# DB2 Documentation

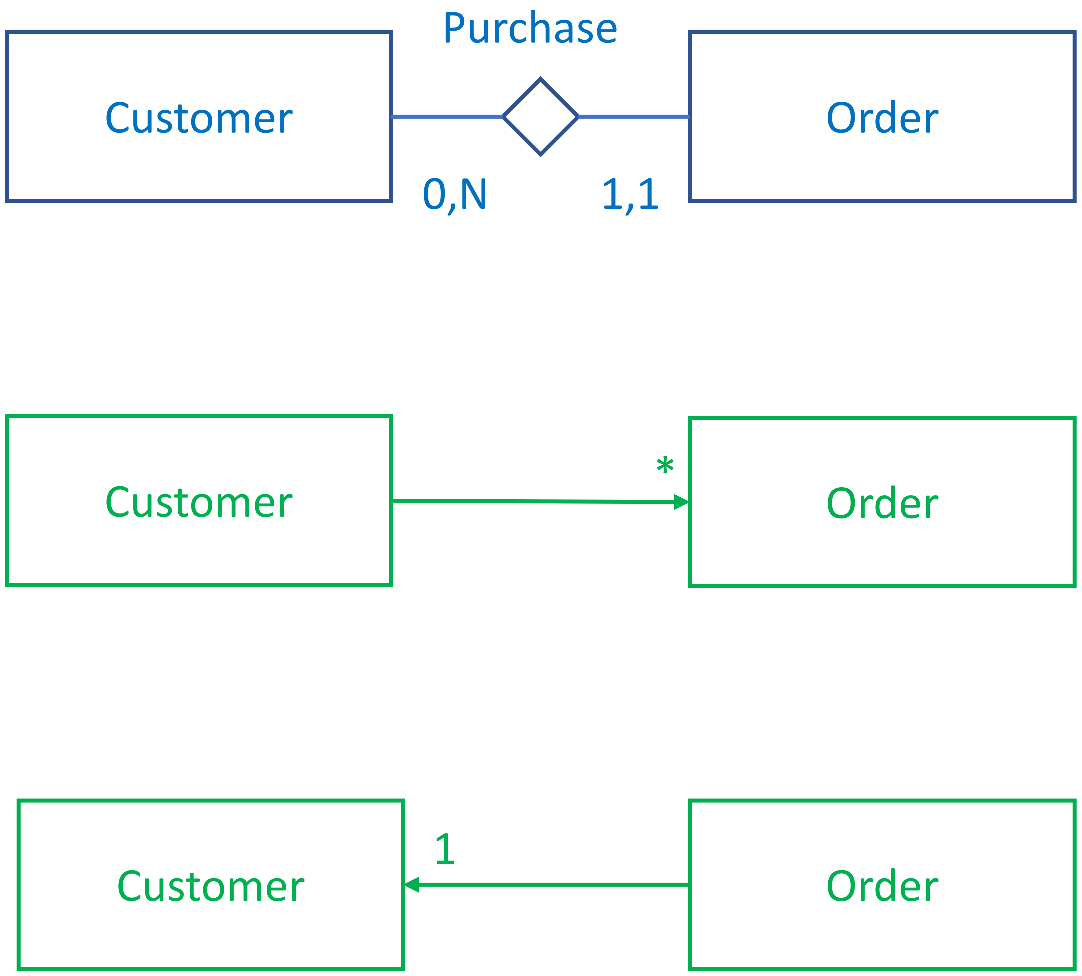
# ER

# Logical Data Model

# Trigger Design and Code

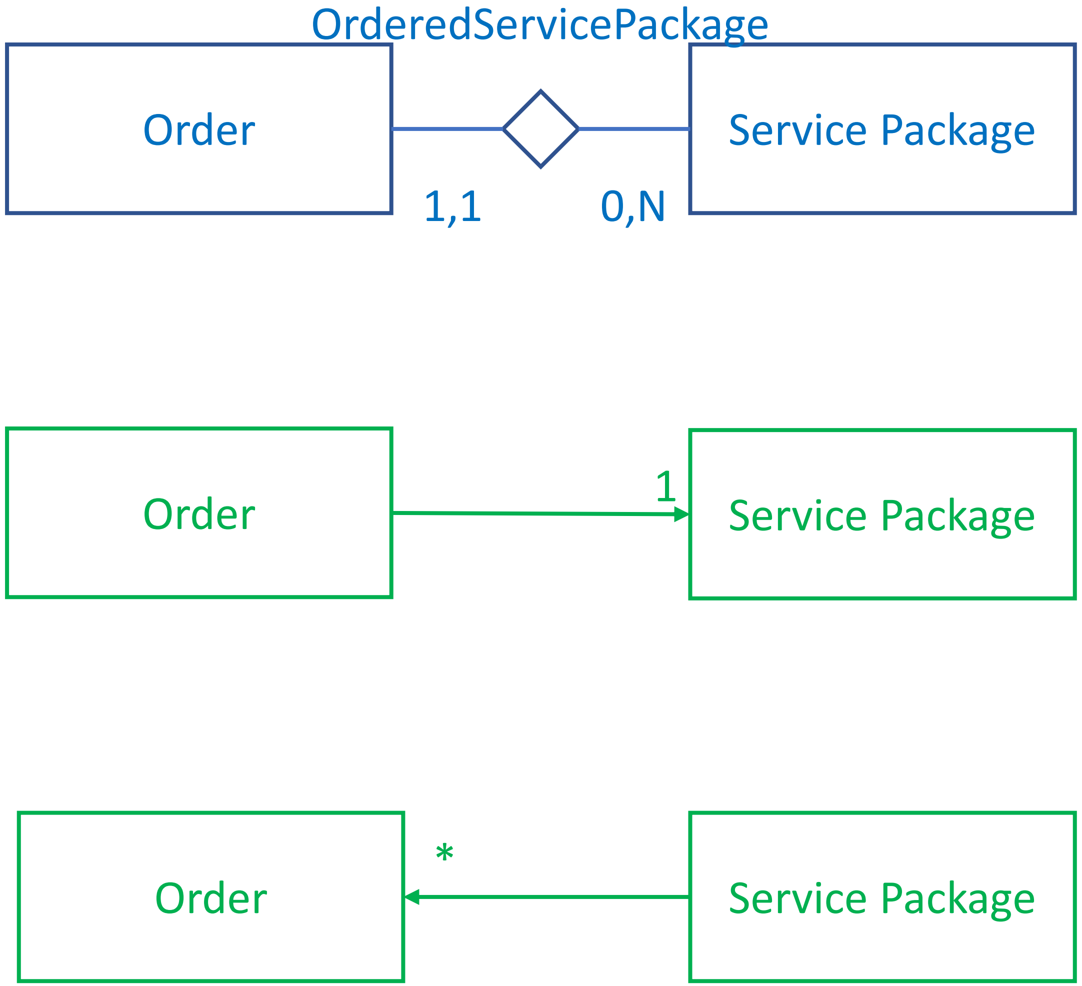
# ORM Relationship Design

# Relationship «Purchase»



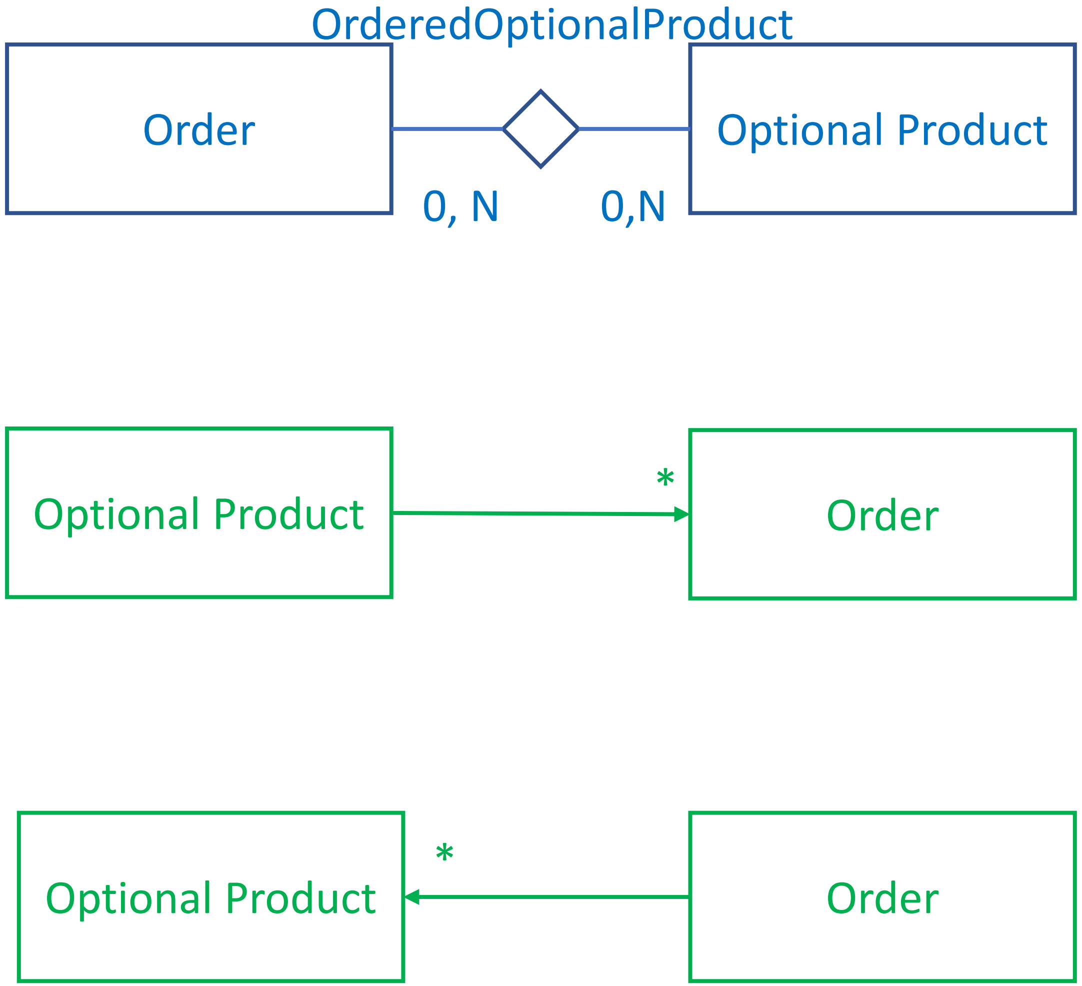
- Customer → Order @OneToMany is necessary to show a list of the rejected orders
- Order → Customer @ManyToOne is not necessary, because there’s no scenario where from the Order we have to retrieve the Customer
