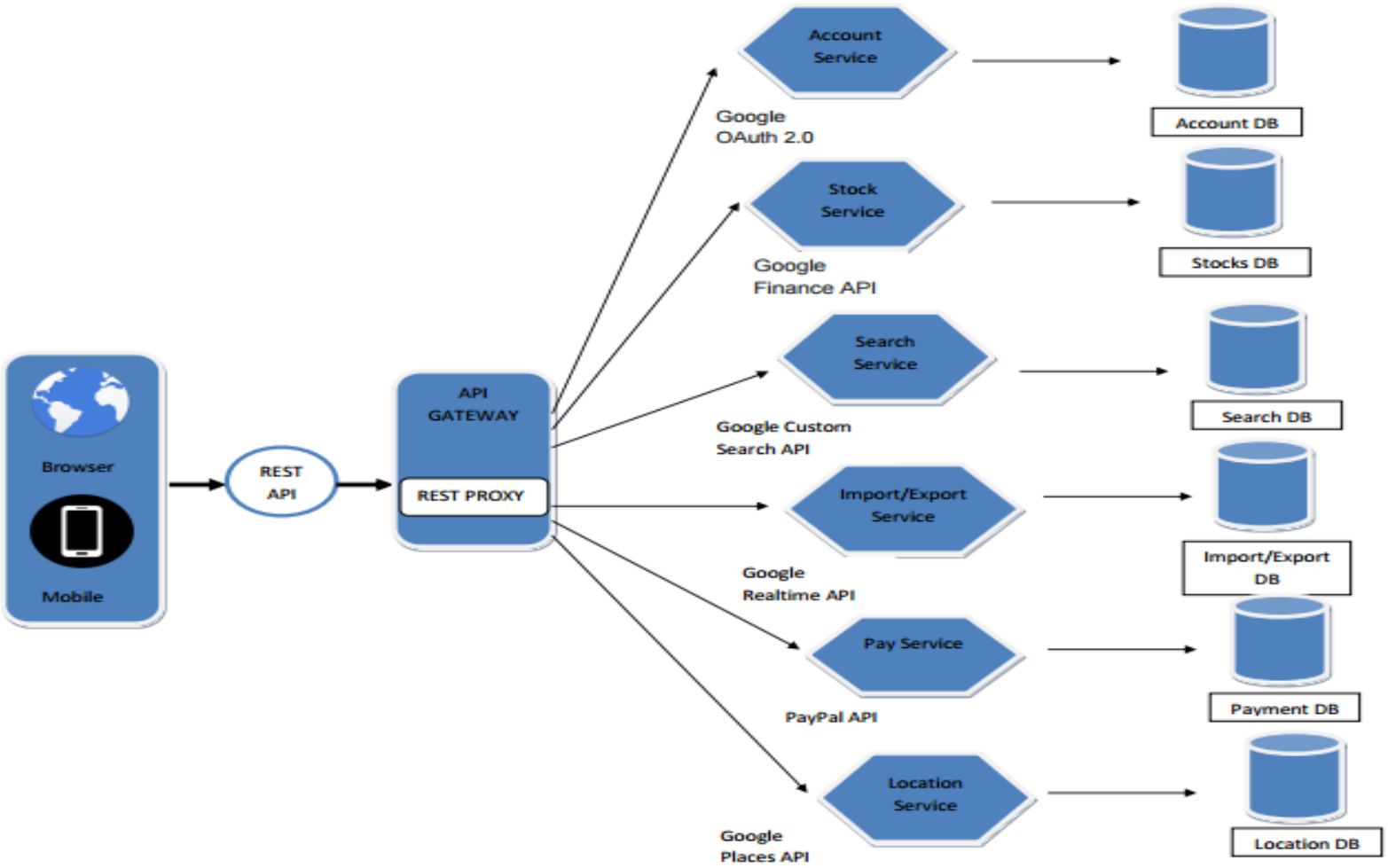
Part 2. SOFTWARE ARCHITECTURE

Regarding the software design of the application, for web app we adopted the microservices architecture as it is also presented also in the following figure:



As it can be seen, the **User Interface** is represented by the interface from the Browser or from the Mobile. The 'Client request' is passing through the User Interface and through the '**REST API**' which is giving access to the user to the Web resources. This request is after that passing through the '**Rest Proxy**' which is used for the improvement of the user response time.

