University of Balamand

Faculty of Sciences
Department of Computer Science
CSIS290 – Final project

E-**Valet** Final Report

Supervisor:	Joumana Dargham, **PhD**
Moderator:	Dani Nini, **MSc**
Presented by:	Nabih Bassil **and** Elie Youssef
Presented on:	30/5/2019

Spring 2019

Table of contents:

Project Abstract

Welcome to E-Valet.

This application was created for the purposes of graduating with a Bachelor of Science in Computer Science degree but as this application has many customers and has a potential to grow, the development team may decide to continue with this app outside the context of a university degree.

In this document, various aspects of the development of E-Valet mobile application will be discussed by defining some concepts and ideas, explaining choices made during the development and displaying graphical documentation on some aspects where pictures make ideas clear.

Acknowledgments

I wish to express sincere appreciation to Doctors Joumana Dargham and Dani Nini for their assistance in creating this application by giving us advice and technical support in developing E-Valet to turn this idea into a feasible and useful solution. I wish to acknowledge my gratitude to Mr. Jad Makhoul, whose familiarity with the needs and ideas of the targeted audience of E-Valet was helpful during the early programming phase.I also thank the faculty members of the department for their valuable input.

Approval Page

“E-Valet mobile application” a project prepared by Nabih J. Bassil and Elie G. Youssef in partial fulfillment of the requirements for the Bachelor of Science in Computer Science. This project has been approved and accepted by:

______________________________ ____________

Name, Supervisor Date

______________________________ ____________

Name, Moderator Date

______________________________ ____________

Name, Moderator Date

1. Introduction:

a. The Idea:

Valet parking employees in large cities are having difficulties retrieving parked cars because of the large number of cars and the overly busy state of lebanese streets. This faulty process is causing a significant wait time for car owners and an unnecessary effort from the valet employee to remember where they parked the cars.

We plan to provide an app that streamlines valet car retrieval. It will take the locations of the cars and manage them into ID based lists, so that whenever a car is to be retrieved you can just look up the ID on the app and it will give you the location of the car as well as directions to that location using Google Maps.

b. The Need:

We decided to pursue this project and develop E-Valet because we found out from our field analysis with the help of valet parking companies that an app like E-Valet is in need as technology has not caught up with this service sector, so companies could be potential buyers for an app like this as it is very time and effort saving which is what Valet companies are trying to improve in their day to day tasks.

c. The Objectives:

Ensure a stable and reliable app that will serve various valet parking employees simultaneously with their day to day jobs by saving locations of parked cars and additional information such as description and image to the database which will keep refreshing to provide consistency, and with the retrieval of parked car by retrieving the coordinates of the parked cars from the database and displaying them on Google Maps which will guide the employee to where the car is parked and the route for retrieval.

d. The Motivation

When we were brainstorming for ideas to make a final project, we desired an app that was unique and smart and it’s simplicity to show its usefulness. When we came across the E-Valet idea it was perfect for what we were demanding as this app is not a copycat of any application in the market and is a solution to a problem faced daily by valet parking employees which not only affects their productivity but also customers like ourselves waiting for our cars to arrive. Finding are retrieving cars in large and populated cities like beirut is not an easy task and few people think about the struggles employees go through to retrieve cars and that created an opening and an opportunity to be unique. Finding a creative solution for a problem that not everyone thought about or gave any importance motivated us to be innovators as we combined different components from various applications such as maps and databases to create E-Valet.

2. Feasibility Study:

Organizational feasibility:

- Development Team: Nabih Bassil – Elie Youssef
- Project Supervisor: Dr. Joumana Dergham
- Market references: Mr. Jad Makhoul (manager of a Valet parking company). We will be getting information on how he and his employees work to help us find the most efficient way to help them.
- Moderator: Dr Dani Nini.
- Other sources of input: UOB doctors and staff members.

Economic feasibility:

Our app will not incur any tangible costs as everything we are going to use is free. As for intangible costs, the project will take approximately 350 hours of work plus a certain amount of time to get familiar with all the technologies to be used in the project.

Technical feasibility:

We chose to go with React Native as our main programming language since we find it to be a new and exciting language to work with and learn. We already have prior experience with React by working with the technology in previous courses which gave us some experience with the React philosophy which can be applied to React Native, their advantages are not for the near future but for the far future as React is currently one of the most requested technologies in the market and has an active community that can help with technical difficulties.

We will also be making use of the Google Maps API provided by Google as our map service since we find it to be the most accurate and easily attainable with our current tools. Google maps is the most used map API out there.
For the database, we decided to go with Firebase by Google which is a cloud real time DB that has an excellent synchronization between all users by providing real time updates constantly. We haven't worked with firebase in our previous courses as we used nuBuilder at that time, but there are a lot of tutorials online which helps us get to know firebase plus it isn't that hard as firebase design is very simple and clear.

3. Similar Work:

In our research, we found some apps that are closely similar to the app we are trying to make and some apps that are broadly associated:

1. Close apps:

We found a lot of mobile apps on the play store and the app store, some are cross platform and some are unique to a specific platform. These apps include: Find my car smarter, Anchor pointer, Honk, Parking pin (IOS) and Car finder AR, Parkify, Car locator (android). These apps all have in common that only one user only can locate a specific car where he has to input details of the car and such which is time consuming and not efficient and contradicts with our vision in our project.

