

Software Design Specification

Hotel Management System

Prepared by Aleem Maknojia, Ali Asar Khowaja,
Bilal Pervez & Hammad Suleman Khan

SZABIST University

4th January, 2023

Software Design Specification

1. Introduction	4
Purpose of this document	4
Scope of the development project	4
Definitions, acronyms, and abbreviations	4
References	4
Overview of document	4
2. System architecture description	5
Section Overview	5
General Constraints	5
Data Design	5
Program Structure	8
Alternatives Considered	8
3. Detailed description of components	9
Section Overview	9
Component and Detail	9
4. User Interface Design	13
Section Overview	13
Interface Design Rules	13
Detailed Description	13
5. Appendices	20
Class Diagram	20
Object Diagram	21
Statechart Diagram	22
Activity Diagram	23
Login	23
Generate Reports	24
Update Hotel Information	25
Add Employee	26
Update Employee	27
Add Room	28
Update Room	29
Add Booking	30
Update Booking	31
Check-In	32
Check-Out	33

Request Services	34
Sequence	35
Login	35
Generate Report	35
Update Hotel Information	36
Add Employee	36
Update Employee	37
Add Room	37
Update Room	38
Add Booking	39
Update Booking	39
Check-in	40
Check-Out	40
Request Services	41
Collaboration	42
Login	42
Generate Reports	42
Update Hotel Information	43
Add Employee	43
Update Employee	44
Add Room	44
Update Room	45
Add Booking	45
Update Booking	46
Check-in	46
Check-Out	47
Request Services	47
Component Diagram	48
Deployment diagram	49
Use-Case Diagram	50

1. Introduction

Purpose of this document

This is our project about HOTEL MANAGEMENT SYSTEM to manage bookings, employees, rooms and hotels efficiently in order to provide the best service. The design of our project is based on SAAS applications where we provide Software as Service. Firstly, we are the super admins of this application and the hotel owners can get our application on a subscription based model. We will be registering the hotels in a database and authenticating them to use this application.

Scope of the development project

The motivation behind the idea is that nowadays the world is getting digitized so in order to remove all manual and paper work we have created a system which would help us to manage data more accurately and efficiently. Moreover, this would give an ability to keep records of guests and employees which could be further used for analytics.

We have provided full rights to the user where he can add, update, search and delete information when required. Best database design is used in order to increase efficiency of managing data from various hotels.

This application is very useful for a long run. This can help hotels manage their staff and guests with ease just by a single click as this provides functionality of generating reports. Moreover, data is managed efficiently and accurately which would help users to fetch everything easily.

Definitions, acronyms, and abbreviations

SDS: Software Design Specification
HMS: Hotel Management System
GUNA: DDL File for UI implementation

References

- Database System Concepts - Abraham Silberschatz, Henry F. Korth, S. Sudarshan
- Database Systems - Ramez Elmasri , Shamkant B. Navathe
- <https://www.javatpoint.com/software-requirement-specifications>
- <https://www.geeksforgeeks.org/software-requirement-specification-srs-format/>

Overview of document

A short description of how the rest of the SDS is organized and what can be found in the rest of the document. This is not simply a table of contents. Motivate and briefly describe the various parts!

2. System architecture description

Section Overview

In this section we have provided a few general constraints based on the hardware used, the programming language, the platform and the environment constraints. We have also provided the database and application structure. Other than that we have also added the alternative structures we considered,

General Constraints

As per client's request the system has to be designed in Java. It has to have the flexibility of running on multiple platforms however, it will specifically run on a windows system in our case. Interface had to be designed that follows easy and common conventions so that its usability could be increased. The software must run locally i.e. on the local area network of the hotel.

The application server should be able to run on the following system:

Dell PowerEdge T30

CPU: Intel Xeon E3-1225; v5

Graphics: Intel HD Graphics P530;

RAM: 64GB

Storage: Up to six SATA HDD

Connectivity: 10 x USB ports, HDMI, DisplayPort, VGA, serial, PS2, Gigabit Ethernet

Dimensions (W x D x H): 17.5 x 43.5 x 36cm

And for network constraints, the following devices should be considered:

MikroTik hEX S – Network Router.

Netgate SG-1100 – Network Firewall Device.

Firewalla Blue Plus – Cybersecurity Device.

Data Design

Describe the structure of any databases, external files, and internal data structures. You may wish to include references to appendices containing ER diagrams, data, or file formats.

Tables:

- **Authentication**

AuthId INT IDENTITY (1,1),

Username NVARCHAR (30) UNIQUE NOT NULL,

Password NVARCHAR (30) NOT NULL,

CONSTRAINT PK_AuthId PRIMARY KEY (AuthId)

- **Login**

LoginId INT IDENTITY (1,1),
Username NVARCHAR (30) NOT NULL UNIQUE,
Password NVARCHAR (30) NOT NULL,
EmployeeId INT,
SecurityQuestion NVARCHAR(100),
Answer NVARCHAR(50),
NewUser CHAR(5) CHECK (NewUser IN ('Yes', 'No')) DEFAULT 'Yes',
CONSTRAINT PK_LoginId PRIMARY KEY (LoginId),
CONSTRAINT FK_EmployeeId_Login FOREIGN KEY (EmployeeId)
REFERENCES Hotels.Employees (EmployeeId) ON DELETE CASCADE ON
UPDATE NO ACTION

- **RoomType**

RoomTypeId INT IDENTITY (1,1),
Name NVARCHAR (50) NOT NULL,
Cost INT NOT NULL,
SmokeFriendly NVARCHAR (5) NOT NULL,
PetFriendly NVARCHAR (5) NOT NULL,
CONSTRAINT PK_RoomTypeId PRIMARY KEY (RoomTypeId)

- **Room**

RoomId INT IDENTITY (1,1),
RoomNumber NVARCHAR (10) NOT NULL,
RoomTypeId INT,
Available NVARCHAR (5) NOT NULL CHECK (Available IN ('Yes', 'No'))
DEFAULT 'Yes',
CONSTRAINT PK_RoomId PRIMARY KEY (RoomId),
CONSTRAINT FK_RoomTypeId_Room FOREIGN KEY (RoomTypeId)
REFERENCES Rooms.RoomType (RoomTypeId) ON DELETE CASCADE ON
UPDATE NO ACTION

- **Employee**

EmployeeId INT IDENTITY (1,1),
EmployeeFirstName NVARCHAR (50) NOT NULL,
EmployeeLastName NVARCHAR (50) NOT NULL,
EmployeeDesignation NVARCHAR (50) NOT NULL,
EmployeeContactNumber NVARCHAR (15) NOT NULL,
EmployeeEmailAddress NVARCHAR(50) NOT NULL UNIQUE,
EmployeeJoiningDate DATE NOT NULL,
AddressLine NVARCHAR(50) NOT NULL,
Street NVARCHAR(50) NOT NULL,
City NVARCHAR(20) NOT NULL,
Zip NVARCHAR(50) NOT NULL,
CNIC NVARCHAR(20) NOT NULL UNIQUE,
CONSTRAINT PK_EmployeeId PRIMARY KEY (EmployeeId),

- **Booking**

BookingId INT IDENTITY (1,1),
BookingDate DATE NOT NULL,
StayDuration INT NOT NULL,
CheckInDate DATE NOT NULL,
CheckOutDate DATE NOT NULL,
BookingAmount INT NOT NULL,
EmployeeId INT NOT NULL,
GuestId INT NOT NULL,
Status NVARCHAR (10) NOT NULL CHECK (Status IN ('Checkin',
'Checkout')) DEFAULT 'Checkin',
CONSTRAINT PK_BookingId PRIMARY KEY (BookingId),
CONSTRAINT FK_EmployeeId_Booking FOREIGN KEY (EmployeeId)
REFERENCES Hotels.Employees (EmployeeId) ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT FK_GuestId_Booking FOREIGN KEY (GuestId)
REFERENCES Hotels.Guests(GuestId) ON DELETE NO ACTION ON
UPDATE NO ACTION

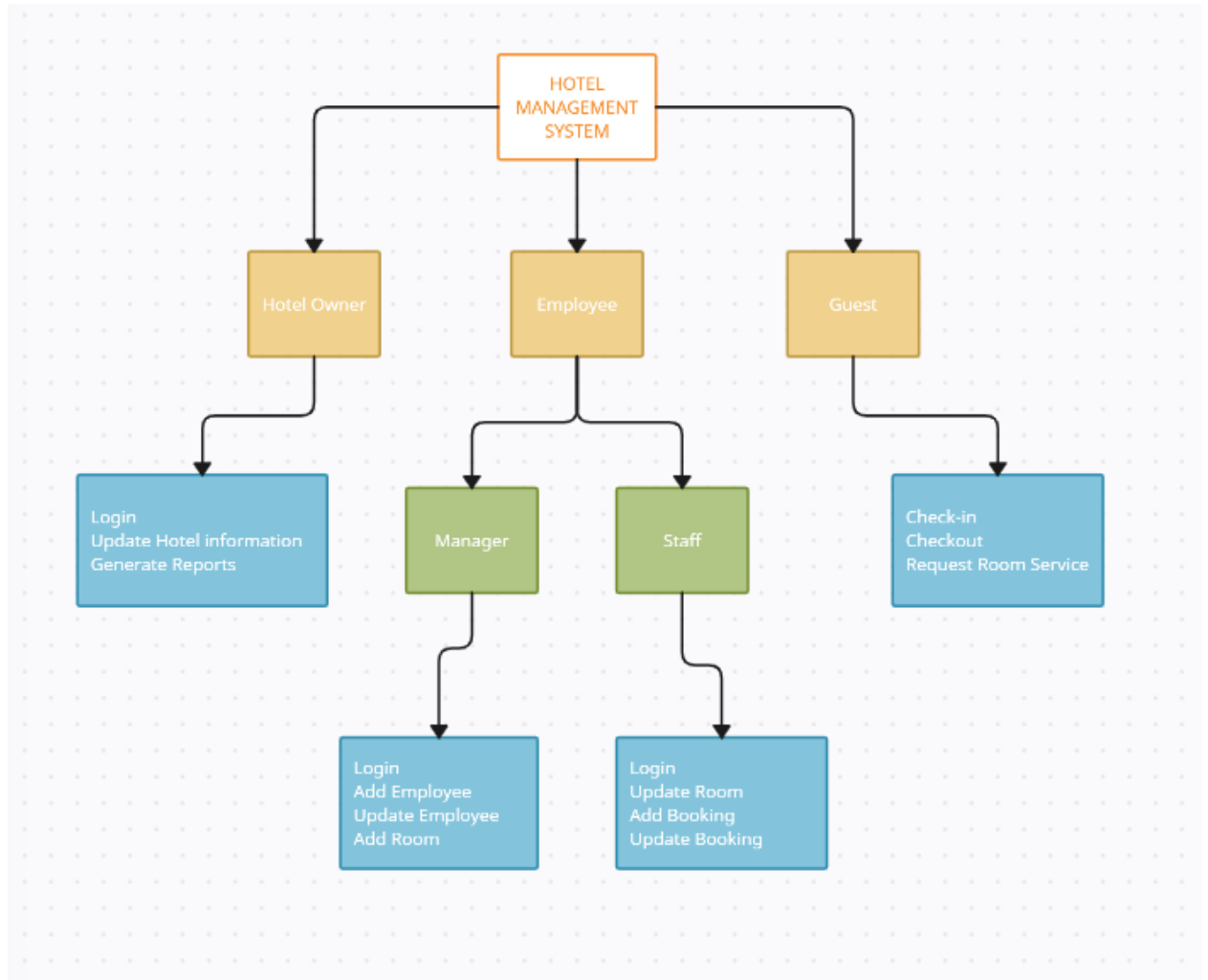
- **Payments**

PaymentId INT IDENTITY (1,1),
PaymentStatus NVARCHAR (6) NOT NULL,
PaymentType NVARCHAR(50) NOT NULL,
PaymentAmount INT NOT NULL,
BookingId INT NOT NULL,
CONSTRAINT PK_PaymentId PRIMARY KEY (PaymentId),
CONSTRAINT FK_BookingId_Payments FOREIGN KEY(BookingId)
REFERENCES Bookings.Booking(BookingId) ON DELETE CASCADE ON
UPDATE CASCADE

- **Guests**

GuestId INT IDENTITY (1,1),
GuestFirstName NVARCHAR (50) NOT NULL,
GuestLastName NVARCHAR (50) NOT NULL,
GuestEmailAddress NVARCHAR(50) NOT NULL ,
GuestContactNumber NVARCHAR (15) NOT NULL,
GuestPassportNumber NVARCHAR(20) NOT NULL,
GuestCnic NVARCHAR (30) NOT NULL,
AddressLine NVARCHAR(50) NOT NULL,
Street NVARCHAR(50) NOT NULL,
City NVARCHAR(20) NOT NULL,
Zip NVARCHAR(50) NOT NULL,
Status NVARCHAR(20) CHECK (Status IN ('Reserved', 'Not Reserved'))
NOT NULL DEFAULT 'Not Reserved',
CONSTRAINT PK_GuestId PRIMARY KEY (GuestId),

Program Structure



Alternatives Considered

We considered a NoSQL database for our system but it soon became apparent that it would not be a good choice to go that route as the relations are not strict enough and we cannot define well constrained data models in that route.

3. Detailed description of components

Section Overview

This section has three components of the many in the system. The purpose of showing the components is to give an understanding of how different modules and apis are being used to provide functionality as a whole to the system; how the inputs are being manipulated to produce outputs or to enable use of different features.

Component and Detail

1. Login.cs
2. Employee.cs
3. Report Generation API

Identification	Login.cs
Type	class
Purpose	Login will be used to authenticate users and generate a unique token that will be used throughout the whole processing until the user is logged out.
Function	Login will take user id and password. It will check it against the database records. It then checks what kind of authentication to be provided depending on the type of user e.g. if the employee is a manager it will generate a specific token with additional privileges. In every case a token will be generated that will be used throughout the duration of use until the user logs out. Logs will be updated to keep the record of login activity.
Subordinates	The constituents or attributes of login are login id, username and password. For the first time there will be a security question and its answer. Initially the values of security question and answer are null in the database. When the user logs in for the first time these security question and answer fields will pop up to be filled to proceed to the dashboard once these fields have been filled and the database is updated.
Processing	<ul style="list-style-type: none">- Start- Show login page- Enter id and password- If new user then select secret question/answer- Update database- Else If user id/ password wrong prompt incorrect entered- Else take to dashboard
Data	<p>Most of the fields are set to empty initially. The use, semantics and format is better understood by looking at the database table that stores the login authentication data below.</p> <p>LoginId INT IDENTITY (1,1), Username NVARCHAR (30) NOT NULL UNIQUE,</p>

	Password NVARCHAR (30) NOT NULL, EmployeeId INT, SecurityQuestion NVARCHAR(100), Answer NVARCHAR(50),
--	--

Identification	employees.cs
Type	class
Purpose	Provides a UI to the manager to be able to manage employees
Function	This component requires the login component and allows the manager to <ul style="list-style-type: none"> - Add Employees - Remove Employees - Update Employees - View Employees
Subordinates	The DELETE sql query is used to satisfy the delete function The INSERT sql query is used to satisfy the add function The UPDATE sql query is used to satisfy the update function The SELECT sql query is used to satisfy the update function
Processing	<ul style="list-style-type: none"> - Add Employees: Runs the INSERT query by using the parameters provided by the manager in a ui form - Remove Employees Runs the DELETE query by using the parameters provided by the manager in a ui form - Update Employees Runs the UPDATE query by using the parameters provided by the manager in a ui form - View Employees Runs the SELECT query by using the filters provided by the manager
Data	Most of the data cannot be null. The use, semantics and format is better understood by looking at the database table that stores the login authentication data below. EmployeeId INT IDENTITY (1,1), EmployeeFirstName NVARCHAR (50) NOT NULL, EmployeeLastName NVARCHAR (50) NOT NULL, EmployeeDesignation NVARCHAR (50) NOT NULL, EmployeeContactNumber NVARCHAR (15) NOT NULL, EmployeeEmailAddress NVARCHAR(50) NOT NULL UNIQUE, EmployeeJoiningDate DATE NOT NULL, AddressLine NVARCHAR(50) NOT NULL, Street NVARCHAR(50) NOT NULL, City NVARCHAR(20) NOT NULL, Zip NVARCHAR(50) NOT NULL, DepartmentId INT, HotelId INT NOT NULL, CNIC NVARCHAR(20) NOT NULL UNIQUE,