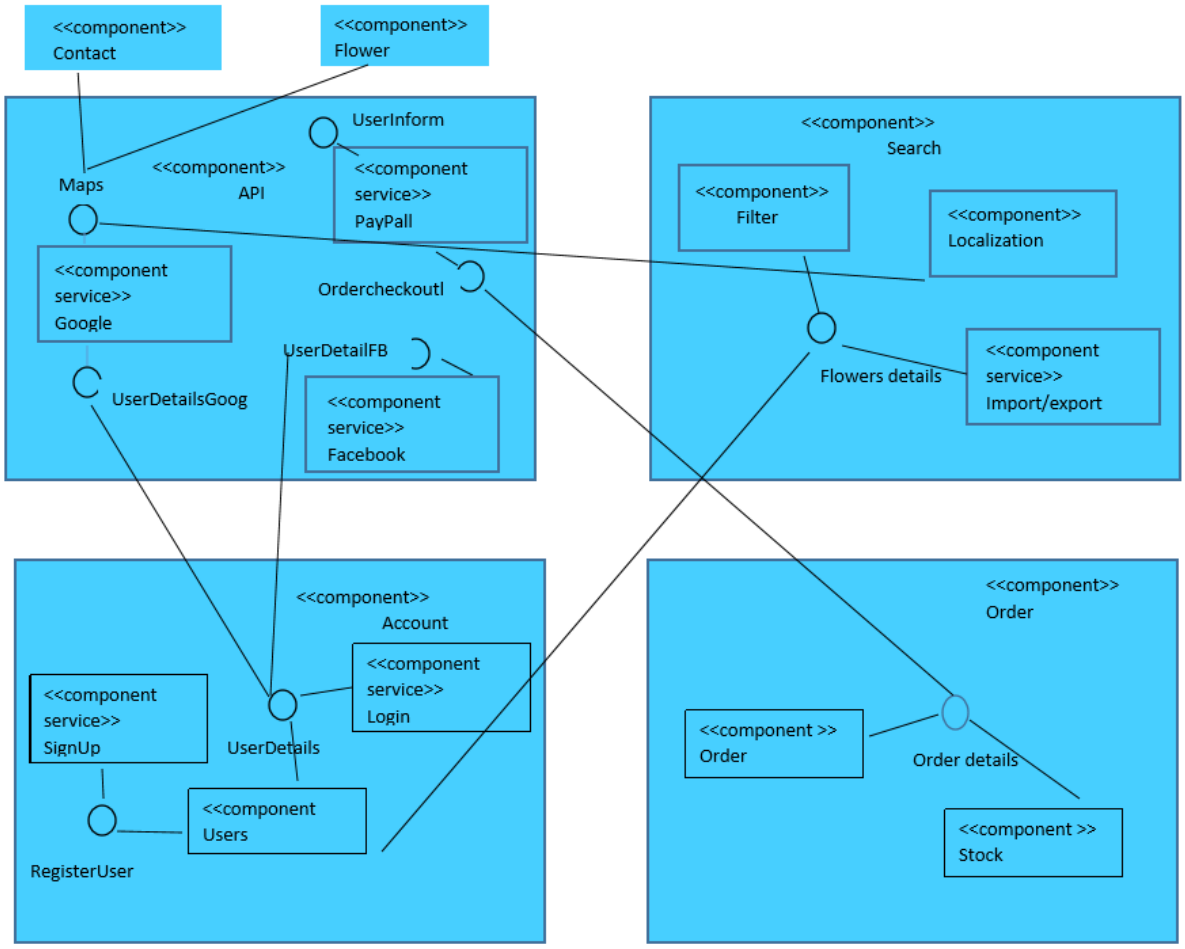
In what comes next, the system is requesting a few services that **APIs** can offer in order for our app to cover all the needs of the clients in order to be able to use the Web app and also to achieve the main objectives that they had when using our app. In our application, the user has the possibility to either create a new user – where we will use the '**Account Service'** – or simply navigate throught the app as an unregistered user. Of course, the permissions will be different for these 2 types of users. The user accounts and all the informations related to our users will be kept on the DataBase.

Regarding the '**Stocks Service'**, it's main objective is to keep the stock for both the flower that we have in Our store and the one that are used as exhibits for our Botanical Garden. As you may expect, when a specific product is sold from our store, the stock is updated and also for specific species of seasonal flowers. Moreover, the payments that will be done for the sold flowers from our store will be managed using the '**Payment Service**'.