- Unidirectional 1:N → Use JPQL queries to retrieve the Customer’s rejected orders

# Relationship «OrderedServicePackage»



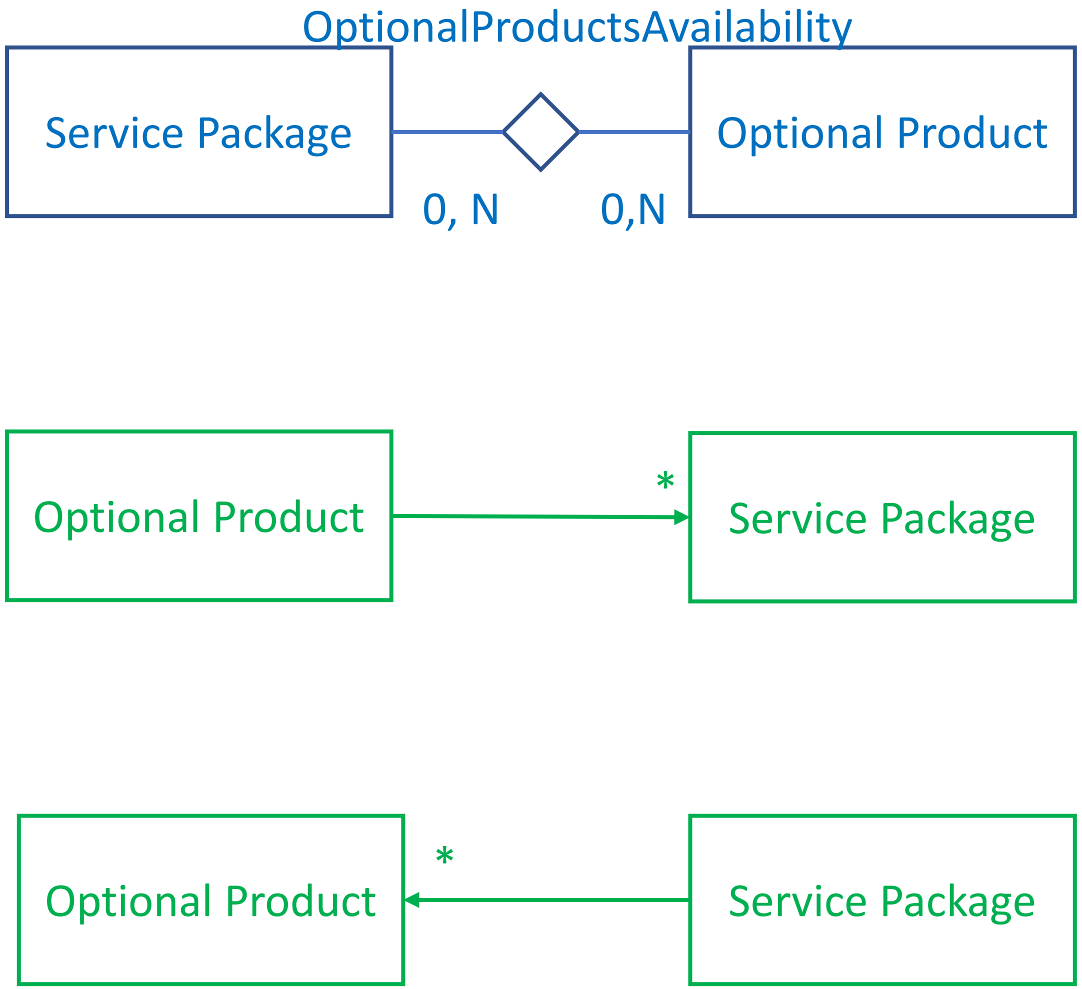
- Order → Service Package @ManyToOne is necessary to show the Customer's selected Service Package in the rejected Order
- Service Package → Order @OneToMany is not necessary, because there's no scenario where from the Service Package we have to retrieve the Order
- Unidirectional N:1 → @ManyToOne in Order entity
- FetchType.EAGER
- No Cascading