Identification	Generate report Api
Type	API
Purpose	This will provide a brief report on all the transactions done within the hotel premises which includes booking payments, etc.
Function	This would be using three joins query to fetch data where sub-queries will be used in order to fetch data from the database. This would show the bookings with id, employee id who did the booking, payment, booking date, and room id for every guest.
Subordinates	The database will have tables such as Payment, Booking, Employee and Guest which would contain relevant information. Foreign keys and table primary keys would be used to fetch useful information to generate reports. Three joins would be used with two full join and one inner join with subquery to fetch info and then the data would be converted to excel sheets which would help users to perform more complex calculations.
Processing	<p>Here is the query:</p> <pre> SELECT Bookings.Booking.BookingId, Bookings.Booking.BookingDate, Bookings.Booking.StayDuration, Bookings.Booking.CheckInDate, Bookings.Booking.CheckOutDate, Bookings.Booking.Status, Hotels.Guests.GuestId, Hotels.Guests.GuestFirstName, Hotels.Guests.GuestLastName, Hotels.Guests.GuestContactNumber, Hotels.Guests.GuestPassportNumber, Hotels.Guests.HotelId, Hotels.Employees.EmployeeId, Hotels.Employees.EmployeeFirstName, Hotels.Employees.EmployeeLastName, Hotels.Employees.EmployeeContactNumber, Bookings.Payments.PaymentId, Bookings.Payments.PaymentStatus,Bookings.Payments.Payment Amount FROM Bookings.Booking INNER JOIN Hotels.Employees ON Bookings.Booking.EmployeeId = Hotels.Employees.EmployeeId FULL JOIN Hotels.Guests ON Bookings.Booking.GuestId = Hotels.Guests.GuestId FULL JOIN Bookings.Payments ON Bookings.Booking.BookingId = Bookings.Payments.BookingId WHERE Bookings.Booking.HotelId = @hotelId; </pre>
Data	<ol style="list-style-type: none"> 1. BookingId 2. BookingDate 3. BookingDuration 4. CheckInDate 5. CheckoutDate 6. Status 7. GuestId 8. GuestFirstName 9. GuestLastName 10. GuestContact

	<div>11. EmployeeId</div> <div>12. EmployeeName</div> <div>13. PaymentId</div> <div>14. PaymentAmount</div> <div>15. Payment Status</div> <div>After running the query these all information would be translated to excel sheet.</div>
--	--

4. User Interface Design

Section Overview

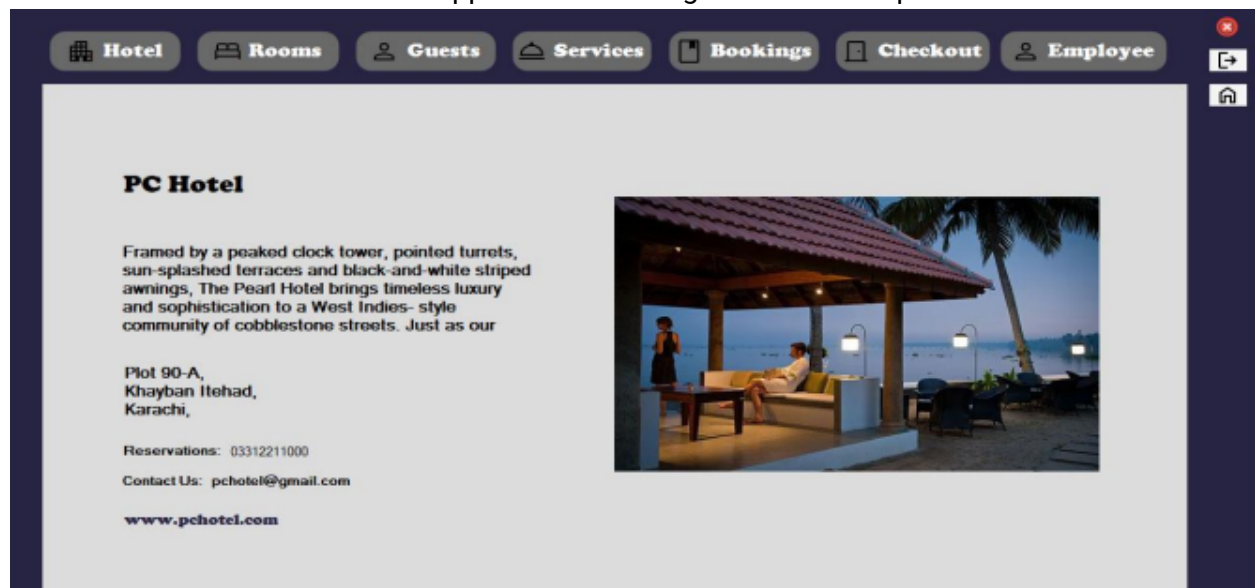
This section contains the GUI standards that we are using as well as the screen shots of the interface and brief explanation of what it is used for. This provides a detailed description of the flow of the application and how it will work. Apart from that we have designed the UI in a way that it is self explanatory and easy to use by the users.

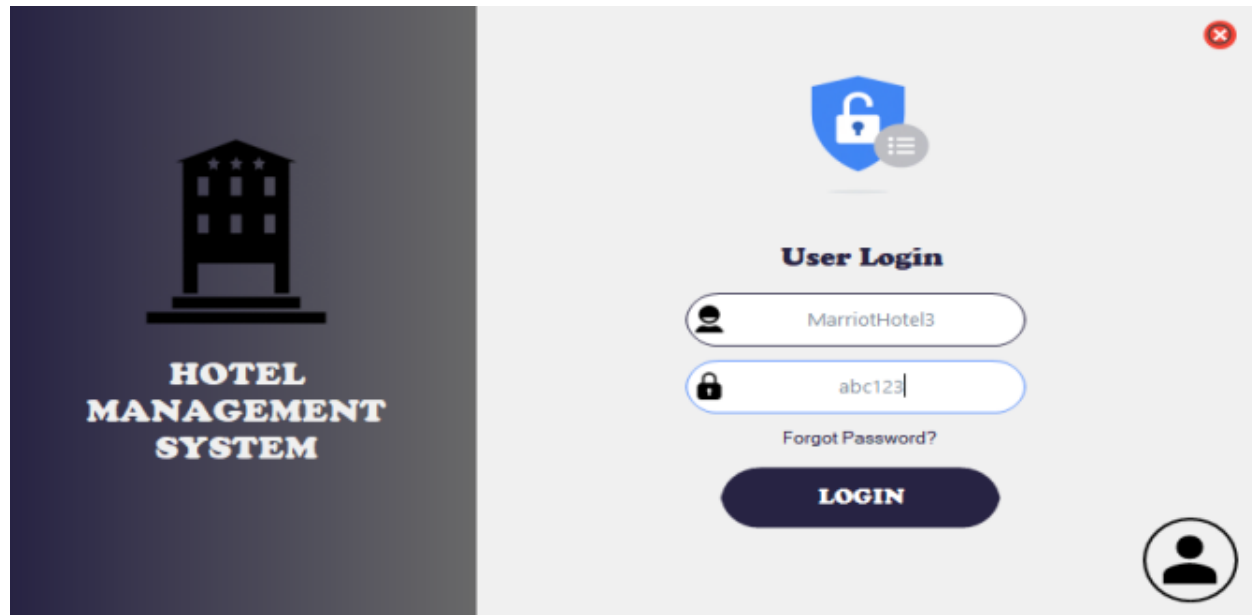
Interface Design Rules

We are using MS Windows GUI standard. Here every screen will have buttons, input text areas and some form of information display section like a data table. There will also be pop ups messages for confirmation or invalid input or response.

Detailed Description

Provide a detailed description of the user interface including screen images. You may prefer to reference an appendix containing the screen snapshots.





Multiple hotels can be accessed as we can see now Marriot Hotel user is accessing this application.

Hotel

Rooms

Guests

Services

Bookings

Checkout

Employee

Rooms

Add Room Types

RoomId	RoomNumber	HotelId	RoomTypeId	Available
2	2	3	3	Yes
3	3	3	1	Yes
4	4	3	1	Yes
5	5	3	3	Yes
6	9	3	2	Yes
7	7	3	2	Yes
8	8	3	2	Yes

Room Id

Room No

Available

Room Type

Room Cost

ADD

UPDATE

CLEAR

DELETE

Here we can add rooms and room type. CRUD operations can be applied here.

Room Types

ID	Name	Cost	SmokeFriendly	PetFriendly
1	Gold	10000	Yes	No
2	Silver	7000	No	No
3	Platinum	14000	Yes	Yes

Room Type Id

Room Type Name

Cost

Smoke Friendly

Pet Friendly

ADD

UPDATE

CLEAR

DELETE

Room types can be add where we can select options as provided.

Hotel

Rooms

Guests

Services

Bookings

Checkout

Employee

Rooms

Add Room Types

RoomId	RoomNumber	HotelId	RoomTypeId	Available
2	2	3	3	Yes
3	3	3	1	Yes
4	4	3	1	Yes
5	5	3	3	Yes
6	9	3	2	Yes
7	7	3	2	Yes
8	8	3	2	Yes

Room Id

Room No

Available

Room Type

Room Cost

ADD

UPDATE

CLEAR

DELETE

Here we can add rooms and room type. CRUD operations can be applied here.

Room Types

ID	Name	Cost	SmokeFriendly	PetFriendly
1	Gold	10000	Yes	No
2	Silver	7000	No	No
3	Platinum	14000	Yes	Yes

Room Type Id

Room Type Name

Cost

Smoke Friendly

Pet Friendly

ADD

UPDATE

CLEAR

DELETE

Room types can be add where we can select options as provided.

Hotel

Rooms

Guests

Services

Bookings

Checkout

Employee

Filter

Booking Id

Q

Check-in Date

2022-05-26

Services

☐ Boating
☐ Hot spa
☐ Massage

Promo Id

ADD

CLEAR

Guest Id

11

13

14

Room Type

Room Id

Total Amount

\$

UPDATE

DELETE

Bookings

ID	BookingDate	CheckInDate	CheckOutDate	GuestId	DiscountId	EmployeeId	Status
2032	17/06/2022	26/05/2022	28/05/2022	11	1	1	Checkout
2033	17/06/2022	17/06/2022	24/06/2022	14	2	1	Checkout
2034	17/06/2022	17/06/2022	20/06/2022	13	1	1	Checkout
2035	17/06/2022	26/05/2022	28/05/2022	14	1	1	Checkout
2038	17/06/2022	26/05/2022	26/05/2022	14		1	Checkout

Bookings can be added here which will show the total amount after all calculations.

Hotel

Rooms

Guests

Services

Bookings

Checkout

Employee

Filter

Booking Id

Q

Check-in Date

2022-05-26

Services

☐ Boating
☐ Hot spa
☒ Massage

Promo Id

1

ADD

CLEAR

Guest Id

13

Check-out Date

2022-05-30

Room Type

Silver

Room Id

6

Total Amount

27899

\$

UPDATE

DELETE

Bookings

ID	BookingDate	CheckInDate	CheckOutDate	GuestId	DiscountId	EmployeeId	Status
2032	17/06/2022	26/05/2022	28/05/2022	11	1	1	Checkout
2033	17/06/2022	17/06/2022	24/06/2022	14	2	1	Checkout
2034	17/06/2022	17/06/2022	20/06/2022	13	1	1	Checkout
2035	17/06/2022	26/05/2022	28/05/2022	14	1	1	Checkout
2038	17/06/2022	26/05/2022	26/05/2022	14		1	Checkout

Hotel

Rooms

Guests

Services

Bookings

Checkout

Employee

Filter

ID	BookingDate	CheckInDate	CheckOutDate	GuestId	DiscountId	EmployeeId	Status
2032	17/06/2022	26/05/2022	28/05/2022	11	1	1	Checkout
2033	17/06/2022	17/06/2022	24/06/2022	14	2	1	Checkout
2034	17/06/2022	17/06/2022	20/06/2022	13	1	1	Checkout
2035	17/06/2022	26/05/2022	28/05/2022	14			
2038	17/06/2022	26/05/2022	26/05/2022	14			

Success

Booking inserted successfully!

OK

Booking Id

Guest Id

13

Check-in Date

2022-05-26

Check-out Date

2022-05-30

Services

☐ Boating

☐ Hot spa

☒ Massage

Room Type

Silver

Room Id

6

Promo Id

1

Total Amount

27899

ADD

UPDATE

CLEAR

DELETE

Hotel

Rooms

Guests

Services

Bookings

Checkout

Employee

Checkout

ID	Status	TYPE	Amount	BookingId
7	Paid	Cash	32399	2032
8	Paid	Cash	42750	2033
9	Paid	Credit Card	28799	2034
10	Paid	Credit Card	15299	2035
13	Paid	Credit Card	7000	2038

Payment Id

Booking Id

3024

Payment Type

Cash

Status

Paid

Total Payable Amount

27899

Pay

CLEAR

This is checkout screen which will be shown to checkout guests.

Hotel

Rooms

Guests

Services

Bookings

Checkout

Employee

Employees

Add Departments

ID	FName	LName	CNIC	Designation	Contact	Email	JDate	DeptId	HotelId
1	Ali Asar	Khowaja	42101-3...	Manager	0331231...	ali@gma...	28/05/20...	3	3
2	Hammad	Khan	41202-3...	Staff	0331231...	hammad...	28/05/20...	2	3
4	Bilal	Saeed	42101-3...	Manager	0331221...	bilal@g...	26/05/20...	1	3

Employee Id

First Name

Email

CNIC Number

Address Line

Street

Designation

Last Name

Department Name

Zip Code

City

Contact Number

Joining Date

2022-05-25

CLEAR

Here we can add employees and if an employee is from HR, Admin, Accounts department a default username and password will be provided to him.

