

Requirement Collections

- R1: While Ordering the item(s), the customer can choose the nearest location to pick up the order package from the locker.
- R2: One or more items can be contained in one order. An order will be placed in a package before the delivery.
- R3: There can be different size of lockers including extra small, small, medium, large, extra large, & double extra large.
- R4: The locker is assigned to the customer based on the size of the order package.
- R5: When the order package is delivered to the locker location specified by customer, a 6-digit code will be sent to the customer to open the locker.
- R6: The package will be kept or placed inside the locker for three days only.
- R7: If the customer does not pick up the package from their locker within three days, the refund process will be initiated & the customer won't be allowed to pickup the package any longer.

- R8: Only eligible packages can be placed in the locker such that the size of the package must be less than the size of the locker.
- R9: There can be multiple locker at every locker location
- R10: The amazon locker is accessed within a specific time. Every location has its opening & closing time. Therefore, the customer should pick up the package during this time period.
- R11: The item can be returned to the Amazon locker if it doesn't match the expectation of the customer or is faulty, & there is refund policy available for the product.
- R12: To return an item, the customer needs to choose the nearest locker location. The available locker will be assigned to them based on the size & location
- R13: When the customer picks up the order package from the locker, the locker state is changed to closed, & the customer will no longer be able to open the locker with the given code.

System → Our system "Amazon locker"