# Relationship «OrderedOptionalProduct»



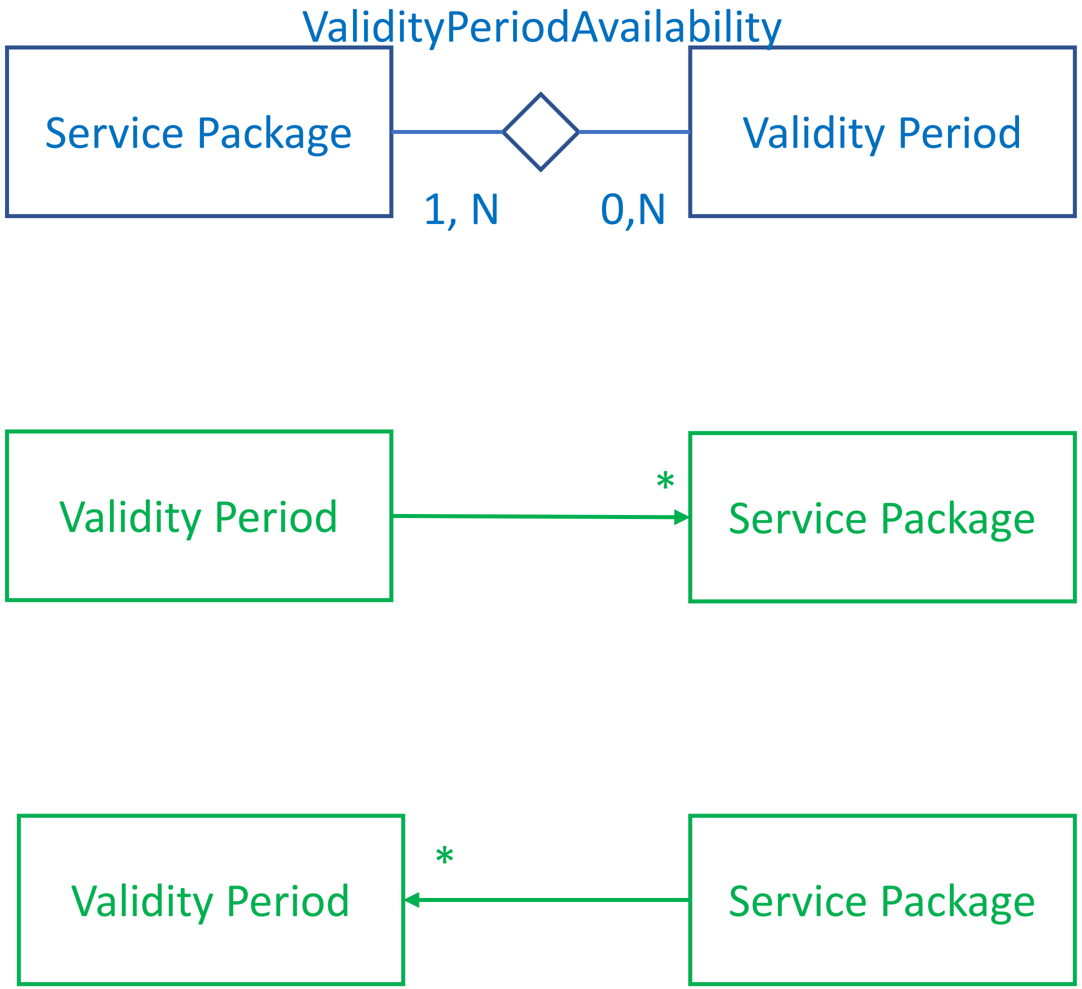
- Order → Optional Product @ManyToMany is necessary to show the Customer's selected Optional Products in the rejected Order
- Service Package → Order @ManyToMany is not necessary, because there's no scenario where from the Optional Products we have to retrieve the Order
- Unidirectional N:M → @ManyToMany in Order entity
- FetchType.EAGER
- No Cascading

# Relationship «OptionalProductsAvailability»



- Service Package → Optional Product @ManyToMany is necessary to show to the Customer the available Optional Products for a certain Service Package
- Optional Product → Service Package @ManyToMany is not necessary, because there's no scenario where from the Optional Products we have to retrieve the Service Package
- Unidirectional N:M → @ManyToMany in Service Package entity
- FetchType.EAGER
- No Cascading