The '**Search Service'** is used for the possibility to filter the content on our app,. in our case we are talking about the exhibits in our botanical garden. The Service will create indexes , load data, execute different queries depending on the filters applied and will return the results. Also, the users have the possibility to Import/Export data from our app. This functionality is facilitated through the '**Import/Export Service'** which is also managed using the DB.

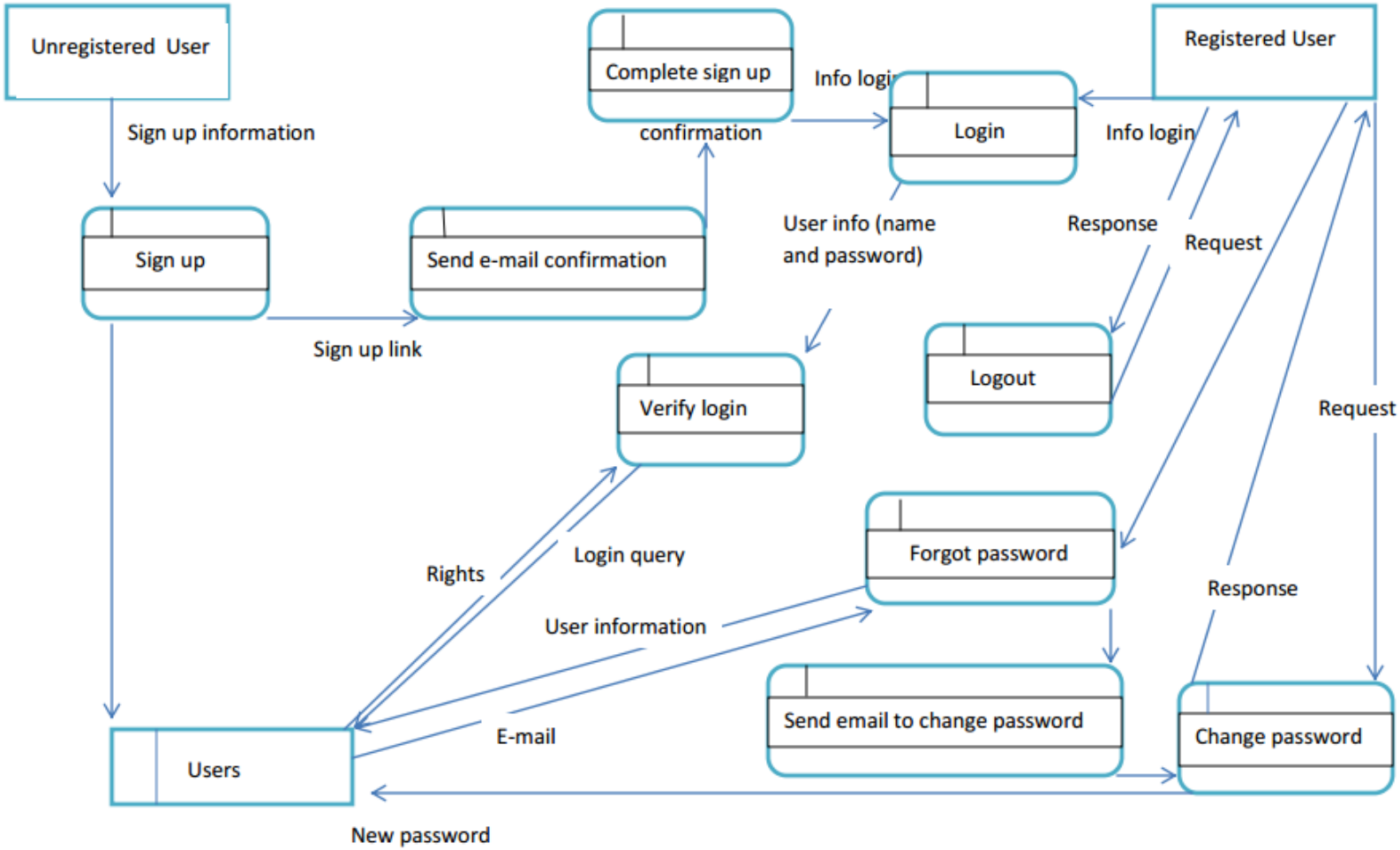
Taking into consideration that the user has also the possibilitu to be located in our Botanic Garden and to be guided to the location that he/she wants to get to inside the Garden, we used the '**Location Services'** provided by Google.

The functionalities of our web app are determined by all its components which are presented in the next diagram through its main modules and the relationships between them.



