<Hotel management>

Analysis and Design Document

Student: Saratean Timeea

**Group: 30233**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <23/04/2019> | <1.0> | First two capitols of this doc | Saratean Timeea |
| <05/05/2019> | <2.0> | Iteration 1.2 | Saratean Timeea |
| <29/05/2019> | <3.0> | Iteration 2 | Saratean Timeea |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams, Package diagram 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

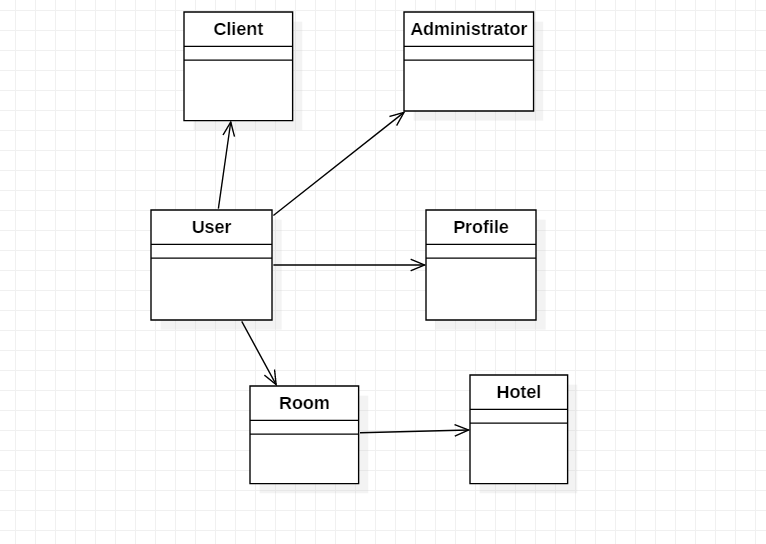
VI. Bibliography 5

# Project Specification

# This is a web project created for every person who wants to book a room to a hotel, but more useful to the hotel manager because it helps him keeping track of empty room, prices, offers.

# Elaboration – Iteration 1.1

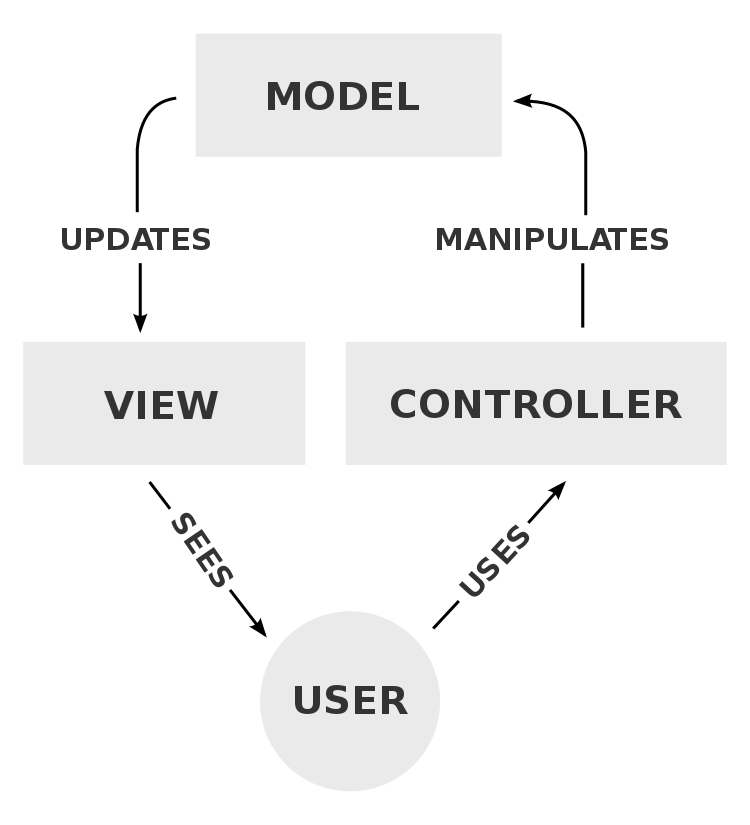
# Domain Model



# Architectural Design

## Conceptual Architecture

**Model–View–Controller** (usually known as MVC) is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to and accepted from the user. The MVC design pattern decouples these major components allowing for efficient code reuse and parallel development.

