Passenger Database Description:

The role of Passengers is to carry information required by the user to take out a payment. Since Passenger is something that executes Payment it has a Many to One Relationship with Passenger and it as well as a Many to Many Relationship with Website where Passengar develops Website. Furthermore, when it comes to the specific type that the Passengar Database is because of its size and relational goals with Website and Passanger, it is NoSQL. As it not only stores: name : String, age : int, email : String, address : String, and accountNum : int openAcc(), closeAcc(), and requestInfo(), but it sends this information to Website and Payment. These variables also would be Public mechanisms because they are foundational to the rest of the program running hence why it must be NoSQL. To be more technical especially since the Passengar is the main user of the application their data is a necessity and needs to be explored in a detailed manner.

Payment Database Description:

The role of Payment is to carry information required by the user to take out a Payment. Since Payment is something done by the Passenger it has a Many to One Relationship with Passenger and specifically is executed by the Passenger. Furthermore, when it comes to the specific type that the Payment Database is because of its size and relational goals, it is SQL. As it only stores: id : int, customId : int, ccv : int, depPayment(), refPayment(). Which are all private variables and methods belonging to Payment, that will not interact with any other database's objects.