Primary Actors →

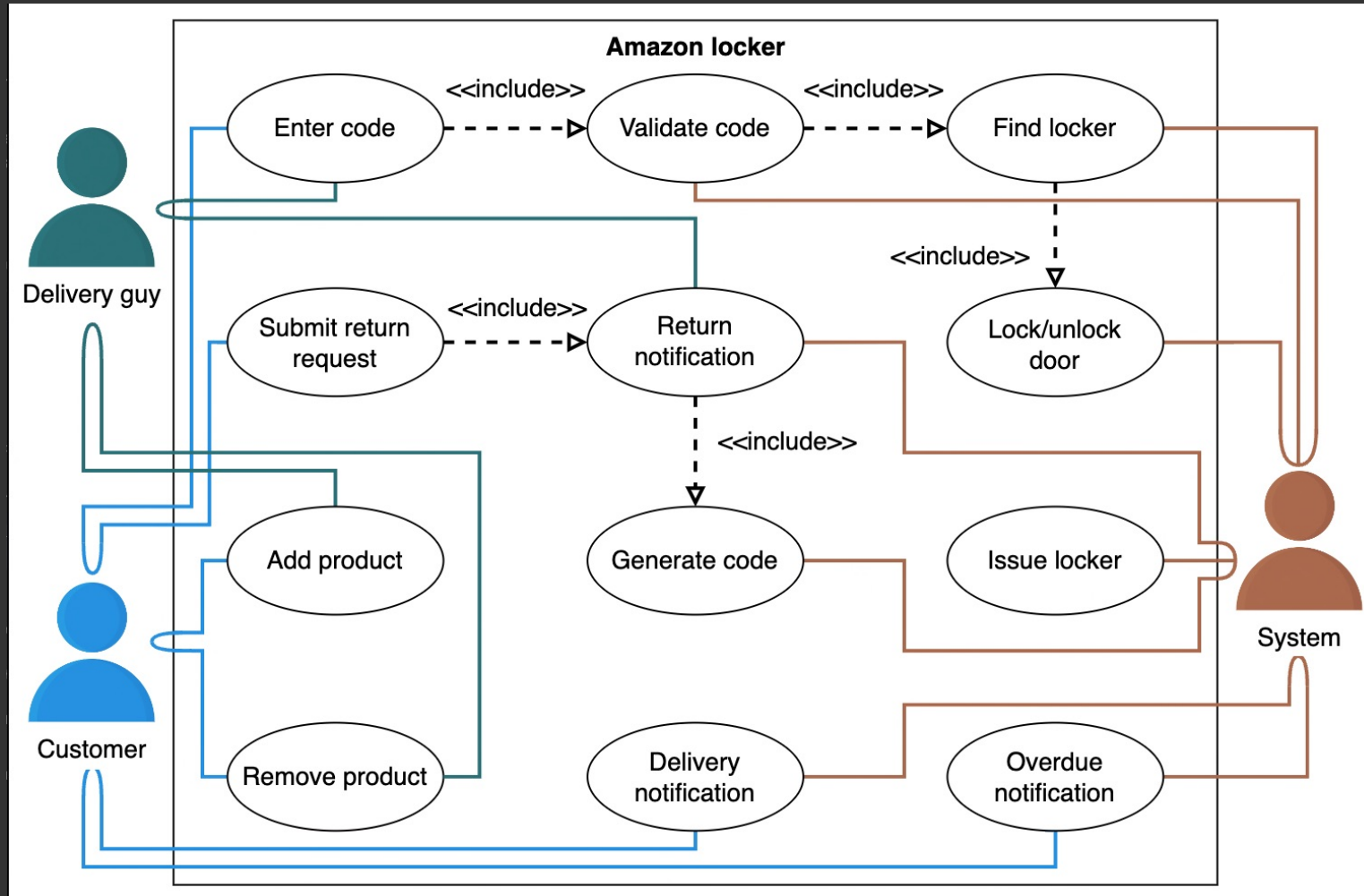
- Customer
- Delivery guy

Secondary actors →

- System

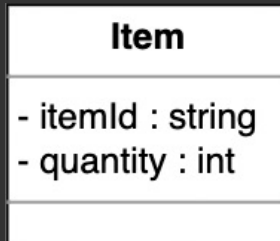
Customer	Delivery Guy	System
Enter code	Enter code	Validate code
Add product	Add product	Find locker
Remove product	Remove product	Lock/unlock door
Delivery notification	Return notification	Return notification
Submit return request		Generate code
Overdue notification		Issue locker
		Overdue notification
		Delivery notification