****

**Model**

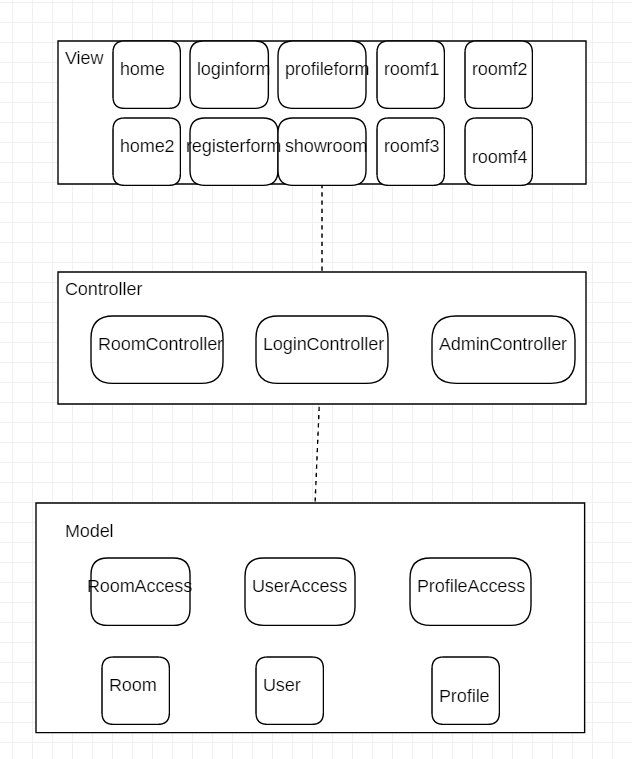
The central component of the pattern. It is the application's dynamic data structure, independent of the user interface.It directly manages the data, logic and rules of the application.

**View**

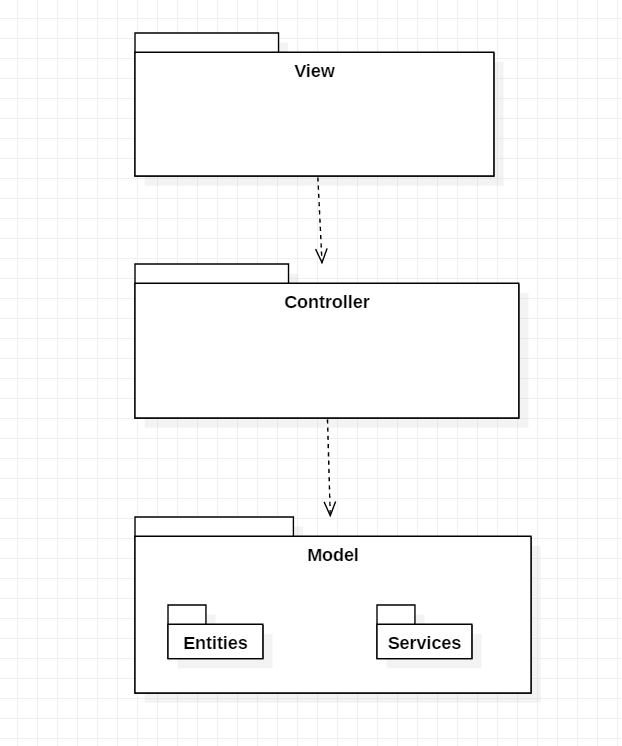
Any representation of information such as a chart, diagram or table. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.

**Controller**

Accepts input and converts it to commands for the model or view.

