Project:  
Building a hotel service management system

Logo

Description automatically generated

Class code:

SE104.L23.PMCL

Members:

Ha Thuc Dang Khoa

Huynh Lu Anh Khoa

Le Van Tri

Instructor:

Nguyen Thi Thanh Truc

UNIVERSITY OF INFORMATION TECHNOLOGY

HO CHI MINH NATIONAL UNIVERSITY

# Table of content

[**Table of content**](#_vrqzpnctcpi4) **1**

[**Table of figure**](#_rnuex4rlchmz) **3**

[**Introduction**](#_733egzjw8d2r) **3**

[Purpose](#_yypvqzgptlxz) 3

[Definition, Acronyms and Abbreviations](#_tv2v21nr5byb) 3

[Overview](#_tv2v21nr5byb) 4

[**System architect**](#_rz0gf6i933c3) **4**

[**Screen flow**](#_rz0gf6i933c3) **5**

[**Common package and mechanism**](#_rz0gf6i933c3) **5**

[Class diagram](#_tbkcjm5tjrv6) 5

[Data transfer object package](#_zaorfat6i9db) 5

[BookingDTO](#_3moiymrcgbgi) 5

[Attribute](#_wvllhe1qd4dp) 5

[Methods](#_ok7erzno3h2n) 6

[CustomerDTO](#_h8cvtrhtbcfq) 6

[Attribute](#_ar7n5cflr5rr) 6

[Methods](#_n19oy28hk3lg) 6

[DepositDTO](#_m7xcn4usmpqf) 7

[Attribute](#_hlijcap8spiw) 7

[Methods](#_c8auty3vbaz9) 7

[PaymentDTO](#_j36eskgtxosq) 8

[Attribute](#_pu1n95f7988v) 8

[Methods](#_wjz4psvbiz2m) 8

[ServiceBillDTO](#_w9m7bxmftkxs) 9

[Attribute](#_wi3bnhl9fd20) 9

[Methods](#_crhsxnajn744) 9

[ServiceDTO](#_6j41rqrn856) 10

[Attribute](#_5aslilz0pla4) 10

[Methods](#_j28d69twmc14) 10

[Business logic layer package](#_cc1bhr7c4hqk) 10

[BookingBLL](#_vz0q5uh65lhr) 10

[Methods](#_j349vniz99v8) 11

[CustomerBLL](#_4i1qg5brf4yq) 12

[Attribute](#_kouxlqk0zv8h) 12

[Methods](#_5gnc51pdzclx) 12

[DepositBLL](#_c6tejt8sfase) 14

[Methods](#_cf9ai89zip1h) 14

[PaymentBLL](#_mn4y6gujosde) 16

[Attribute](#_b8fvem7hco7l) 16

[Methods](#_2p90lunm1n82) 16

[ServiceBillBLL](#_6axfb6wer13n) 17

[Attribute](#_sib1cowzcjp8) 17

[Methods](#_7qs7972yvwhq) 17

[Data access layer package](#_oyl4kat2bf7g) 19

[BookingDAL](#_x7ckquflem56) 19

[Attribute](#_7ipwjv9zhtlq) 19

[Methods](#_xmc1fxckyw7i) 20

[CustomerDAL](#_rudpdzfbxse8) 22

[Attribute](#_kkr4q9ctd7u8) 22

[Methods](#_lkt0gt9q44ew) 22

[DepositDAL](#_gzck7gynzzgb) 24

[Attribute](#_vetntv3cmgnn) 24

[Methods](#_n4lx6kqx7uuh) 24

[PaymentDAL](#_8iovu68vbrj7) 26

[Attribute](#_yq94id1fsr6z) 26

[Methods](#_u93cotvh0rbg) 27

[ServiceBillDAL](#_rjn5909y584m) 28

[Attribute](#_23nkh18htx83) 28

[Methods](#_yfpekyvevizr) 28

[Utils package](#_sjgwpnxixp3v) 31

[DBUtils](#_ultfoal0vugu) 31

[Attribute](#_6nu9ypnt15ii) 31

[Methods](#_ojx4jhdkpfxv) 31

[**Diagrams**](#_rz0gf6i933c3) **32**

[System states](#_7fju1e54paxm) 33

[Sequence diagram](#_vea3jqnd7tx1) 34

[Booking](#_k5ehcugps9dm) 34

[Record service usage](#_ebxn2j2ajvp0) 34

[Check-out](#_m8ey4ythdys8) 34

[Payment](#_q5dzrhzibjcb) 36

[**Database design**](#_rz0gf6i933c3) **36**

[ERDS](#_azs0axeke8t2) 36

[RoomInformation](#_8dzhl8tvrbxa) 37

[Booking](#_zco6k9b217t3) 37

[Customer](#_j8w3wecg7vws) 38

[Staff](#_byeeio42avgj) 38

[Deposit](#_uc2044dvt6ya) 38

[ServiceBill](#_3i6336jnjuht) 39

[Service](#_i65ehfhjtvvn) 39

[Payment](#_u9ux7nrma65t) 40

# Table of figure

[1. System architecture 5](#_Toc76630821)

[2. System's state diagram 32](#_Toc76630822)

[3. Booking sequence 35](#_Toc76630823)

[4. Record service usage sequence 35](#_Toc76630824)

[5. Checkout sequence 36](#_Toc76630825)

[6. Payment sequence 37](#_Toc76630826)

[7. Entity relationship diagram 38](#_Toc76630827)

# Introduction

## Purpose

This documentation is the design documentation for building a hotel service management system for the final project of the Introduction to Software engineering course for Ho Chi Minh University of Information Technology. Course code: SE104.L23.PMCL.

## Definition, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| Abbreviations | Description | Comment |
| PL | Presentation layer |  |
| BLL | Business logic layer |  |
| DAL | Data access layer |  |
| DTO | Data transfer object |  |
| JDBC | Java Database Connection |  |
| P key | Primary key |  |
| F key | Foreign key |  |
| N/A | Not available |  |

## Overview

This document serves as a guideline for designing the hotel service management system. The document, from this point onward, will provide the following information with the according order:

* System architecture: specify the program’s structure
* Screen flow: provide interface design and screen flow
* Common package: this section gives information on how the code will be organized, classes’ method and their interface
* Diagrams: this section will present diagrams’ representing a certain aspect of the system.
* Database design: this section gives information on the databases used by the system.

# System architect

The system is structured using 3 layers: presentation layer(PL), business logic layer(BLL), data access layers(DAL). The users interact directly with the system through the presentation layer using the graphical user interface(GUI). This information will then be packed into data transfer object transfer(DTO) to the business layer. The BLL is the intermediate layer between PL and DAL. All information from PL must go through here to be processed before reaching DAL and vice versa. After the information has been processed, BLL will send a request for necessary to DAL. From here DAL will perform data queries from the database. Acquired data will be transferred to BLL for computation through DAL in a DTO. BLL will then give responses to the user accordingly through the PL.