# Relationship «ValidityPeriodAvailability»



ValidityPeriodAvailability

Service Package

1, N

0,N

Validity Period

Validity Period

*

Service Package

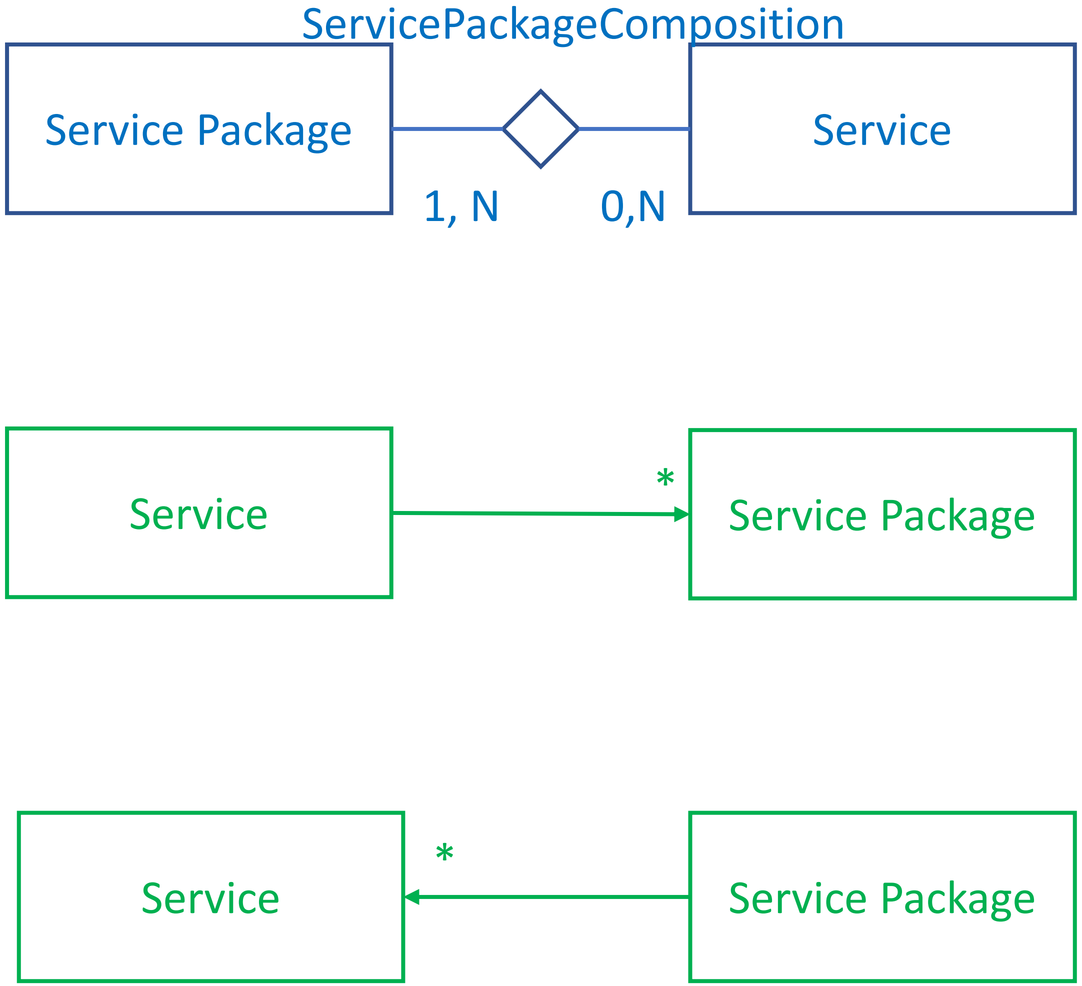
Validity Period

*

Service Package

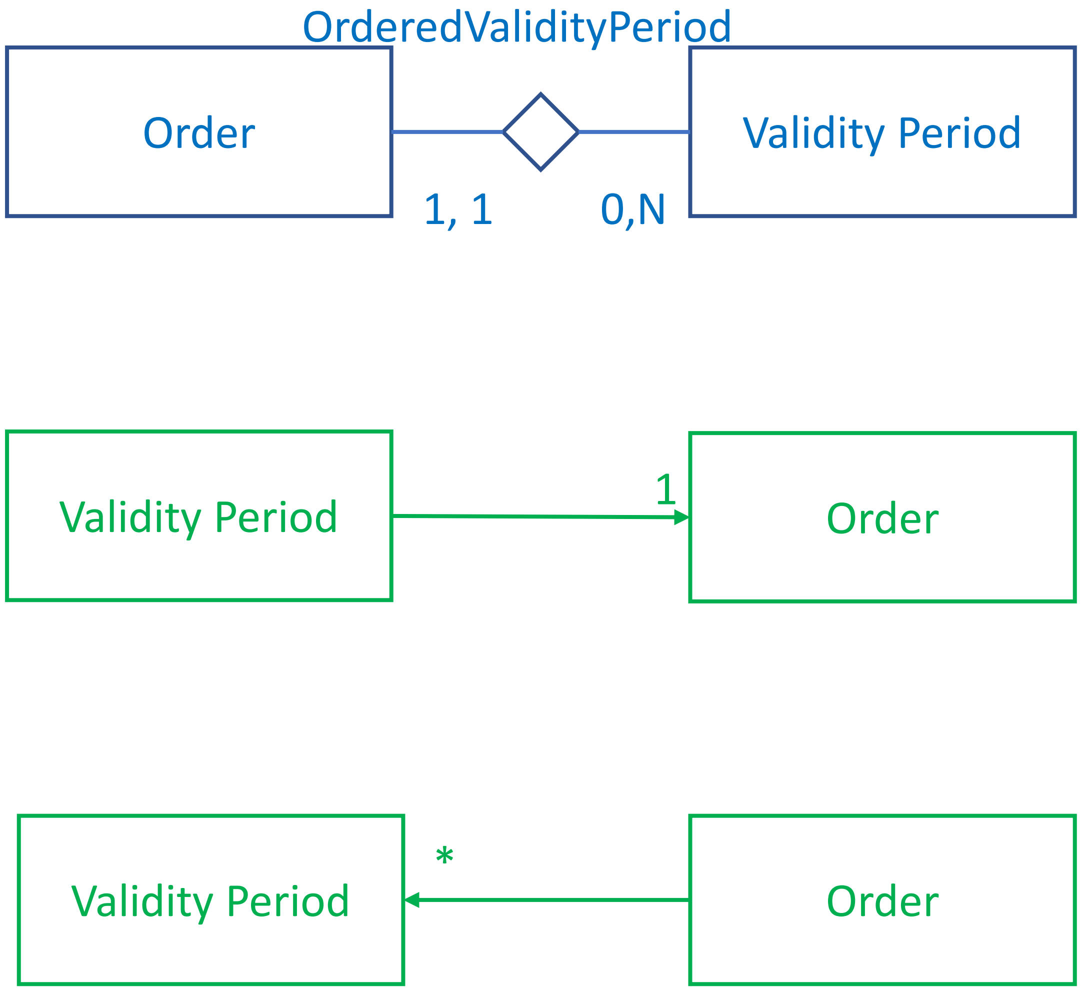
- Service Package → Validity Period @ManyToMany is necessary to show to the Customer the available Validity Period for a certain Service Package
- Validity Period → Service Package @ManyToMany is not necessary, because there's no scenario where from the Validity Period we have to retrieve the Service Package
- Unidirectional N:M → @ManyToMany in Service Package entity
- FetchType.EAGER
- No Cascading

# Relationship «ServicePackageComposition»



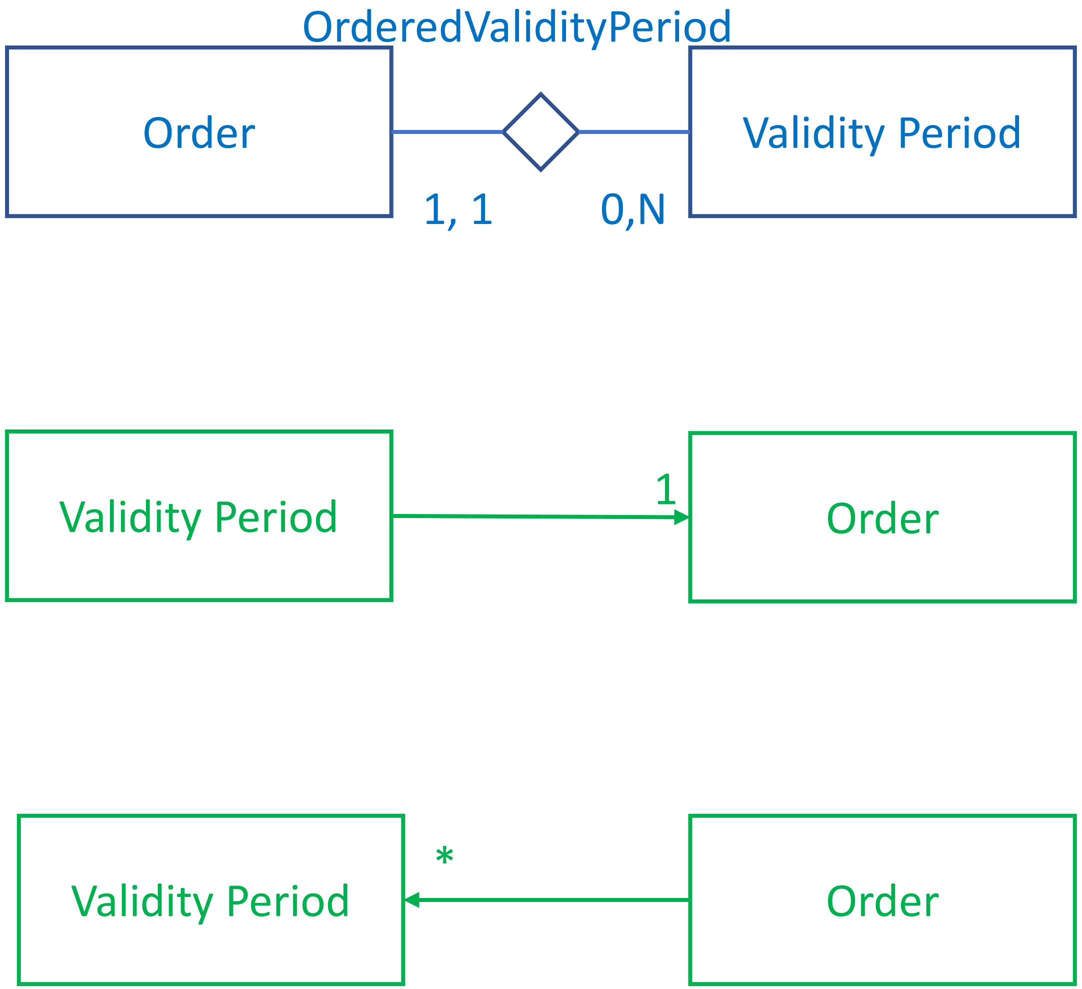
- Service Package → Service @ManyToMany is necessary to show to the Customer the available Services for a certain Service Package
- Service → Service Package @ManyToMany is not necessary, because there's no scenario where from the Service we have to retrieve the Service Package
- Unidirectional N:M → @ManyToMany in Service Package entity
- FetchType.EAGER
- No Cascading