****

## Package Design



## Component and Deployment Diagrams

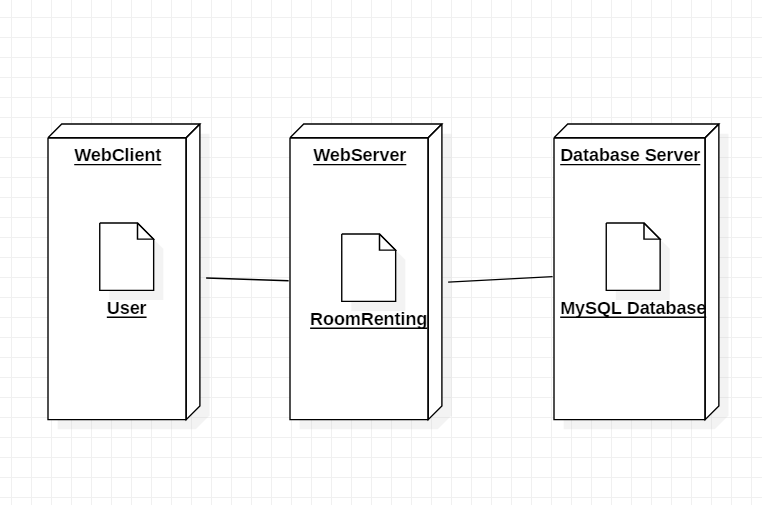
A component is a code module. Component diagram are physically analogs of class diagram.

UML Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

# 

Deployment diagram:

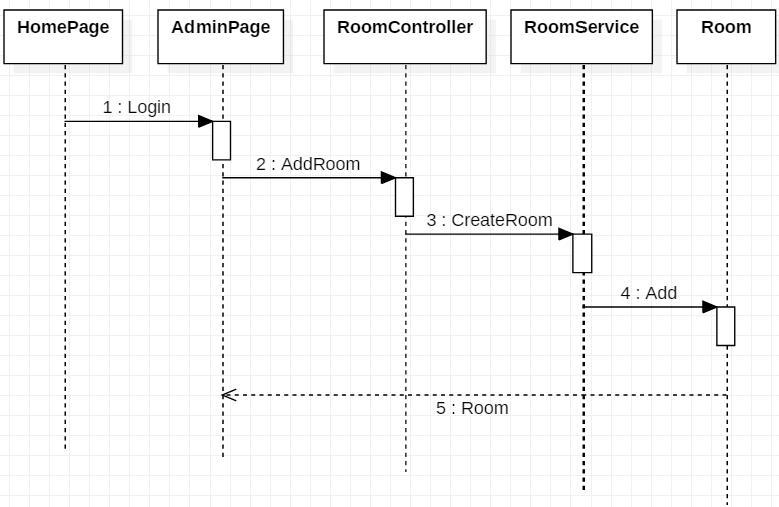
Deployment diagrams show their physical configurations of software and hardware. Deployment Diagram show the structure of the run-time system, capture the hardware that will be used to implement the system and the links between different items of hardware.