Presentation layer

Business logic layer

Data access layer

Database

Send request

Request data(if necessary)

Query data

Return data

Return data

Return response

User

1. System architecture

# Screen flow

# Common package and mechanism

## Class diagram

## Data transfer object package

Each class in this package serves as a medium for transferring data between layers of the system’s architecture.

### BookingDTO

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | bookingID | int | Private | Booking ID |
| 2 | roomID | int | private | Room ID |
| 3 | customerID | int | private | Customer ID |
| 4 | startDate | Date | private | Starting commodation date |
| 5 | endDate | Date | private | Ending commodation date |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getBookingID | Getter for bookingID attribute |
| 2 | setBookingID | Setter for bookingID attribute |
| 3 | getRoomID | Getter for roomID attribute |
| 4 | setRoomID | Setter for roomID attribute |
| 5 | getCustomerID | Getter for customerID attribute |
| 6 | setCustomerID | Setter for customerID attribute |
| 7 | getStartDate | Getter for startDate attribute |
| 8 | setStartDate | Setter for startDate attribute |
| 9 | getEndDate | Getter for endDate attribute |
| 10 | setEndDate | Setter for endDate attribute |

### CustomerDTO

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | customerID | int | Private | Customer ID |
| 2 | customerName | String | private | Customer’s name |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getCustomerID | Getter for bookingID attribute |
| 2 | setCustomerID | Setter for bookingID attribute |
| 3 | getCustomerName | Getter for customerName attribute |
| 4 | setCustomerName | Setter for customerName attribute |

### DepositDTO

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | depositID | int | Private | Deposit ID |
| 2 | bookingID | int | private | Booking ID |
| 3 | customerID | int | private | Customer ID |
| 4 | depositDate | Date | private | Date the deposit’s received |
| 5 | explanation | String | private | Explanation for receiving deposit |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getDepositID | Getter for depositID attribute |
| 2 | setDepositID | Setter for depositID attribute |
| 3 | getBookingID | Getter for bookingID attribute |
| 4 | setBookingID | Setter for bookingID attribute |
| 5 | getCustomerID | Getter for customerID attribute |
| 6 | setCustomerID | Setter for customerID attribute |
| 7 | getDepositDate | Getter for depositDate attribute |
| 8 | setDepositDate | Setter for depositDate attribute |
| 9 | getExplanation | Getter for explanation attribute |
| 10 | setExplanation | Setter for explanation attribute |

### PaymentDTO

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | paymentID | int | Private | Payment ID |
| 2 | bookingID | int | private | Booking ID |
| 3 | amount | int | private | Payment amount |
| 4 | paymentTypeName | String | private | Chosen payment type |
| 5 | paymentDate | Date | private | The date payment’s received |
| 6 | paymentStatus | boolean | private | Payment’s status |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getPaymentID | Getter for paymentID attribute |
| 2 | getBookingID | Getter for bookingID attribute |
| 3 | getAmount | Getter for amount attribute |
| 4 | getPaymentTypeName | Getter for paymentTypeName attribute |
| 5 | getPaymentDate | Getter for paymentDate attribute |
| 6 | isPaymentStatus | Check payment status |
| 7 | setPaymentID | Setter for paymentID attribute |
| 8 | setBookingID | Setter for bookingID attribute |
| 9 | setAmount | Setter for amount attribute |
| 10 | setPaymentTypeName | Setter for paymentTypeName attribute |
| 11 | setPaymentDate | Setter for paymentDate attribute |
| 12 | setPaymentStatus | Setter for paymentStatus attribute |

### ServiceBillDTO

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | serviceBillID | int | Private | Service bill Id |
| 2 | roomID | int | private | Room ID |
| 3 | staffID | int | private | Staff ID |
| 4 | serviceID | int | private | Service ID |
| 5 | quantity | int | private | Service usage’s quantity |
| 6 | serviceDate | Date | private | Date the service’s used |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getServiceBillID | Getter for serviceBillID attribute |
| 2 | getRoomID | Getter for roomIDattribute |
| 3 | getStaffID | Getter for staffID attribute |
| 4 | getServiceID | Getter for serviceID attribute |
| 5 | getQuantity | Getter for quantity attribute |
| 6 | getServiceDate | Getter for serviceDate attribute |
| 7 | setServiceBillID | Setter for serviceBillID attribute |
| 8 | setRoomID | Setter for roomID attribute |
| 9 | setStaffID | Setter for staffID attribute |
| 10 | setServiceID | Setter for serviceID attribute |
| 11 | setQuantity | Setter for quantity attribute |
| 12 | setServiceDate | Setter for serviceDate attribute |

### ServiceDTO

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | serviceID | int | Private | Service ID |
| 2 | serviceName | String | private | Service’s name |
| 3 | serviceUnit | String | private | Service counting unit |
| 4 | serviceDepartment | String | private | Department in charge of the service |
| 5 | serviceUnitPrice | double | private | Price for each unit of service |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getServiceID | Getter for serviceID attribute |
| 2 | getServiceName | Getter for serviceName attribute |
| 3 | getServiceUnit | Getter for serviceUnit attribute |
| 4 | getServiceDepartment | Getter for serviceDepartment attribute |
| 5 | getUnitPrice | Getter for unitPrice attribute |
| 6 | setServiceID | Setter for serviceID attribute |
| 7 | setServiceName | Setter for serviceName attribute |
| 8 | setServiceUnit | Setter for serviceUnit attribute |
| 9 | setServiceDepartment | Setter for serviceDepartment attribute |
| 10 | setUnitPrice | Setter for unitPrice attribute |

## Business logic layer package

### BookingBLL

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | bookingDAL | BookingDAL | Private | Serve as an interface to interact with the DAL. |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getAllBooking | Get a list of all booking record in the database |
| 2 | insertBooking | Transfer booking information to DAL to be stored in the database. |

**getAllBooking method**

*Declaration*

|  |
| --- |
| public ArrayList<BookingDTO> getAllBooking() |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
|  | <return> | ArrayList | None | ArrayList is a class exclusive to Java. It is used to store booking information |

*Implementation*

|  |
| --- |
| Function get all booking record in the databases  Pass in: none  Call bookingDAL.getAllBooking()  Pass out: List of all booking in form of ArrayList Function end |

**insertBooking method**

*Declaration*

|  |
| --- |
| **public int insertBooking(BookingDTO booking)** |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | booking | BookingDTO | None | Booking information to be transferred to DAL. |
|  | <return> | int | None | Insert status code. |

*Implementation*

|  |
| --- |
| Function insert customer information in database  Pass in: BookingDTO object  Call bookingDAL.insertBooking() on BookingDTO object  Pass out: Insert status Function end |