Departments

ID	Name	Description	InitialSalary
1	Admin	Administration.	20000
2	Food	Serves food,	20000
3	HR	Manages people.	25000

Department Id

Department Name

Initial Salary

Description

ADD

UPDATE

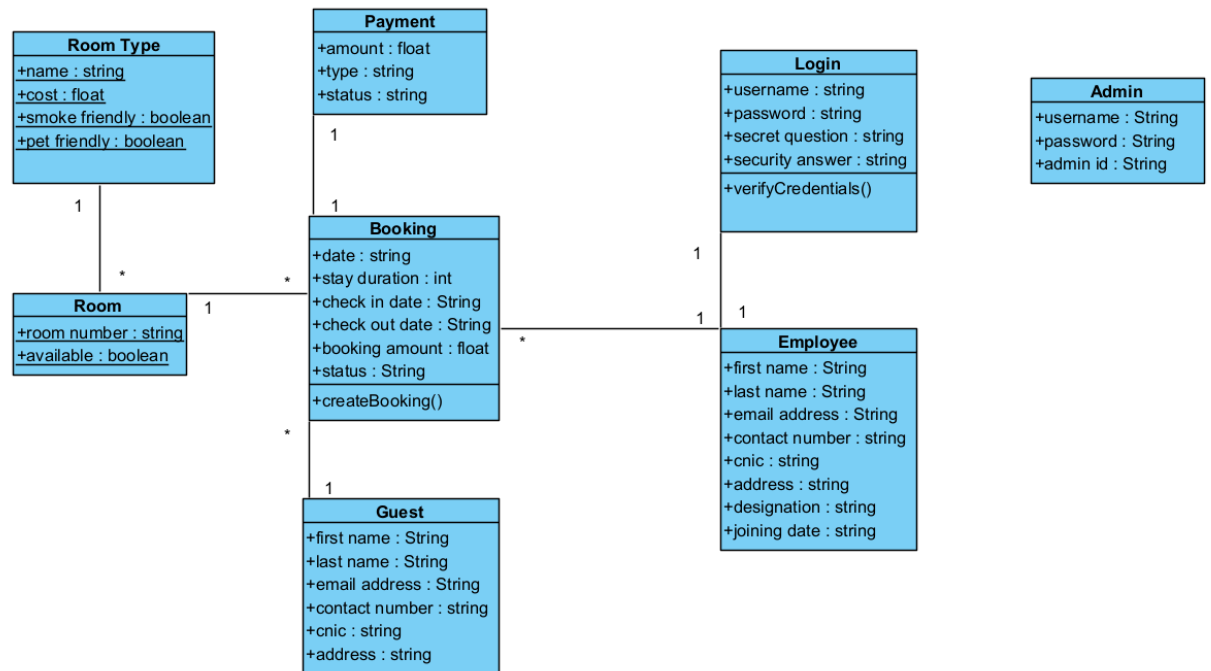
CLEAR

DELETE

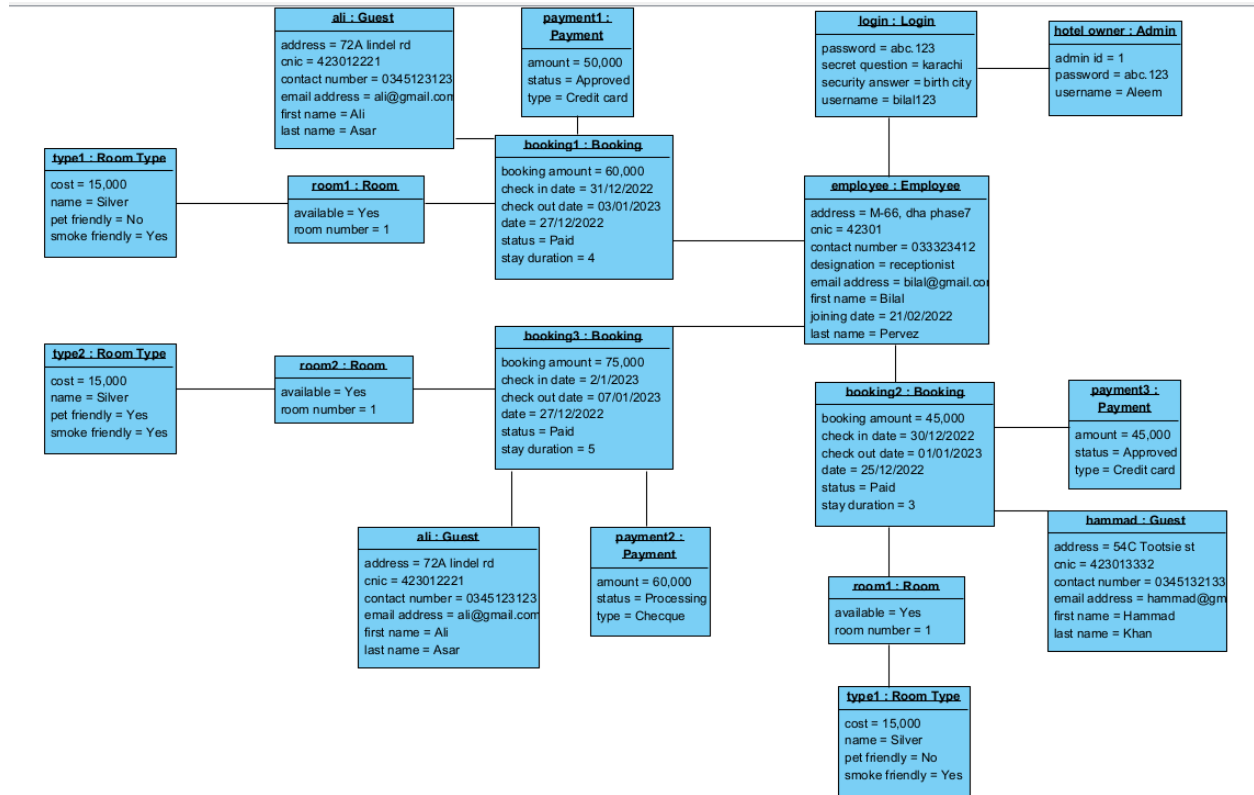
Here we can add departments. CRUD functionality is provided.