2. Broad apps:

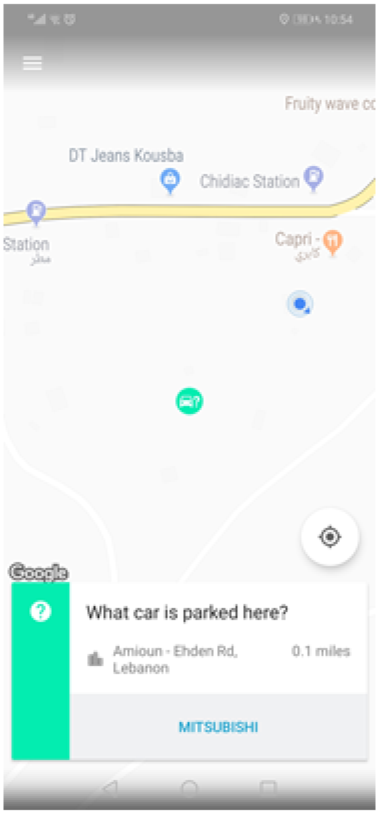
There are some apps that are not related to parking and retrieving cars but still have pins and locating cars and these apps are: Google maps (which we will use in our development), Apple maps and Uber. The latter app shows you the eight closest cars to your location and you can pin your destination so a driver can pick you up.

Our app differs from these mentioned above in that it will let multiple users manipulate the map and the pins put by other users when parking or retrieving a certain vehicle. The challenge in this application is keeping the database consistent and real time with all users simultaneously. Also, providing the maximum performance and accuracy of the pins to the cars parked is another challenge to keep in mind as it may affect the speed of which parked cars can be retrieved.

Similar apps review:

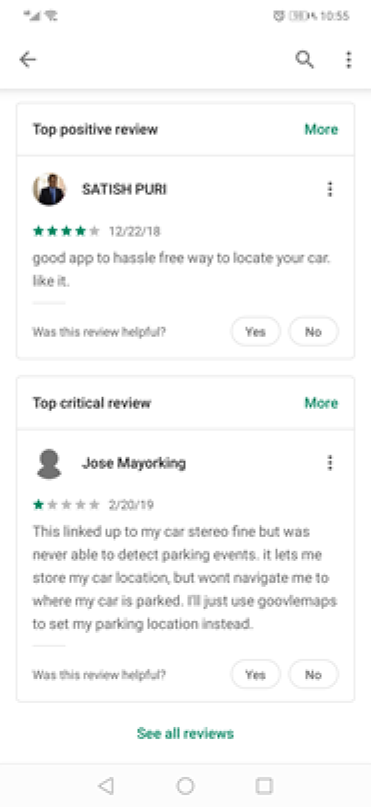
In Addition, we reviewed some of these similar applications to find out how can we be different from our competitors.

1. Parkify:



What car is parked here?

Amioun - Ehden Rd, Lebanon 0.1 miles



Top positive review More

SATISH PURI

★★★★☆ 12/22/18

good app to hassle free way to locate your car. like it.

Was this review helpful? Yes No

Top critical review More

Jose Mayorking

★☆☆☆☆ 2/20/19

This linked up to my car stereo fine but was never able to detect parking events. it lets me store my car location, but wont navigate me to where my car is parked. I'll just use goovlemaps to set my parking location instead.

Was this review helpful? Yes No

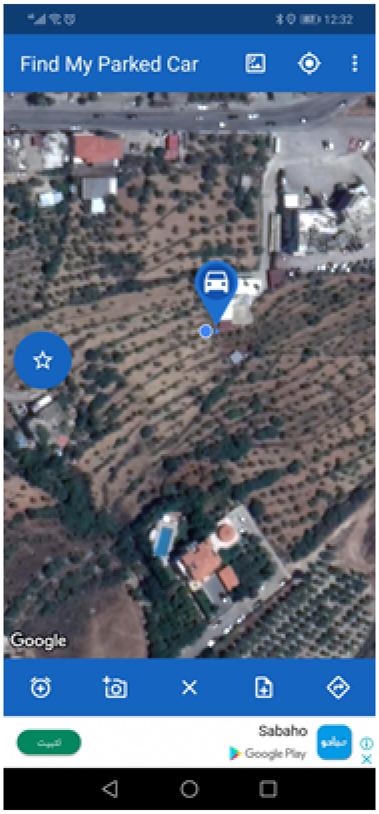
See all reviews

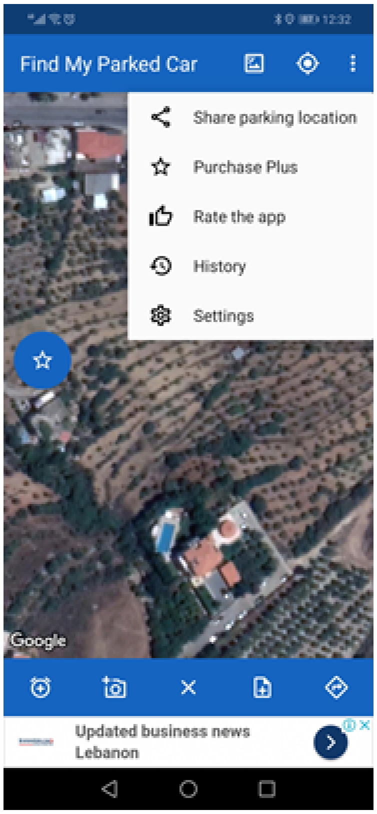
Parkify has multiple car locations available, editing information about the car, and sharing location via whatsapp.