Use Case Diagram

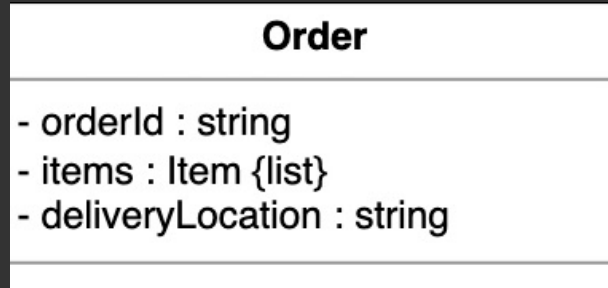


Class Diagram of Amazon Locker Services

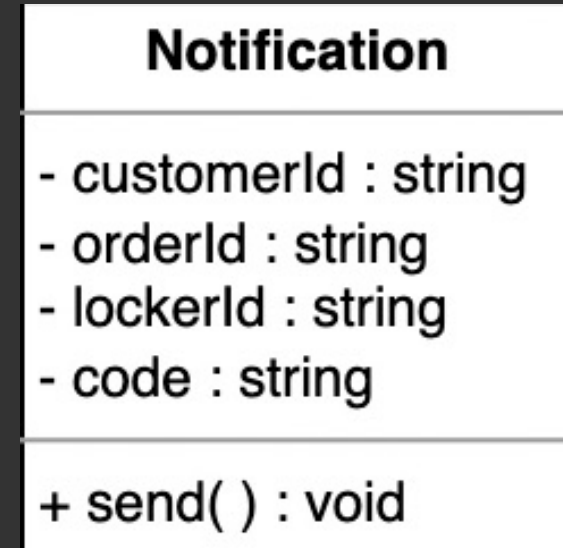
Item



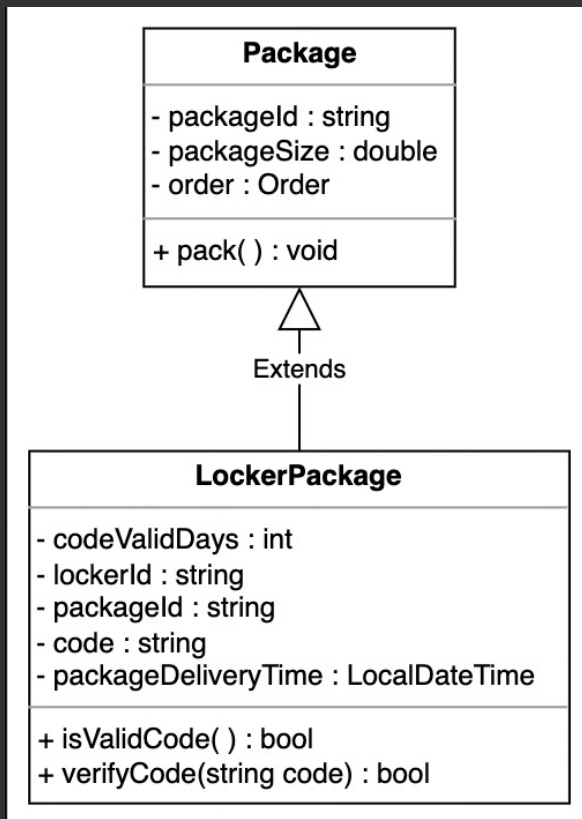
Order



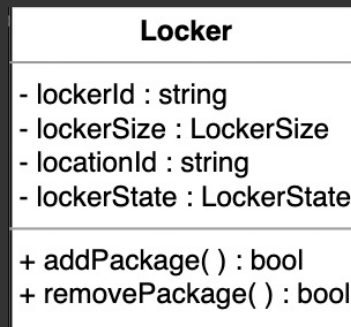
Notification



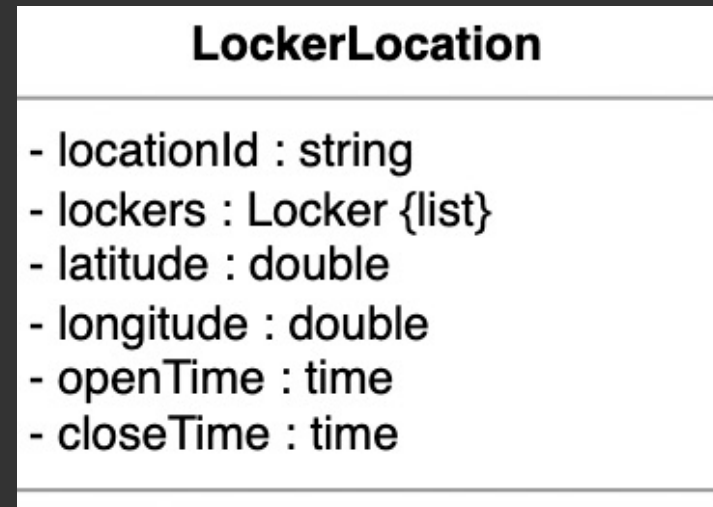
Package



Locker



Locker Location



Locker Service

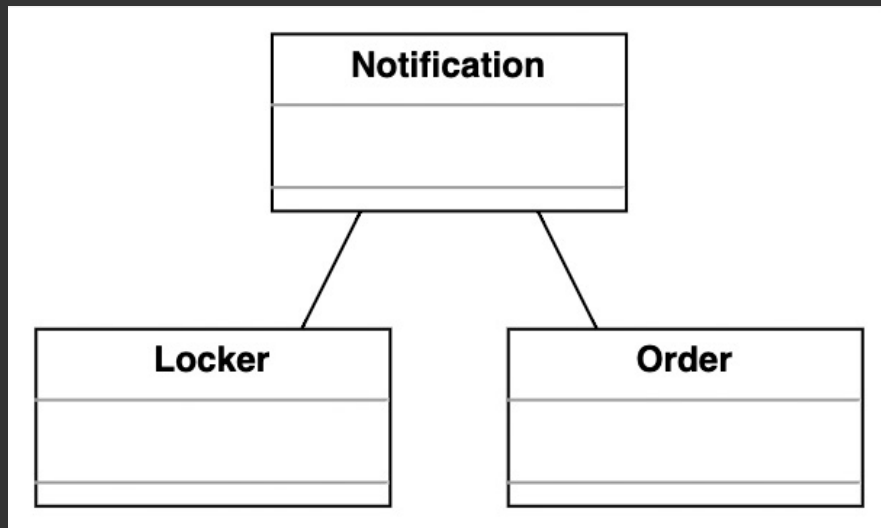
LockerService
- locations : LockerLocation {list}