### CustomerBLL

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | customerDAL | CustomerDAL | Private | Serve as an interface to interact with the DAL. |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getAllCustomer | Get a list of all customer in the database |
| 2 | insertCustomer | Transfer customer information to DAL to be stored in the database. |

**getAllCustomer**

*Declaration*

|  |
| --- |
| public ArrayList<CustomerDTO> getAllCustomer() |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
|  | <return> | Array | None | Return a list of all customer in the database |

*Implementation*

|  |
| --- |
| Function get all customer from databases  Pass in: None  Call this.customerDAL.getAllCustomer()  Pass out: List of all customer in form of ArrayList Function end |

**insertCustomer**

*Declaration*

|  |
| --- |
| **public int insertCustomer(CustomerDTO customer)** |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | customer | CustomerDTO | None | Customer information to be transferred to DAL. |
|  | <return> | int | None | Insert status code. |

*Implementation*

|  |
| --- |
| Function insert customer information in database  Pass in: CustomerDTO object  Call this.customerDAL.insertCustomer() on CustomerDTO object  Pass out: Insert status Function end |

### DepositBLL

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | depositDAL | DepositDAL | Private | Serve as an interface to interact with the DAL. |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getAllDeposit | Get a list of all deposit in the database |
| 2 | insertDeposit | Transfer deposit information to DAL to be stored in the database. |
| 3 | deleteDeposit | Delete a deposit record in the database. |

**getAllDeposit**

*Declaration*

|  |
| --- |
| public ArrayList<DepositDTO> getAllDeposit() |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
|  | <return> | Array | None | Return a list of all deposit in the database |

*Implementation*

|  |
| --- |
| Function get all deposit from databases  Pass in: None  Call this.depositDAL.getAllDeposit()  Pass out: Array of depositDTO Function end |

**insertDeposit**

*Declaration*

|  |
| --- |
| public int insertDeposit(DepositDTO deposit) |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | deposit | DepositDTO | None | Deposit information to be stored in the database. |
|  | <return> | int | None | Insert status code |

*Implementation*

|  |
| --- |
| Function get all customer from databases  Pass in: None  Call this.depositDAL.getAllDeposit()  Pass out: Insert status code Function end |

**deleteDeposit**

*Declaration*

|  |
| --- |
| public int deleteDeposit(DepositDTO deposit) |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | deposit | DepositDTO | None | Deposit information to be deleted in the database. |
|  | <return> | int | None | Delete status code |

*Implementation*

|  |
| --- |
| Function get all customer from databases  Pass in: None  Call this.depositDAL.getAllDeposit()  Pass out: Delete status code Function end |

### PaymentBLL

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | paymentDAL | PaymentDAL | Private | Serve as an interface to interact with the DAL. |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getAllPayment | Get a list of all payment record in the database |
| 2 | insertPayment | Transfer payment information to DAL to be stored in the database. |

**getAllPayment**

*Declaration*

|  |
| --- |
| public ArrayList<PaymentDTO> getAllPayment() |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
|  | <return> | Array | None | Return a list of all payment record in the database |

*Implementation*

|  |
| --- |
| Function get all payment record from databases  Pass in: None  Call this.paymentDAL.getAllPayment()  Pass out: List of payment record in form of ArrayList Function end |

**insertPayment**

*Declaration*

|  |
| --- |
| public int insertPayment(PaymentDTO payment) |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | payment | PaymentDTo | None | Payment information to be recorded in the database |
|  | <return> | Array | None | Insert status code |

*Implementation*

|  |
| --- |
| Function get all payment record from databases  Pass in: None  Call this.paymentDAL.getAllPayment()  Pass out: Insert status code Function end |

### ServiceBillBLL

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | serviceBillDAL | ServiceBillDAL | Private | Serve as an interface to interact with the DAL. |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getAllServiceBill | Get a list of all service bill in the database |
| 2 | insertServiceBill | Transfer service bill’s information to DAL to be stored in the database. |
| 3 | deleteServiceBill | Transfer service bill’s information required for deletion |

**getAllServiceBill**

*Declaration*

|  |
| --- |
| public ArrayList<ServiceBillDTO> getAllServiceBill() |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
|  | <return> | Array | None | Return a list of all service bill record in the database |

*Implementation*

|  |
| --- |
| Function get all service bill record from databases  Pass in: None  Call this.serviceDAL.getAllServiceBill()  Pass out: List of service bill in form of ArrayList Function end |

**insertServiceBill**

*Declaration*

|  |
| --- |
| public int insertServiceBill(ServiceBillDTO serviceBill) |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | serviceBill | ServiceBillDTO | None | Service bill to be recorded in the database |
|  | <return> | int | None | Insert status code |

*Implementation*

|  |
| --- |
| Function insert service bill information into database  Pass in: ServiceBillDTO object  Call this.serviceBillDAL.insertServiceBill() on ServiceBillDTO object  Pass out: Insert status code Function end |

**deleteServiceBill**

*Declaration*

|  |
| --- |
| public int deleteServiceBill(ServiceBillDTO serviceBill) |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | serviceBill | ServiceBillDTO | None | Service bill to be deleted in the database |
|  | <return> | int | None | Insert status code |

*Implementation*

|  |
| --- |
| Function delete service bill from databases  Pass in: ServiceBillDTO object  Call this.serviceBillDAL.insertServiceBill() on ServiceBillDTO object  Pass out: Delete status code Function end |

## Data access layer package

### BookingDAL

This class communicates with the Booking entity and performs queries.

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | conn | Connection | Private | Store connection to database |
| 2 | dbu | DBUtils | Private | Instance of DBUtils class |
| 3 | pres | PreparedStatement | Private | Instance of PreparedStatement class, directly communication through instance of Connection class. |
| 4 | rs | ResultSet | Private | Result after executing the query. |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getAllBooking | Get a list of Booking information from the database. |
| 2 | insertBooking | Insert new Booking records into databases. |
| 3 | saveBooking | Update room information. |

**getAllBooking**

*Declaration*

|  |
| --- |
| public ArrayList<BookingDTO> getAllBooking() |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
|  | <return> | ArrayList | Empty array list | ArrayList contains a list of bookings retrieved from the database. |

*Implementation*

|  |
| --- |
| Function get booking information from databases  Pass in: None  Initialize an empty ArrayList  Establish connection  Create PreparedStatement  Get ResultSet using PreparedStatement  Parsing ResultSet and insert into ArrayList  Pass out: Result ArrayList Function end |

**insertBooking**

*Declaration*

|  |
| --- |
| public int insertBooking(BookingDTO booking) |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | booking | BookingDTO | None | Instance of BookingDTO class. Storing information to be inserted into the database. |
|  | <return> | int | 0 | Insert status code. |

*Implementation*

|  |
| --- |
| Function insert booking information into databases  Pass in: None  Initialize status code  Establish connection  Create PreparedStatement  Get insert status code using PreparedStatement  Pass out: Status code Function end |

**saveBooking**

*Declaration*

|  |
| --- |
| public int saveBooking(BookingDTO booking) |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | booking | BookingDTO | None | Instance of BookingDTO, storing booking information to be inserted into the database. |
|  | <return> | int | 0 | Update status code |

*Implementation*

|  |
| --- |
| Function update room information in databases  Pass in: None  Initialize status code  Establish connection  Create PreparedStatement  Get update status code after executing PreparedStatement  Pass out: status code Function end |

### CustomerDAL

This class communicates with the Customer entity and performs queries.