However, it has ads, and the multiple car locations are only for one user.It gives you the option to sign in with google or facebook to keep your cars in sync online, however that also only works with one account so that wouldn’t work in our case where we need the locations to be common to multiple users. And while you can share locations via whatsapp, that doesn’t solve the problem since it is not feasible to maintain many car locations by sharing them on a whatsapp group via google pins. Also, when testing, we deleted a car but it was still showing on the map and we couldn't interact with it. So that's unacceptable when dealing with many deletions and new insertions. It claims to have directions to your car location but we found out that that wasn't the case.

It is great for what it's designed for, but we don't believe it can be an alternative to our app, which will have a common map for all valet employees, and dynamic updates which will make sure our map represents the latest state possible of our parked cars.

2. **Find my Parked Car:**



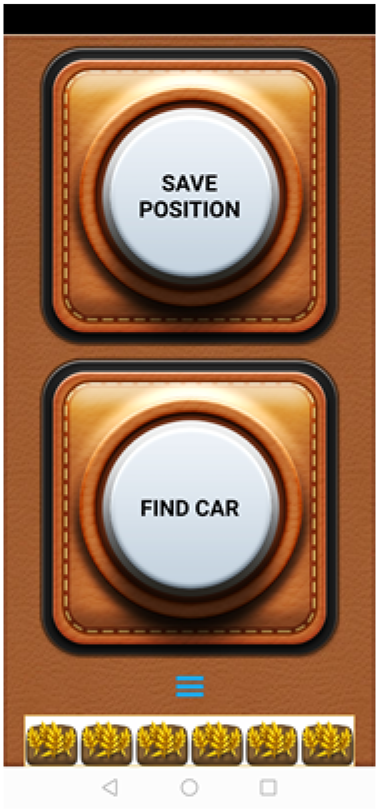


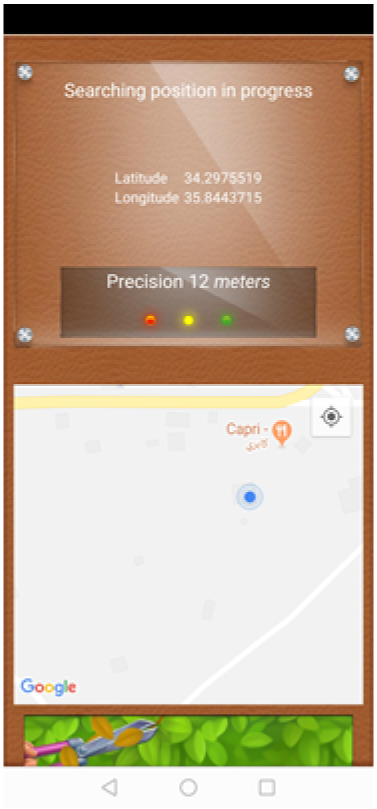
This is a very well designed app. It has a built in alarm(for parking meters), you can add a picture and notes to the parking location and it provides directions to the parking space. You can share the parking location via whatsapp as well.

However, it is designed for a single person and not for a valet crew, so it only has one car location available and adding a new location requires that you delete the other location. And it also features ads (which you can pay to get rid of).

3. <u>Car Find</u>

This app was able to save a position of one car and locate it later, but as we've said, only one car is not enough for a valet parking crew, therefore this app cannot be an alternative to our app. It also features ads.




Searching position in progress

Latitude 34.2975519
Longitude 35.8443715

Precision 12 meters

All these applications reviewed above are made for personal use where a single user can save a location and this location cannot be shared within a team while E-Valet is created for business use and differs from these applications as it has the ability to share locations and additional information across a team of multiple users that can read and write on the data shared.

4. Domain analysis:

Today’s valet parking job consists of a team of employees and not just a single person. They have one way with dealing with their jobs, one of the crew parks the car in a certain location that could be several blocks away from the restaurant or pub etc. this mentioned location is of course unknown to all other colleagues. So when retrieving the car, which can be several hours after its original parking, all employees except the one who parked the car have only one way of locating the demand vehicle which is walking down every block using an inaccurate description of the location given by his colleague and press the lock button on the key controller until they see the lights blink to find the requested car, Another mishap is occurring in this old and obsolete process, the one employee who parked the car may not remember where he parked the car because employees deal with a huge pressure of parking numerous cars especially in rush hours which are lunch and dinner times in the fastest way possible , the location of this particular vehicle can be vague, so the aforementioned employee will have to do the same process mentioned above and search for the car using the key controller.

Another challenge employees are facing is dealing with older car models that don’t have a key controller that makes the car blink, which makes the valet employees job even more difficult than it originally was.