Enumerations

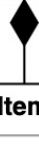
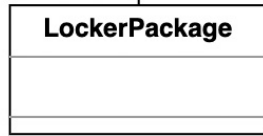
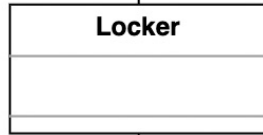
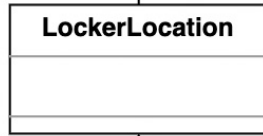
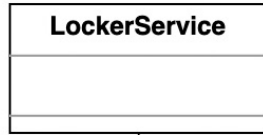
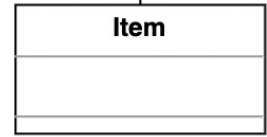
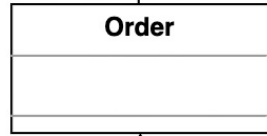
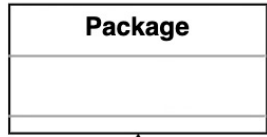
LockerSize
ExtraSmall
Small
Medium
Large
ExtraLarge
DoubleExtraLarge

LockerState
Closed
Booked
Available

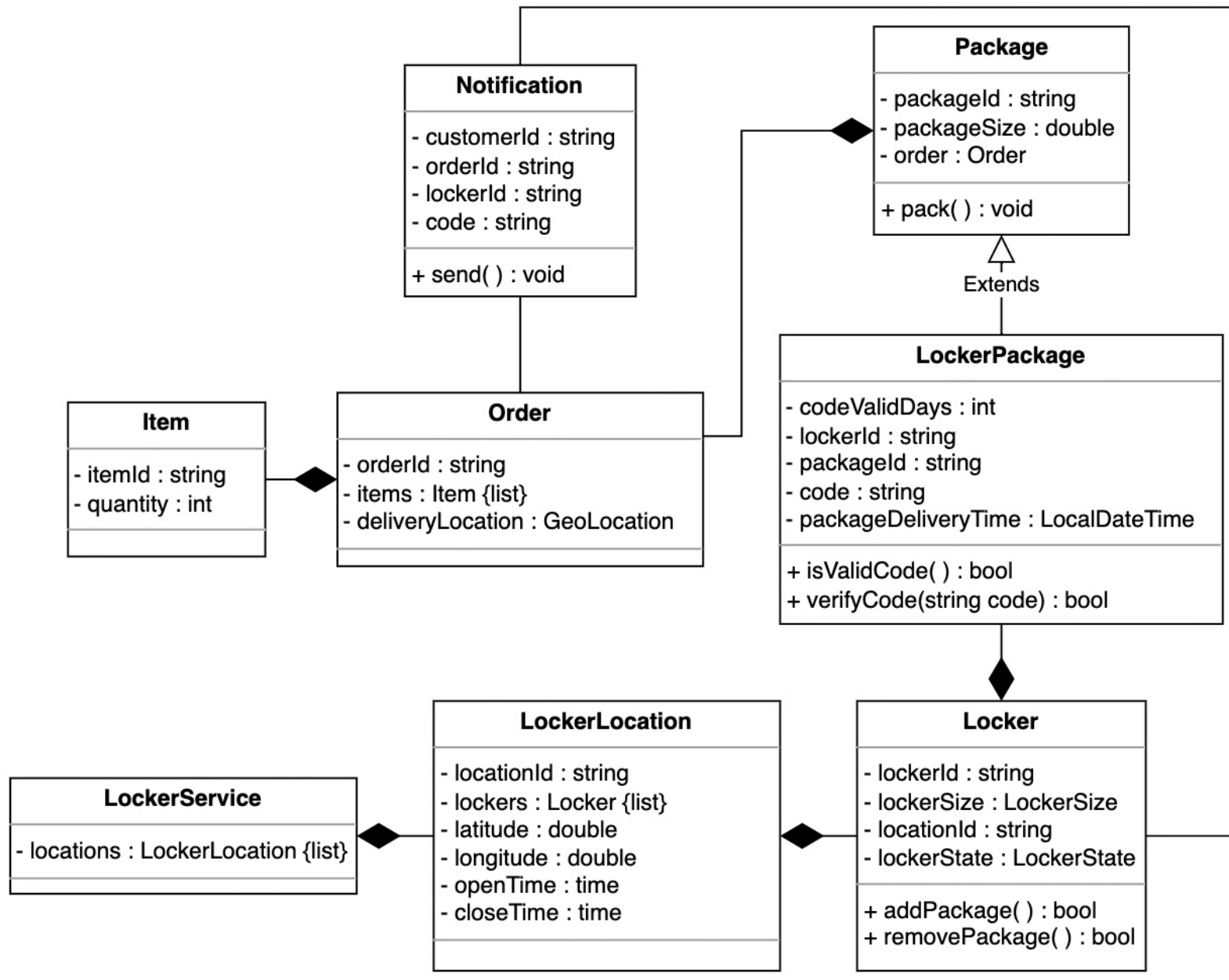
Relationship between the classes



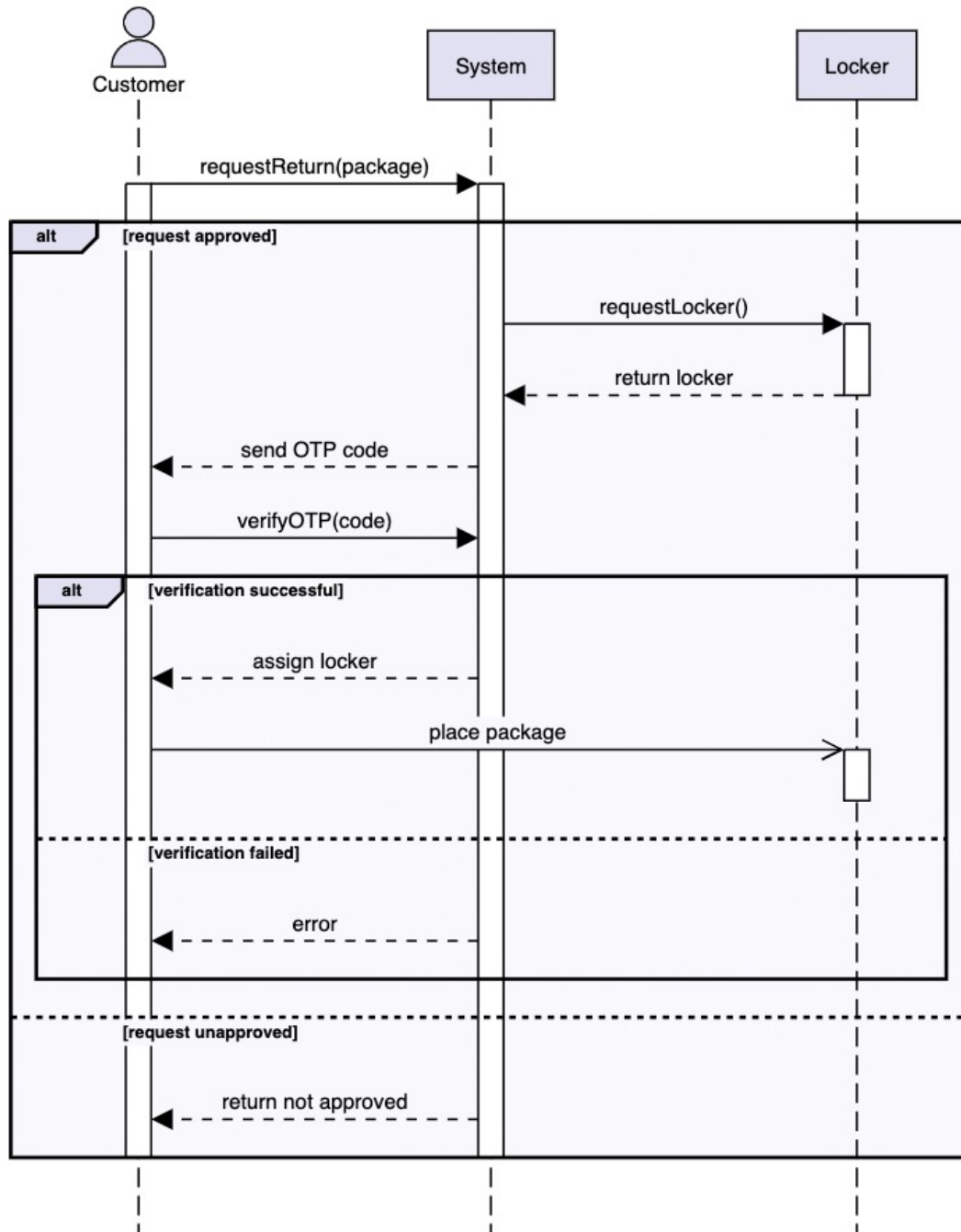
Composition



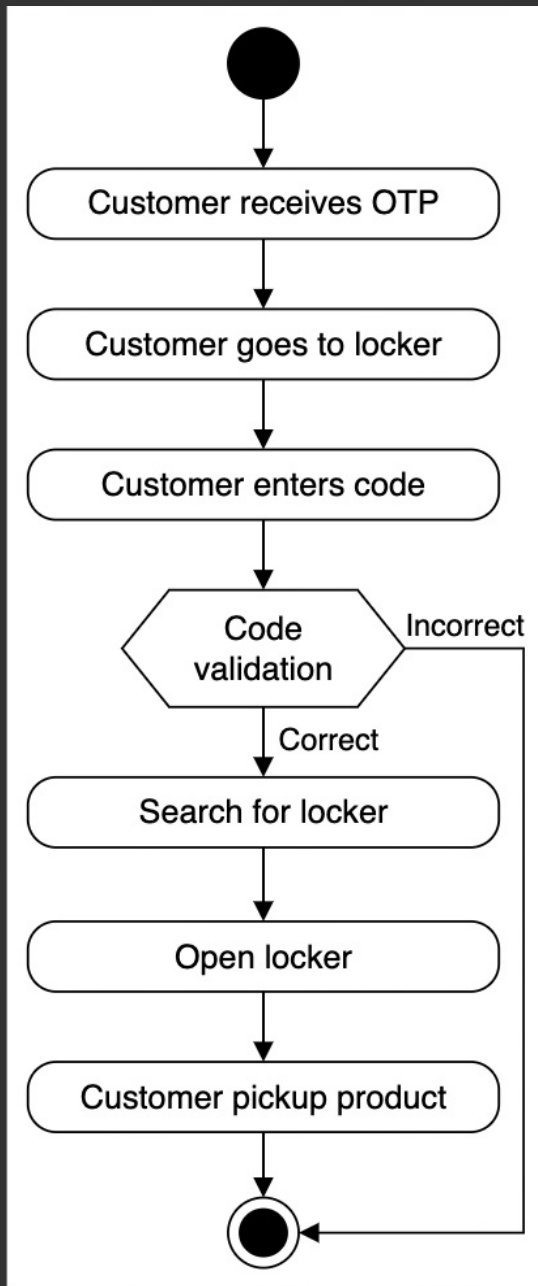
class Diagram



sd return package



✓ Sequence diagram of Return Package



Activity Diagram for product pickup from the locker.

Code for Amazon Locker Service

Enumerations

```
// definition of enumerations used in the Amazon Locker service
public enum LockerSize {
    EXTRA_SMALL,
    SMALL,
    MEDIUM,
    LARGE,
    EXTRA_LARGE,
