and Login Page screenshots

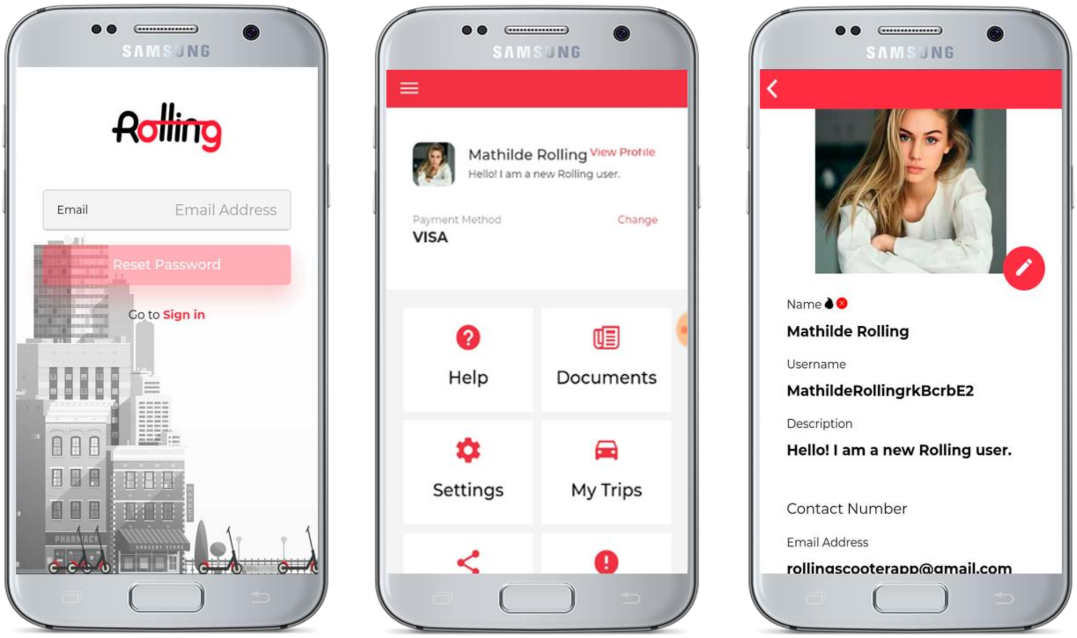


Fig. 17: Forgot Password Page, Profile Page and Edit Profile Page screenshots

A2. DESKTOP APPLICATION

In this section, we can find some of the diagrams and screenshots associated with Desktop Application.