5. Appendices

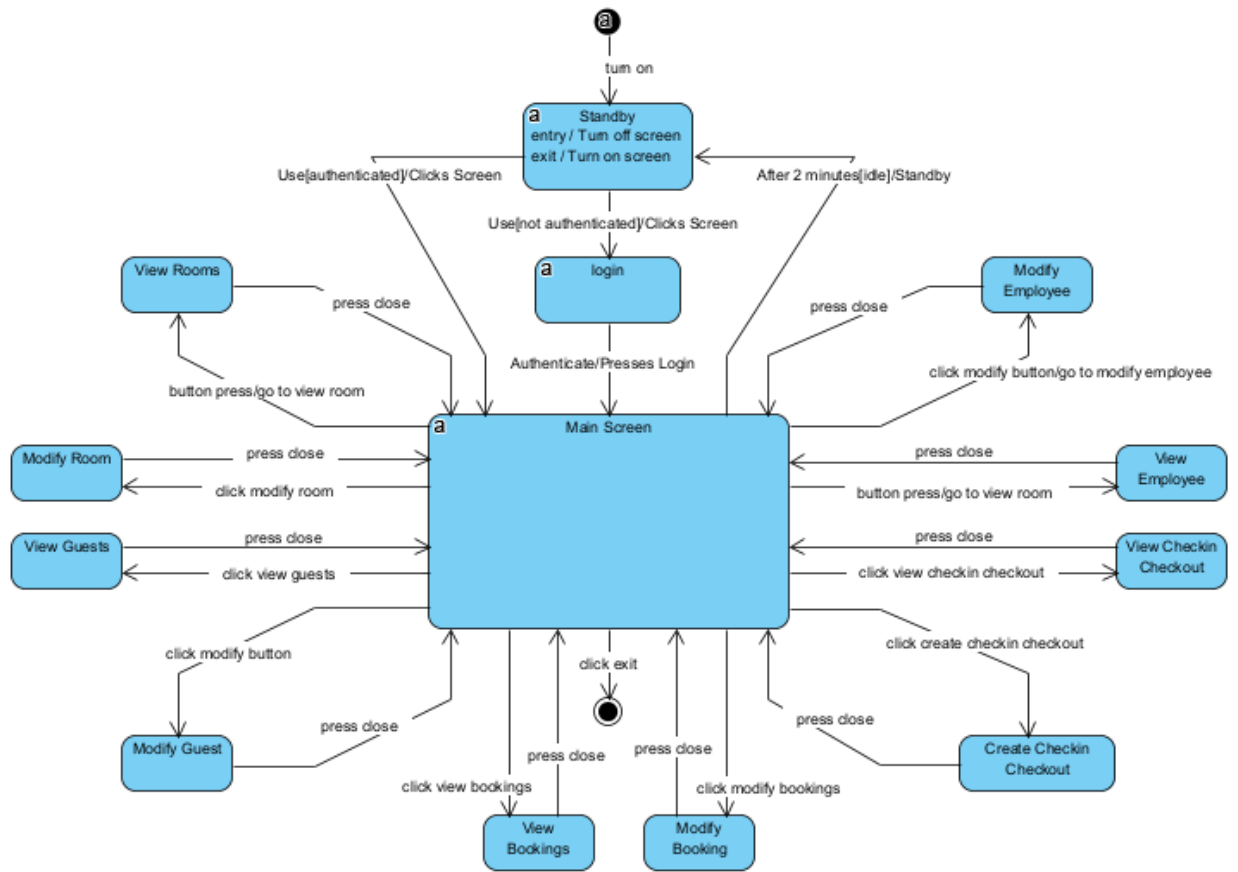
Class Diagram



Object Diagram

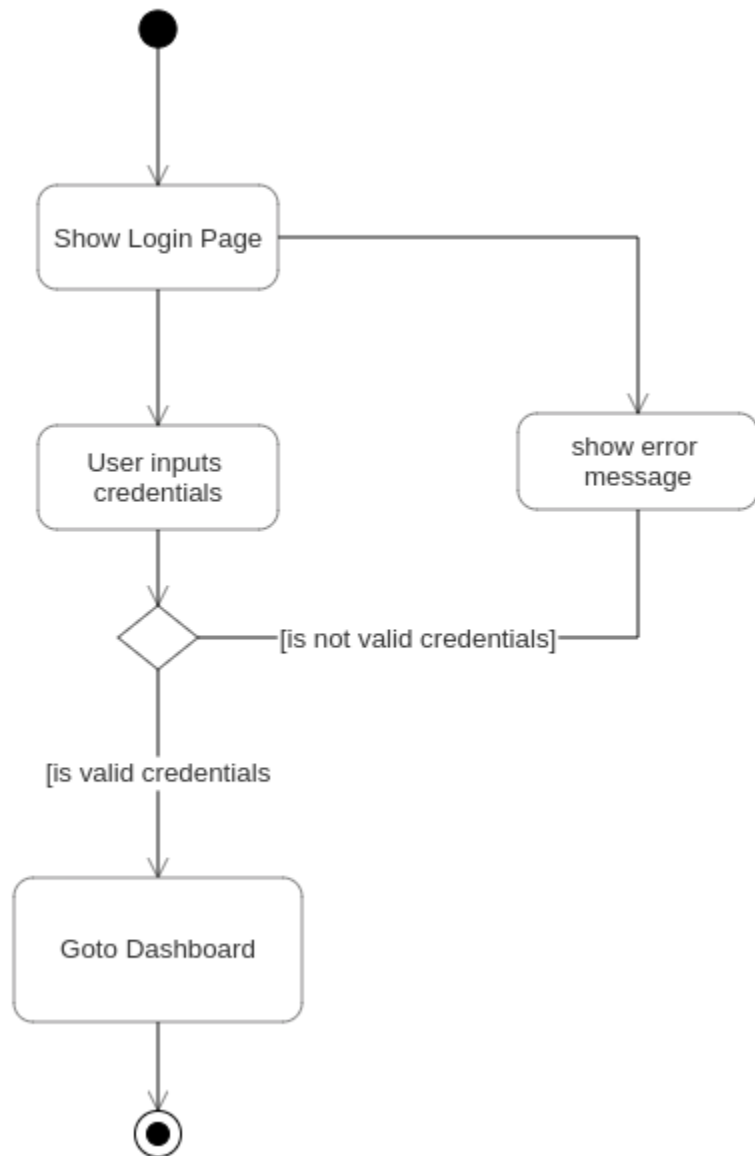


Statechart Diagram

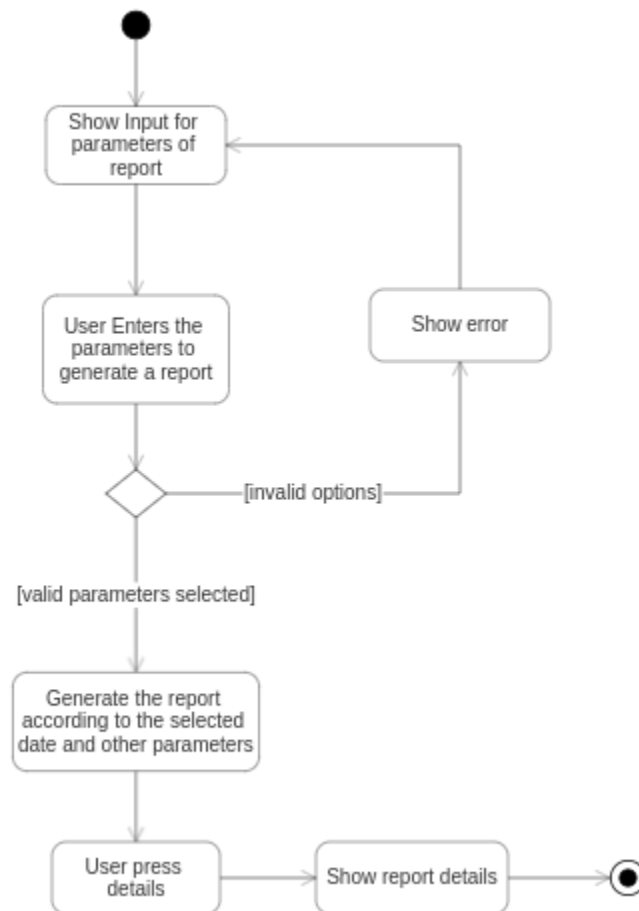


Activity Diagram

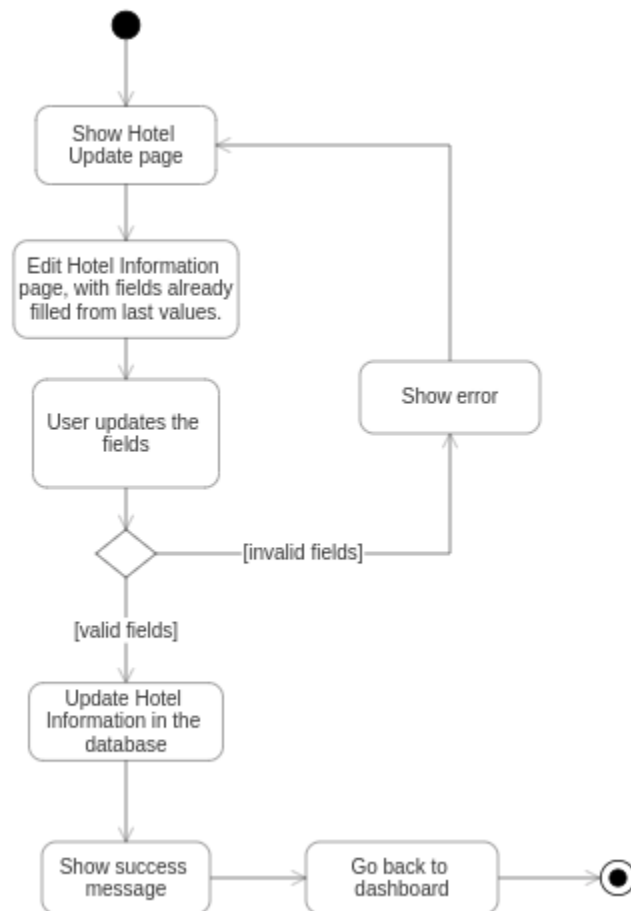
Login



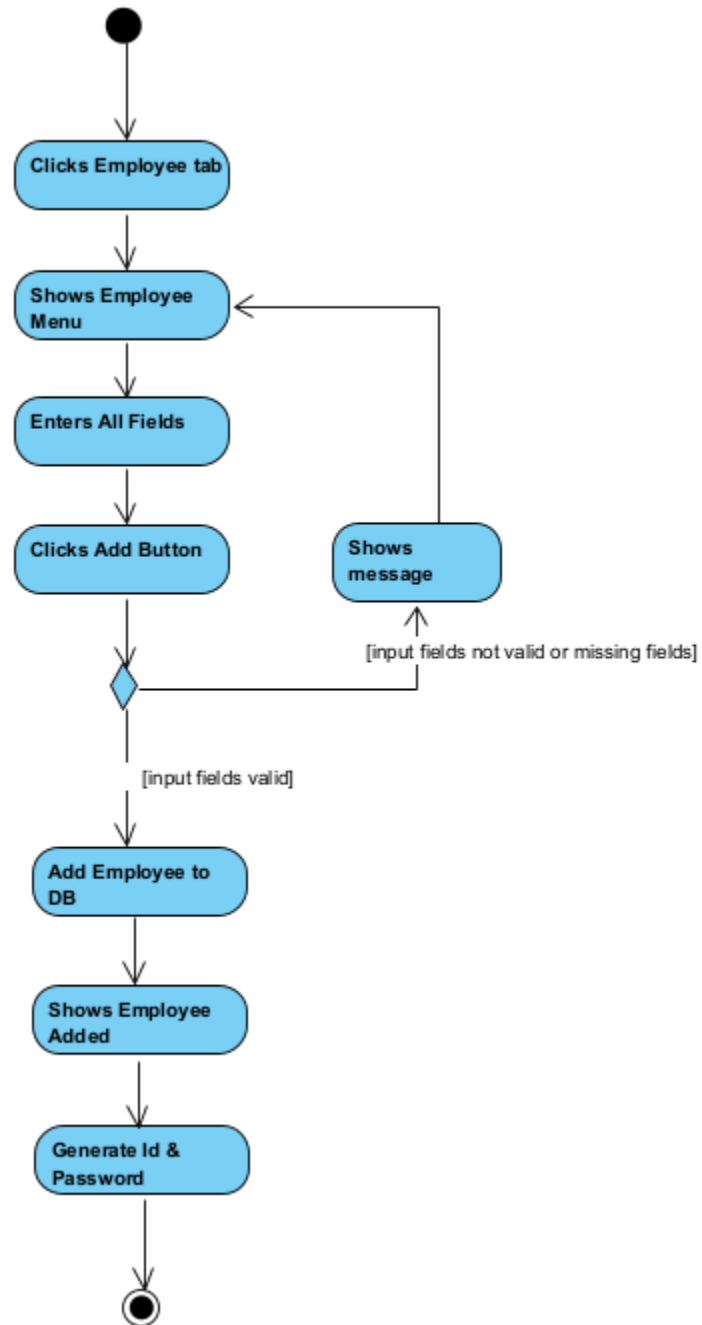
Generate Reports



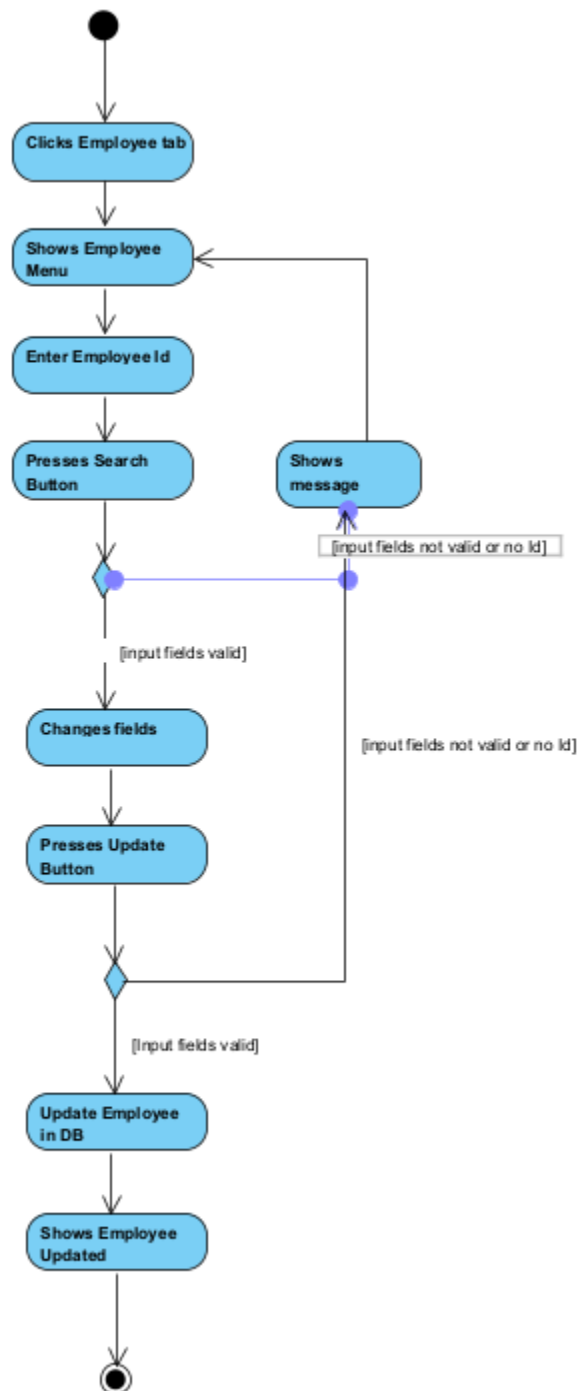
Update Hotel Information



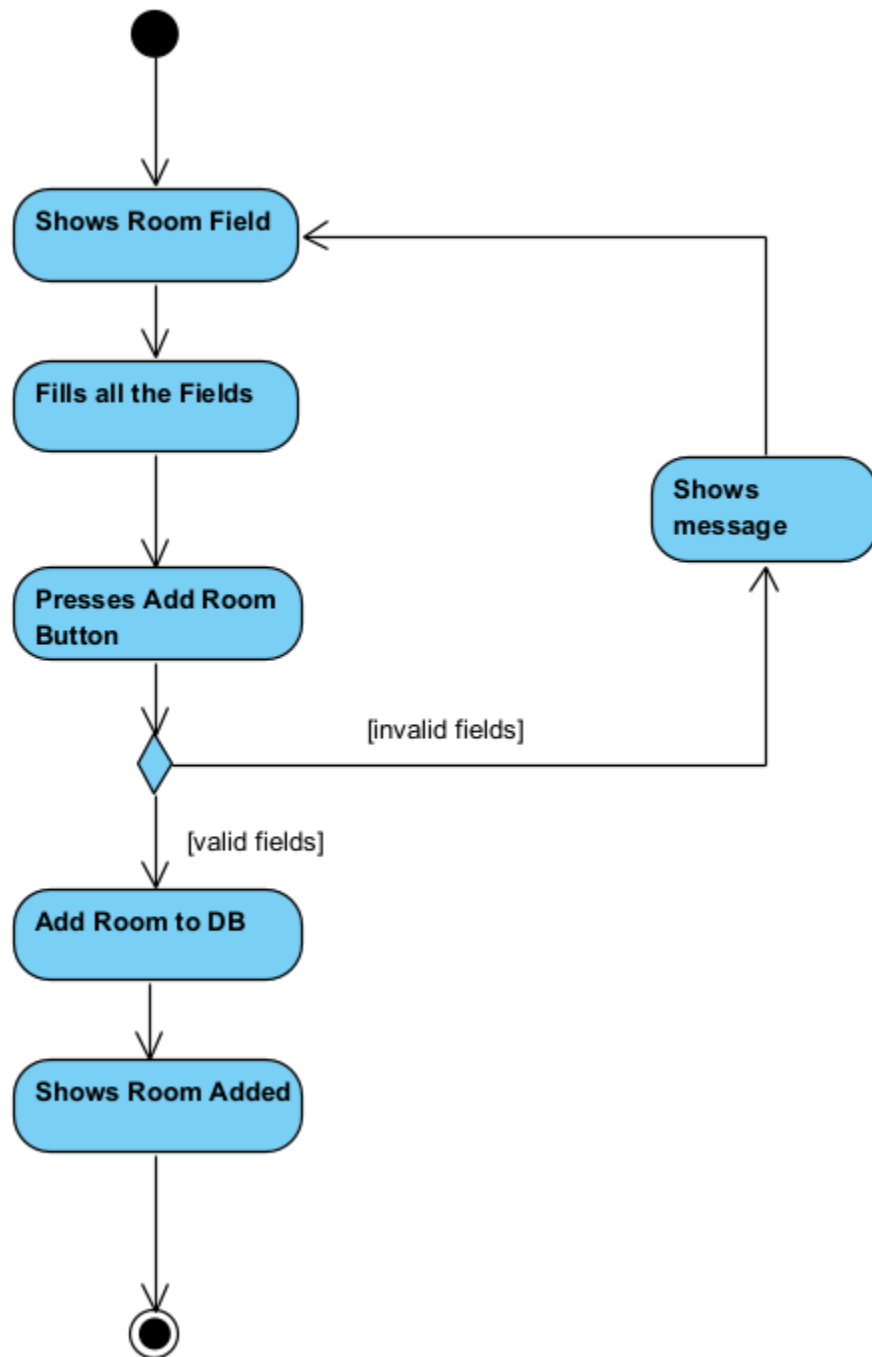
Add Employee



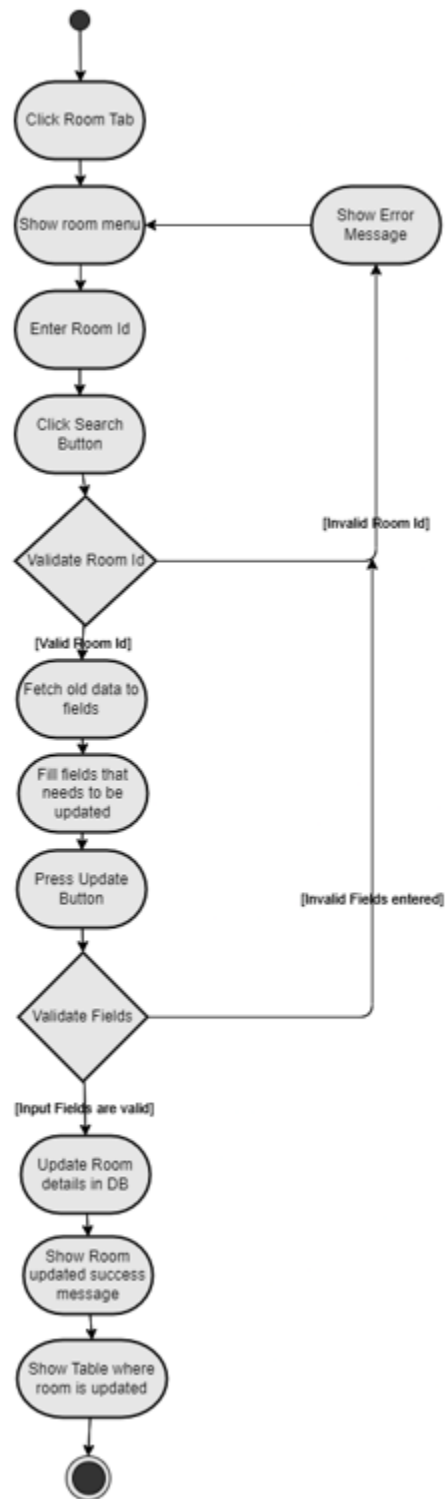