This process is very time consuming as the search process takes a significant amount of time which is far more than going directly to a known place where the car is parked, and it is demanding much more effort from the

employee than necessary which can tire him quickly leading to more waiting times as walking and searching for a location is energy consuming.
In some more advanced places, electronic cards that have a button on them are distributed to customers so that the latter can send a beep using the button to the valet 10 minutes before they exit the place so the car could be ready and waiting for them and even this process has many failures as the electronic cards are not very reliable and sometimes break and don't beep a signal to the valet employee, other times customers lose their electronic cards with is causing a significant loss to the valet company as if one piece of this hardware went missing the remaining of the hardware has to be replaced as well.

5. SRS document:

I. Introduction:

The purpose of this document is to define a GPS based system called *E-Valet* that helps valet parking employees ease their car parking management. In addition, we are being aided by a valet company which is giving us their needs and concerns which helps us pinpoint their problems to better our system for their eventual use. The application will be convenient and easy to use so it doesn’t hinder the employees’ job but it helps them decrease customer waiting times and employee effort to retrieve numerous parked cars. The persons who have access to this document are the development team or administrators and any manager looking to purchase this product.

A. Definitions, acronyms, and abbreviations:

Term	Definition
GPS	Global Positioning System
Administrator	System administrator who is given specific permission for managing and controlling the system
Manager	A supervisor of a company most probably a valet parking one
User	Person who interacts with the application
OS	Operating System

II. Overall Description:

A. Product perspective:

The mobile application will be used to put a pin on location on the map representing a parked car and the retrieval of the mentioned car using the map.

The app will need to communicate with a GPS application on the employee’s phone, which in turn communicates with a physical GPS device to find the location of the employee when parking a car, Since this is a data-centric application, it will need to communicate with a database. The app will communicate with the database by storing and retrieving coordinates and information about a certain car.

B. User needs:

The intended users of this system are valet parking employees and not the general public. When a certain employee parks a car on a random block he will press the “pin location” button and a pop up will show to let him add additional info if he wants to, after this process the location coordinates of this particular vehicle will be saved in the database with all the info entered by the employee. Now for the retrieval process, the employee will add the ID of the desired car, then the location of this car would show on the map to visualize where the car is parked to retrieve it. When the car is delivered, the car ID with all information will be deleted to provide concurrency to the database.

C. Assumptions and Dependencies:

One factor that may inhibit the application to work normally is the hardware of the phone the employee is using. If the phone is outdated, the app may not work properly and cause performance issues.

Another factor would be the internet connection, as our data will be saved over the internet on firebase. If the valet employee is not connected to the internet or the connection is weak the application may malfunction.

D. Implementation constraints:

- ➢ Linking pins to appropriate data.

- ➢ Dropping pins representing cars on Google Maps.
- ➢ Providing the maximum performance(accuracy to find cars, fast and responsive).

E. <u>**User Characteristics:**</u>

There will be 2 types of users accessing this application: the valet employees and the administrators.

- The valet parking employees are the first type of users, they will use the system to park and retrieve cars as mentioned above as well as delete cars after their retrieval.
- The administrators are managing the overall system by setting up the accounts for the valet employees.

F. <u>**Operating environment:**</u>

- <u>Database:</u> The database we will be operating with and saving our data on is Firebase. It is a Real Time sync cloud-hosted NoSQL database that can operate on both mobile and web apps.

- <u>Operating systems:</u> E-Valet will be developed on both IOS and Android mobile OS.

- <u>Platform:</u> The application will be developed using React-Native.

III. Specific requirements:

A. Interfaces:

1. Hardware interface:

Since the app doesn't have any special hardware, it does not have any direct hardware interfaces. The physical GPS is managed by the GPS application in the mobile phone and the hardware connection to the database server is managed by the OS on the mobile phone

2. Software interface:

The system communicates with the GPS application in order to get geographical information about where the employee is located, as for the database, there will be reading are writing operations between it and the system consisting of writing information such as ID, location and description.

3. Communication interface:

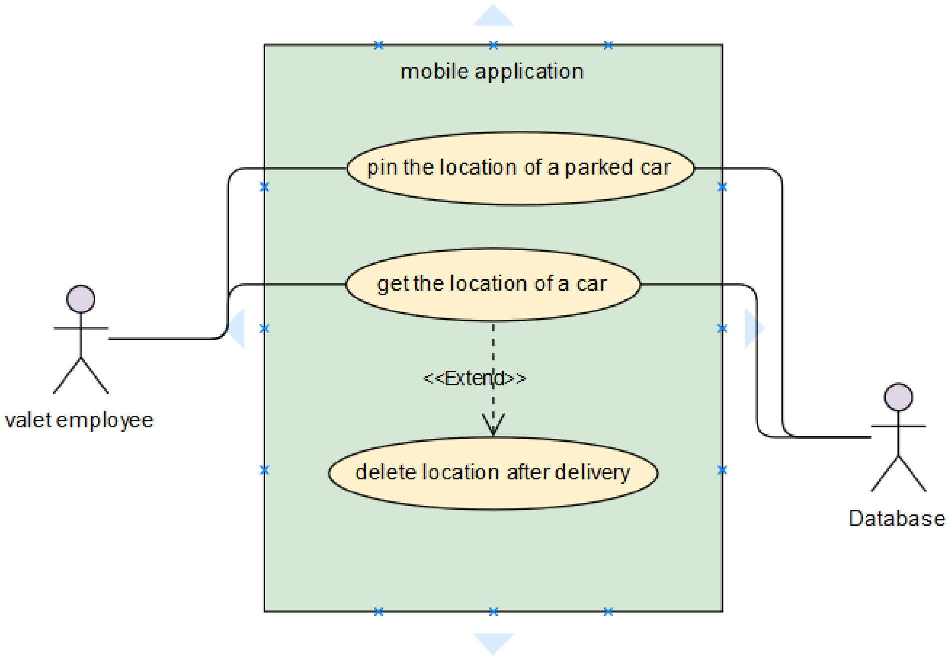
Communication between different parts of the system from GPS to app to database is handled by the OS.