In the diagram are presented the main service components: Search, Stock and Account Services and the REST API used to create the connection between them. In our case, Google, Facebook and PayPal for the payment of the Orders.

In what comes next we are going to present the flows of the information within the system and we will attach some data flows diagrams:



As in the case of many other web applications, the user are of 2 types in the case of this diagram. One type is related to the users that do not have an account and the users that already have one.

Firstly, we will start with the users that do not have an already existing account and plan on creating a new one. In order to sign up each and everyone of them will have to fill in the ‘Sign up’ form and Submit it. As a results of this action, the system will send an e-mail confirmation than will need to be accepted by the user in order for the sign up to be complete.

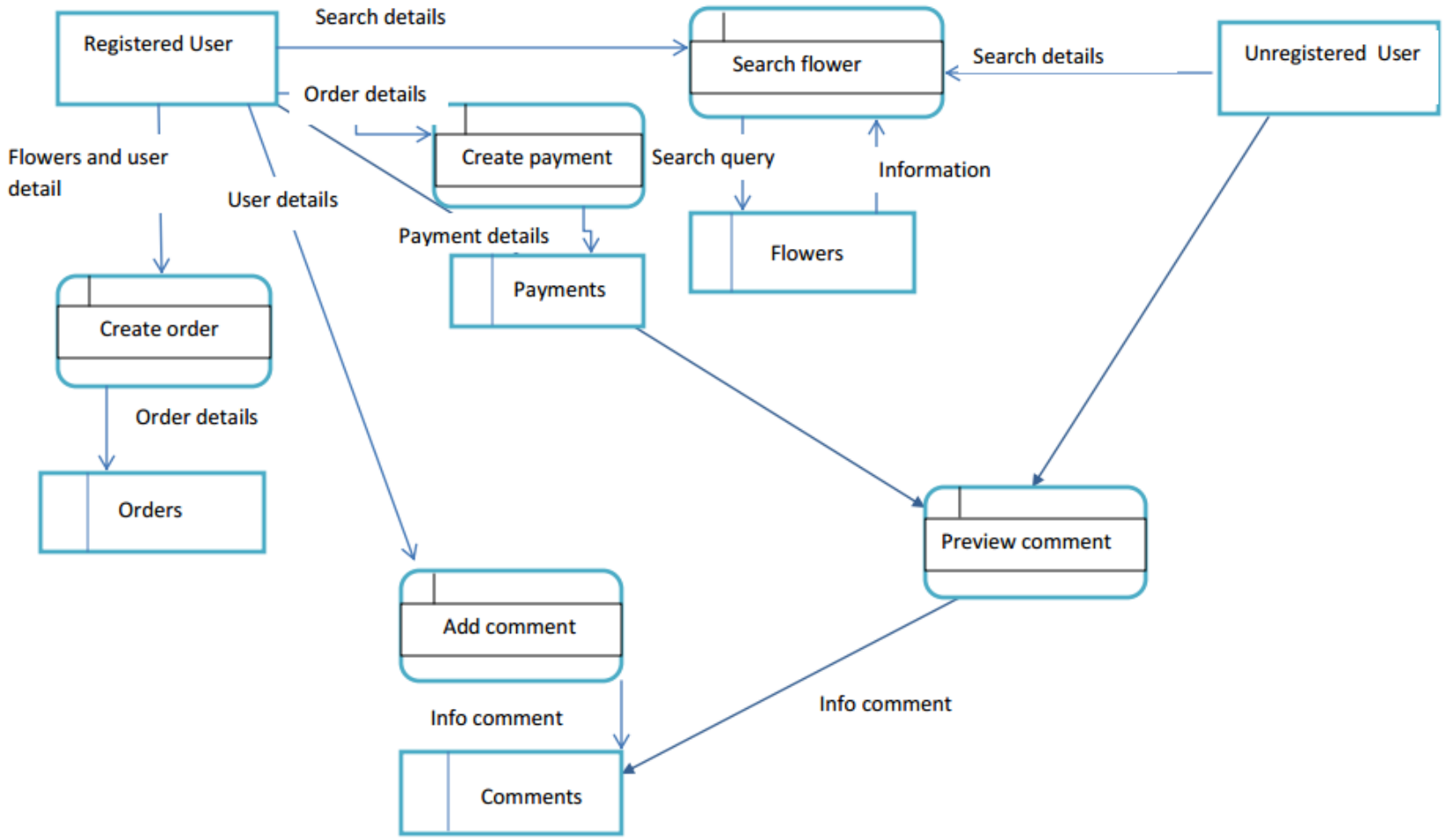
And here is where the second type of user is taken into consideration. This time the user has an account created. In this case, the user has the possibility to Login using the form. Once the ‘Login’ form is filled in, the system will check the information provided also in the Users Table where all the information related to the Users are managed. In case the login can not be done because maybe the user forgot the password or the user simply wants to change the password as a security measure, the system provides the possibility to change the password via e-mail and confirmation. Once Logged in, of course, the user has also the possibility to Log out.