Update Employee



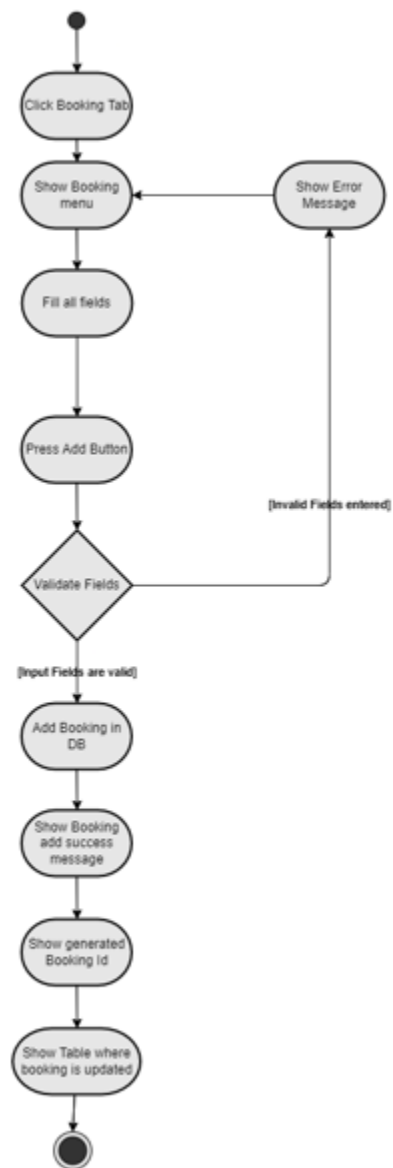
Add Room



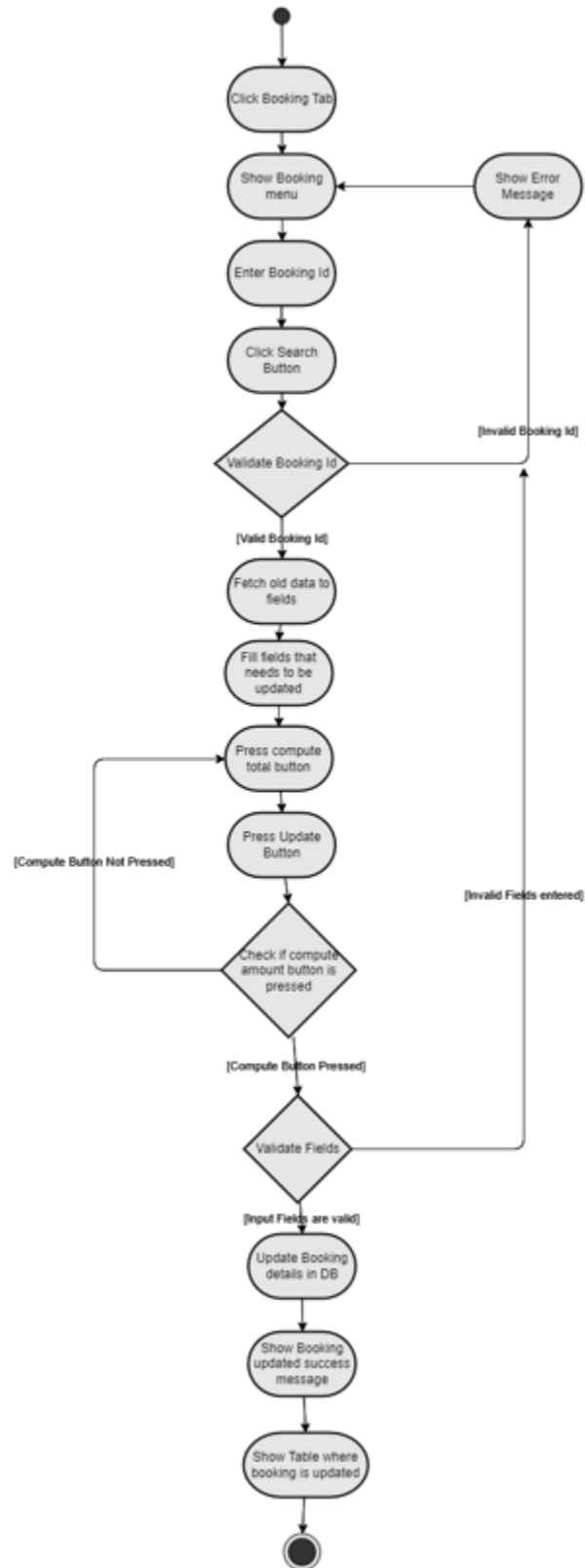
Update Room



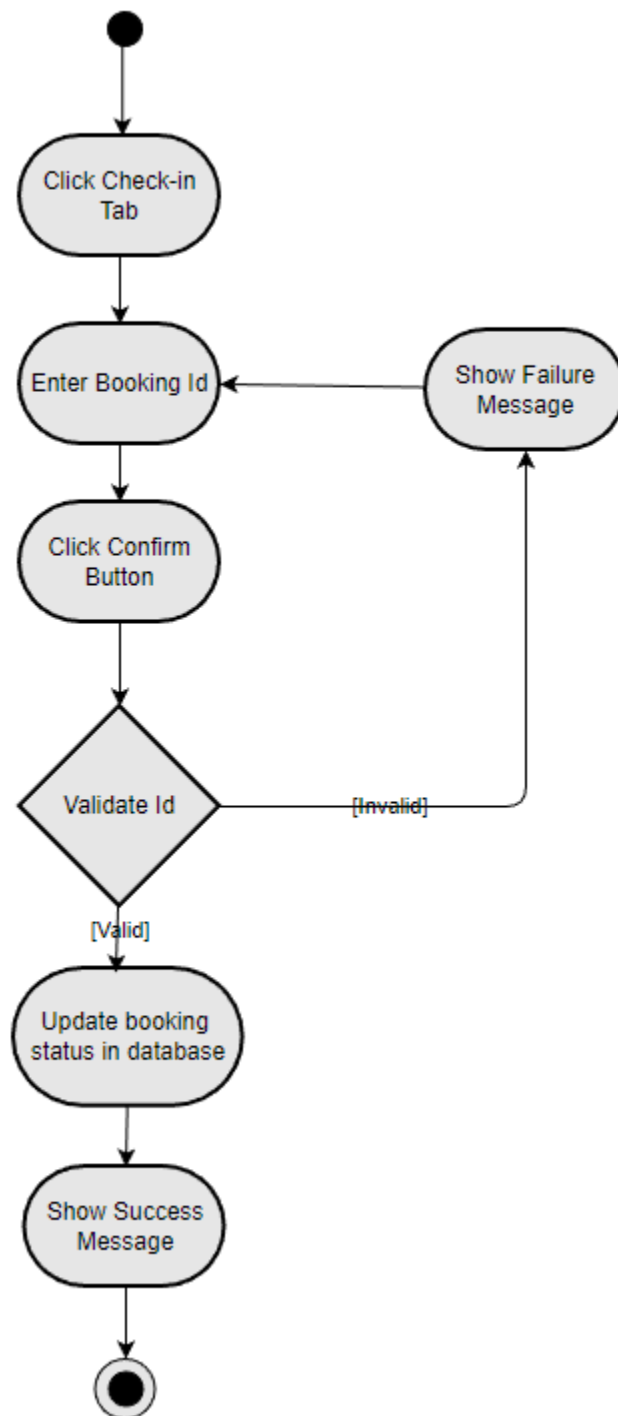
Add Booking



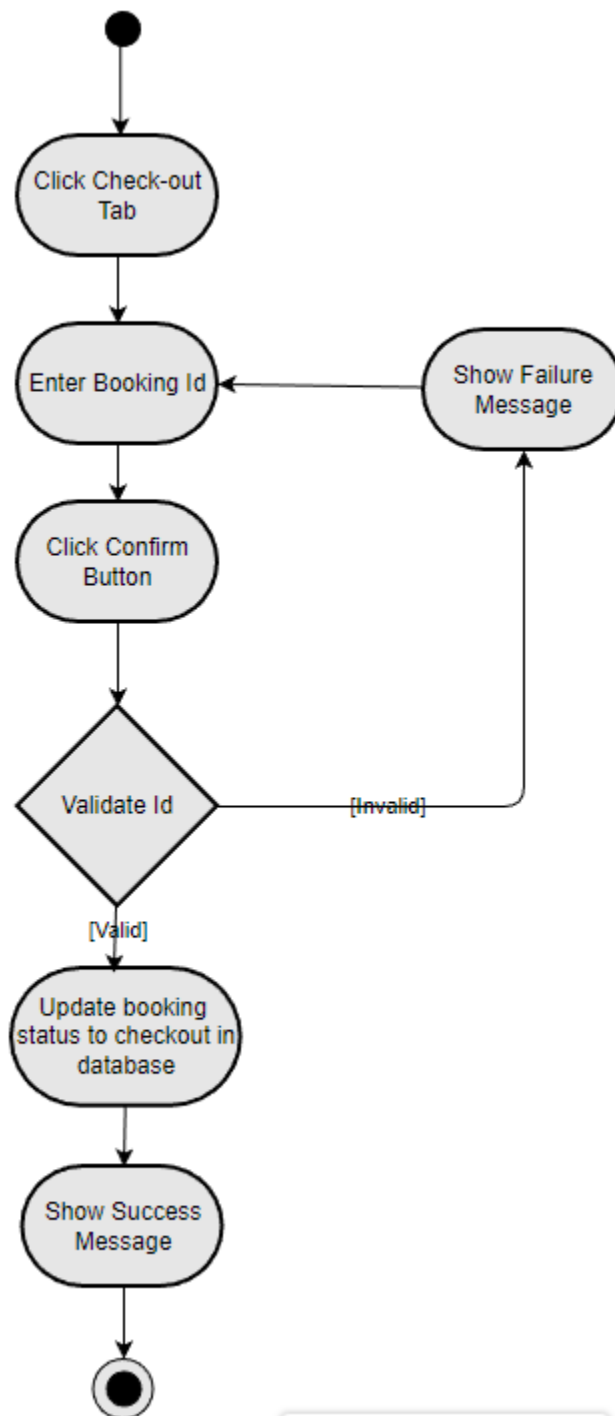
Update Booking



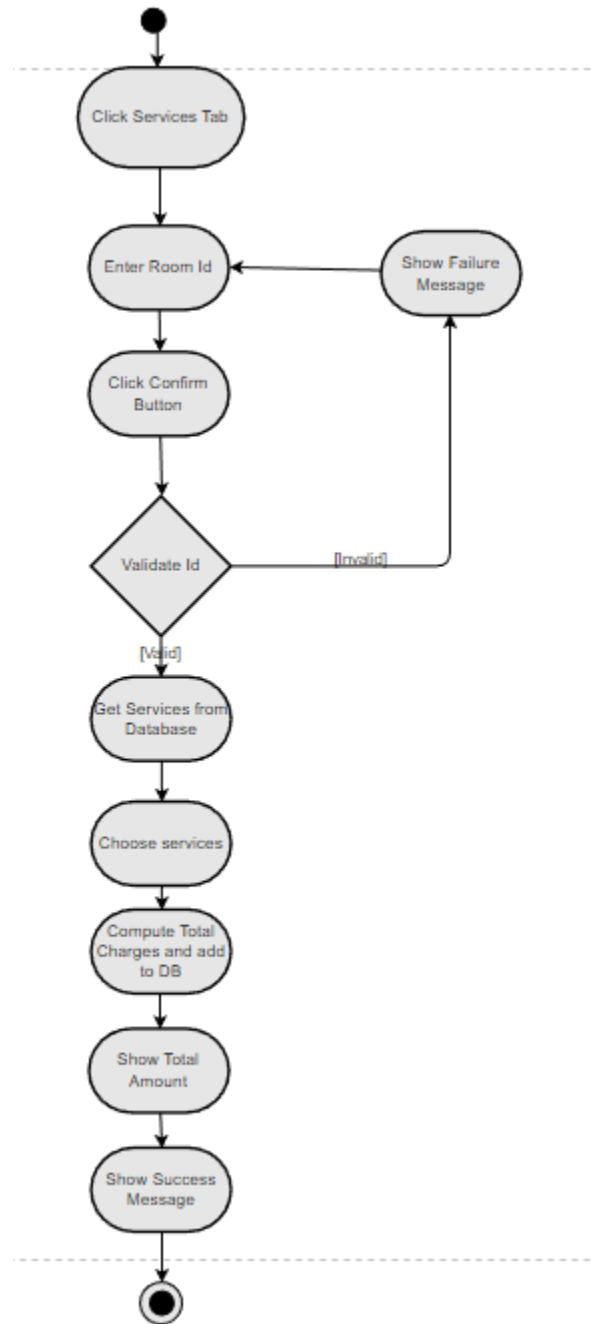
Check-In



Check-Out

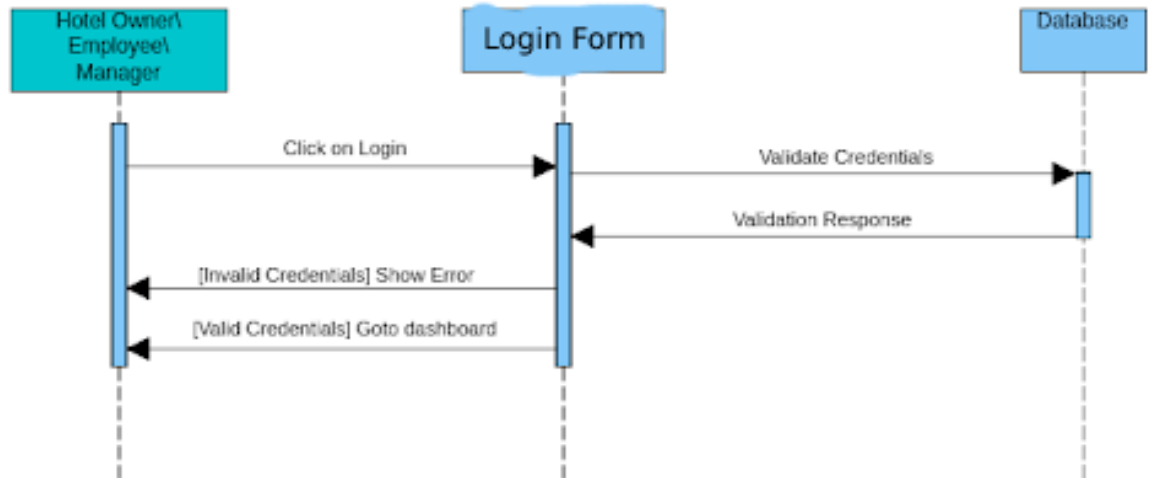


Request Services

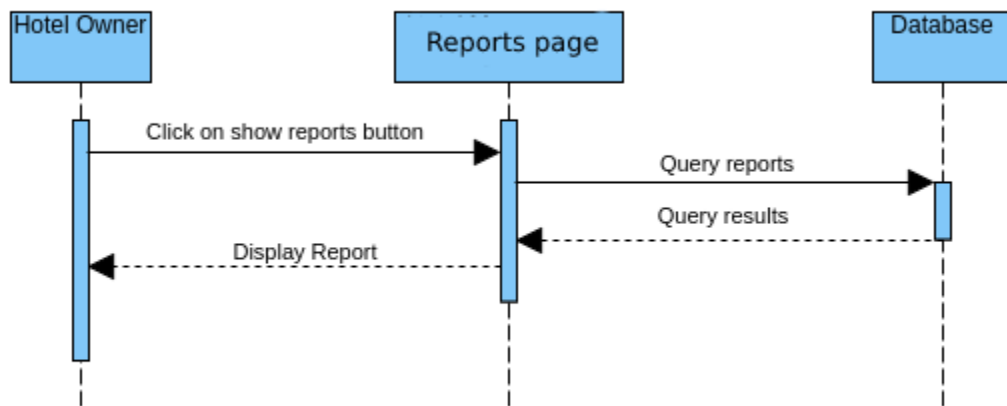


Sequence

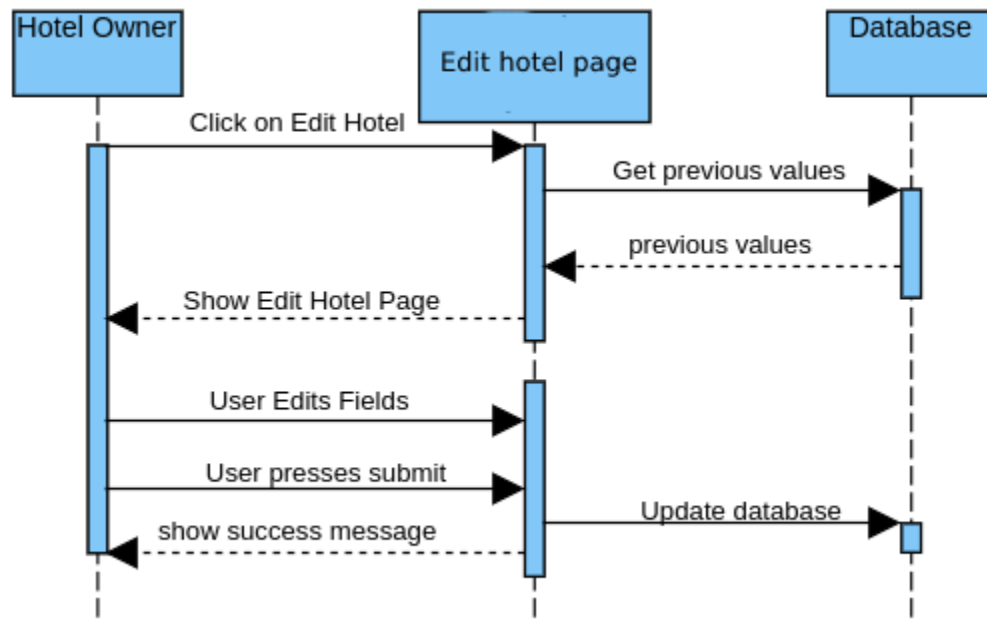