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | conn | Connection | Private | Store connection to database |
| 2 | dbu | DBUtils | Private | Instance of DBUtils class |
| 3 | pres | PreparedStatement | Private | Instance of PreparedStatement class, directly communication through instance of Connection class. |
| 4 | rs | ResultSet | Private | Result after executing the query. |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getAllCustomer | Get a list of Customer information from the database. |
| 2 | insertCustomer | Insert new Customer information into databases. |

**getAllCustomer**

*Declaration*

|  |
| --- |
| public int getAllCustomer() |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
|  | <return> | ArrayList | Empty ArrayList | List of all customers retrieved from the database. |

*Implementation*

|  |
| --- |
| Function get all customer from databases  Pass in: None  Initialize an empty ArrayList  Establish connection  Create PreparedStatement  Get ResultSet after executing PreparedStatement  Parsing ResultSet and inserting them into ArrayList  Pass out: ArrayList Function end |

**insertCustomer**

*Declaration*

|  |
| --- |
| public int insertCustomer(CustomerDTO customer) |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | customer | CustomerDTO | None | Instance of CustomerDTO to be inserted into the database. |
|  | <return> | int | 0 | Insert status |

*Implementation*

|  |
| --- |
| Function get all customer from databases  Pass in: None  Initialize status code  Establish connection  Create PreparedStatement  Get status code after executing PreparedStatement  Pass out: status code Function end |

### DepositDAL

This class communicates with the Deposit entity and performs queries.

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | conn | Connection | Private | Store connection to database |
| 2 | dbu | DBUtils | Private | Instance of DBUtils class |
| 3 | pres | PreparedStatement | Private | Instance of PreparedStatement class, directly communication through instance of Connection class. |
| 4 | rs | ResultSet | Private | Result after executing the query. |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getAllDeposit | Get a list of Deposit records from the database. |
| 2 | insertDeposit | Insert new Deposit records into the database. |
| 3 | deleteDeposit | Delete Deposit records from the database |

**getAllDeposit**

*Declaration*

|  |
| --- |
| public ArrayList<CustomerDTO> getAllDeposit() |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
|  | <return> | ArrayList | Empty ArrayList | List of all deposit records retrieved from the database. |

*Implementation*

|  |
| --- |
| Function get deposit record from databases  Pass in: None  Initialize an empty ArrayList  Establish connection  Create PreparedStatement  Get ResultSet after executing PreparedStatement  Parsing ResultSet and inserting them into ArrayList  Pass out: ArrayList Function end |

**insertDeposit**

*Declaration*

|  |
| --- |
| public int insertDeposit(DepositDTO deposit) |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | deposit | DepositDTO | None | Instance of DepositDTO class storing deposit information to be inserted into the database. |
|  | <return> | int | 0 | Insert status code |

*Implementation*

|  |
| --- |
| Function insert deposit into databases  Pass in: None  Initialize status code  Establish connection  Create PreparedStatement  Get status code after executing PreparedStatement  Pass out: Status code  Function end |

**deleteDeposit**

*Declaration*

|  |
| --- |
| public int deleteDeposit(DepositDTO deposit) |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | deposit | DepositDTO | None | Instance of DepositDTO class storing deposit information to be deleted from the database. |
|  | <return> | int | 0 | Delete status code |

*Implementation*

|  |
| --- |
| Function delete deposit from databases  Pass in: None  Initialize status code  Establish connection  Create PreparedStatement  Get status code after executing PreparedStatement  Pass out: Status code  Function end |

### PaymentDAL

This class communicates with the Deposit entity and performs queries.

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | conn | Connection | Private | Store connection to database |
| 2 | dbu | DBUtils | Private | Instance of DBUtils class |
| 3 | pres | PreparedStatement | Private | Instance of PreparedStatement class, directly communication through instance of Connection class. |
| 4 | rs | ResultSet | Private | Result after executing the query. |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getAllPayment | Get a list of Payment records from the database. |
| 2 | insertPayment | Insert new Payment records into the database. |

**getAllPayment**

*Declaration*

|  |
| --- |
| public ArrayList<PaymentDTO> getAllPayment() |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
|  | <return> | ArrayList | Empty ArrayList | List of all payment records retrieved from the database. |

*Implementation*

|  |
| --- |
| Function get payment record from databases  Pass in: None  Initialize an empty ArrayList  Establish connection  Create PreparedStatement  Get ResultSet after executing PreparedStatement  Parsing ResultSet and inserting them into ArrayList  Pass out: ArrayList Function end |

**insertPayment**

*Declaration*

|  |
| --- |
| public int insertPayment(PaymentDTO payment) |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | payment | PaymentDTO | None | Instance of PaymentDTO class, storing payment information to be inserted into the database. |
|  | <return> | int | 0 | Insert status code |

*Implementation*

|  |
| --- |
| Function insert payment record into databases  Pass in: None  Initialize status code  Establish connection  Create PreparedStatement  Get status code after executing PreparedStatement  Pass out: Status code Function end |

### ServiceBillDAL

This class communicates with the ServiceBill entity and performs queries.