# Relationship «OrderedValidityPeriod»



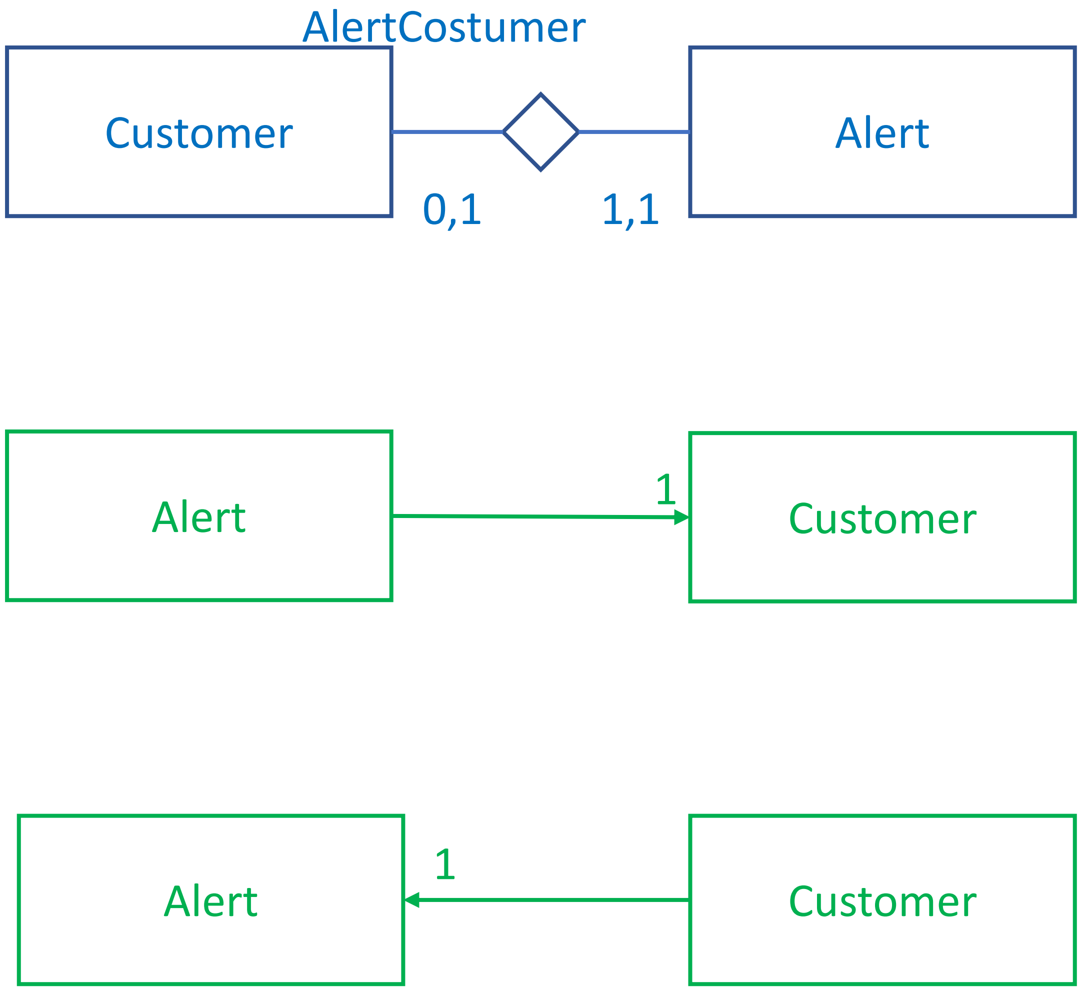
- Order → Validity Period @ManyToOne is necessary to show to the Customer the Validity Period for a certain Order
- Validity Period → Order @OneToMany is not necessary, because there’s no scenario where from the Validity Period we have to retrieve the Order
- Unidirectional N:1 → @ManyToOne in Order entity
- FetchType.EAGER
- No Cascading

# Relationship «OrderedValidityPeriod»



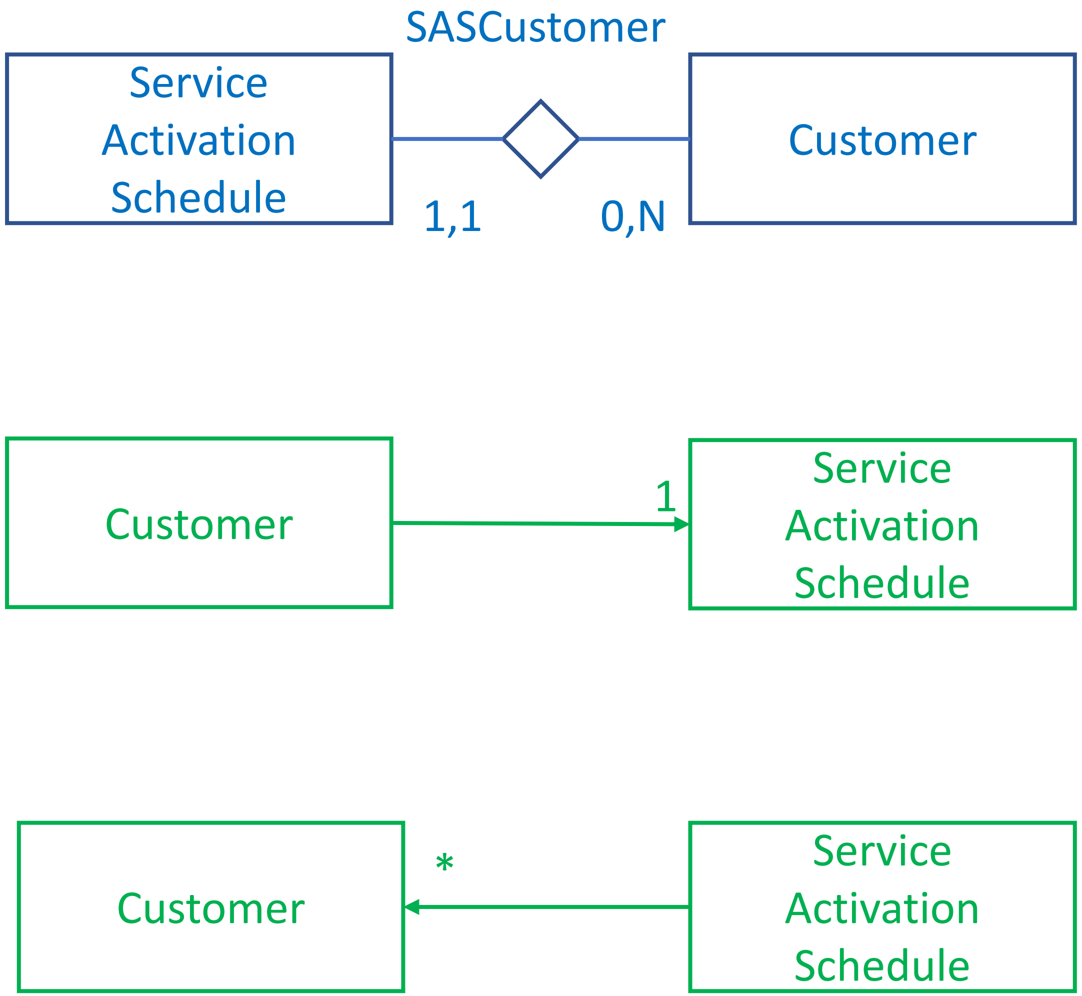
- Order → Validity Period @ManyToOne is necessary to show to the Customer the Validity Period for a certain Order
- Validity Period → Order @OneToMany is not necessary, because there’s no scenario where from the Validity Period we have to retrieve the Order
- Unidirectional N:1 → @ManyToOne in Order entity
- FetchType.EAGER
- No Cascading

