# **Class Design For Rent Bike**

#### <<Control>> RentBikeController

+ rentbike: rentBikeInfo

- + rentBike(bikeld: Strin, bikeName: String, deposit: float, remainingTime: float, battery: int, rentBikeDate: date): Map<String,String> + pauseTimer(biketId: String, bikeName: String>: boolean + resumeRent(biketId: String, bikeName: String): void

### **Attribute**

#	Name	Data type	Default value	Description
1	rentBike	rentBikeInfo	NULL	Represent the bike information that user want to rent

## **Operation**

#	Name	Return type	Description
1	rentBike	Map <string,string></string,string>	User chooses the bike that he wants to rent
2	pauseTimer	Boolean	User pause rent bike and then he can return bike
3	resume	Void	After user pause rent bike, he can resume rent

#### Parameter:

bikeName: The name of bike

bikeld: The id of bike

deposit: The deposit of rental bike

remainingTime: The remain time of rental bike

battery: The battery of rental bike rentBikeDate: The order date

### Exception:

PlaceOrderException - if responded with a pre-defined error code

 UnrecognizedException - if responded with an unknown error code or something goes wrong

#### Method:

None

#### State:

None

#### Illustration:

```
stage.setScene(sce
stage.show();

com.example.placerushorder.RentBike
public boolean pauseTimer(String bikeId,
String bikeName)
throws Exception

Choose rush delivery or not

Params: bikeId - The id of bike
bikeName - The name of bike
Returns: Pause time to rent bike

Throws: Exception

**Throws: Exception**

**Throws:
```

```
Scene scene = nev
                             public Bike rentBike(String bikeId,
    stage.setTitle("H
    stage.setScene(sc
                                                  float remainingTime,
    stage.show();
                                                  int battery,
                                                  Date rentBikeDate)
                             throws Exception
                             Place order in cart and then create order
ublic static void ma
    RentBike rBike =
                             Params: bikeId – The id of rental bike
                                     bikeName - The name of rental bike
    rBike.rentBike()
                                     deposit – The deposit of rental bike
    rBike.pauseTimer(
                                     remaining Time - The remain time of rental bike
                                     battery – The battery of rental bike
                             Returns: Bike The rental bike
```