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | conn | Connection | Private | Store connection to database |
| 2 | dbu | DBUtils | Private | Instance of DBUtils class |
| 3 | pres | PreparedStatement | Private | Instance of PreparedStatement class, directly communication through instance of Connection class. |
| 4 | rs | ResultSet | Private | Result after executing the query. |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | getAllServiceBill | Get a list of service bills from the database. |
| 2 | insertServiceBill | Insert a new service bill into the database. |
| 3 | deleteServiceBill | Delete a service from the database. |

**getAllServiceBill**

*Declaration*

|  |
| --- |
| public ArrayList<ServiceBillDTO> getAllServiceBill() |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
|  | <return> | ArrayList | Empty ArrayList | List of all service bill retrieved from the database |

*Implementation*

|  |
| --- |
| Function get all service bill from databases  Pass in: None  Initialize empty ArrayList  Establish connection  Create PreparedStatement  Get ResultSet after executing PreparedStatement  Parsing ResultSet and inserting ServiceBill into ArrayList  Close connection  Close PreparedStatement  Pass out: ArrayList Function end |

**insertServiceBill**

*Declaration*

|  |
| --- |
| public int insertServiceBill(ServiceBillDTO serviceBill) |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | serviceBill | ServiceBillDTO | None | Instance of ServiceDTO class, storing service information to be inserted into the database. |
|  | <return> | int | 0 | Insert status code |

*Implementation*

|  |
| --- |
| Function insert service bill information into the databases  Pass in: None  Initialize status code  Establish connection  Create PreparedStatement  Get status code after executing PreparedStatement  Parsing ResultSet and inserting ServiceBill into ArrayList  Close connection  Close PreparedStatement  Pass out: Status code Function end |

**deleteServiceBill**

*Declaration*

|  |
| --- |
| public int deleteServiceBill(ServiceBillDTO serviceBill) |

*Parameters and return*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
| 1 | serviceBill | ServiceBillDTO | None | Instance of ServiceDTO class, storing service information to be deleted from the database. |
|  | <return> | int | 0 | Delete status code |

*Implementation*

|  |
| --- |
| Function delete service bill information from the databases  Pass in: None  Initialize status code  Establish connection  Create PreparedStatement  Get status code after executing PreparedStatement  Parsing ResultSet and inserting ServiceBill into ArrayList  Close connection  Close PreparedStatement  Pass out: Status code Function end |

## Utils package

### DBUtils

This class incharge of the database's utility like creating connections.

#### Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Attribute | Type | Scope | Description |
| 1 | conn | Connection | Private | Store connection to database |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 1 | createConn | Create and return a database connection |

**createConn**

*Declaration*

|  |
| --- |
| public int createConn() |

*Parameters and return*

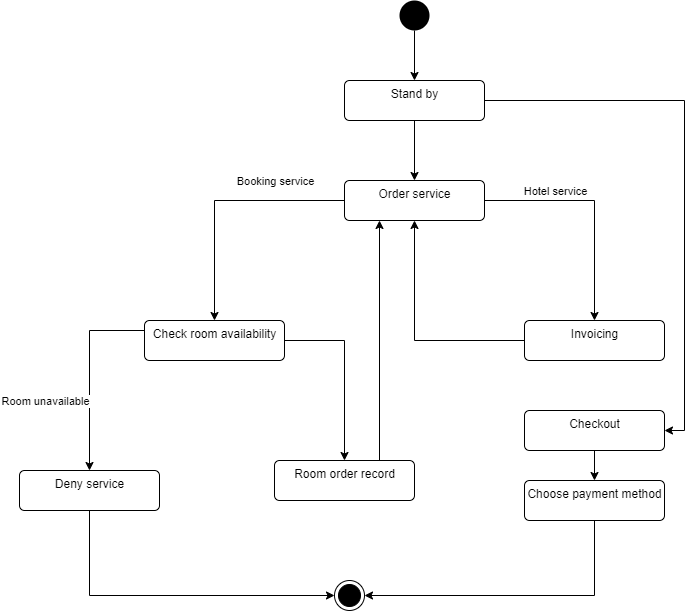
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Parameter | Type | Default | Description |
|  | <return> | Connection | None | Connection to database |

*Implementation*

|  |
| --- |
| Function delete service bill from databases  Pass in: None  Create new instance of JDBC Driver using Class.forName()  Create new connection using DriverManager.getConnection()  Print out connection creation’s result  Pass out: Connection to database Function end |