# Relationship «AlertCustomer»



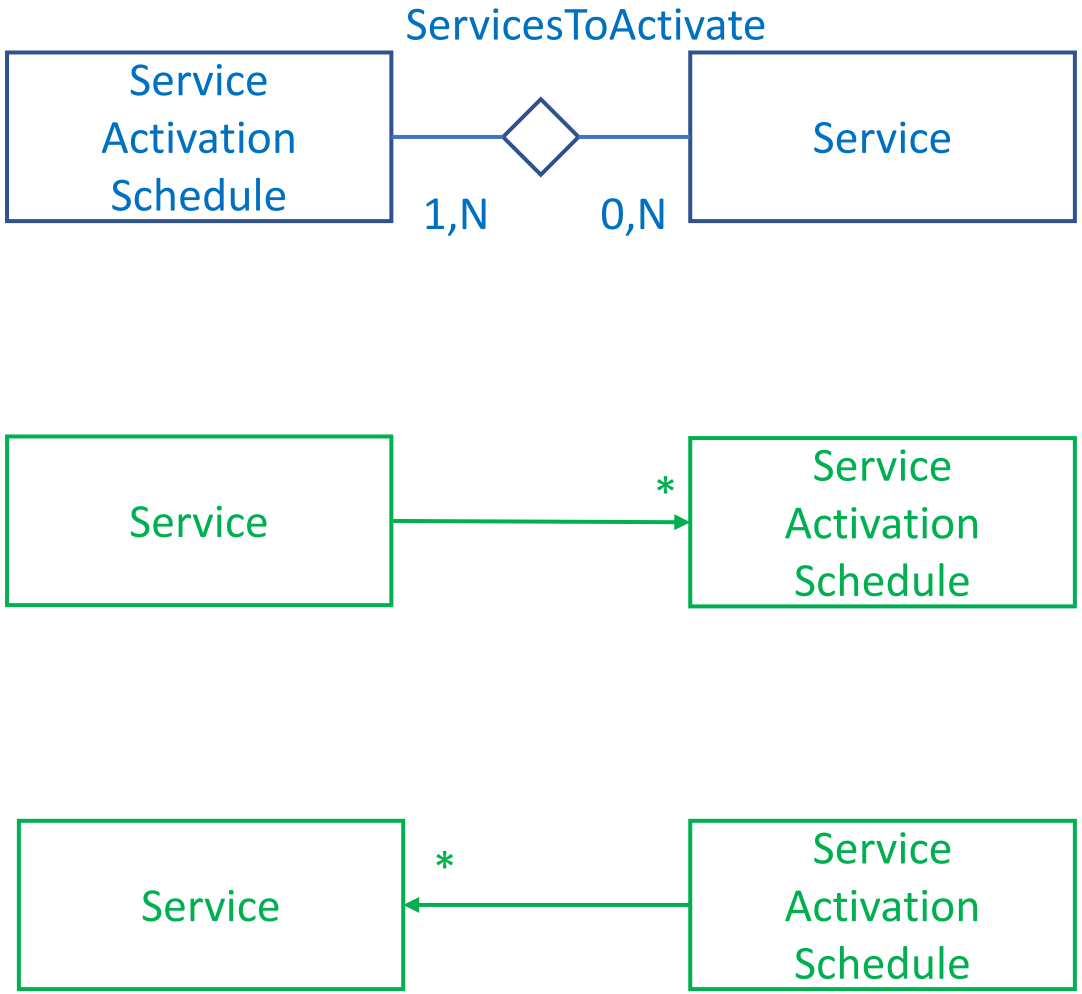
- Customer → Alert @OneToOne is necessary to show to the Customer the Alert for the failed Orders
- Alert → Customer @OneToOne is not necessary, because there’s no scenario where from the Alert we have to retrieve the Customer
- Unidirectional 1:1 → @OneToOne in Customer entity, but mapped in Alert
- FetchType.EAGER
- No Cascading

# Relationship «SASCustomer»


SASCustomer
Service Activation Schedule
Customer
1,1
0,N
Customer
1
Service Activation Schedule
Customer
*
Service Activation Schedule

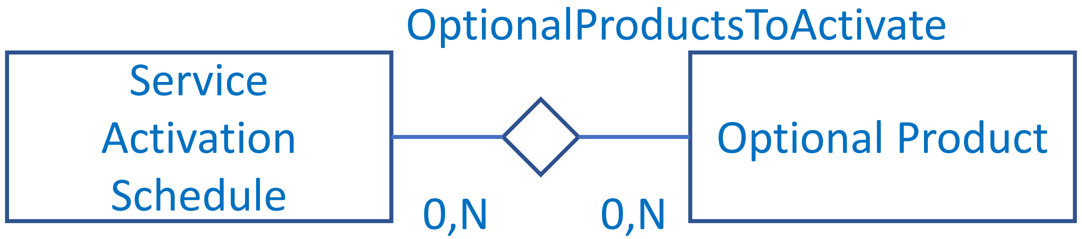
- Customer → Service Activation Schedule @OneToMany is necessary to show to the Customer the Service Activation Schedule
- Service Activation Schedule → Customer @ManyToOne is not necessary, because there’s no scenario where from the Service Activation Schedule we have to retrieve the Customer
- Unidirectional 1:N → @OneToMany in Customer entity, mapped as a bidirectional relationship (@ManyToOne in Service Activation Schedule)
- FetchType.EAGER
- No Cascading