Login



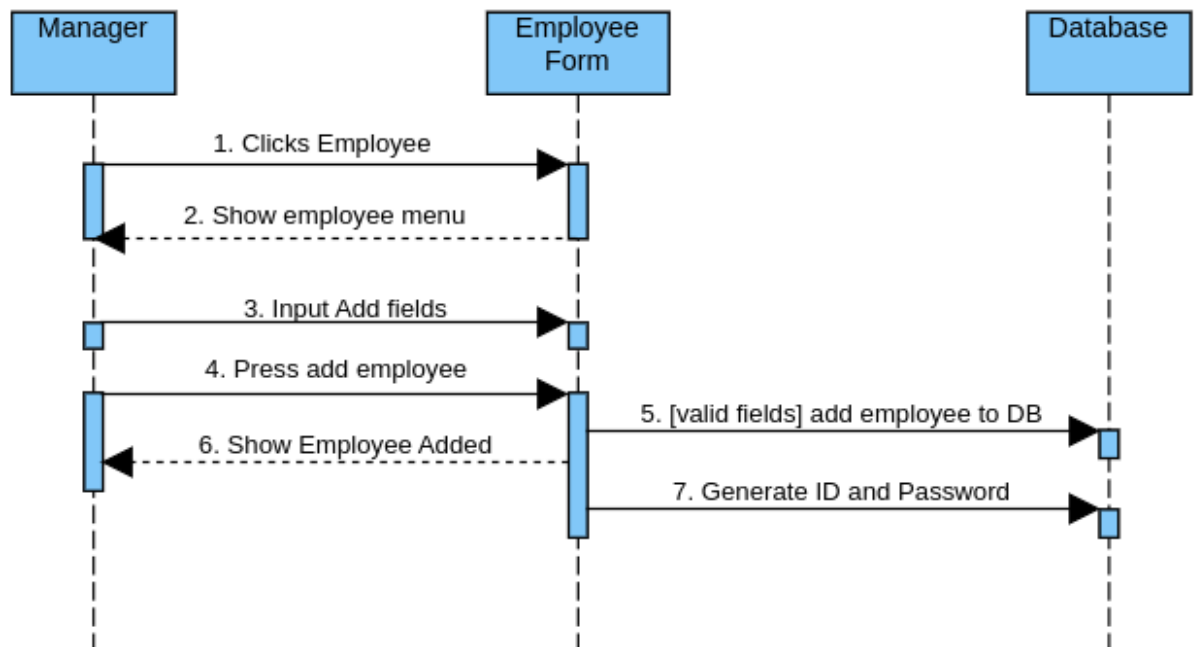
Generate Report



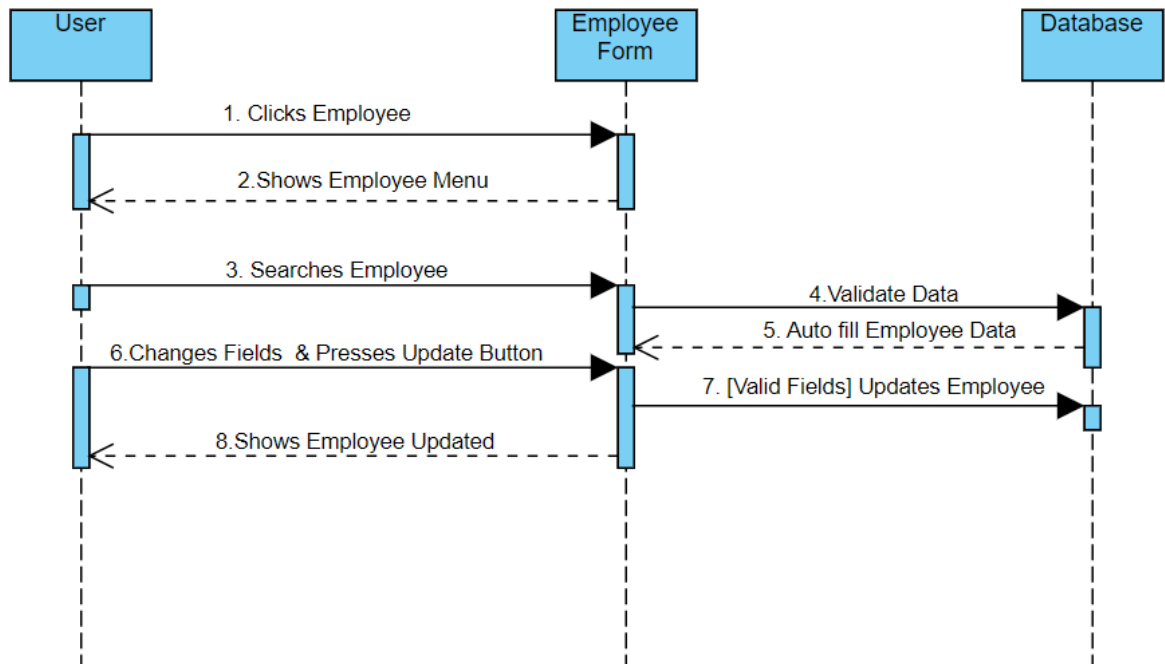
Update Hotel Information



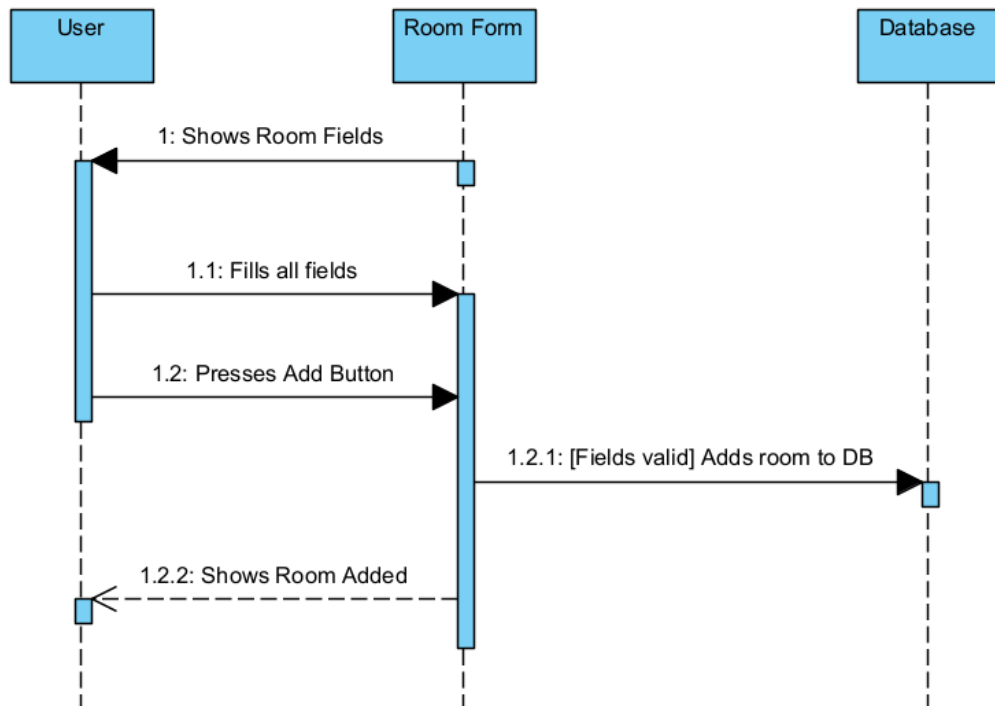
Add Employee



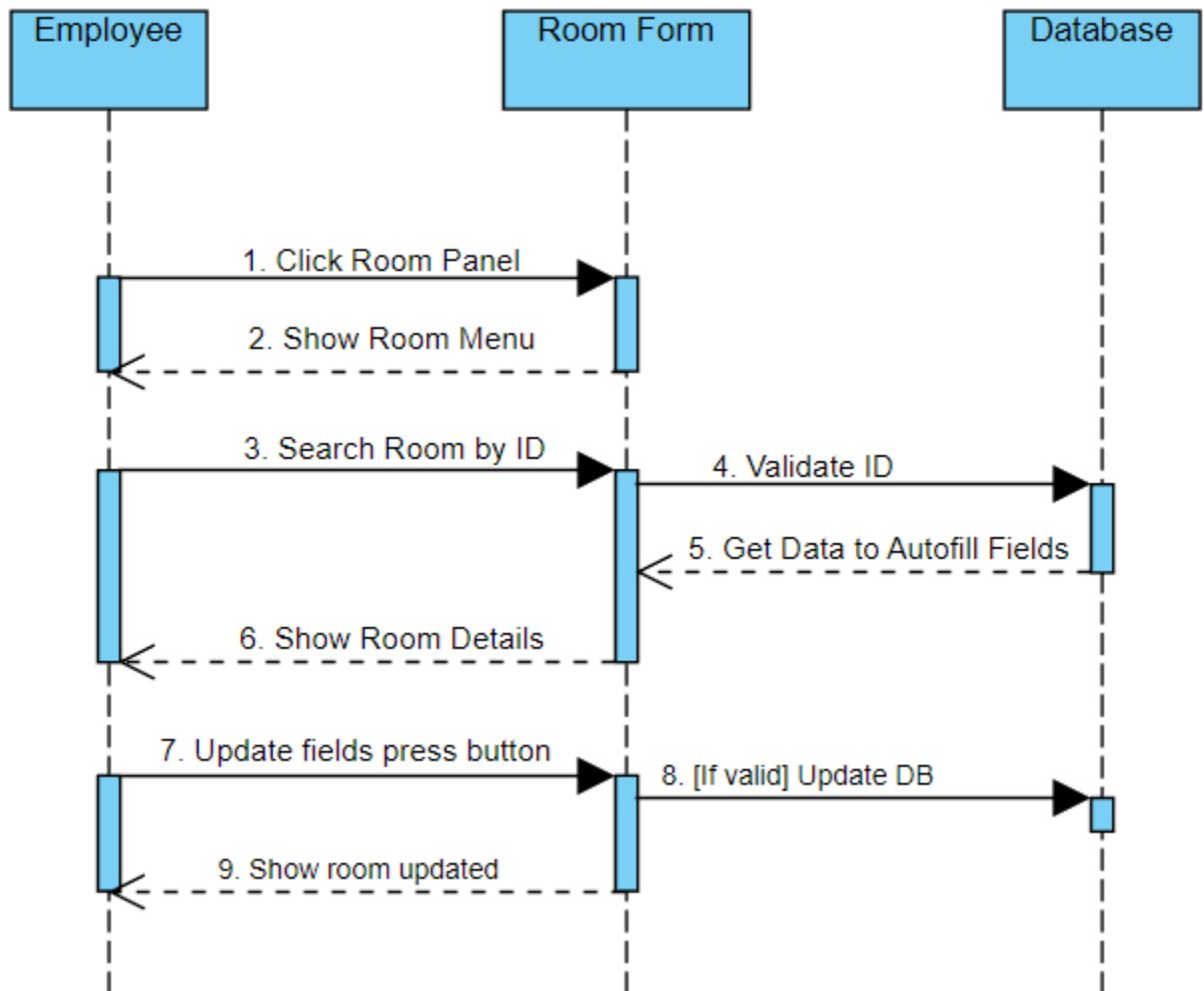
Update Employee



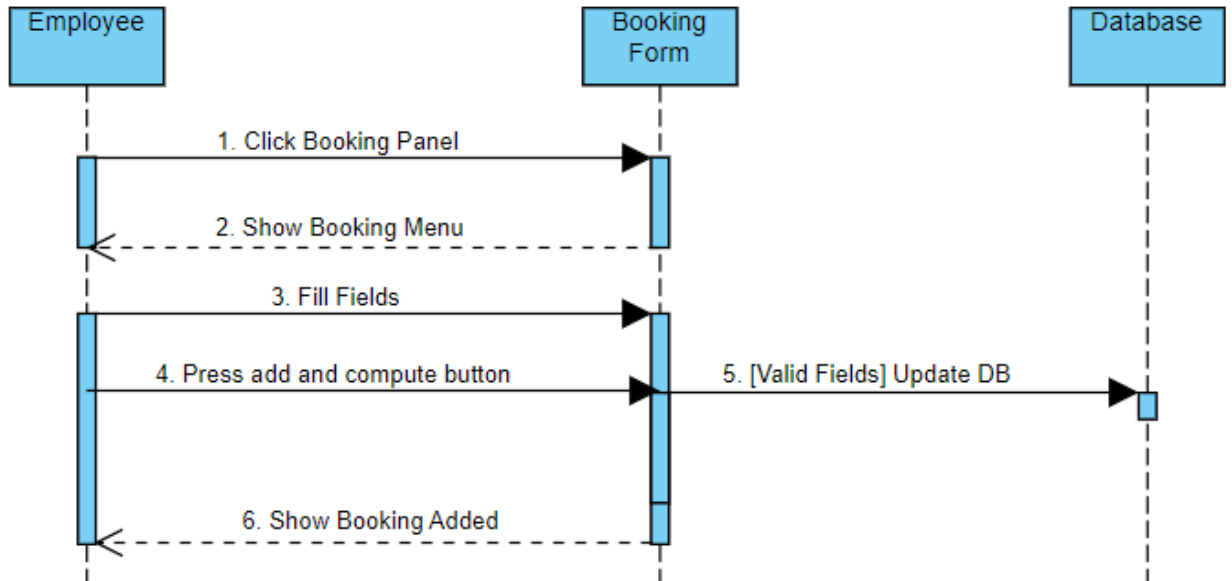
Add Room



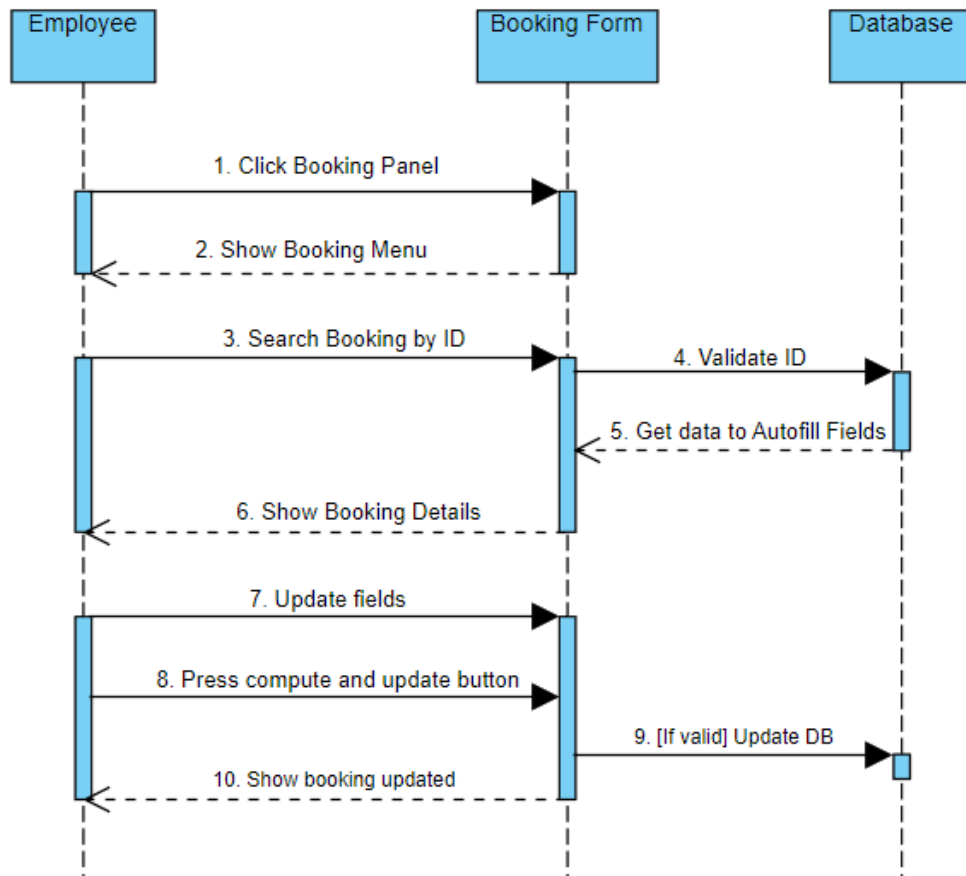
Update Room



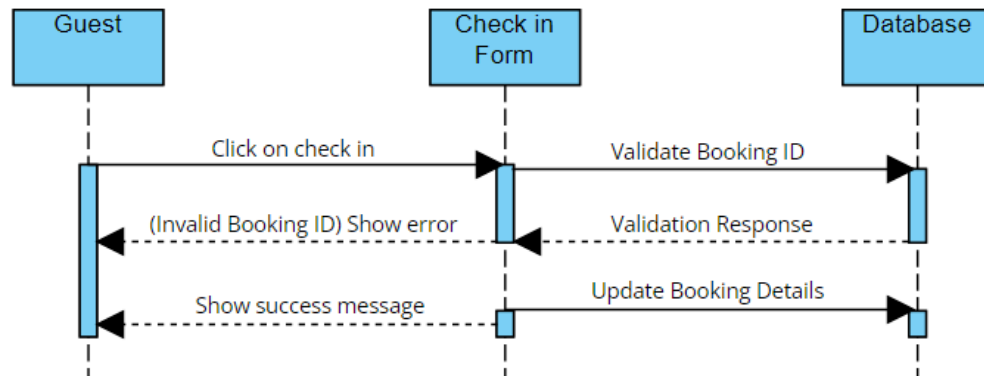
Add Booking



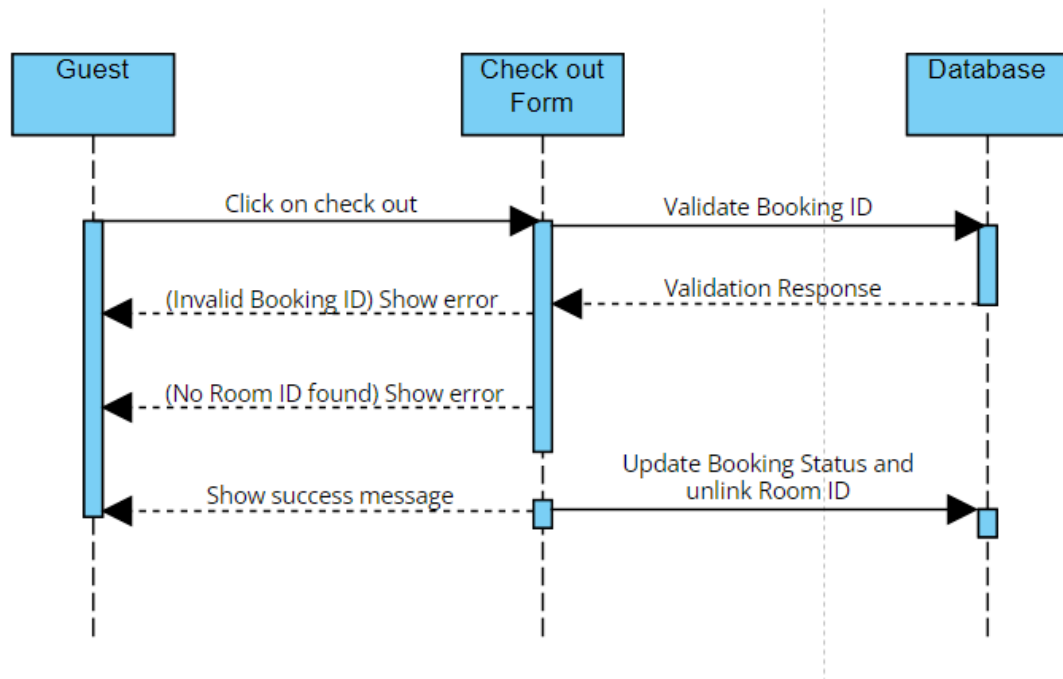