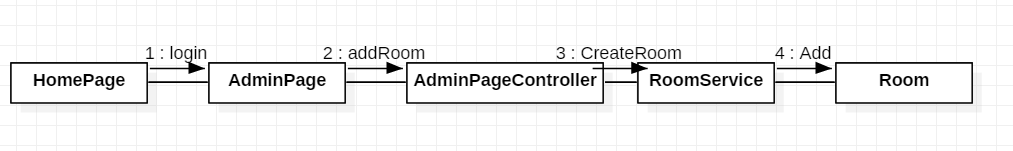
# Elaboration – Iteration 1.2

# Design Model

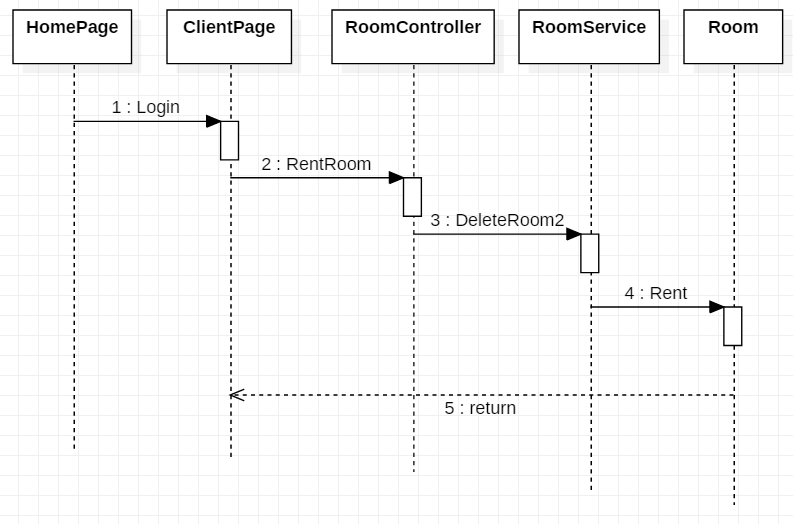
## Dynamic Behavior

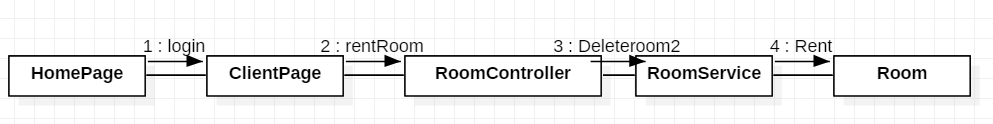
1.Adding a new room

## 

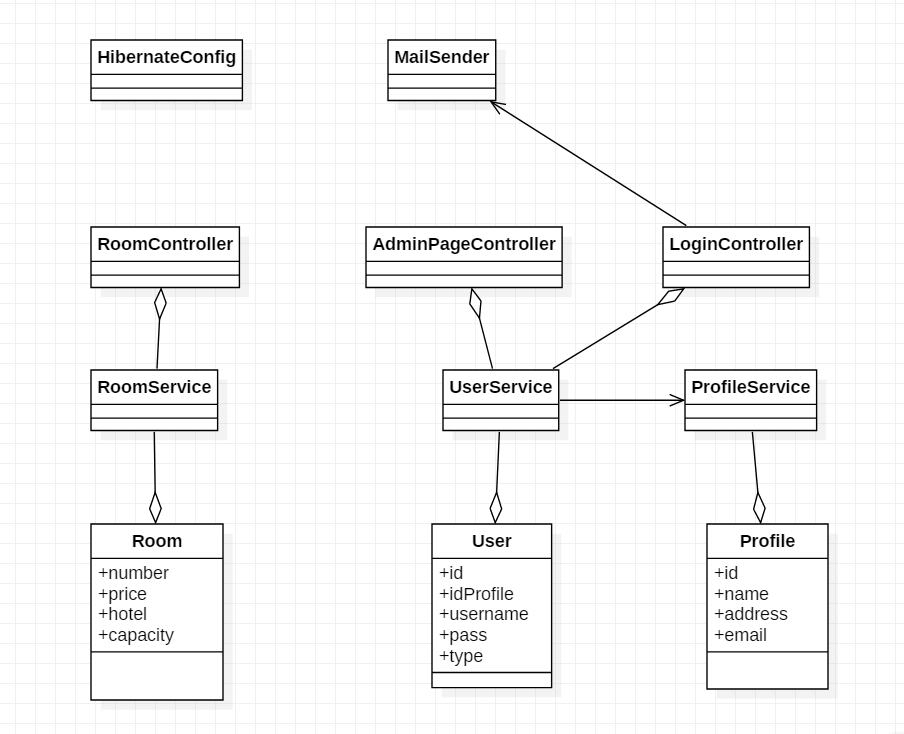


1. Rent a room

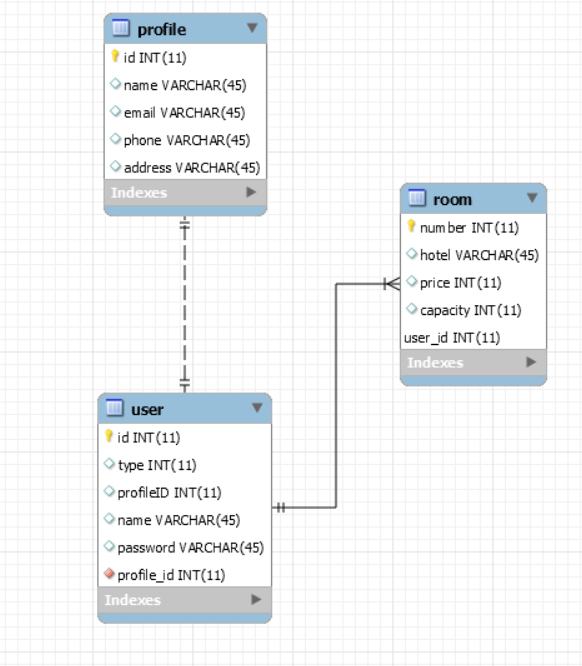




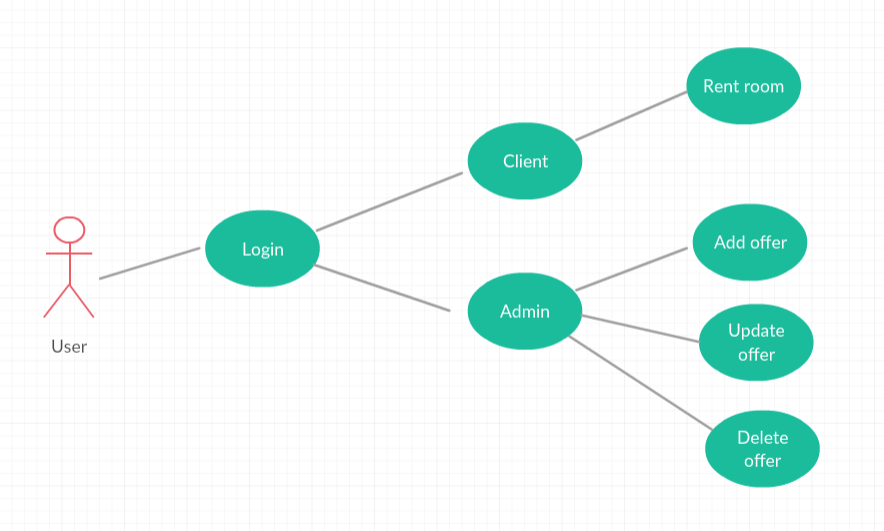
## Class Design



# Data Model

**

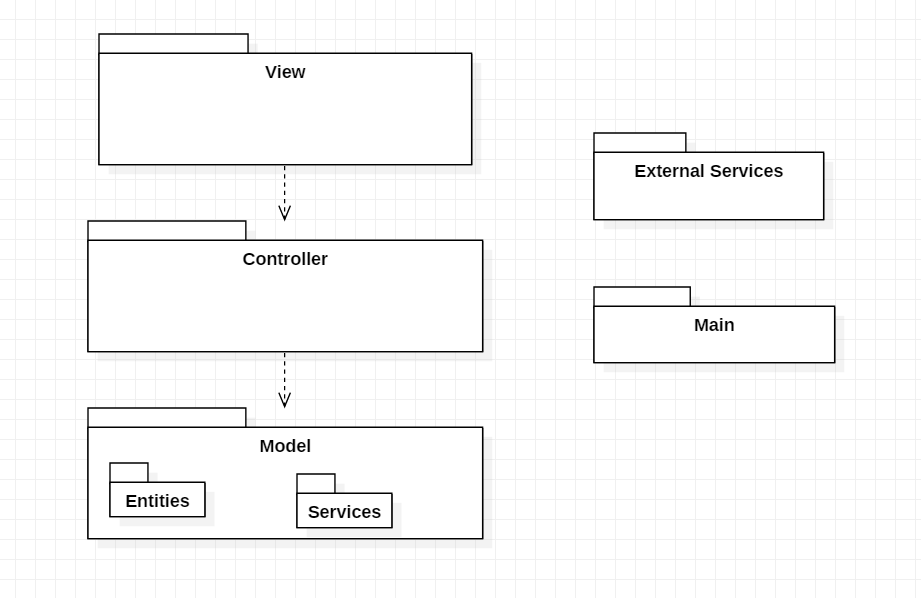
# Unit Testing

**

# Elaboration – Iteration 2

# Architectural Design Refinement

Below I have the refined package design diagram, it suffers some modifications regarding to structure and adding some others packages like External Services and Main.

****

# Design Model Refinement

## The modification made was adding the Mail Sender Class for sending you a mail when you register to the application.

# Construction and Transition

# Future improvements

In terms of future improvements, the app could use an upgrade on the visual side, something to make it look more user friendly and modern. Another improvement can be adding more hotels, so you can choose the hotel you want to stay in.

# Bibliography

<https://www.uml-diagrams.org/communication-diagrams.html>

<https://www.javatpoint.com/spring-mvc-tutorial>