4. User interface

There are several interfaces in E-Valet, some are used to park cars and interact with the database by inserting Information such as ID, description and an optional picture to it, while the car retrieval part incorporates Google maps and the database by looking up the location on the Database and opening it in Google maps. More can be read in the specific interfaces part of this report.

B. Functional requirements:

As seen in the use case diagram below, the first functionality consists of putting a pin on the location of the car once the employee parks it, then a prompt pops to let him add information about the car if he wishes to include additional info. Another functionality includes getting the location of an already parked car to retrieve it for the employee's customer, after finding the car location on the map and retrieving it he can delete the pin of the delivered car to keep only current parked cars on the map to provide a realistic count on the cars under supervision. One requirement not mentioned in the use case shown below is the ability for the valet employee to login/logout of accounts created by E-Valet administrators.


mobile application
pin the location of a parked car
get the location of a car
<<Extend>>
delete location after delivery
valet employee
Database

C. Non-functional requirements:

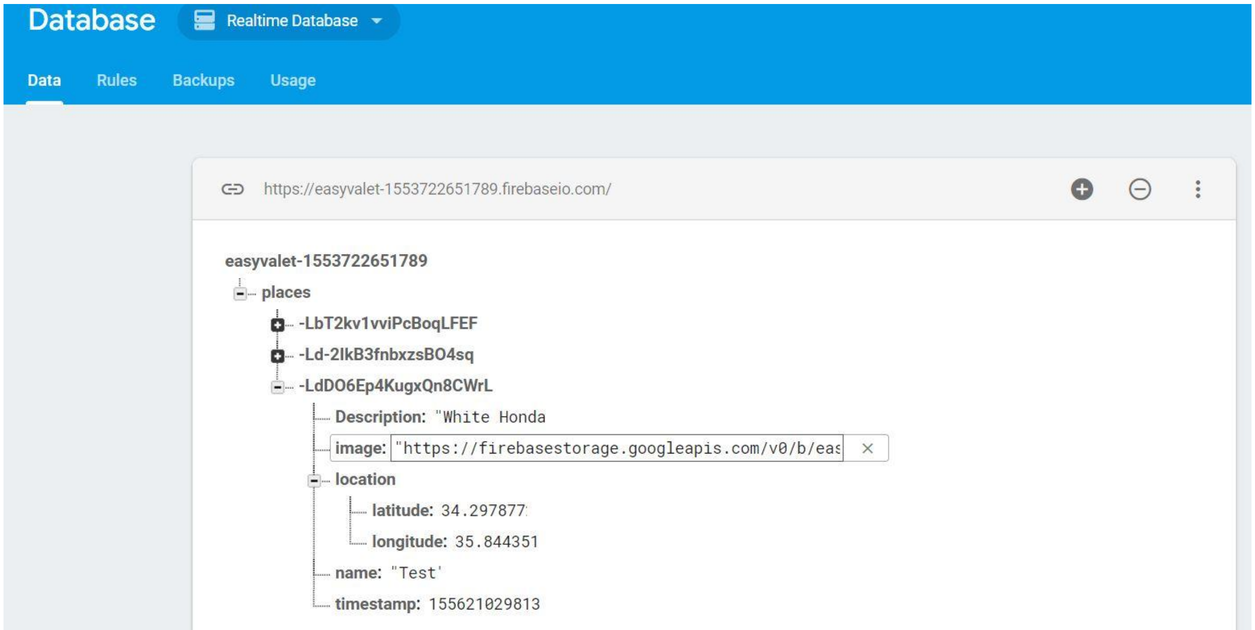
Usability: The application is very easy to use as it’s straightforward and is especially designed to be easily operable to save time in learning the system and working with it, as employees will encounter rush hours, this system should help them complete their tasks faster and not act as a burden to them and slow them down.

Performance: The system should display location or save location of a parked car as fast as possible, in addition, the system should show the location of the car accurately to avoid employees searching for a car in the wrong street.

Availability: the app will always be available for peak customer times where employees will have the most work to do which are lunch times (12:00 pm - 3:00 pm) and dinner/pub times (8:00 pm - 2:00 am).