Update Booking



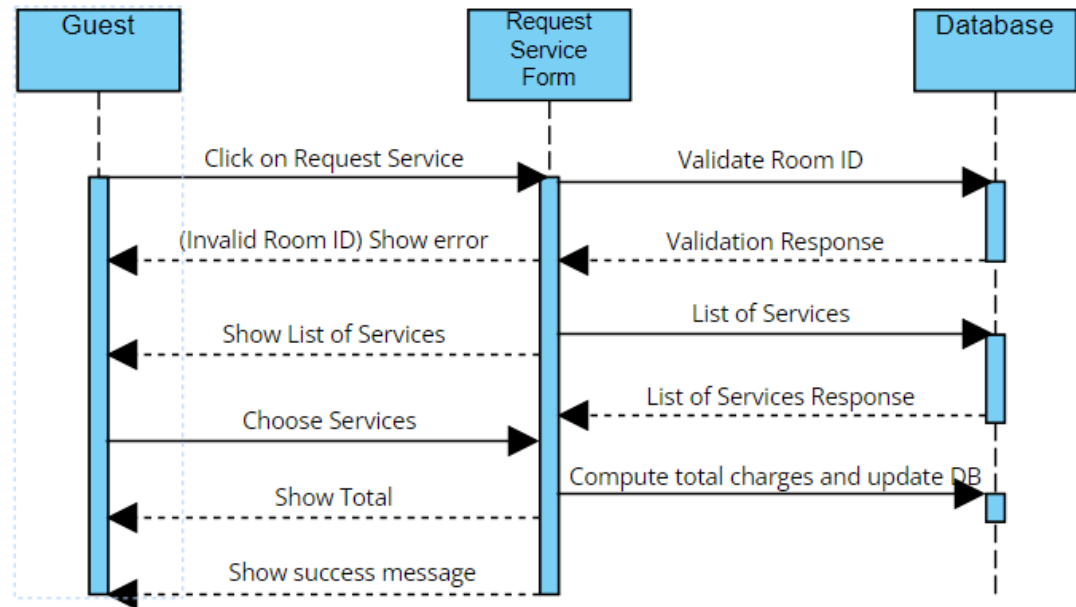
Check-in



Check-Out

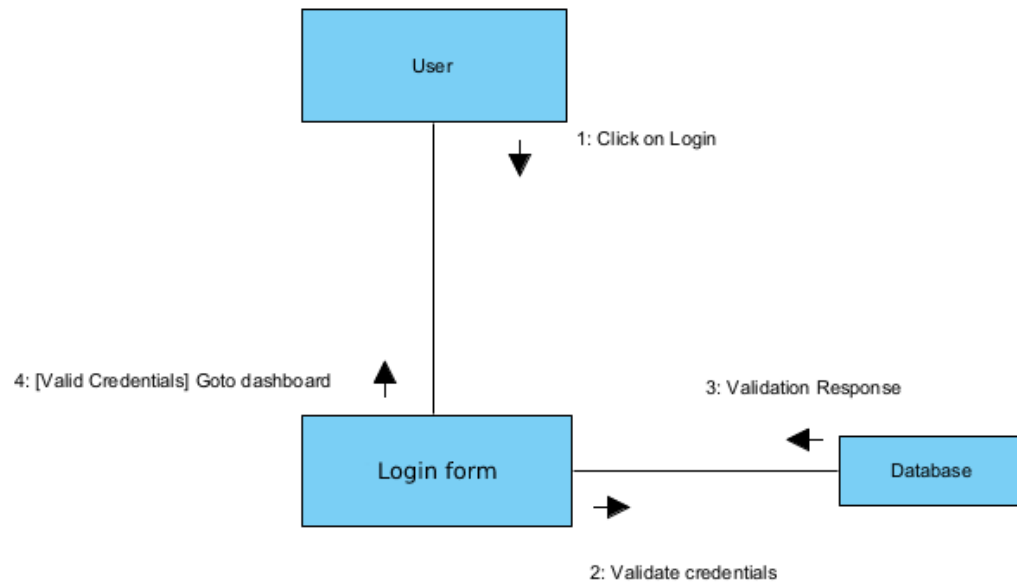


Request Services

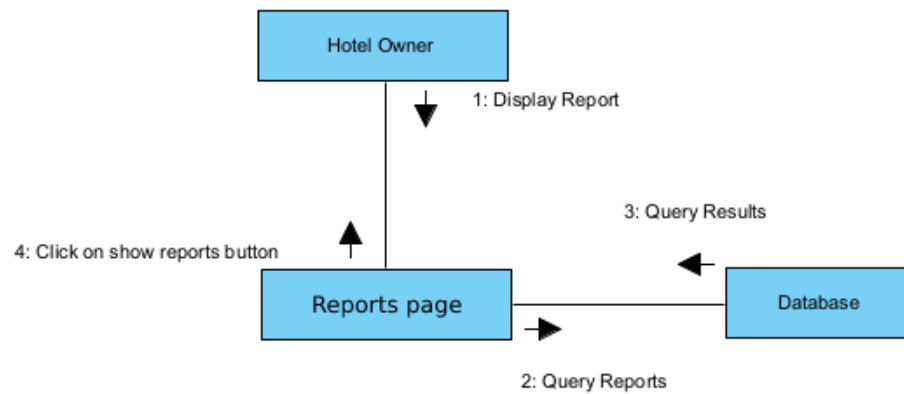


Collaboration

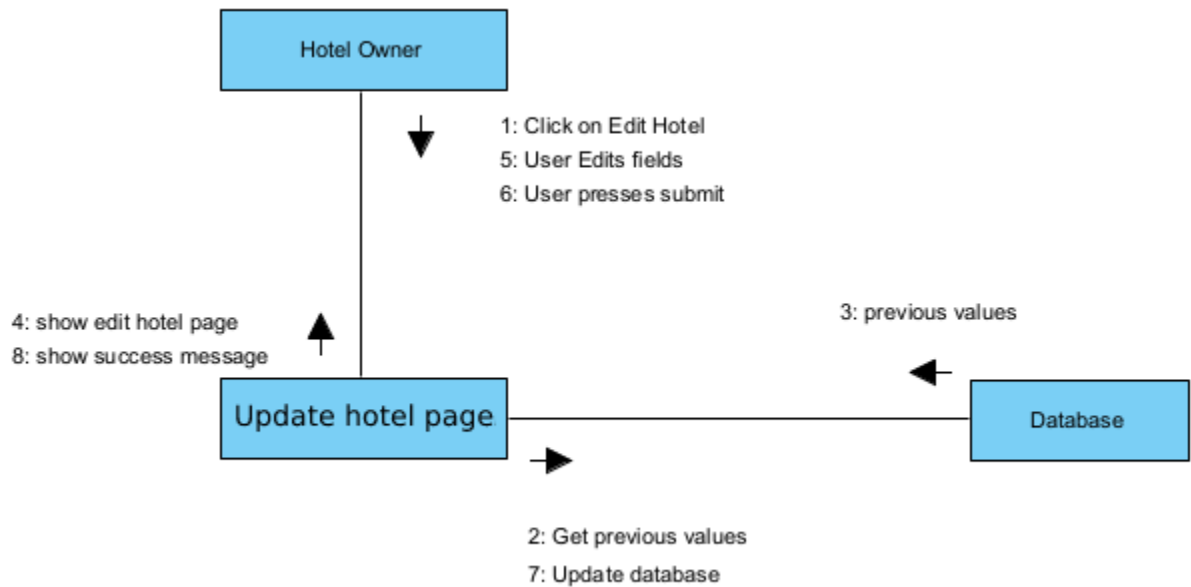
Login



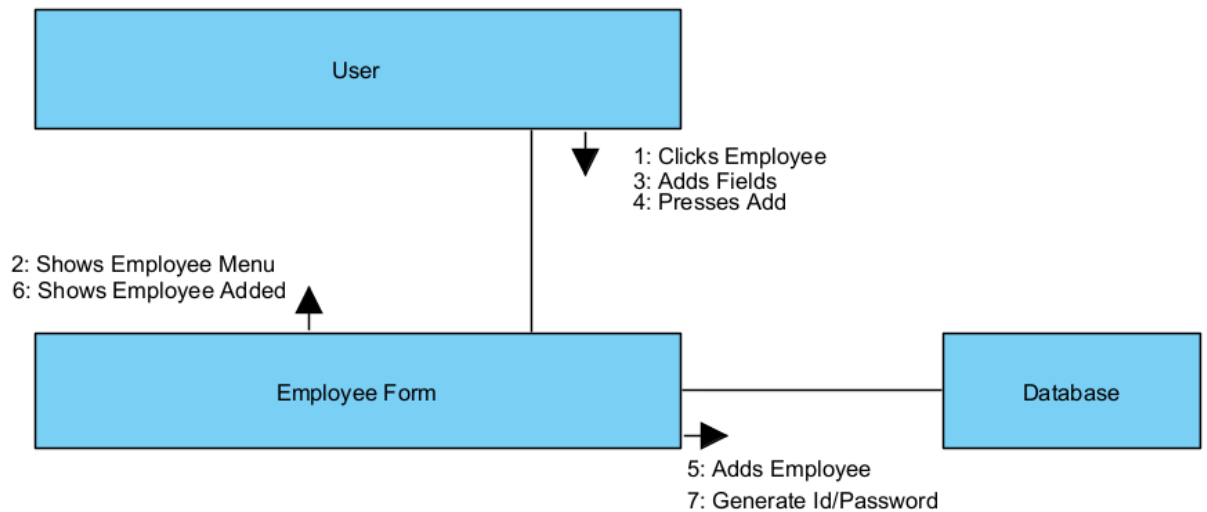
Generate Reports



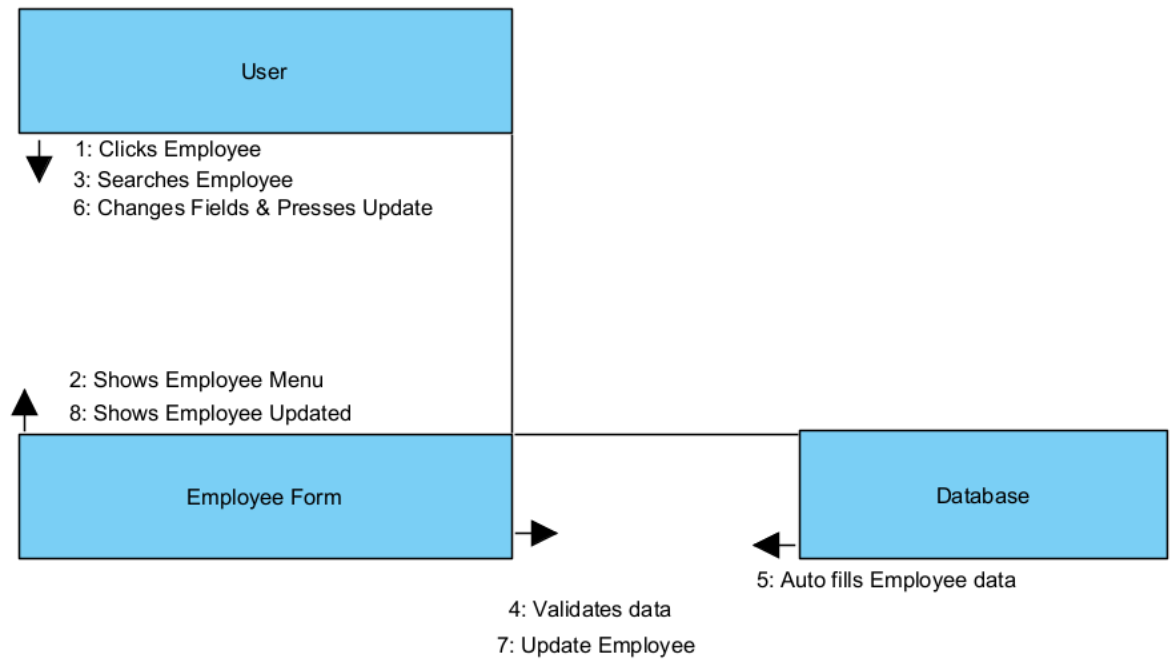
Update Hotel Information



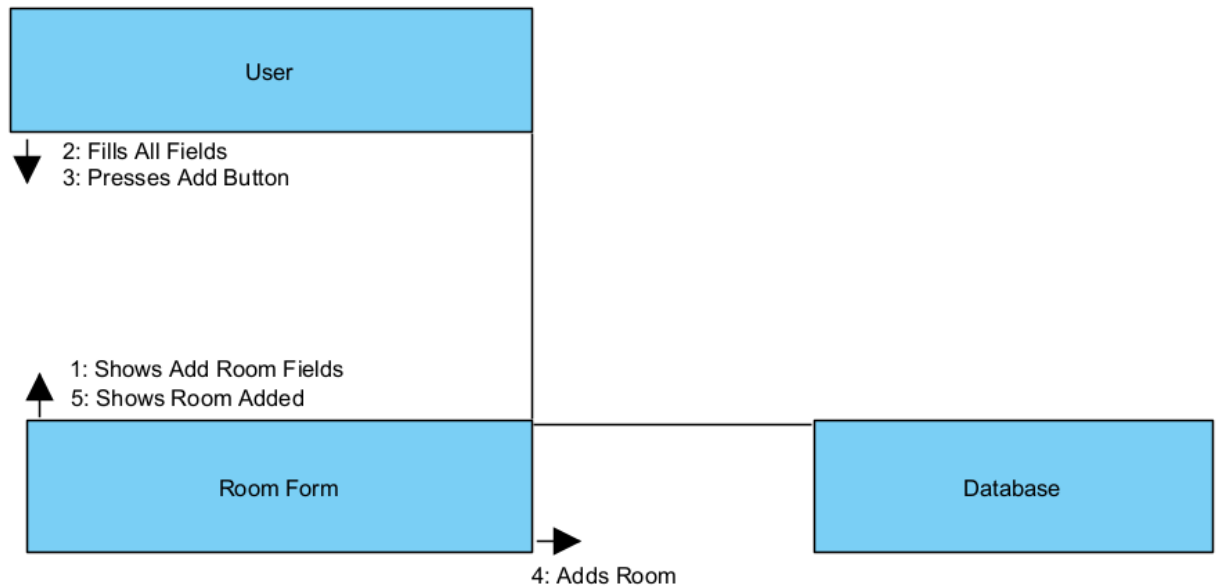
Add Employee



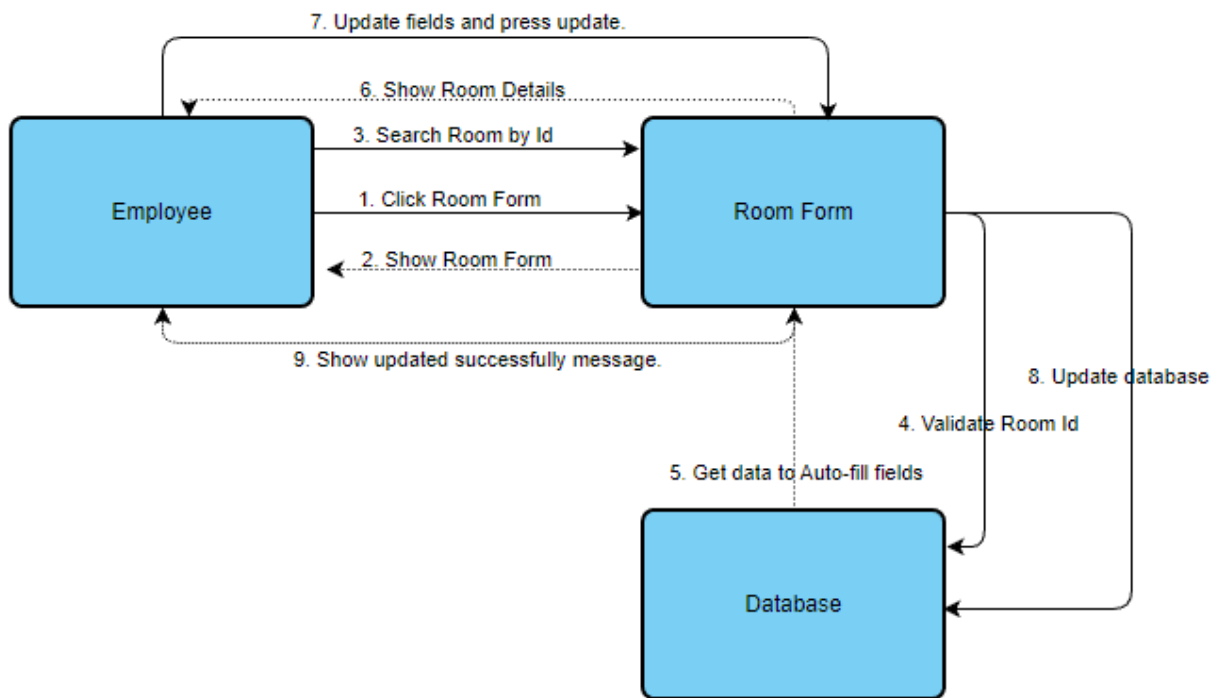
