

**Online Hotel Booking Application  
Analysis and Design Document  
Student: Andrei Branga  
Group: 30432**

Online Hotel Booking Application	Version: 1.1
	Date: 08/05/2018
Proj-doc	

## Revision History

Date	Version	Description	Author
11/04/2018	1.0	First iteration	Andrei Branga
08/05/2018	1.1	Second iteration	Andrei Branga

Online Hotel Booking Application	Version: 1.1
	Date: 08/05/2018
Proj-doc	

## Table of Contents

I.	Project Specification	4
II.	Elaboration – Iteration 1.1	4
1.	Domain Model	4
2.	Architectural Design	5
2.1	Conceptual Architecture	5
2.2	Package Design	6
2.3	Component and Deployment Diagrams	7
III.	Elaboration – Iteration 1.2	8
1.	Design Model	8
1.1	Dynamic Behavior	8
1.2	Class Design	9
2.	Data Model	10
3.	Unit Testing	10
IV.	Elaboration – Iteration 2	4
1.	Architectural Design Refinement	4
2.	Design Model Refinement	4
V.	Construction and Transition	11
1.	System Testing	11
2.	Future improvements	11
VI.	Bibliography	11

Online Hotel Booking Application	Version: 1.1
	Date: 08/05/2018
Proj-doc	

## I. Project Specification

There are many web applications available which makes people's work quicker. Here we introduce new web application where user can book rooms via our handy web app. This application allows users to book hotel rooms. Using this system user can view and check for various rooms available and simultaneously book them by making online payment via credit card. The system also provides user with additional facilities like Jacuzzi, swimming, meals and additional bed addition along with their associated charges. The system calculates the total cost on booking the services. Once the user makes the payment, system will provide online receipt to the user. User can view the room booking in an effective graphical user interface. Since room bookings will be displayed in effective graphical user interface user will get to know which rooms are booked and how many rooms are available for booking. Using this application user can select the room according to his preference. The rooms which are already will be disabled and the rooms which are available user just have to select it and then proceed to payment option. Once user makes the payment system will generate receipt and it will be sent to respective users email id and it will be reported to the admin, when user visits the hotel, he must show the receipt for the accommodation. The application also contains an admin side for the app owners and hotel owner-side where each hotel manager can manage the hotel booking page.

The application will have three types of users:

- ADMIN
- HOTEL MANAGER
- REGULAR USER

Each of them will have different functionalities available.

The app will be delivered as a cloud - based solution.