In the next part of the technical report, we will present the rights that the register/nonregistered users have among the functionalities of our web application.

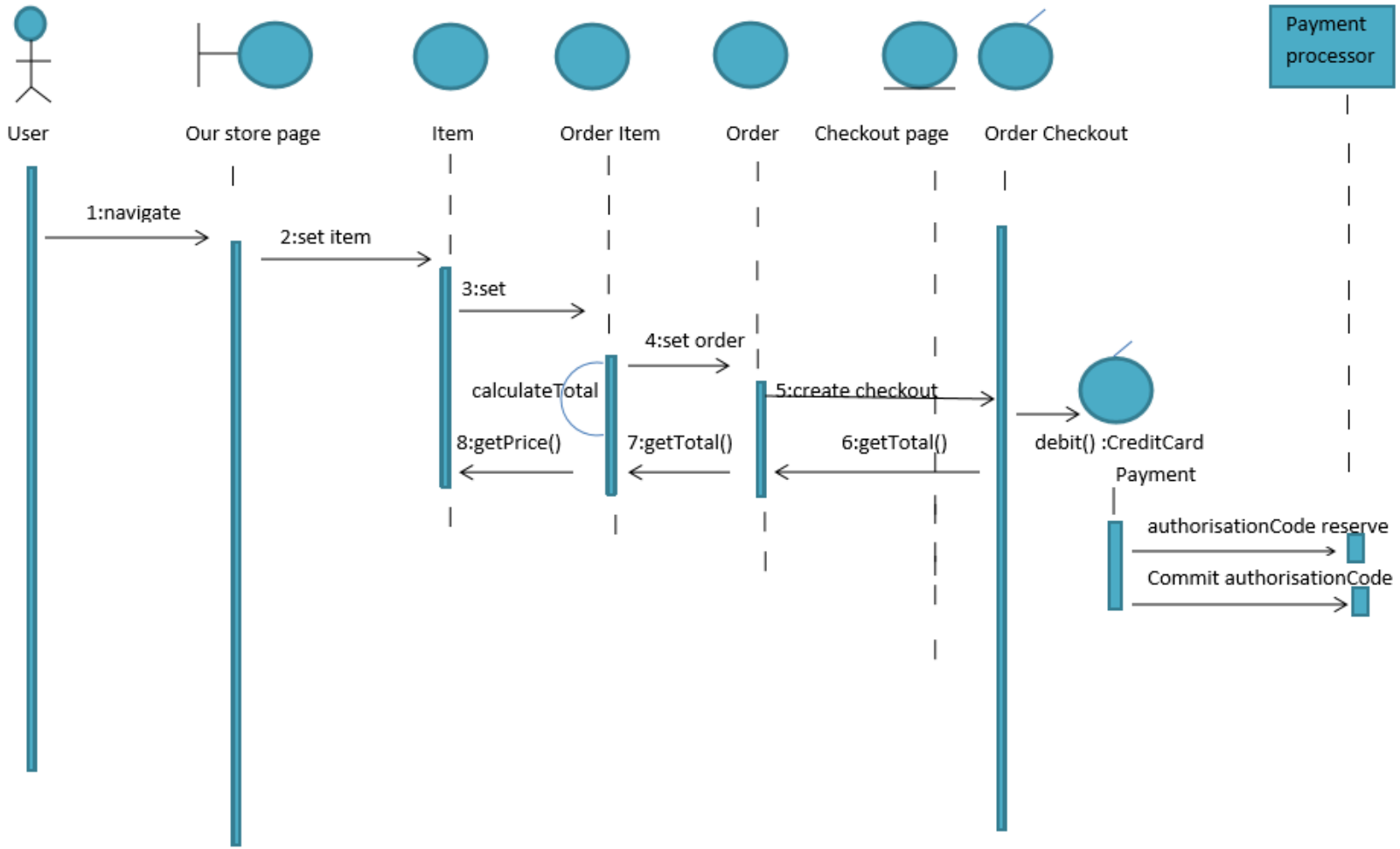
In what will be seen in the below figure, we will describe the main functionalities of our application taking into consideration the rights of the users. This time we have 2 other types of uers, the logged in and the one that are not logged in which will be refered to as visitors.