6. Data Manipulation:

In this document, we will be presenting how data flows in the application from the point where it is called and manipulated until the user is done with it. We will address the source of the previously this data, its fields and their meanings.



```
https://easyvalet-1553722651789.firebaseio.com/

easyvalet-1553722651789
  places
    -LbT2kv1vviPcBoqLFEF
    -Ld-2lkB3fnbxzsBO4sq
    -LdDO6Ep4KugxQn8CWrL
      Description: "White Honda
      image: "https://firebasestorage.googleapis.com/v0/b/eas
      location
        latitude: 34.297877
        longitude: 35.844351
      name: "Test'
      timestamp: 155621029813
```

Data in E-Valet is used in 2 separate actions with the users, either when parking a car where the user writes data to the database or when the user wants to retrieve a car he reads from the database then deletes the entry read.

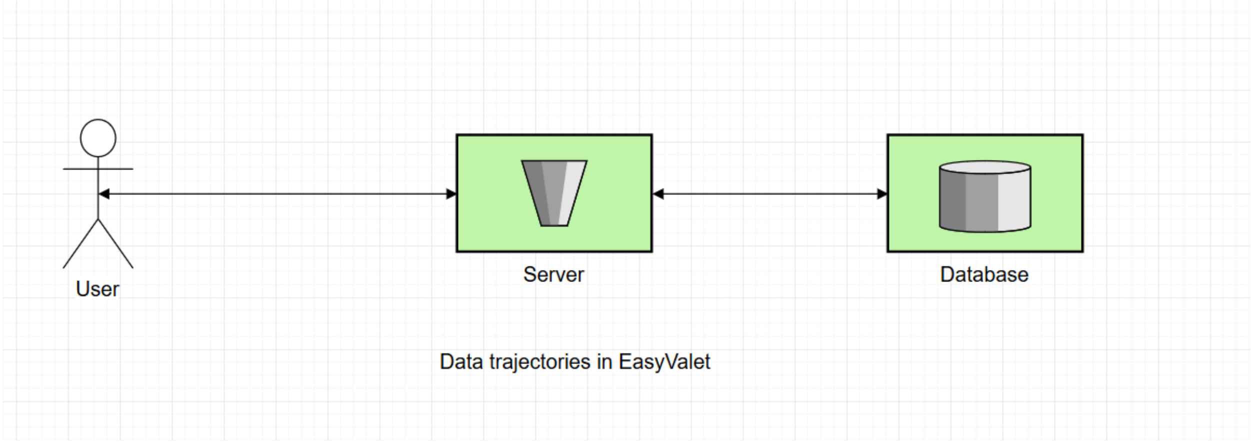
Data is first instantiated when a valet parking employee parks a car and wants to save its location. As the interface document explains, the user gets his current location through the GPS on his device, adds an ID to the car; photo and a description added are optional. All this data sent is in JSON (Javascript Object Notation) format from the user to the server which will send it to our Database.

In the image above, this is what the data looks like in Firebase (the database we are using in this project). Under “*places*” which is the table the data resides in, we have different entries, each represents a different vehicle under the company’s service. The different fields in each object are:

- Description: (String) an optional field where users can add a brief description of the car being serviced.
- Image: (BLOB) an optional field, users can take a snapshot of the car using their device’s camera.
- Location: (Int) This field is consisted of 2 sub-fields: longitude (a horizontal geographic coordinate that specifies the east–west position of a point on the Earth's surface) and latitude (a vertical geographic coordinate that specifies the north–south position of a point on the Earth's surface). The data present in this field are retrieved from Google maps when the user requests his current location.
- Name: (String) represents the ID of the parked car, this number is not random but is inspired from the receipt given to the customer when employees take his car, this field is important as it differentiates a car from another.
- Timestamp: (int) the time when the data entry has been created.

These objects can be seen in the application when users wish to retrieve cars. A list of car IDs is first seen, the additional info is found when an ID is clicked . Once the user retrieves a vehicle, his last interaction with the database is done which is deleting the entry completed which can easily be done. Once deleted by the user, the deleted entry will not be shown in the List of parked cars anymore as it’s not present anymore in the database (refer to interface document for visual interpretation).

In the picture below you can see graphically how the data can flow when users interact with the application



Data trajectories in EasyValet

7. Tools and Frameworks:

In this section, we will be displaying all the major technologies used to develop E-Valet by defining what they are, what they are used for and some alternative choices we could have chosen.

- ### Main Frameworks:

React-Native is our main programming framework for the front-end of the application. We already have prior experience with React by working with the technology in previous courses which gave us some experience with the React philosophy which can be applied to React-Native. React is currently one of the most sought-after languages in the market nowadays which encouraged us to use this technology in our project, another advantage to React-Native is that apps built

using this technology ensures speed and agility with responsiveness and a great native app-based user experience, since in the end all the code will be compiled into native code.
Alternatives to React-Native are Apache Cordova, Mobile Angular and Ionic Framework. The main difference between React-Native and its alternatives are that React offers HTML and DOM support while their alternatives don’t. React-Native being created by a large company such as Facebook has a very active community which gives out different plugins every day to make developing with this technology easier.

Accompanying React-Native is Redux. The latter is a state management library, and is often used with React-Native to simplify data flow within an app. It helps write applications that behave consistently, run in different environments (client, server, and native). Redux has three fundamental principles: Single source of truth (The state of your whole application is stored in an object tree within a single store), State is read-only (The only way to change the state is to emit an action) and Changes are made with pure functions (To specify how the state tree is transformed by actions, you write pure reducers).