## II. Elaboration – Iteration 1.1

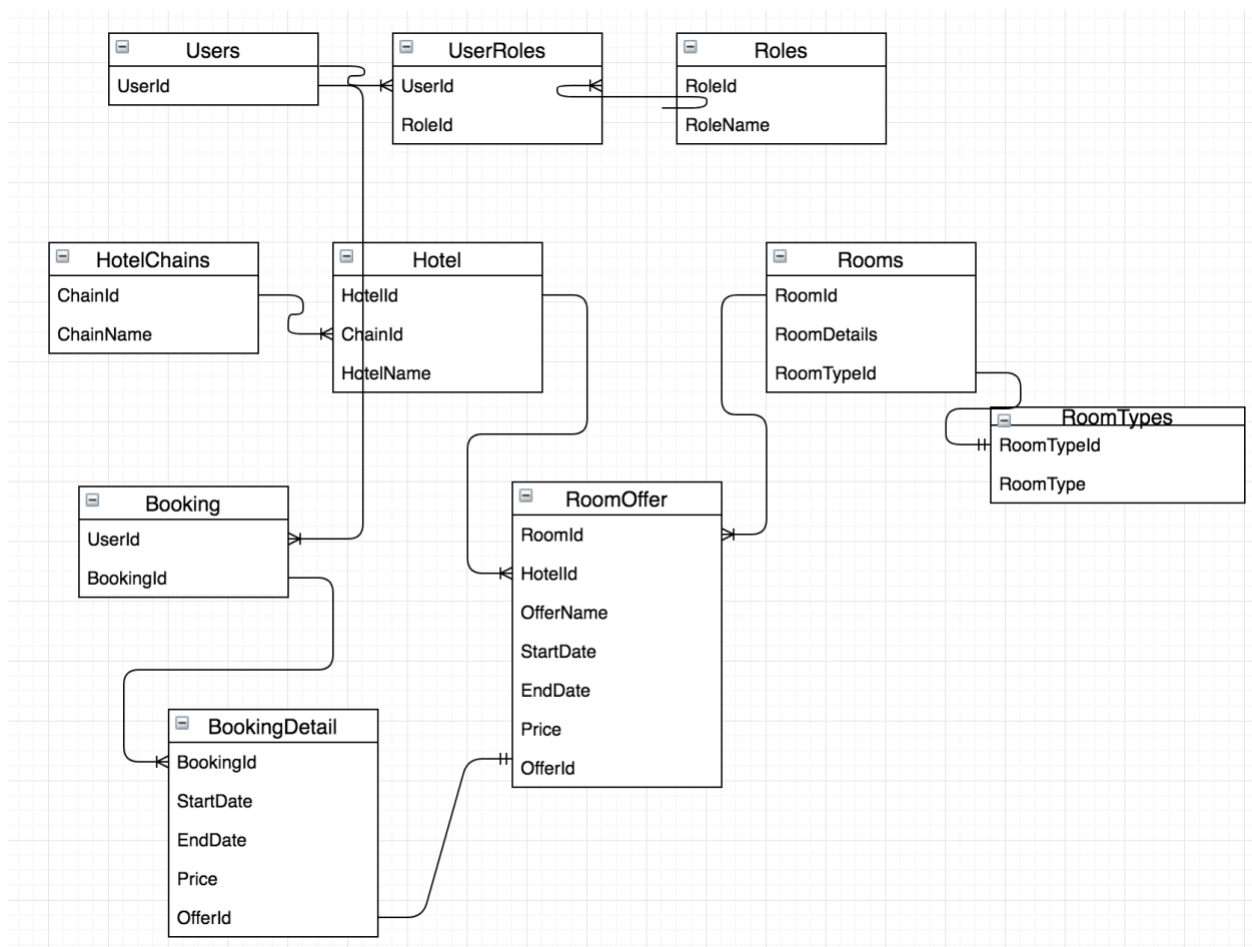
### 1. Domain Model

Each type of user (admin, hotel manager, regular user) will have an account associated. The hotel manager and regular user will have a user profile attached.

The hotel manager can have multiple hotels assigned to his account and can manage each one of them. He will be able to define what types of rooms the hotel has, the price etc. He can also see and manage the bookings, see reports.

The regular user can see the hotels available for one given location. Each location has many hotels attached. Bookings may be accessed by the user, new ones can be made or active ones edited.

Online Hotel Booking Application	Version: 1.1
	Date: 08/05/2018
Proj-doc	



So, I have identified the data tables: Accounts, User profiles, Roles, Hotels, Hotel Chain, Rooms, Availability, Locations, Bookings, Payments and also the auxiliary tables necessary for the table relations.

## 2. Architectural Design

### 2.1 Conceptual Architecture

The main architecture used in the project is MVC:

View renders the data from the Model in response to the request made to the model by controlled events made by user interaction.

Model View Controller is a design approach to separate the application object model from GUI, originally invented around 80s. Then later on it has become a widely accepted common design pattern. The main objective behind this pattern is to decouple the view of the data (presentation layer) from the actual data processing so that the same model can be used for various views. This is achieved by using three different types of objects that interact with each other in loosely coupled manner with their discreet set of tasks.

#### Presentation layer

Presentation of the web pages, and user interaction. This is represented by the MVC project. It will handle the data through the business layer.