# Diagrams

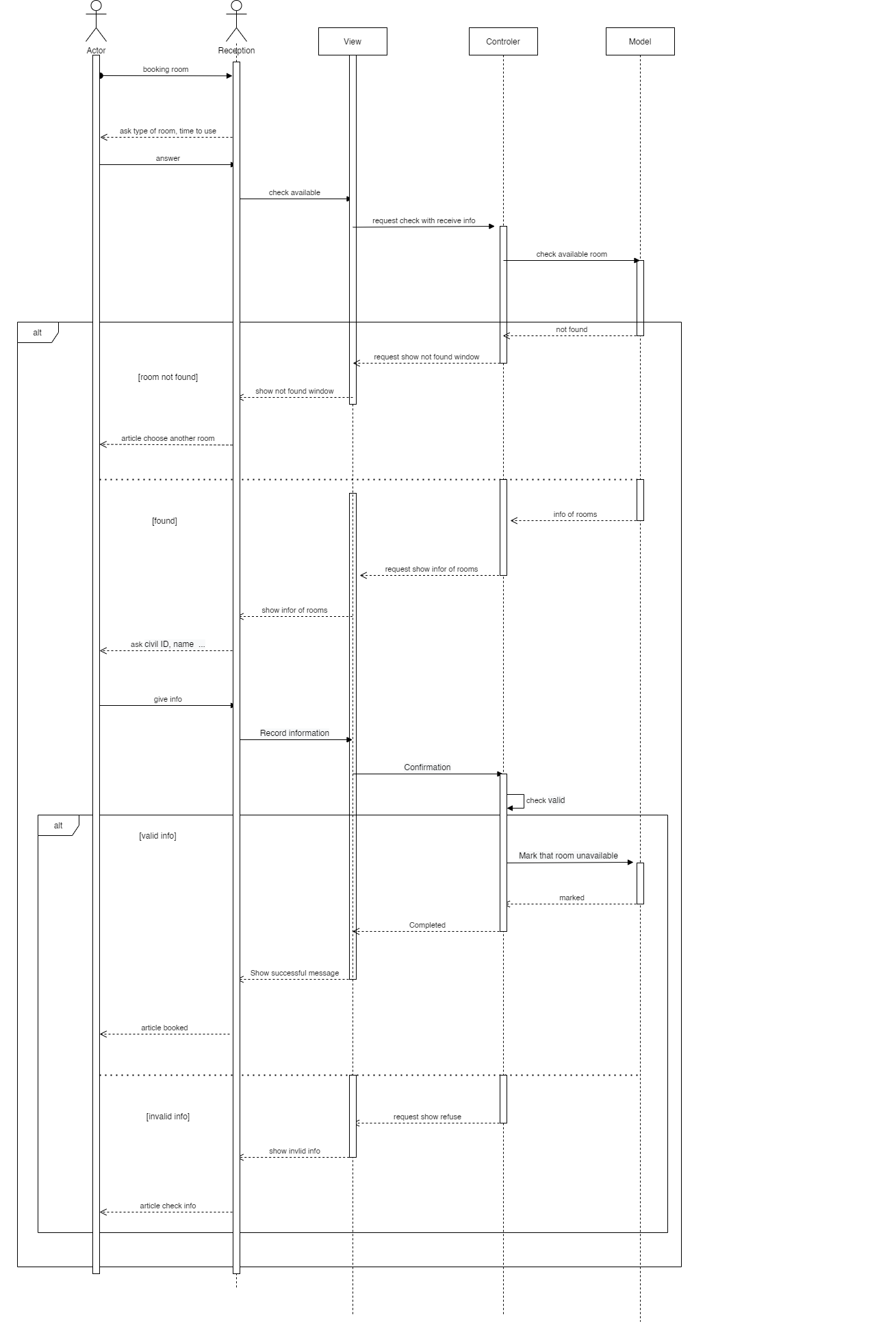
## System states



2. System's state diagram

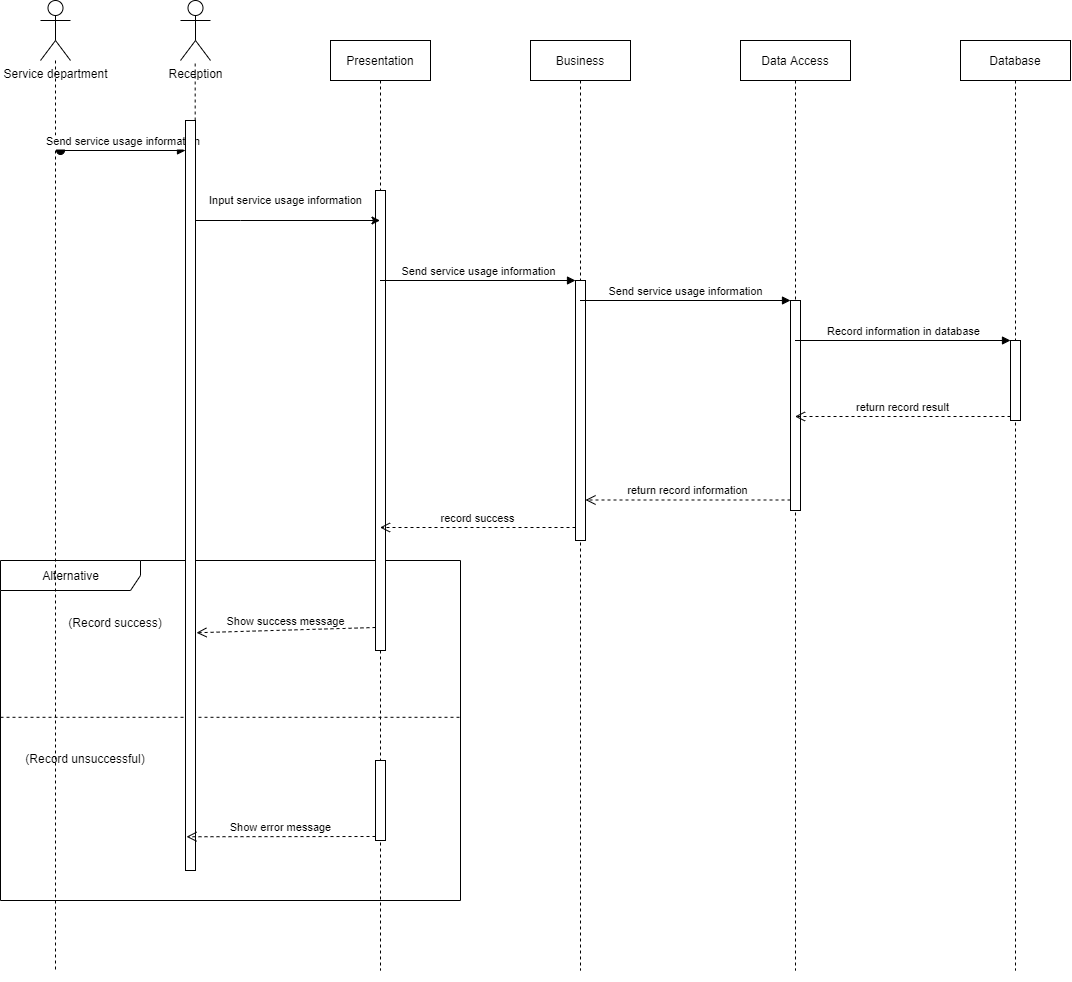
## Sequence diagram

### Booking



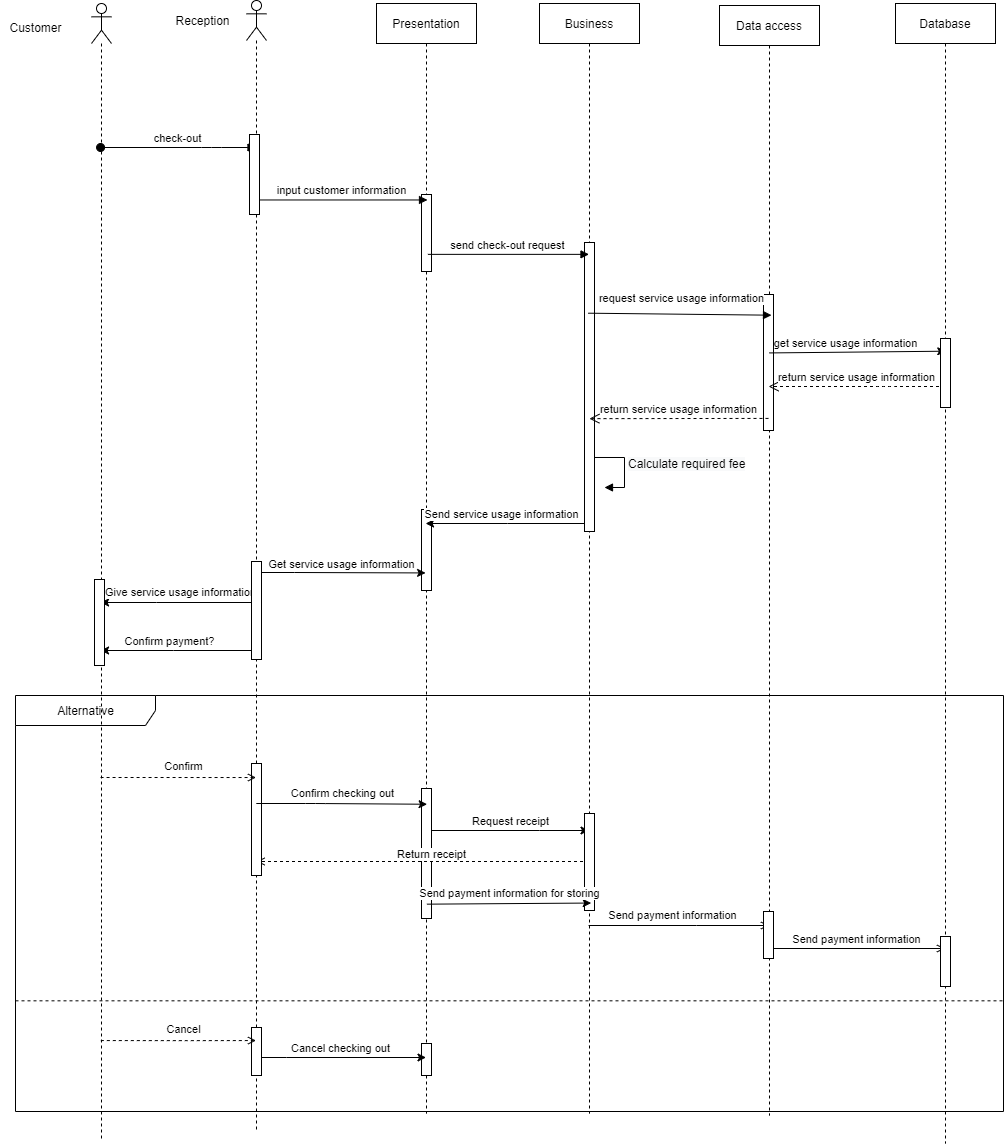
3. Booking sequence

### Record service usage



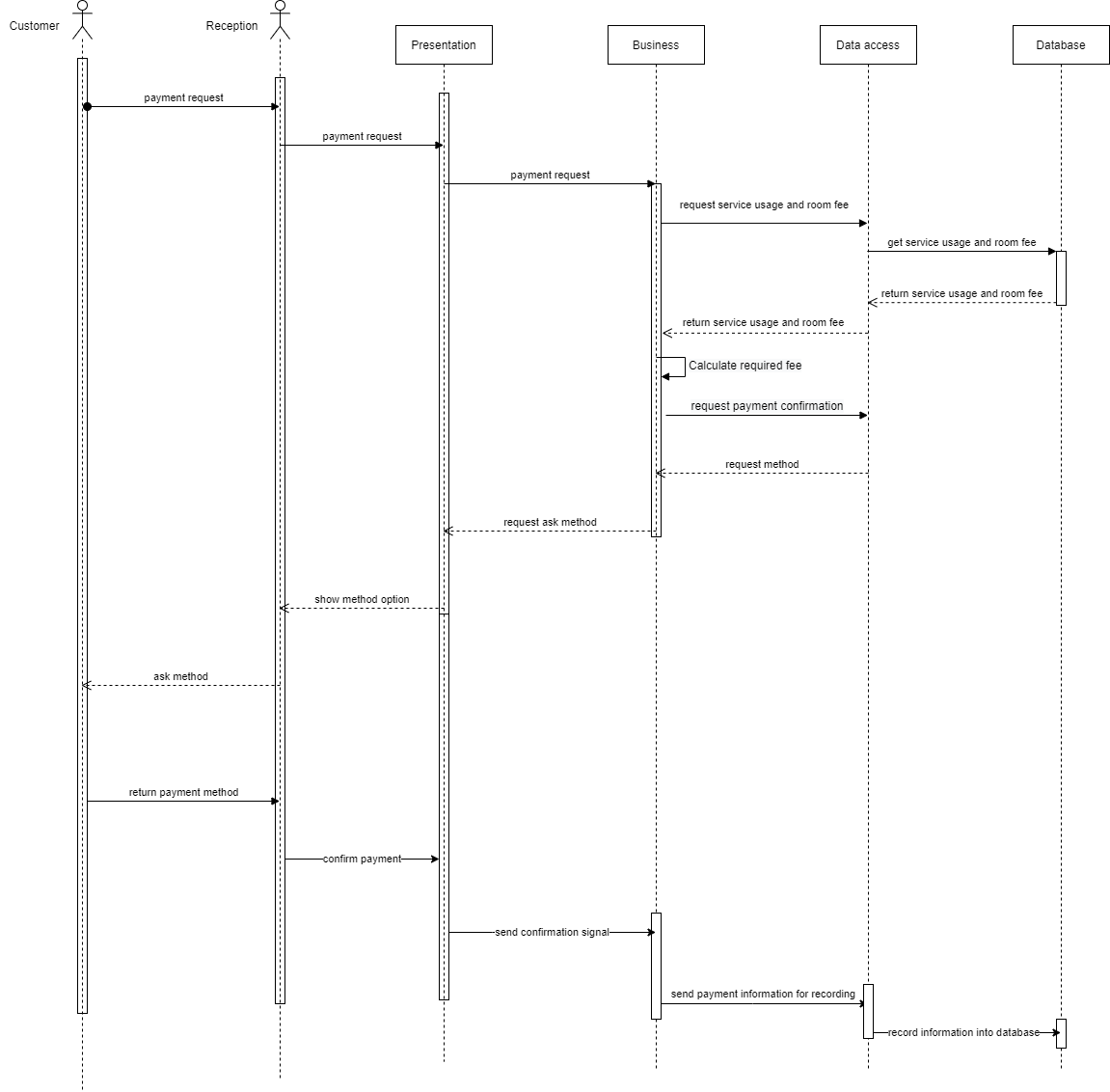
4. Record service usage sequence

### Check-out



5. Checkout sequence

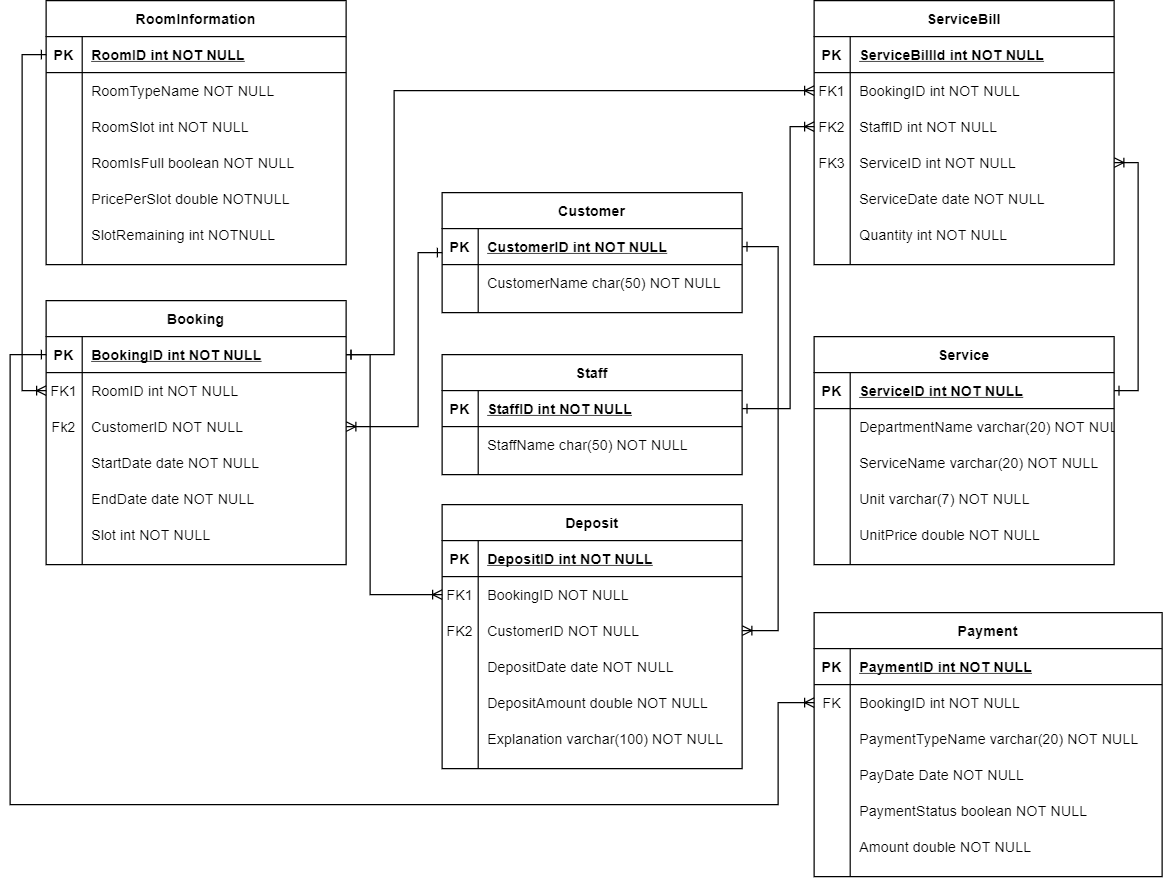
### Payment



6. Payment sequence

# Database design

## ERDS



7. Entity relationship diagram

|  |  |  |
| --- | --- | --- |
| No | Table | Description |
| 1 | RoomInformation | Information for each room of the hotel |
| 2 | Booking | Booking’s information |
| 3 | Customer | Customer’s information |
| 4 | Staff | Hotel staff’s information |
| 5 | Deposit | Store deposit information |
| 6 | ServiceBill | Store service bill’s detail |
| 7 | Service | Hotel’s service detail |
| 8 | Payment | Payment’s detail |

## RoomInformation

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field name | Type | Length | Size  (bytes) | nullable | Unique | P/F key | Default | Description |
| 1 | RoomTypeName | varchar | 20 | Variable | No | Yes | P key | None | Hotel room’s type |
| 2 | RoomSlot | int | N/A | 4 | No | No | N/A | None | Hotel room’s slot |
| 3 | RoomIsFull | boolean | N/A | 1 | No | No | N/A | None | Hotel room’ status |
| 4 | PricePerSlot | double | N/A | 8 | No | No | N/A | None | Price for each room slot |
| 5 | SlotRemaining | int | N/A | 4 | No | No | N/A | None | Remaining slot in each room |

## Booking

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field name | Type | Length | Size  (bytes) | nullable | Unique | P/F key | Default | Description |
| 1 | BookingID | int | N/A | 4 | No | Yes | P key | None | Booking identifier |