- **Database:**

Firebase by Google is used to store the data manipulated in E-Valet. Firebase is a cloud real time DB that has an excellent synchronization between all users by providing real time updates constantly to users. Advantages for Firebase are:

- Fast, secure, static, and production-grade hosting for developers. It allows developers to efficiently deploy web apps and static content.
- Instant UI Libraries and SDKs for authenticating clients across application using usernames or passwords which helps with integrating several sign-in techniques for developers for their apps to allow users to sign in.

Alternatives to Firebase are MongoDB, Kuzzle, Parse and CloudBoost. These apps shown here are all open source NoSQL databases but we chose Firebase instead of those listed above for it’s mentioned advantages and for its reliability as a Google product.

- **Other Technologies:**

Google Maps is a service that provides detailed information about geographical regions and sites around the world. In addition to conventional road maps, Google Maps offers aerial, satellite and street views of many places. Maps for Mobile offers a location service that uses GPS to track the location of the mobile device. We are Using Google Maps as it’s API is the easiest to integrate in apps and is widely and commonly used among developers in their mobile apps.

Alternatives to Google Maps are MapQuest and Waze.

8. Various Interfaces:

In this section, we will portray all the different interfaces interfaces in the application and how they are used.

Figure 1 is the first interface the user will get once he enters E-Valet. On this UI, users can sign in by entering their email address and password or if they don't already have an account, they can press the *"switch to Sign Up"* button where a *"confirm password"* field will appear and the user would enter his email and password twice to complete the signup process and access the application.

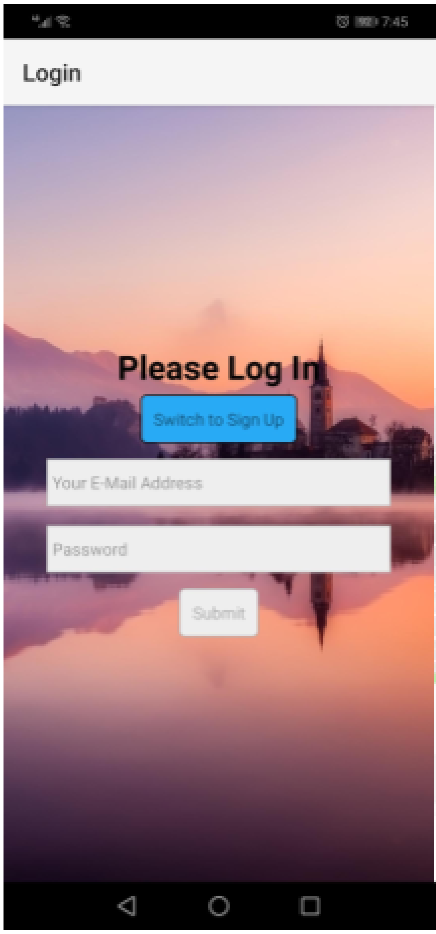


Figure 1: Sign in/ Sign up

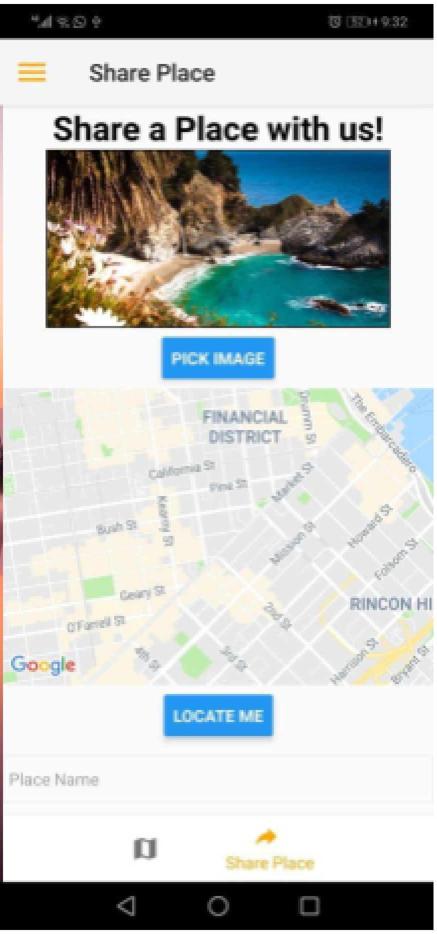


Figure 2: adding a location

Once inside the application, the first interface that will come up is **Figure 2**. On this interface, users will be able to add a location to the list of saved locations on the database, each saved object can hold an image which can be of the parked car by pressing the *"Pick Image"* button which will open his phone camera, the *"Locate Me"* button on this interface uses the GPS on the device to locate the current location of the user which will be used when the car is parked to identify the location of the car once parked, this location will show up on the map above the button. Under these 2 options, the user will add the ID of the parked car which will be the coupon given to the owner of the parked car. All these info will be added to a list of all cars currently working with. This concludes the process of parking cars.