#### Business layer

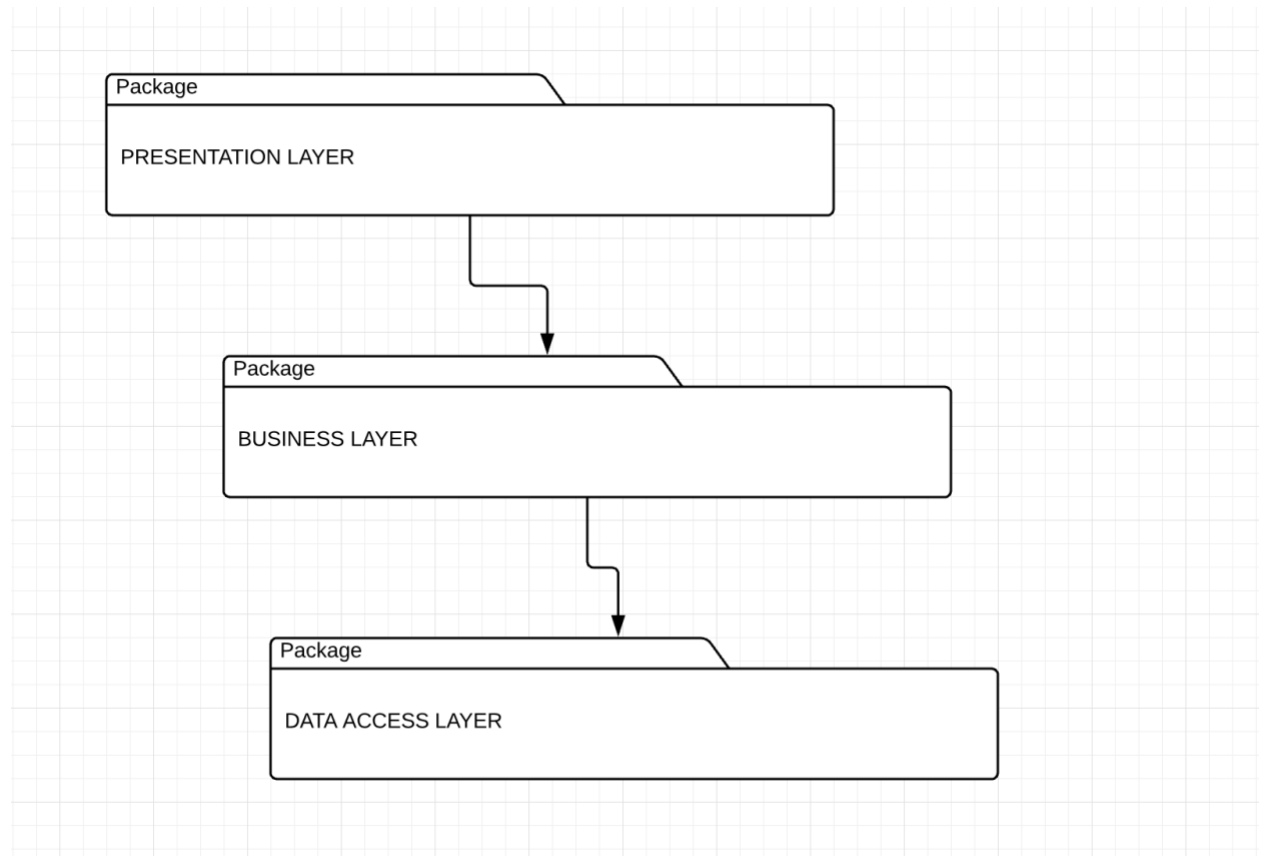
The logic behind the accessibility, all business logic and data manipulation happens here.

Online Hotel Booking Application	Version: 1.1
	Date: 08/05/2018
Proj-doc	

### Persistent layer

This is the presentation layer for the Data. This includes the DAO (Data Access Object) presentation, ORM (Object Relational Mappings) and Other modes of presenting persistent data in the application level. In more meaningful words this demonstrates the persistent data in RAM. Which usually stays in Disks at the below layer.