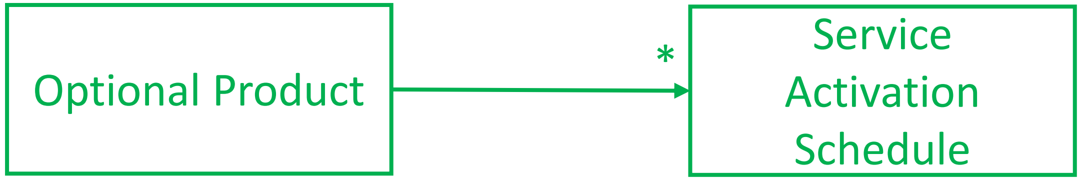
# Relationship «ServicesToActivate»

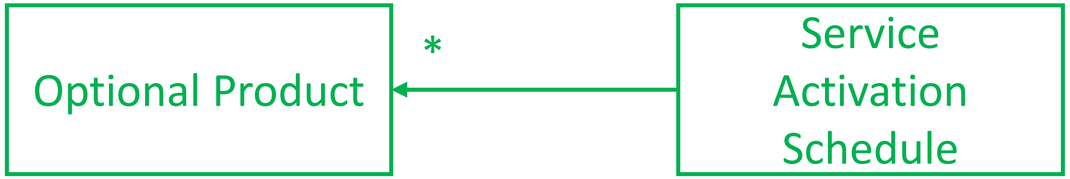

ServicesToActivate
Service Activation Schedule
Service
1,N
0,N
Service
*
Service Activation Schedule
Service
*
Service Activation Schedule

- Service Activation Schedule → Service @ManyToMany is necessary to show to the Customer the Services contained by the Service Activation Schedule
- Service → Service Activation Schedule @ManyToMany is not necessary, because there’s no scenario where from the Service we have to retrieve the Service Activation Schedule
- Unidirectional N:M → @ManyToMany in Service Activation Schedule entity
- FetchType.EAGER
- No Cascading

# Relationship «OptionalProductsToActivate»







- Service Activation Schedule → Optional Product @ManyToMany is necessary to show to the Customer the Optional Products contained by the Service Activation Schedule
- Optional Product→ Service Activation Schedule @ManyToMany is not necessary, because there's no scenario where from the Optional Product we have to retrieve the Service Activation Schedule
- Unidirectional N:M → @ManyToMany in Service Activation Schedule entity
- FetchType.EAGER
- No Cascading

# Entitities Code

# «User» Entity

```
@Entity
@NamedQuery(name="User.checkCredentials", query="SELECT u FROM User u WHERE u.username=? AND
u.password=?2")
//NAMEDQUERY ORDINI RIGETTATI

Public class User implements Serializable{
Private static final long serialVersionUID=1L;

@Id
Private String username;

Private String email;
Private String password;
Private boolean insolvent;
Private UserType type; //UserType is a Enum Class with EMPLOYEE and CUSTOMER as values

@OneToMany(fetch=FetchType.EAGER, mappedBy="customer", cascade=?
Private List<ServiceActivationSchedule> serviceActivationSchedules;
@OneToOne(mappedBy="customer")
Private Alert alert;
}
```

# «Order» Entity

```
@Entity
@NamedQuery(name="User.checkCredentials", query="SELECT u FROM User u WHERE u.username=? AND
u.password=?2")
//NAMEDQUERY ORDINI RIGETTATI

Public class User implements Serializable{
Private static final long serialVersionUID=1L;

@Id
Private String username;

Private String email;
Private String password;
Private boolean insolvent;
Private UserType type; //UserType is a Enum Class with EMPLOYEE and CUSTOMER as values

@OneToMany(fetch=FetchType.EAGER, mappedBy="customer", cascade=?
Private List<ServiceActivationSchedule> serviceActivationSchedules;
@OneToOne(mappedBy="customer")
Private Alert alert;
}
```

# «Service Package» Entity

```
@Entity
@NamedQuery(name="User.checkCredentials", query="SELECT u FROM User u WHERE u.username=? AND
u.password=?2")
//NAMEDQUERY ORDINI RIGETTATI

Public class User implements Serializable{
Private static final long serialVersionUID=1L;

@Id
Private String username;

Private String email;
Private String password;
Private boolean insolvent;
Private UserType type; //UserType is a Enum Class with EMPLOYEE and CUSTOMER as values

@OneToMany(fetch=FetchType.EAGER, mappedBy="customer", cascade=?
Private List<ServiceActivationSchedule> serviceActivationSchedules;
@OneToOne(mappedBy="customer")
Private Alert alert;
}
```

# «Optional Product» Entity

```
@Entity
@NamedQuery(name="User.checkCredentials", query="SELECT u FROM User u WHERE u.username=? AND
u.password=?2")
//NAMEDQUERY ORDINI RIGETTATI

Public class User implements Serializable{
Private static final long serialVersionUID=1L;

@Id
Private String username;

Private String email;
Private String password;
Private boolean insolvent;
Private UserType type; //UserType is a Enum Class with EMPLOYEE and CUSTOMER as values

@OneToMany(fetch=FetchType.EAGER, mappedBy="customer", cascade=?
Private List<ServiceActivationSchedule> serviceActivationSchedules;
@OneToOne(mappedBy="customer")
Private Alert alert;
}
```

# «Service» Entity

```
@Entity
@NamedQuery(name="User.checkCredentials", query="SELECT u FROM User u WHERE u.username=? AND
u.password=?2")
//NAMEDQUERY ORDINI RIGETTATI

Public class User implements Serializable{
Private static final long serialVersionUID=1L;

@Id
Private String username;

Private String email;
Private String password;
Private boolean insolvent;
Private UserType type; //UserType is a Enum Class with EMPLOYEE and CUSTOMER as values

@OneToMany(fetch=FetchType.EAGER, mappedBy="customer", cascade=?
Private List<ServiceActivationSchedule> serviceActivationSchedules;
@OneToOne(mappedBy="customer")
Private Alert alert;
}
```

# «Service Activation Schedule» Entity

```
@Entity
@NamedQuery(name="User.checkCredentials", query="SELECT u FROM User u WHERE u.username=? AND
u.password=?2")
//NAMEDQUERY ORDINI RIGETTATI

Public class User implements Serializable{
Private static final long serialVersionUID=1L;

@Id
Private String username;

Private String email;
Private String password;
Private boolean insolvent;
Private UserType type; //UserType is a Enum Class with EMPLOYEE and CUSTOMER as values

@OneToMany(fetch=FetchType.EAGER, mappedBy="customer", cascade=?
Private List<ServiceActivationSchedule> serviceActivationSchedules;
@OneToOne(mappedBy="customer")
Private Alert alert;
}
```

