Nick Forleo

SWENG 837

1. **System Operation Contract (SOC)**

**Name:** sendCustomerData({name: string, address: string, email: string,   
 phoneNumber: string, licenseNumber: int, idNumber: int, date: datetime})

**Responsibilities:** Enter and record personal customer information necessary to create a   
 reservation (whole parameter object to be referenced as *customerData* in   
 the SOC to reduce confusion and *data* in the diagrams to save   
 space)

**Type:** System

**Cross References:** U.C.1, U.C.6

**Exceptions:** Assume this is the first reservation for a new customer. If customerData  
 exists then CustomerDetails is updated with the delta of what is passed   
 in. Reports error if entered customerData not validated

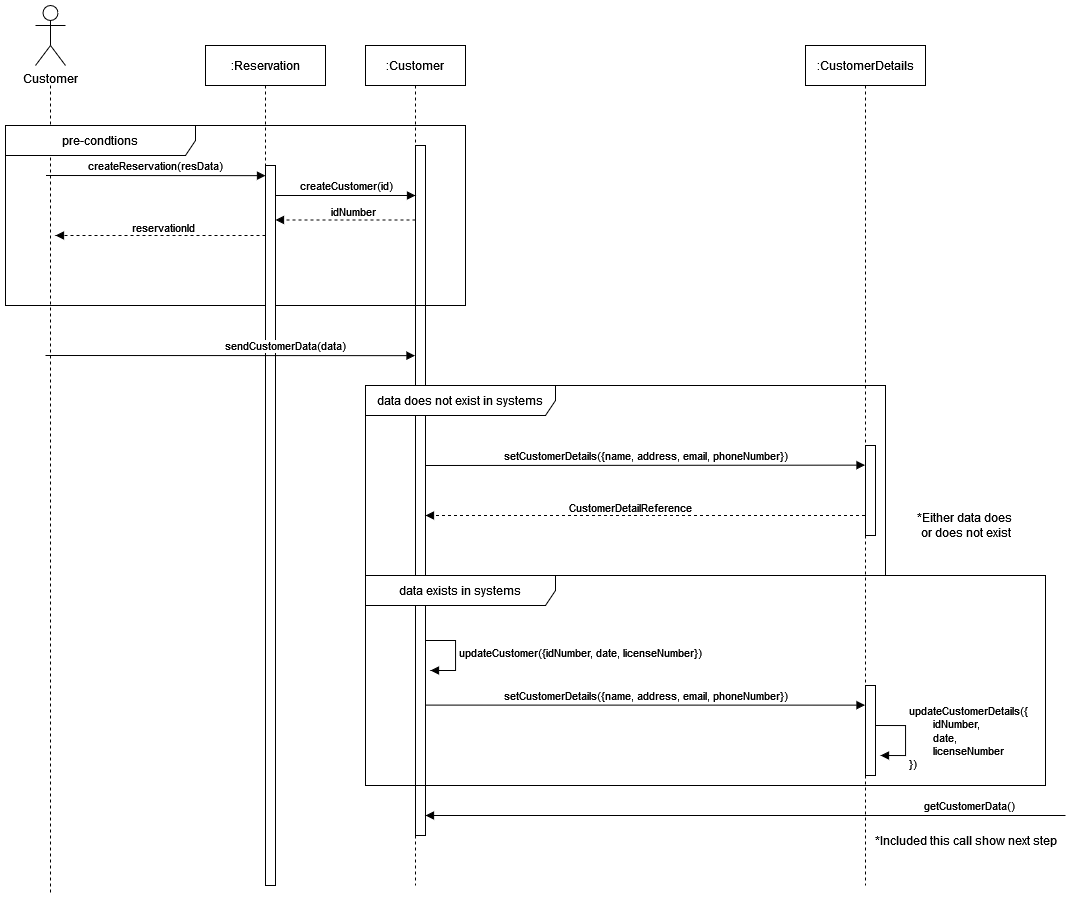
**Output:** N/A - doesn’t return anything