Next, when retrieving parked vehicles, Users deal with 2 interfaces. The first interface shown in **Figure 3** shows the List of all cars parked by Valet employees at a certain point, so in this illustration, *"1"* and *"Testing"* are two different objects

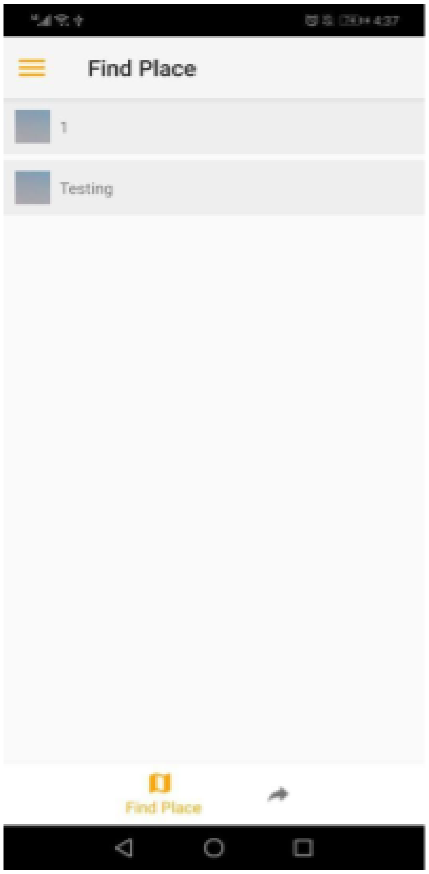


Figure 3: List of parked cars

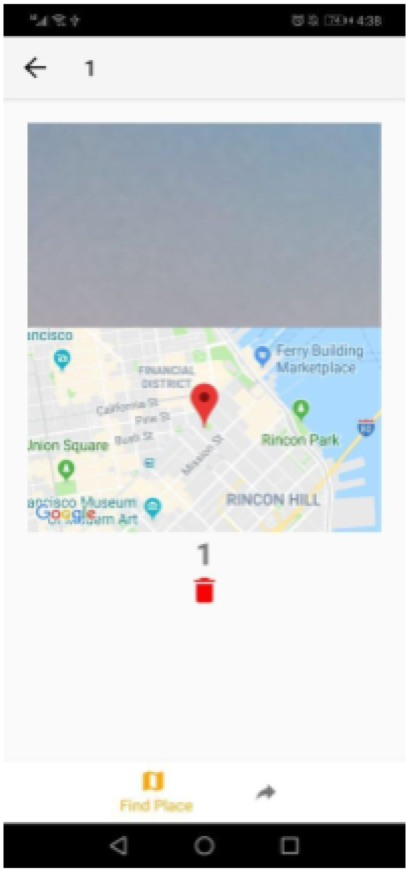


Figure 4: individual object

representing vehicles currently being manipulated by the valet employees. Each of these 2 objects is clickable, **Figure 4** represents the interface shown once the user clicks on one of the objects already mentioned, in this example vehicle “*1*” is pressed, this interface is used when an employee wants to retrieve a certain vehicle, so on this interface he can see a map representing the location of the parked car, below it the car ID of this car which is as mentioned before the coupon number that the customer has and below that a little red bin which is pressed once the car has been delivered to its owner so that the object of the car can be deleted and won’t be seen in Figure 3 anymore.

9. Challenges:

With every project, there will always come challenges developers will have to face. E-Valet is no different as there are numerous challenges we faced during the development of this application and they are cited as such:

- Devise an appropriate data structure to readily identify the cars (ID, Description and Location).
- Ensure a reliable and accurate representation of car locations.
- Ensure regularity of updates to the cloud of which cars have been parked and/or retrieved.
- Provide some way to easily communicate to the customer the location or the ETA of their car.

10. Future improvements

- Have a communication system between the car owner and the valet employee. The car owner will have an option to request the delivery of his car 5 minutes prior to his arrival to the retrieval point so when he comes out of the place where he was spending time in his car would be ready and waiting. This functionality can improve delivery speed and provide customer satisfaction from this process.

- Ability for owner to track the location of his own car. In this feature, car owners can access E-Valet to check where his car was parked and if it has been moved. This feature will provide integrity and trust to employees amongst fears from car owners that employees are using their cars in an unethical manner.

- Add timestamps and reports for amount of cars parked to help the business owner assess his business. An important feature that can help business owners assess the profitability of his business and try to improve the park and retrieve process.

11. Conclusion:

In conclusion, E-Valet is a mobile application designed to aid valet parking employees in parking and retrieving cars usually in populated urban areas. This report has presented the various sections in the development of E-Valet mobile application. The first part is an introduction of the idea and the objectives of this application, later we discussed similar close and broad applications while reviewing some of them. That section is followed up by a domain analysis and an SRS document. Furthermore, in the section that follows, we discuss the database used and how the data flows and is manipulated in the application. In section 6, we define all the technologies used to build E-Valet followed by the various interfaces users will use to work on the application, in the section above this, we see the various challenges E-Valet dealt with during its development.

E-Valet will not be available on Google Play Store nor IOS app store as E-Valet will be exclusive to companies wishing to use this app and will directly purchase this software from the development team.

Finally, E-Valet has the potential to exceed the concept of a university final project as this application solves real life problems and customers are open to purchase and integrate this kind of technology into their daily jobs to improve the way the do things.