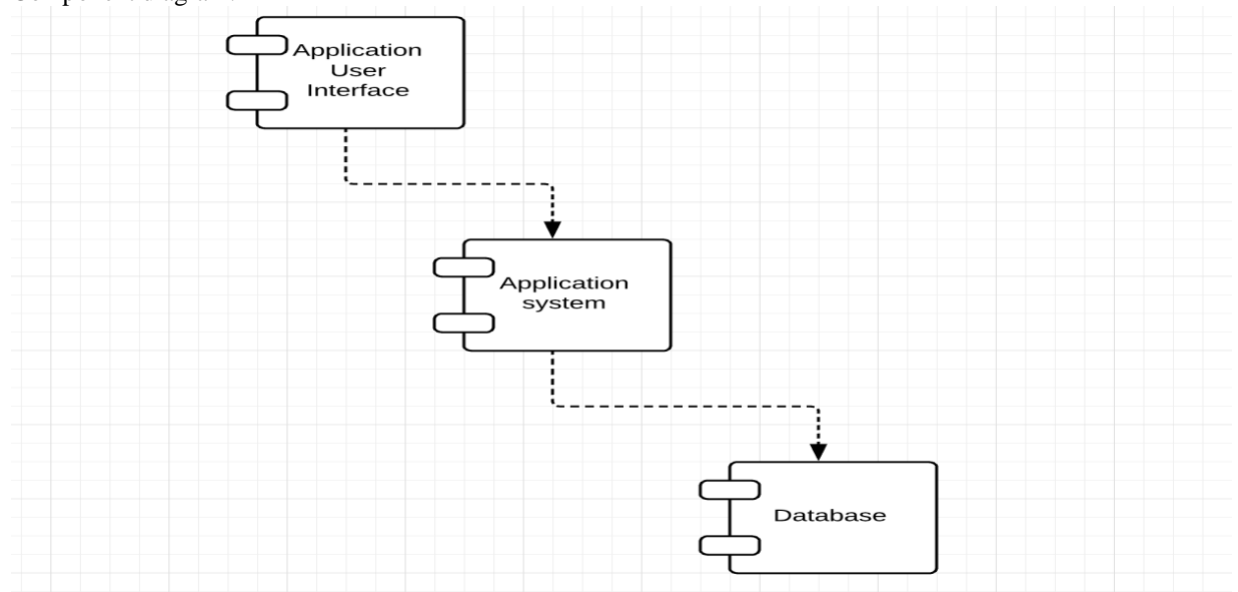
## 2.2 Package Design



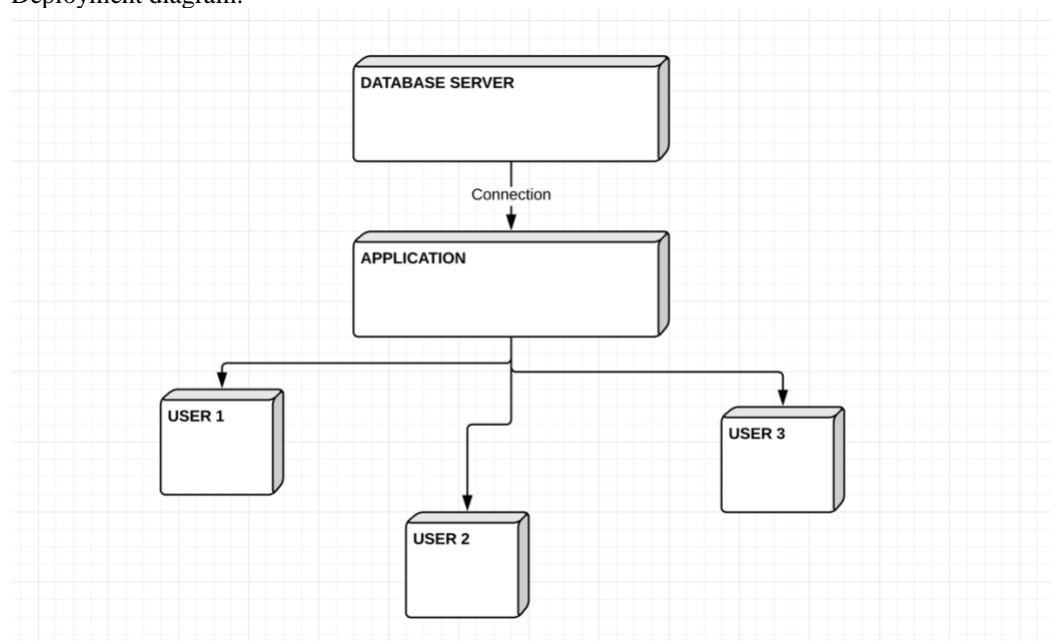
Online Hotel Booking Application	Version: 1.1
	Date: 08/05/2018
Proj-doc	

## 2.3 Component and Deployment Diagrams

Component diagram:



Deployment diagram:



Online Hotel Booking Application	Version: 1.1
	Date: 08/05/2018
Proj-doc	

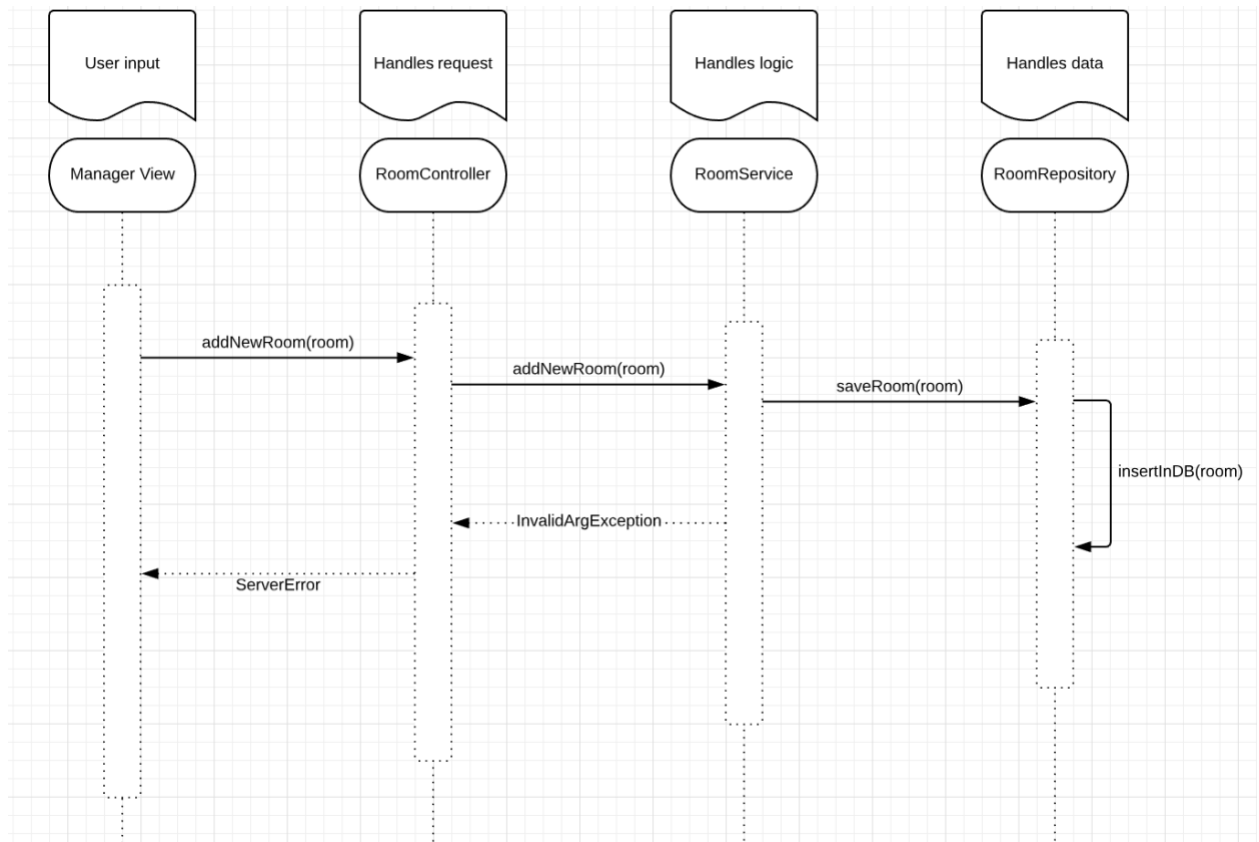
### III. Elaboration – Iteration 1.2

#### 1. Design Model

##### 1.1 Dynamic Behavior