    DOUBLE_EXTRA_LARGE
}
public enum LockerState {
    CLOSED,
    BOOKED,
    AVAILABLE
}
```

Package & LockerPackage

```
public class Package {
    private String packageId;
    private double packageSize;
    private Order order;

    public void pack();
}

public class LockerPackage extends Package {
    private int codeValidDays;
    private String lockerId;
    private String packageId;
    private String code;
    private Date packageDeliveryTime;

    public boolean isValidCode();
    public boolean verifyCode(String code);
}
```

Item & Order

```
public class Item {
    private String itemId;
    private int quantity;
}

public class Order {
    private String orderId;
    private List<Item> items;
    private String deliveryLocation;
}
```

Locker & LockerLocation

```
public class Locker {
    private String lockerId;
    private LockerSize lockerSize;
    private String locationId;
    private LockerState lockerState;

    public boolean addPackage();
    public boolean removePackage();
}

public class LockerLocation {
    private String name;
    private List<Locker> lockers;
    private double longitude;
    private double latitude;
    private Date openTime;
    private Date closeTime;
}
```

Locker Service & Notification

```
public class LockerService {
    private List<LockerLocation> locations;

    // The LockerService is a Singleton class that ensures it will have only one active instance at a time
    private static LockerService lockerService = null;

    // Created a static method to access the Singleton instance of LockerService class
    public static LockerService getInstance() {
        if (lockerService == null) {
            lockerService = new LockerService();
        }
        return lockerService;
    }
}

public class Notification {
    private String customerId;
    private String orderId;
    private String lockerId;
    private String code;

    public void send();
}
```