1. **Introduction**

**1.1 Purpose**

Our team is focus on specifying the accomplishment of the service PowerEnJoy, this is a system which supply the electric cars to facilitate the transportation of citizens. In particular, the PowerEnJoy will offer the accessible electric cars in the reachable areas in the city and admeasure them rationally to users; the system also set rider clauses to standardize the driving behaviors of users by provide discounts and add compensations.

This document aims to :

* Describe the software and hardware architecture of the system we build in the design part
* Make some reasonable adjust based on the design part, improve the system model into a more realizable direction.
* Focus on the technical approach on the system realization, analyze the deep level about the algorithm
* Draw a more clear blueprint not only on the interface aspect but also on the use interaction.

This document is intend to be a guide of the developers who will implement our system to explain and clarify their work.

## 1.2Problem Description

In particular, the system wants to:

1) Provide an easy and efficient approach to users

2) Guarantee an energetic and accessible car service

Anyone can register to be a user by supply the valid driving license information and credit card information.

After the log in, users are able to search the available cars around the location they are or around the certain position they inputted. System response the search request by enumerates the available cars around them in an available queue as well as the basic information. The basic information of cars include: the accurate location, the distance to start position, dump energy, and passenger capacity.

Users are able to reserve at most one car every time, they can cancel the reservation in free within a certain time after the reservation request. The user can’t pick the car in the one hour time span should pay one euro, and the reservation will be released as well.

The system will move away the car from the available queue as soon as the car is reserved, it will move back if user cancels the reservation or the car doesn’t be picked up in one hour.

If the user accesses the car with time limit, he /she can send request to pick up the car, system will unlock the door and let user enter. The system will start charging the car as soon as user ignites the engine, and the charging information will be notified through the screen in the car.

The user is able to choose the “money saving option” after picking up the car, and the system will response the optimal terminal station to park the car.

Users are able to terminate the ride in the safe area, the charging will stop and the car will be locked when all passengers leave the car, no matter how engine works.

If the user parks the car in the unsafe area and leave, it is necessary to lock the car for the reason of security; the system will consider that as a complete driving process with unsafe parking.

Users can recharge the car in the power gird station by plugging the car into power grid.

System estimates the driving behaviors in this ride according to the rider clauses as follow to reset payment:

If the user took more than 2 passengers, apply 10% discount.

1. If the user parked the car in safe area with more than 50% dump energy, apply 20% discount
2. If the user recharged the car after parking, apply 30% discount.
3. If the place user parked car is in a longer distance than 3KM to the nearest power grid station, apply 30% compensation.
4. If the user parked the car with less than 20% dump energy, apply 30% discount.
5. If the user parks the car in an unsafe area, apply 30% compensation.

## 1.3 Glossary

1. Car: the cars that supplied for the car-sharing service in the PowerEnJoy system.
2. Car information: the basic information that helps guests and users to make decisions, include the dump energy, location information, distance to the setting location, the passenger capacity.
3. Starting position: the current position of user or the positions user inputted to start a ride.
4. Available car: the car has dump energy more than 50% and be parked in the safe area.
5. Available queue: a queue that maintains available cars
6. Sensors: the GPS and power plug senor, weight sensor, display screen, battery sensor, door state sensor, locks of door in the car, and the sensor on the power grid.
7. System: the whole system which include the electric devices and the PowerEnJoy system background.
8. Ride: in this system, the ride process is start with ignite the engine and end with all passengers leave the car.

## 1.4Used Tools

The tools in used to create this RASD are:

*StarUML*

To create UML diagrams

*Balsamiq mock-up*

To create the mock-ups of the interfaces

*Dropbox and Google Docs*

To share the files and synchronize our work

Queue management

This algorithm needs the position information about all the available cars in the city, as well as the accurate start position information from the user. Every time user launches a request for the available cars,the Available Queue Manger gets the coordinates of available cars in the city, then it starts the research of an available queue of cars around the start position which insert by user or oriented by GPS.

cars in the queue will be updated when a new request of searching is launched. It can be added with the cars which is released from a ride or a reservation, it can also be removed by reservation.

Cars in the available queue are enumerated in the increase order in “distance to the start position”and the decrease order both in dump energy of car. The available car which is far than 3 KM are be consider as unreachable cars, and won’t be enumerated in the available queue.

The research is done in according to the algorithm bellow, written in pseudo code:

Function searchAvailableQueue

StartPosition start<- GPSofMobileAPP(request.gps)/InsertionFromMobileAPP()

Array car<-GetArrayOfCar(car)

while Distance(car.location,start)<=3

sort{

cars as car;

by Distance(car.location,start)\*10000000+ (100-car.battery)

}

return cars;

End function

Requirement traceability

All the functional requirement should be accomplished by the interfaces of those components in the PowerEnjoy system, so in the following table, we map the functional requirements with the corresponding components which realize the function.

Guest

|  |  |  |
| --- | --- | --- |
| Reference | requirement | component |
| [1] | Register to be a user | Account manager,bank system and traffic system |
| [2] | Get the notification about register | Notification manager |

User

|  |  |  |
| --- | --- | --- |
| Reference | requirement | component |
| [3] | Log in | Account manager |
| [4] | Search for available cars | Available queue manager  Google API |
| [5] | Make a reservation | Reservation manager |
| [6] | Cancel reservation | Reservation manager |
| [7] | Check current reservation | Reservation manager |
| [8] | pick up the car | Mobile APP and  reservation manager |
| [9] | Access the saving money option | Google API, optimal path calculator and On-Board system |
| [10] | Get the discount/compensation | Payment manager |