#### **SEQUENCE DIAGRAM**

*Scenario: Hotel Manager adds a new room type to his hotel account*





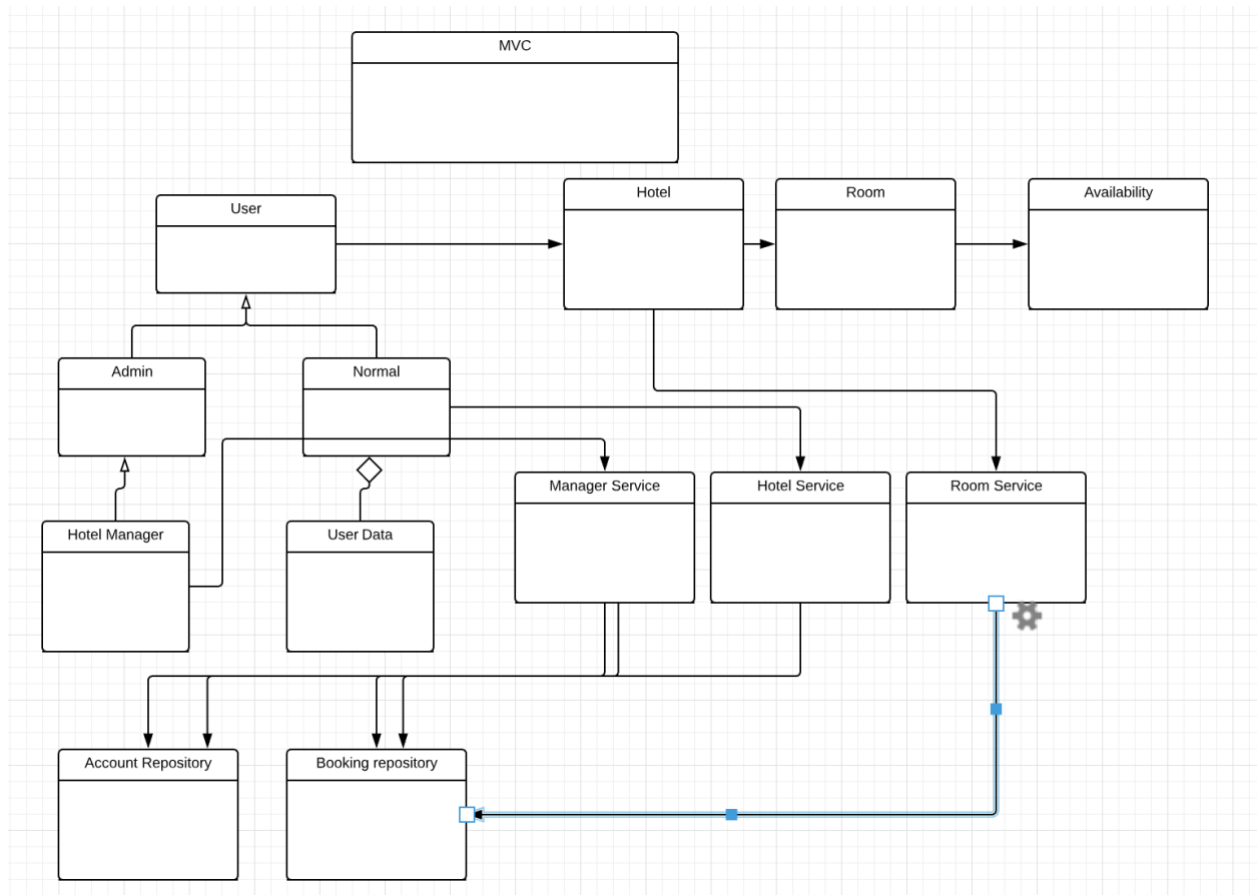
Online Hotel Booking Application	Version: 1.1
	Date: 08/05/2018
Proj-doc	