Update Employee



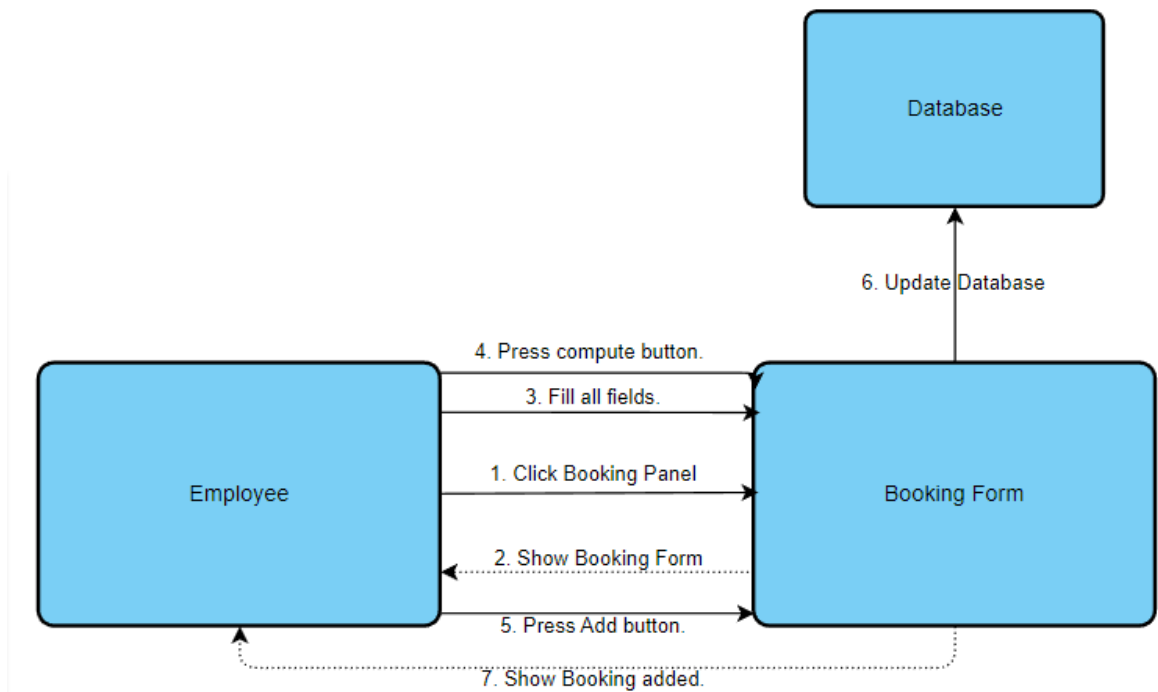
Add Room



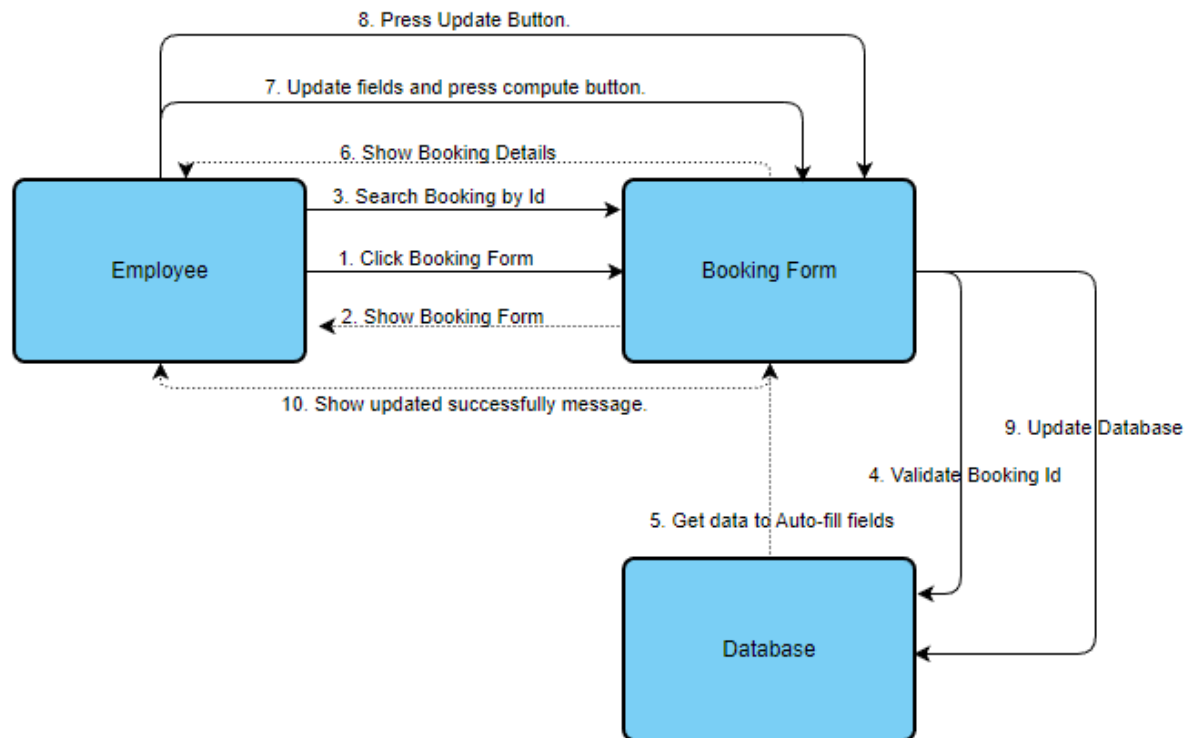
Update Room



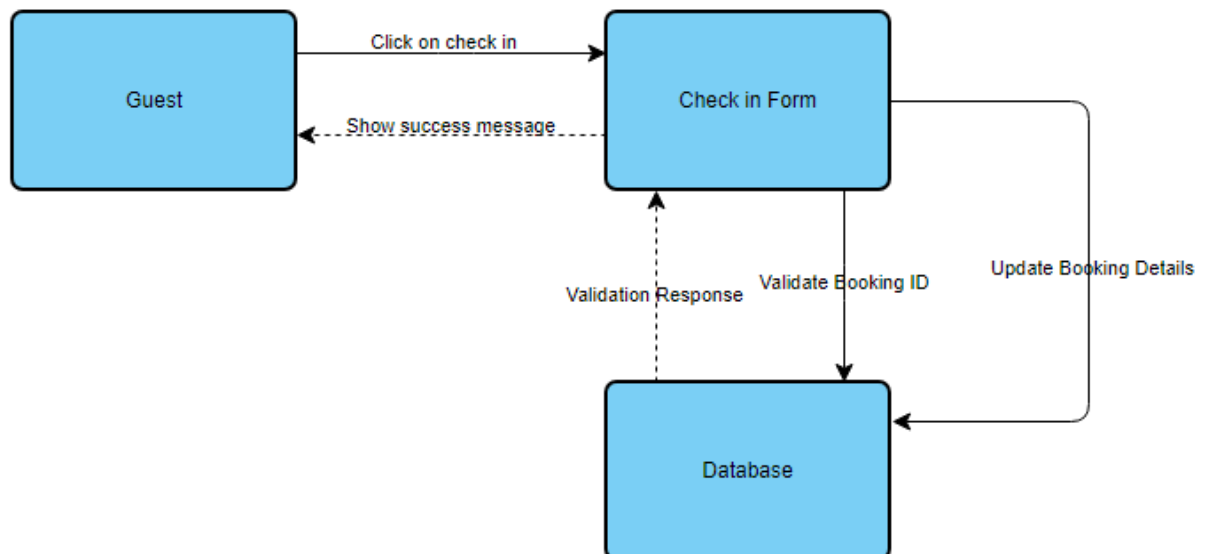
Add Booking



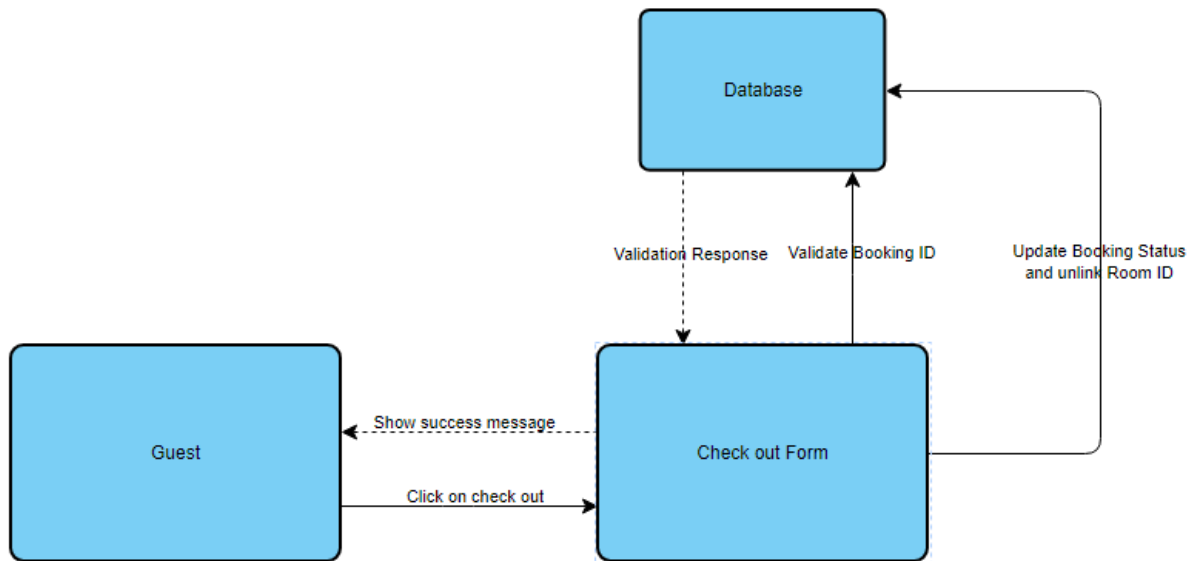
Update Booking



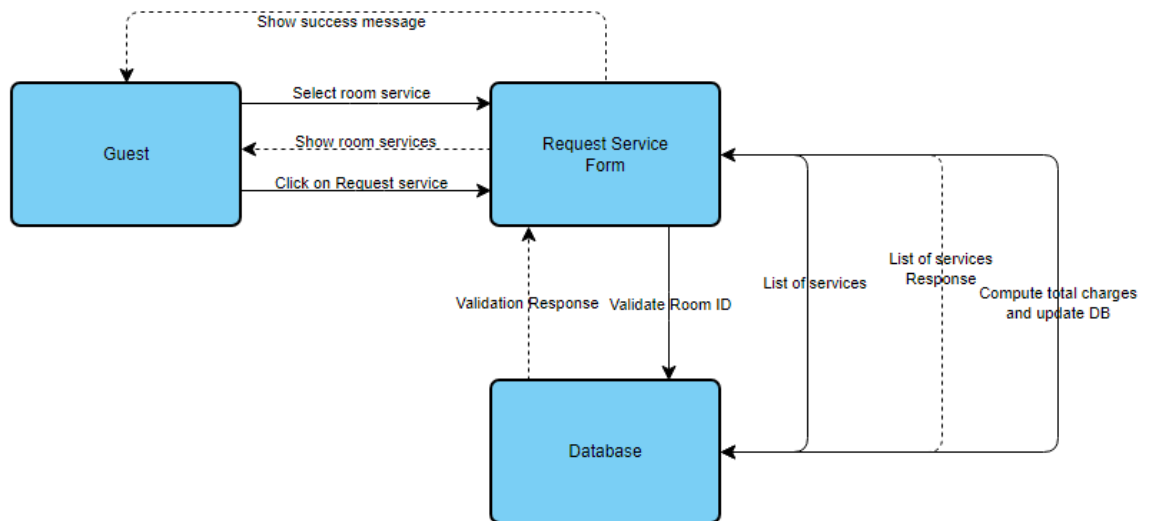
Check-in



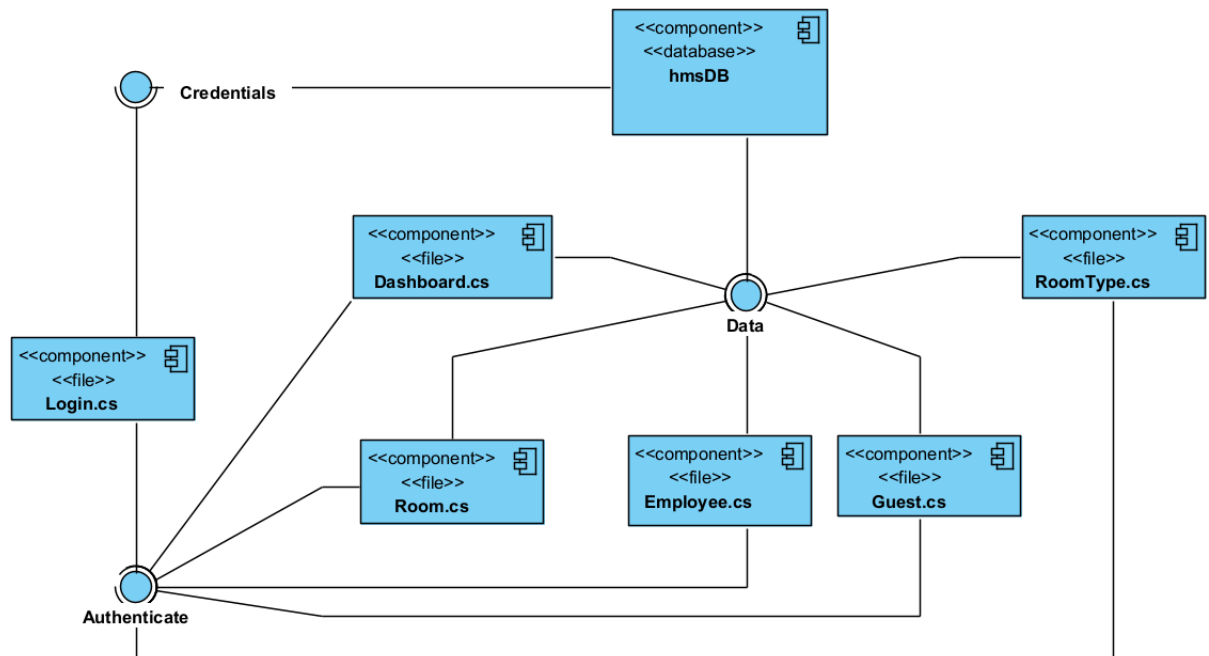
Check-Out



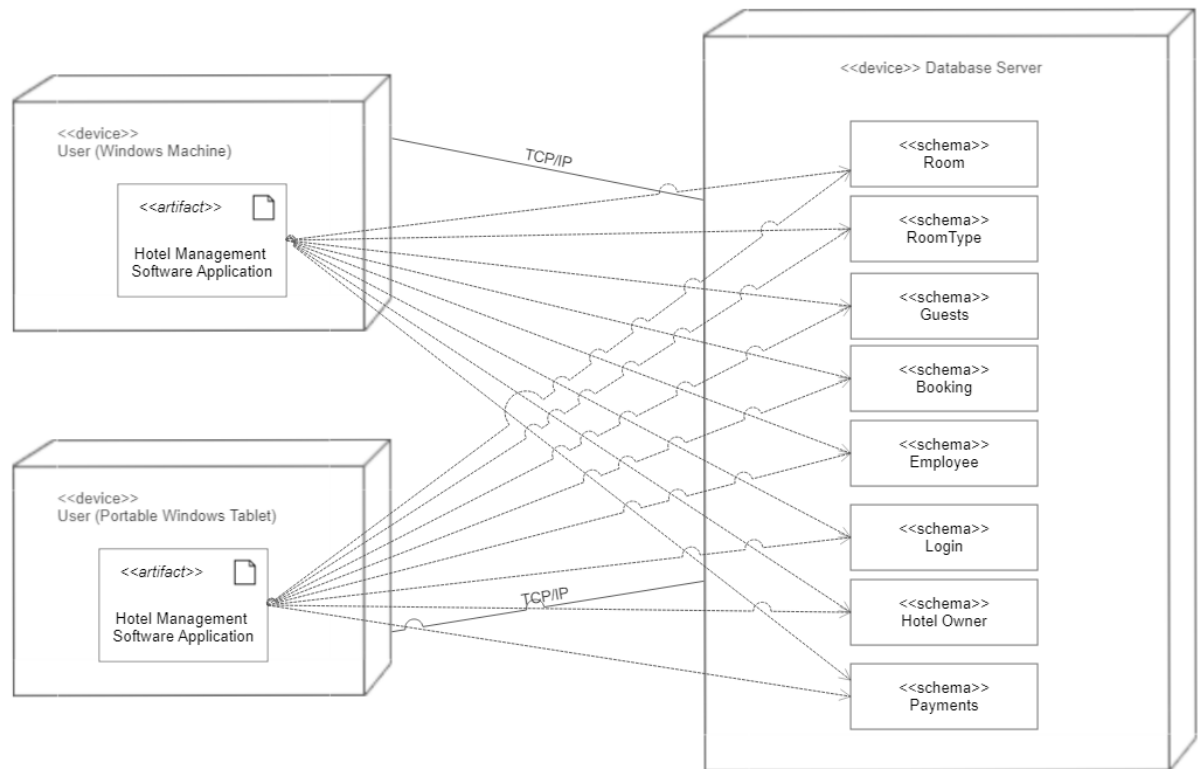
Request Services



Component Diagram



Deployment diagram



Use-Case Diagram