| 2 | RoomID | int | N/A | 4 | No | No | F key | None | Hotel room booked |
| 3 | CustomerID | int | N/A | 4 | No | No | F key | None | Customer booking the room |
| 4 | StartDate | date | N/A | 3 | No | No | N/A | None | Rent start date |
| 5 | EndDate | date | N/A | 3 | No | No | N/A | None | Rent end date |
| 6 | Slot | int | N/A | 4 | No | No | N/A | None | Slot booked |

## Customer

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field name | Type | Length | Size  (bytes) | nullable | Unique | P/F key | Default | Description |
| 1 | CustomerId | int | N/A | 4 | No | Yes | P key | None | Customer identifier |
| 2 | CustomerName | varchar | 50 | variable | No | No | N/A | None | Customer’s name |

## Staff

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field name | Type | Length | Size  (bytes) | nullable | Unique | P/F key | Default | Description |
| 1 | StaffID | int | N/A | 4 | No | Yes | P key | None | Hotel staff identifier |
| 2 | StaffName | varchar | N/A | Variable | No | No | N/A | None | Hotel staff’s name |

## Deposit

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field name | Type | Length | Size  (bytes) | nullable | Unique | P/F key | Default | Description |
| 1 | DepositID | int | N/A | 4 | No | Yes | P key | None | Deposit identifier |
| 2 | BookingID | int | N/A | 4 | No | No | F key | None | Booking to which the deposit belongs |
| 3 | CustomerID | int | N/A | 4 | No | No | F key | None | Customer to which the deposit belongs |
| 4 | DepositDate | date | N/A | 3 | No | No | N/A | None | Deposit date |
| 5 | Explanation | varchar | 100 | Variable | No | No | N/A | None | Explanation for the deposit |

## ServiceBill

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field name | Type | Length | Size  (bytes) | nullable | Unique | P/F key | Default | Description |