### COLLABORATION DIAGRAM

*Scenario: Hotel Manager adds a new room type to his hotel account*



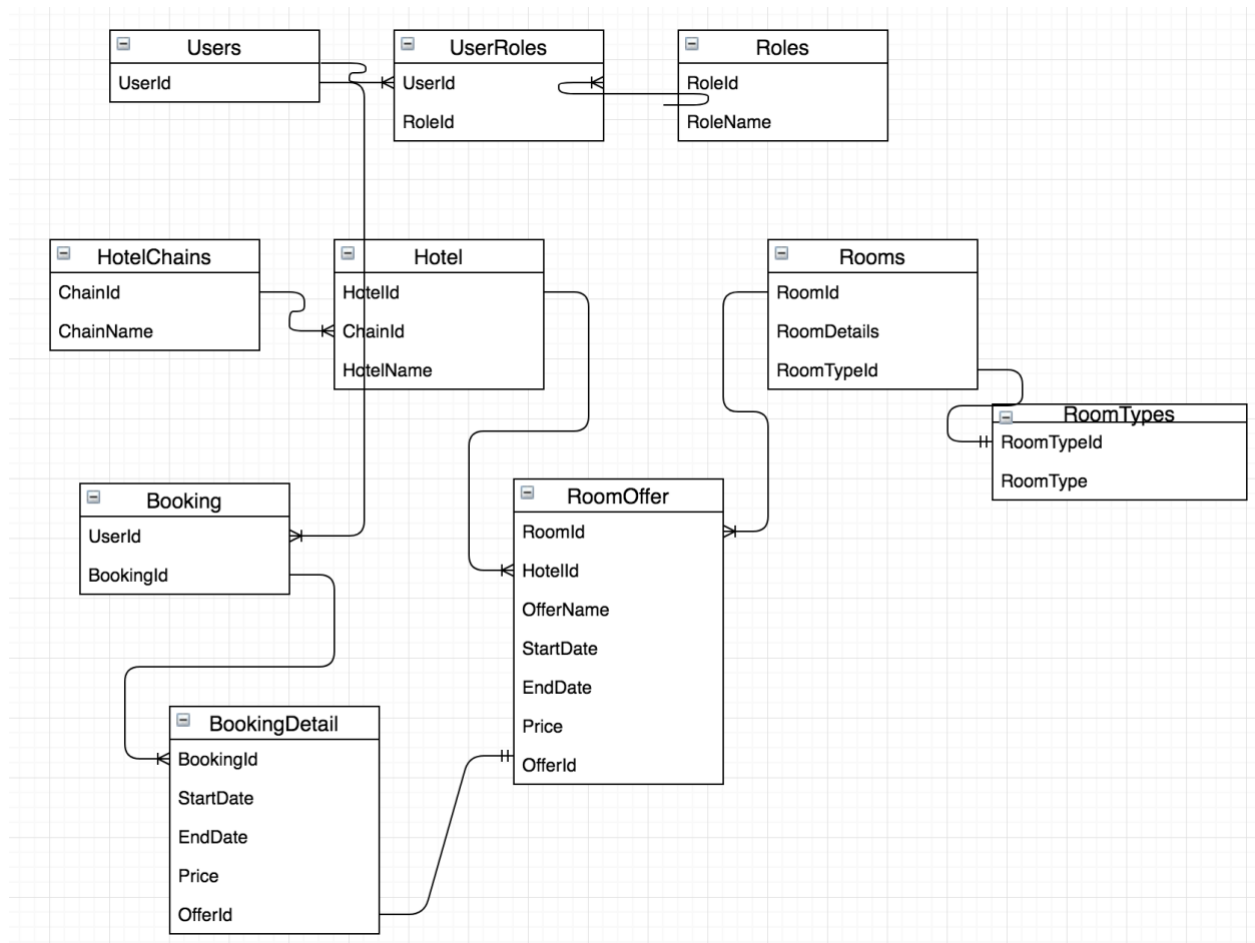
## 1.2 Class Design



Online Hotel Booking Application	Version: 1.1
	Date: 08/05/2018
Proj-doc	

## 2. Data Model

Below I present the data model diagram. This was used to design the application around it.



## 3. Unit Testing

Unit testing does have steep learning curve. The development team needs to learn what unit testing is, how to unit test, what to unit test and how to use automated software tools to facilitate the process on an on-going basis. The great benefit to unit testing is that the earlier a problem is identified, the fewer compound errors occur. A compound error is one that doesn't seem to break anything at first, but eventually conflicts with something down the line and results in a problem.

Online Hotel Booking Application	Version: 1.1
	Date: 08/05/2018
Proj-doc	

## IV. Construction and Transition

### 1. System Testing

The unit test framework that I use and Visual Studio IntelliSense will guide me through writing the code for your unit tests for a code project. To run in Test Explorer, most frameworks require that you add specific attributes to identify unit test methods. The frameworks also provide a way—usually through assert statements or method attributes—to indicate whether the test method has passed or failed. Other attributes identify optional setup methods that are at class initialization and before each test method and teardown methods that are run after each test method and before the class is destroyed.

The AAA (Arrange, Act, Assert) pattern is a common way of writing unit tests for a method under test.

The Arrange section of a unit test method initializes objects and sets the value of the data that is passed to the method under test.

The Act section invokes the method under test with the arranged parameters.

The Assert section verifies that the action of the method under test behaves as expected.

### 2. Future improvements

#### Browser Support

Though ASP .NET MVC works flawlessly on most of the popular browsers such as Chrome, Opera and Samsung Internet it is still unsupported completely in browsers like Safari.

#### Device Support

ASP .NET MVC works flawlessly on most of the android devices but it is still not completely supported with iOS devices. However, acknowledging the potential and popularity of this type of web applications, it is speculated that Apple is working on to make it more supported on iOS too.

## V. Bibliography

<https://msdn.microsoft.com/>

<https://demos.devexpress.com/mvc/>