

# **DELIVERABLE 1**

## **WEB PROJECT**

GEIADE - 21/03/2022

***Àngel Baró Noguero - 49255763K***

***Hector del Egidio Gonzalez - 48256267J***

***Jordi Esquerda Cullerès - 48057037D***

***Gerard Mallol Descàrrega - 41566688F***

***Pau Torné Enrich - 48055969E***

## **Index deliverable 1**

1.	PROPOSED MODEL IMPLEMENTATION.....	2
2.	GITHUB REPOSITORY, HEROKU AND DOCKER.....	3
3.	DEVELOPMENT PROCESS.....	4

# 1. Proposed Model Implementation

Our proposed model is about booking and order managing the different services that our platform “CityMove” offer, that are booking three types of vehicles to go around the city.

As an administrator of the platform, you will be able to add different routes, different prices to each vehicle and new vehicles of each type.

From our UML diagram we have transformed the different classes into Django classes. These ones are:

- **Class User:** This class is provided by Django and it has different attributes like username, email, if the user is active, etc. When a client registers to our page an instance of User is created.
- **Class MyUser:** This is a subclass of User. With this class what we get is a few more attributes, as they are the vehicle and the route that the user is taking, apart from his username, email, etc.
- **Class Transport:** This class represents the different vehicles that our clients can reserve. No matter which vehicle is, all of them have common attributes, that are an id to identify them, an initial price and if they are available or not to use. From this class now we have three subclasses, that are the three different types of vehicles that the platform offers.
- **Class Bicycle:** This is a subclass of Transport, that represents one of the different vehicles that the platform is offering. The class has an extra attribute that is the size, depending on if you are a child or an adult.
- **Class Scooter:** This is a subclass of Transport, that represents one of the different vehicles that the platform is offering. The class has an extra attribute that is the range of the vehicle.

- **Class ElectricScooter:** This is a subclass of Transport, that represents one of the different vehicles that the platform is offering. The class has an extra attribute that is the range of the vehicle.
- **Class Route:** This class represents the different routes that a user can do. After the client has made a route, with this class we will get where the client started and where it finished. It also shows the id of the transport the client is using, when the service started and how many kilometers did he make.
- **Class Record:** This class represents the information of each service that the different clients have received, showing the route that they made, the vehicle that they used, etc.

## 2. GitHub repository, Heroku and Docker

Our GitHub repository is <https://github.com/angelbn01/DjangoPracticeFinal> . Firstly, we had another repository that was [https://github.com/Hdelegido/Django\\_project](https://github.com/Hdelegido/Django_project) but we had some troubles with it so we decided to create another one. We also add this link as it contains some of the commits that we have made during the implementation of the project, in case someone wants to see the first commits that we made.

We worked all together by dividing the different tasks of the project and helping between us in all situations, as we had many troubles related with the configuration of the project and with heroku.

As far as Heroku is concerned, after many attempts, we have not been able to open the web page address successfully because we get an error that the provided address is not within the allowed host.

The program, when the command `git push heroku master` is executed, processes it correctly and creates the link of what would be deployed in the application. The problem is that, when we enter it, it appears that the address we want to enter is not

available in the hosts. We have made several changes to try to fix this issue but none have been successful.

This problem will be solved for the next deliveries.

### 3. Development Process

The first thing we did was the configuration of the virtual environment and Django. After that we divided the different tasks and started implementing the different classes in the models.py file. We also started by implementing the home.html file to represent the different classes.

The Django admin interface allows their users to manage the database with a graphic interface, so once we had the classes implemented, we used the command *python manage.py makemigrations* in order to create a migration of the changes that we made, that were the implementation of the classes. We also used the command *python manage.py migrate* to apply these migrations.

To access the admin site we had to create a superuser, which will be enabled by default in Django. We did this with the command *python manage.py createsuperuser*. Now we are able to access our local domain where the admin site is and add different vehicles or routes, for example.

As in this project we want all the information registered in the database to be showed into the home page, in the file views.py we created the function home, that included the template home.html and with this we are able to print all the information of the different classes in the home page, by using a loop to print the different attributes.