Fig. 18: Architecture of the FrontEnd and BackEnd of the Desktop Application

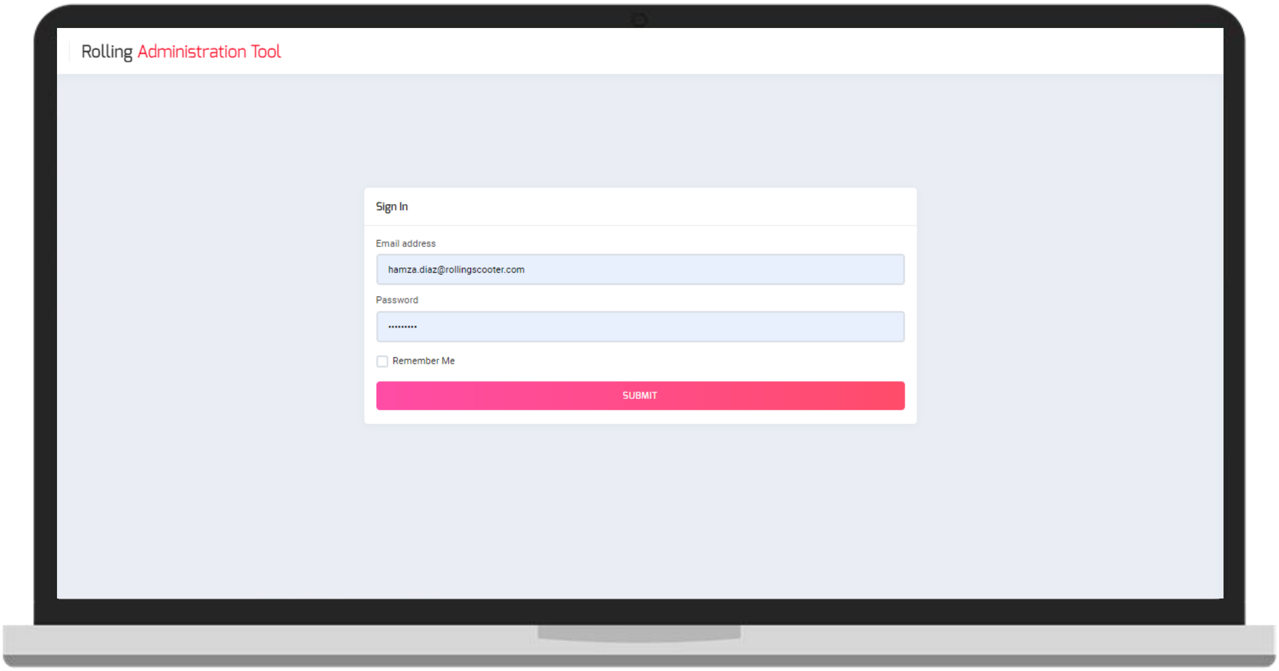


Fig. 19: LoginPage of the Desktop Application

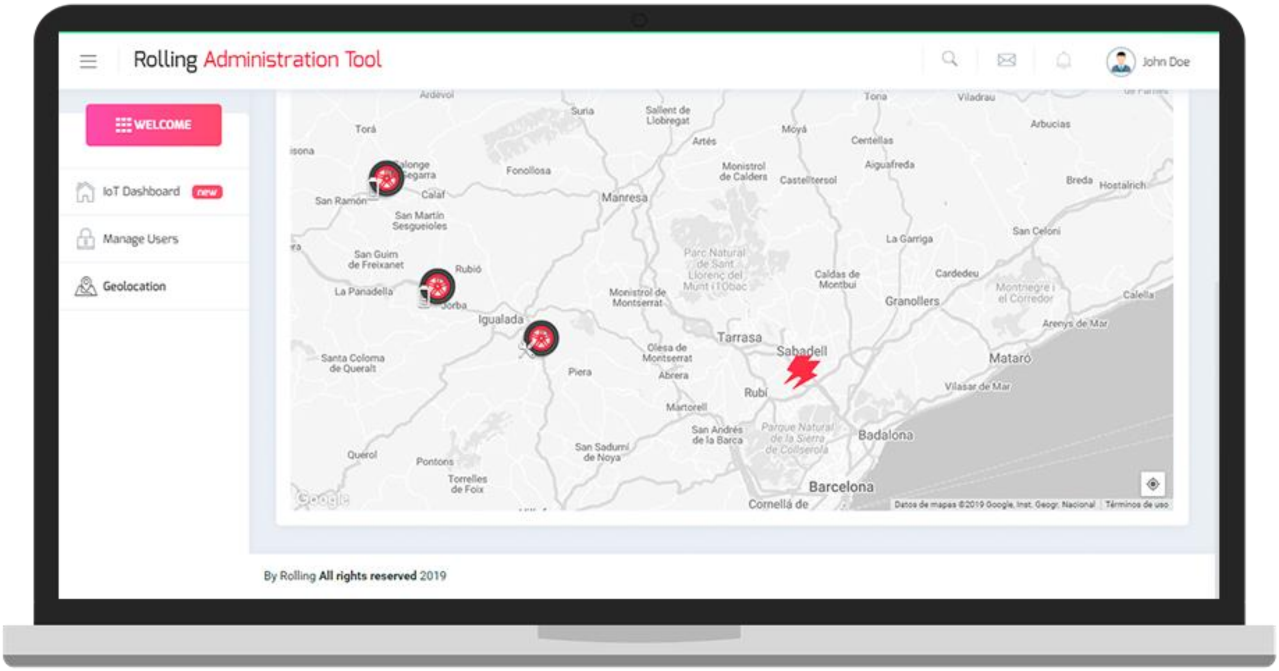


Fig. 20: Real-time map page of the Desktop Application

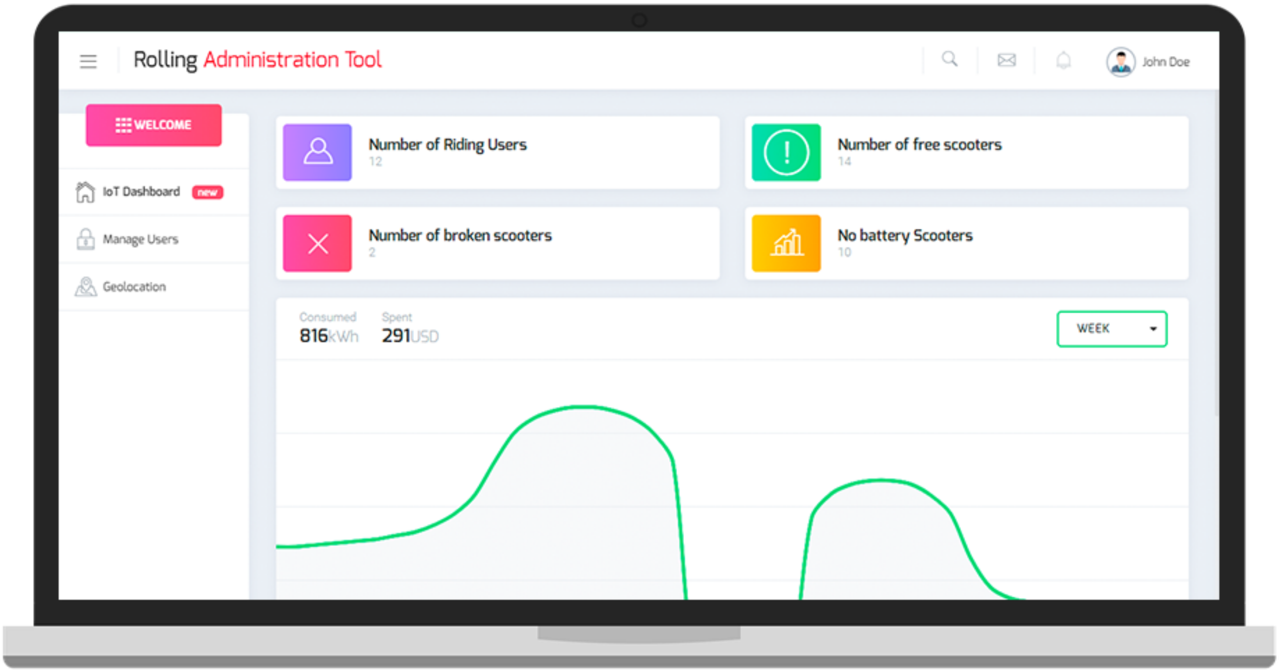


Fig. 21: KPI and charts Page (Mockup since we only have 1 e-scooter but ready for the production)