| 1 | ServiceBillID | int | N/A | 4 | No | Yes | P key | None | Service bill identifier |
| 2 | BookingID | int | N/A | 4 | No | No | F key | None | Booking to which the service belongs |
| 3 | StaffID | int | N/A | 4 | No | No | F key | None | ID of staff in charge of the service |
| 4 | ServiceID | int | N/A | 4 | No | No | F key | None | ID of service used |
| 5 | ServiceDate | date | N/A | 3 | No | No | N/A | None | Date of which the service is used |
| 5 | Quantity | int | N/A | 4 | No | No | N/A | None | Quantity of used service |

## Service

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field name | Type | Length | Size  (bytes) | Null | Unique | P/F key | Default | Description |
| 1 | ServiceId | int | N/A | 4 | No | Yes | P key | None | Service identifier |
| 2 | DepartmentName | varchar | 20 | Variable | No | No | N/A | None | Department to which the service belongs |
| 3 | ServiceName | varchar | 20 | Variable | No | No | N/A | None | Service’s name |
| 4 | Unit | varchar | 7 | Variable | No | No | N/A | None | Unit of the service |
| 5 | UnitPrice | double | N/A | 8 | No | No | N/A | None | Price for each unit of the service |

## Payment

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field name | Type | Length | Size  (bytes) | Null | Unique | P/F key | Default | Description |
| 1 | PaymentID | int | N/A | 4 | No | Yes | P key | None | Payment identifier |
| 2 | BookingID | int | N/A | 4 | No | No | F key | None | Booking to which payment belongs |
| 3 | PaymentTypeName | varchar | N/A | 20 | No | No | N/A | None | Payment type |
| 4 | PayDate | date | N/A | 3 | No | No | N/A | None | Date the payment is paid |
| 5 | PaymentStatus | boolean | N/A | 1 | No | No | N/A | None | Payment’ status |
| 6 | Amount | double | N/A | 8 | No | No | N/A | None | Payment amount |