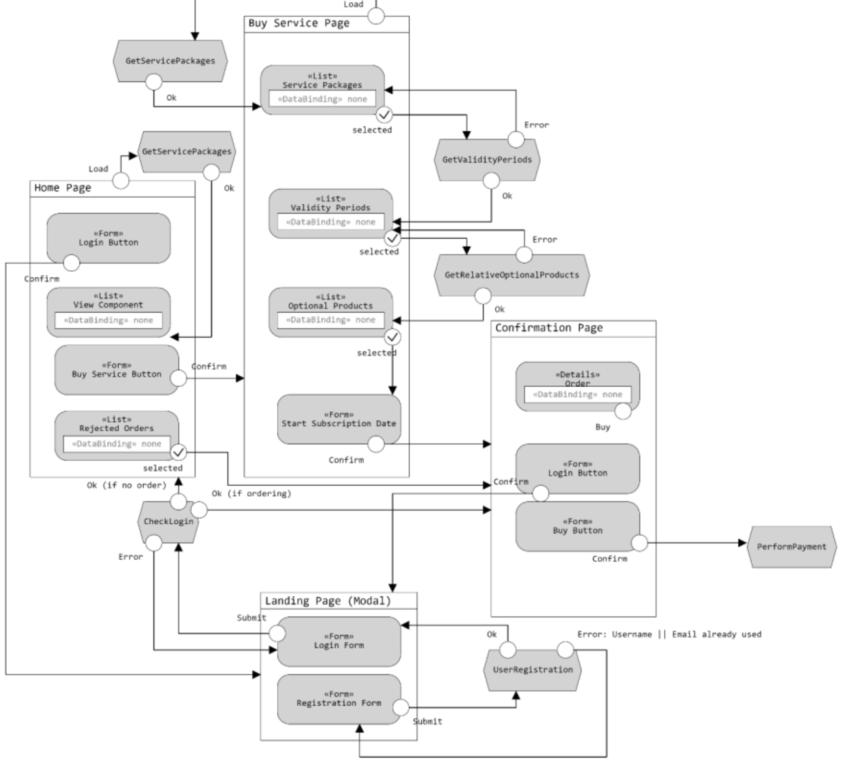
# «Validity Period» Entity

```
@Entity
@NamedQuery(name="User.checkCredentials", query="SELECT u FROM User u WHERE u.username=? AND
u.password=?2")
//NAMEDQUERY ORDINI RIGETTATI

Public class User implements Serializable{
Private static final long serialVersionUID=1L;

@Id
Private String username;

Private String email;
Private String password;
Private boolean insolvent;
Private UserType type; //UserType is a Enum Class with EMPLOYEE and CUSTOMER as values

@OneToMany(fetch=FetchType.EAGER, mappedBy="customer", cascade=?
Private List<ServiceActivationSchedule> serviceActivationSchedules;
@OneToOne(mappedBy="customer")
Private Alert alert;
}
```

# «Alert» Entity

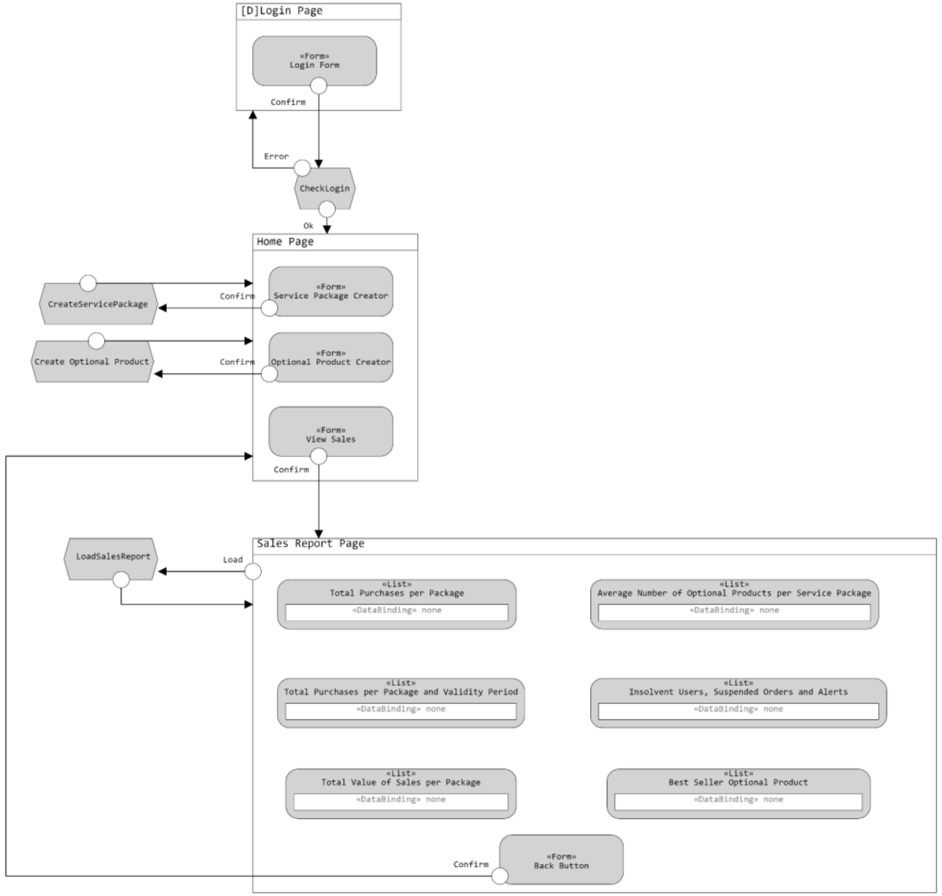
```
@Entity
@NamedQuery(name="User.checkCredentials", query="SELECT u FROM User u WHERE u.username=? AND
u.password=?2")
//NAMEDQUERY ORDINI RIGETTATI

Public class User implements Serializable{
Private static final long serialVersionUID=1L;

@Id
Private String username;

Private String email;
Private String password;
Private boolean insolvent;
Private UserType type; //UserType is a Enum Class with EMPLOYEE and CUSTOMER as values

@OneToMany(fetch=FetchType.EAGER, mappedBy="customer", cascade=?
Private List<ServiceActivationSchedule> serviceActivationSchedules;
@OneToOne(mappedBy="customer")
Private Alert alert;
}
```

# IFML



Customer Side

Employee Side



[D]Login Page
«Form» Login Form
Confirm
Error
CheckLogin
Ok
Home Page
CreateServicePackage
Confirm
«Form» Service Package Creator
Create Optional Product
Confirm
«Form» Optional Product Creator
«Form» View Sales
Confirm
Sales Report Page
LoadSalesReport
Load
«List» Total Purchases per Package
«DataBinding» none
«List» Average Number of Optional Products per Service Package
«DataBinding» none
«List» Total Purchases per Package and Validity Period
«DataBinding» none
«List» Insolvent Users, Suspended Orders and Alerts
«DataBinding» none
«List» Total Value of Sales per Package
«DataBinding» none
«List» Best Seller Optional Product
«DataBinding» none
Confirm
«Form» Back Button

# List of Components

- Client Components
    - Servlets
        - User Registration
        - CheckLogin (Customer)
        - CheckLogin (Employee)
        - GetServicePackage
        - GetValidityPeriods
        - GetRelativeOptionalProducts
        - PerformPayment
        - CreateServicePackage
        - CreateOptionalProduct
        - LoadSalesReport
    - Views
        - Landing Page (Modal)
        - Login Page (Employee)
        - Home Page (Customer)
        - Home Page (Employee)
        - Buy Service Page
        - Confirmation Page (Modal)
        - Sales Report Page
    - Java Beans

- Back End Components
    - Entities
        - User
        - Order
        - Service Package
        - Optional Product
        - Service
        - Validity Period
        - Service Activation Schedule
        - Alert
    - Business Components (EJBs)
        - BC1
            - (Stateless or Stateful)

# Sequence Diagrams