**ER Design**

**Chosen system:**

**Public bicycle rental points in your country**

**FUNCTIONS:**

* The customer will be able to rent a bicycle using his SmartCard with a regular cost.
* The customer will be able to rent only one bicycle at a time.
* Customer has to dock the bicycle in a station before the trip is longer than 30 minutes.
* If the trip is longer than 30 minutes, the customer will pay a fee.
* Maintenance Employees will be able to take multiple bicycles for maintenance or relocation.
* The customer service team will provide telephonic support to customers for any issues with the service.

**USE CASES:**

1. Rent Bicycle (Customer) - Successful:
   * Actors: Customer
   * Description: This use case represents a successful scenario where a customer rents a bicycle using their SmartCard. The customer selects a bicycle, confirms the rental, and the system records the start time of the rental.
2. Rent Bicycle (Customer) - Unsuccessful:
   * Actors: Customer
   * Description: This use case covers scenarios when a customer's attempt to rent a bicycle using their SmartCard is unsuccessful. Possible issues could include an invalid SmartCard, insufficient balance, or technical problems. The system provides appropriate error messages and guidance to the customer.
3. Return Bicycle (Customer) - Successful:
   * Actors: Customer
   * Description: This use case represents a successful scenario where a customer returns a rented bicycle. The system calculates the rental duration, and if the trip is within 30 minutes, no fee is applied. The bicycle is made available for other customers.
4. Return Bicycle (Customer) - Late Return:
   * Actors: Customer
   * Description: In the case of a late return, the customer returns a bicycle after a trip longer than 30 minutes. The system calculates a fee based on the duration exceeding 30 minutes and informs the customer about the additional charge.
5. Rent Bicycle (Employee) - Successful:
   * Actors: Maintenance Employee
   * Description: This use case represents a successful scenario where a maintenance employee rents multiple bicycles for maintenance or relocation purposes. The employee selects the bicycles needed, records the reason for the rental, and the system records the start time of the rental.
6. Rent Bicycle (Employee) - Unsuccessful:
   * Actors: Maintenance Employee
   * Description: This use case covers situations where a maintenance employee's attempt to rent multiple bicycles is unsuccessful. Potential issues include a lack of available bicycles, system unavailability, or other technical problems. The system provides information regarding the reason for the unsuccessful operation.
7. Telephonic Support Request (Customer):
   * Actors: Customer, Customer Service Team
   * Description: This use case allows a customer to request telephonic support from the customer service team for any issues with the service. The customer service team assists the customer by providing guidance and solutions to resolve the problem.

**ENTITIES:**

**Customers:**

Attributes: Customer ID, FirstName, LastName, phoneNumber, SmartCard ID

**Bicycles:**

Attributes: Bicycle ID, Status (e.g., available, rented, in maintenance), Location, SmartCardID

**Rentals:**

Attributes: Rental ID, Customer ID, Bicycle ID, Start Time, End Time, Duration, Fee

**Stations:**

Attributes: Station ID, Location, Capacity

**Bicycles-Stations:**

Attributes: Bicycles-StationsID**,** Bicycle ID, Station ID, Date

**Employees:**

Attributes: Employee ID, FirstName, LastName, Role, phoneNumber

**Maintenance Log:**

Attributes: Log ID, Bicycle ID, Employee ID, Maintenance Type, Date, Description

**Transactions:**

Attributes: Transaction ID, Customer ID, Amount, Date

**Customer Support Requests:**

Attributes: Request ID, Customer ID, Date, Description, Status

**SmartCards:**

Attributes: SmartCard ID, Customer ID, Balance, Activation Date, Expiry Date

**RELATIONSHIPS:**

1. **Customers** to **SmartCards**:
   * The "Customer ID" attribute in the "Customers" entity is related to the "Customer ID" attribute in the "SmartCards" entity. This establishes a one-to-one relationship where each customer can be associated with a SmartCard.
2. **Customers** to **Rentals**:
   * The "Customer ID" attribute in the "Rentals" entity is related to the "Customer ID" attribute in the "Customers" entity. This establishes a one-to-many relationship, indicating that a customer can have multiple rentals.
3. **Bicycles** to **Rentals**:
   * The "Bicycle ID" attribute in the "Rentals" entity is related to the "Bicycle ID" attribute in the "Bicycles" entity. This establishes a one-to-many relationship, indicating that a bicycle can be associated with multiple rentals.
4. **Bicycles** to **Stations**:
   * The "Location" attribute in the "Bicycles" entity is related to the "Station ID" attribute in the "Stations" entity. This indicates that the location of a bicycle is associated with a specific station. It's a one-to-one relationship where each bicycle is at one station.
5. **Bicycles-Stations** to **Bicycles** and **Stations**:
   * The "Bicycle ID" and "Station ID" attributes in the "Bicycles-Stations" entity are related to the corresponding attributes in the "Bicycles" and "Stations" entities. This table is used to represent the many-to-many relationship between bicycles and stations, indicating which bicycles are located at which stations on specific dates.
6. **Employees** to **Maintenance Log**:
   * The "Employee ID" attribute in the "Maintenance Log" entity is related to the "Employee ID" attribute in the "Employees" entity. This establishes a one-to-many relationship, indicating that an employee can be associated with multiple maintenance log entries.
7. **Customers** to **Transactions**:
   * The "Customer ID" attribute in the "Transactions" entity is related to the "Customer ID" attribute in the "Customers" entity. This establishes a one-to-many relationship, indicating that a customer can have multiple transactions.
8. **Customers** to **Customer Support Requests**:
   * The "Customer ID" attribute in the "Customer Support Requests" entity is related to the "Customer ID" attribute in the "Customers" entity. This establishes a one-to-many relationship, indicating that a customer can create multiple support requests.

These relationships allow you to represent how the entities in your system are connected

**ER DIAGRAM:**