The logged in users will have the possibilities to place orders which will be managed in the table with the same name and also pay the order using the order details. All the payment details will be stored in the Payments table. This type of users will also have the possibility to search into the list of exhibits using different filter functions. The search results will provide also more information about the flowers which will be returned as a result of applying the filters. All the details for the exhibits will be stored in the table with the same name. For each of the resulted flowers, the user has the possibility to add different comments which are managed in the Comments table.

As a difference, the visitors only have the possibility to preview the comment added by the logged in users. They will not be able to add any opinion. But they will be able to filter the Store list and also the list of exhibits of the

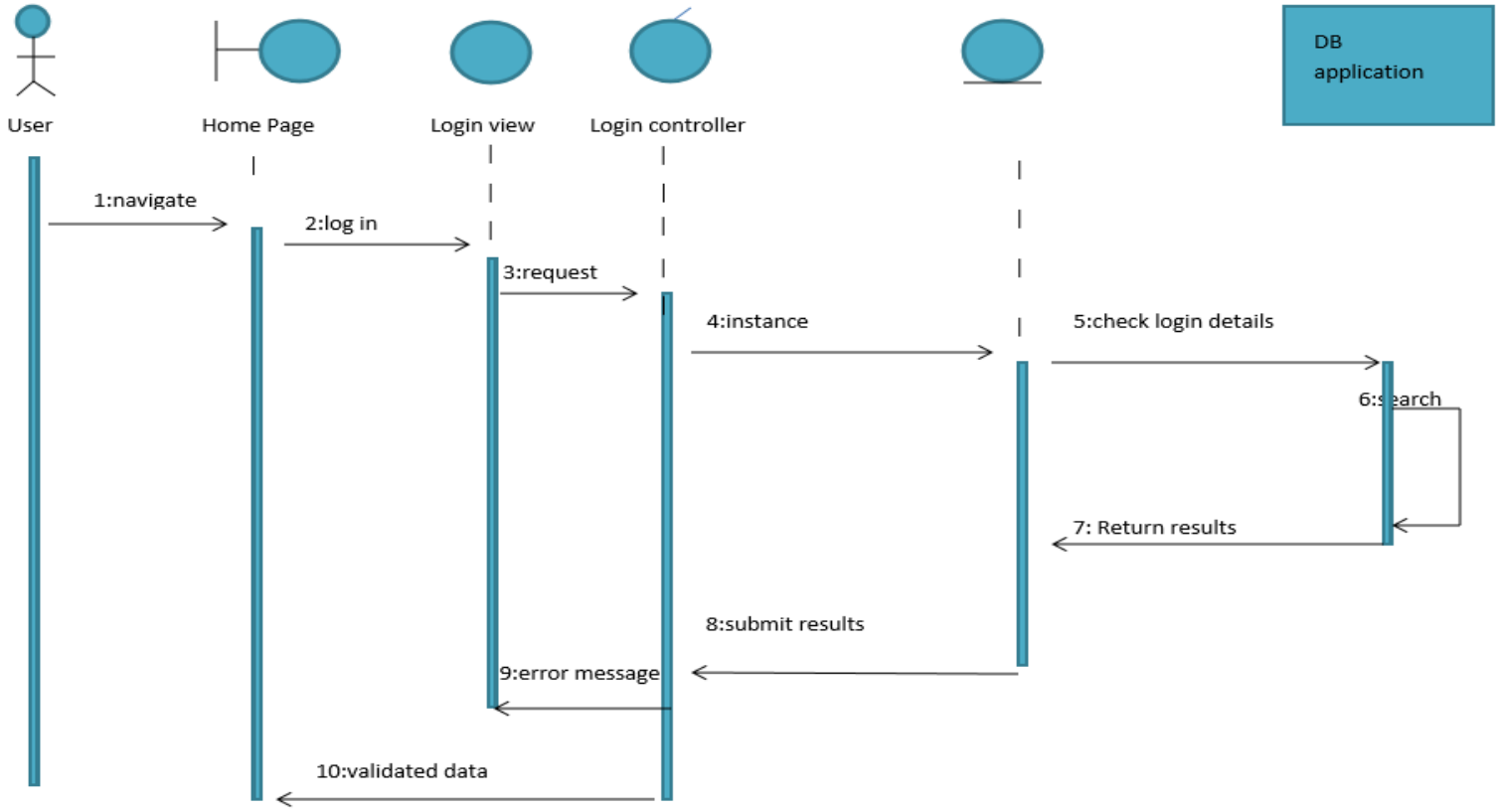


Regarding the data flows and the interactions between the objects, our web application is implemented using the Model-View-Controller (MVC) software design pattern, which is represented through the following sequence diagrams:


User
Our store page
Item
Order Item
Order
Checkout page
Order Checkout
Payment processor
1:navigate
2:set item
3:set
4:set order
calculateTotal
5:create checkout
8:getPrice()
7:getTotal()
6:getTotal()
debit() :CreditCard Payment
authorisationCode reserve
Commit authorisationCode

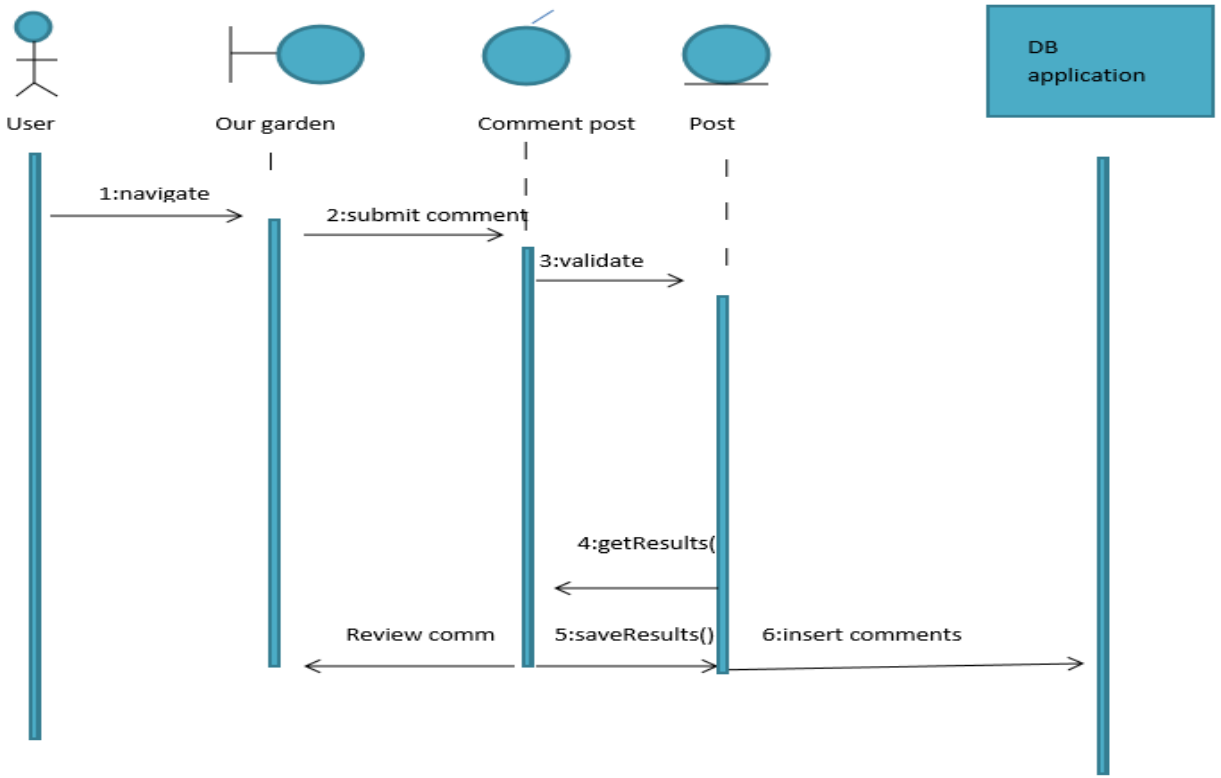
For a better understanding of the diagram we will present the use case on which it is based. In this case, the user while navigating through Our Store navigator and selecting the item, the system is calculating the Total value that should be paid and is creating a check out in order for the payment to be performed. The payment in our case is performed using a third-party transaction.

In what comes next, we are going to present sequence diagram related to the user login.