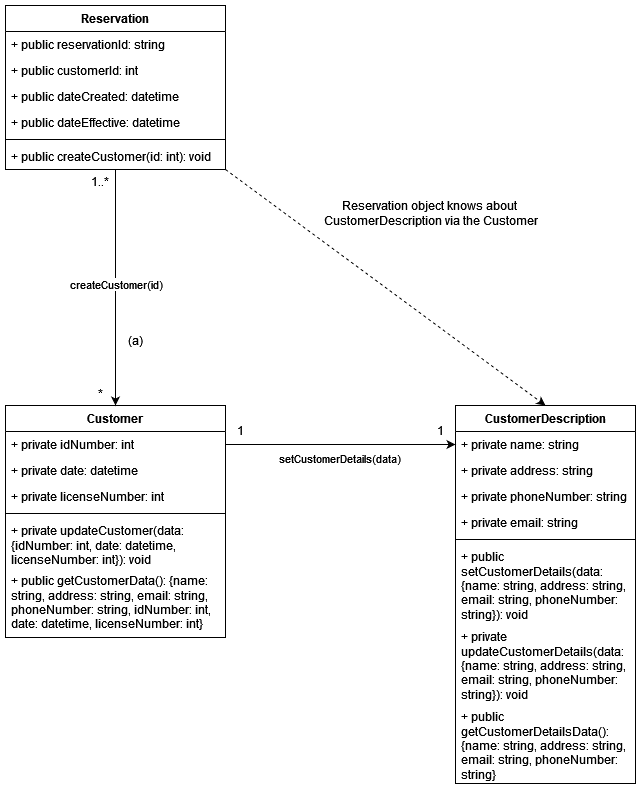
**Pre-conditions:**

* Customer has accessed the system
* Reservation instance exists
* name, string, address, email, and phone number have been provided by the user

**Post-Conditions:**

* If new customerData, Customer attribute and CustomerDetail created (instance creation)
* If customerData is not new, then customerData is associated to the existing Customer attributes and CustomerDetail (association created)
* Customer and CustomerDetails attributes are equal to customerData (attribute modification)

1. **Sequence Diagram (SD)**
2. **Design Class Diagram (DCD)**

****

**Rationale:**

(a) The Domain Model shows navigability Customer → Reservation because logically “The customer requests a reservation. However, programmatically and through the class structure, the reservation object will create the customer. Therefore, I represented it as Reservation → Customer in the DCD.

Indirection is used with the createCustomer() function between Reservation and Customer. A Reservation does not see inside Customer, but rather relies on the update function to create a new Customer. This is meant to reduce coupling between the classes

The following domain classes and domain associations were eliminated:

* CustomerCareRep
* Employee
* Lawyer
* Contract
* ContractDetails
* Vehicle
* VehicleDescription
* CreditReport
* CreditCompany
* InsurancePolicy
* InsuranceCompany
* Invoice
* PaymentCompany

They were removed because they are not relevant for the completion of the SOC/SD.

**4. Partially-Dressed UC**

**Use Case Name:** Create reservation for rental car

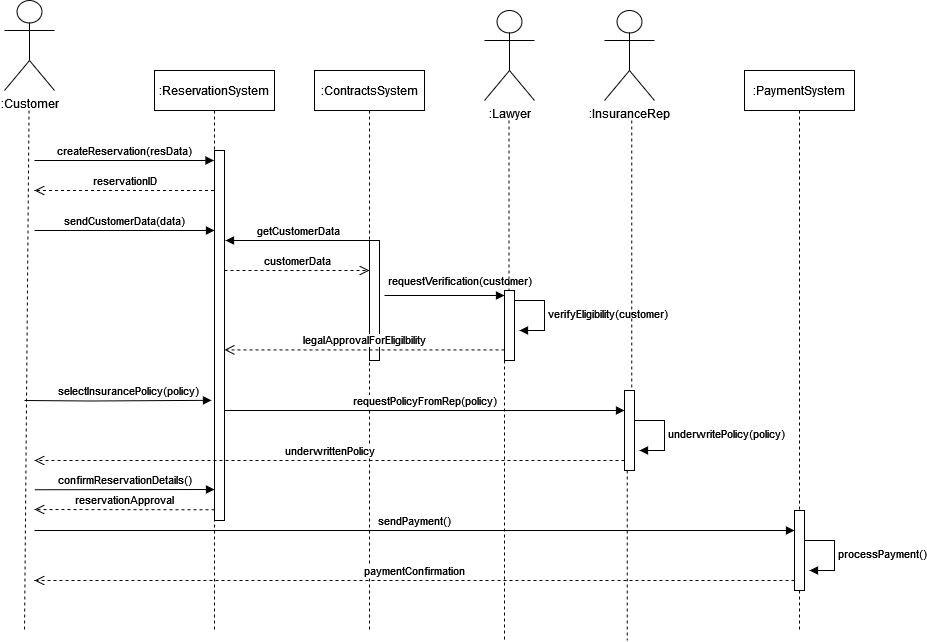
**Goal in Context:** A customer can create a reservation for a rental car using the system.

**Primary Actors:** Customer

**Scenario:**

1. Customer requests a reservation for a rental car and provides data relevant to the reservation.
2. Customer enters necessary personal information.
3. Legal verifies the customer is eligible to rent a car.
4. Customer selects relevant insurance policies.
5. Insurance representative underwrites the customer.
6. Customer confirms reservations and the car rental reservation system approves the reservation.
7. Payment vendor collects money once the reservation is complete.

**5. System Sequence Diagram (SSD)**



**6. Domain Model**