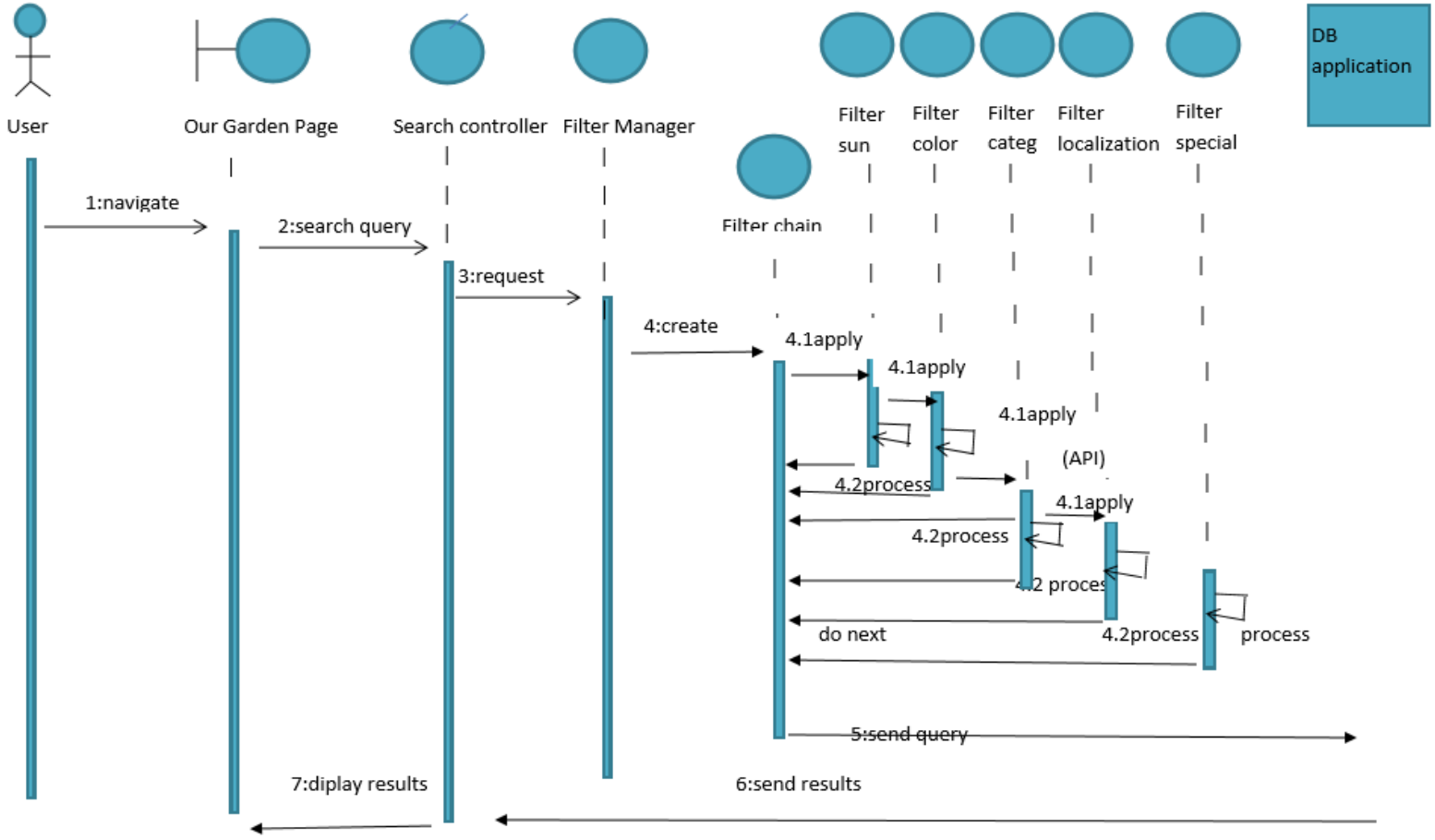
In this case, the user is using the ‘Log in’ function from the Home Page of our app and the request is sent to the system, after the login details are checked and the search in performed in the DB taking into consideration the indexes and is returning the results that can be both as an error message informing the user that the login data is not the correct one and also as a validation message, letting the user perform the login.

Once logged in the user has also the possibility to add any comment at the page for each of the products from our store. This functionality will be presented in the next diagram:



Once the post will be added by the user, the system is the one who validates it and sends the data to the data base in order to be recorded. After validating the comment, it presence will be seen in the Comments list also by the registered users and the visitors of the app.

Moreover, we will attach also the diagram related to the functionality of filtering the data.



Once the user has selected the filters that should be applied, the Controller is taking over the search query and is sending it to the data base through the Filter Manager for each of the filters that our application has. The results are send to the controller and displayed in the View.