Hotel Manager

Analysis and Design Document

Student: Stef Paula

**Group:30231/2**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 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

Design and implement a client-server application for the employees of a hotel. The application should have two types of users (a regular user represented by the receptionist, and an administrator user) which have to provide a username and a password in order to use the application.

The regular user can perform the following operations:

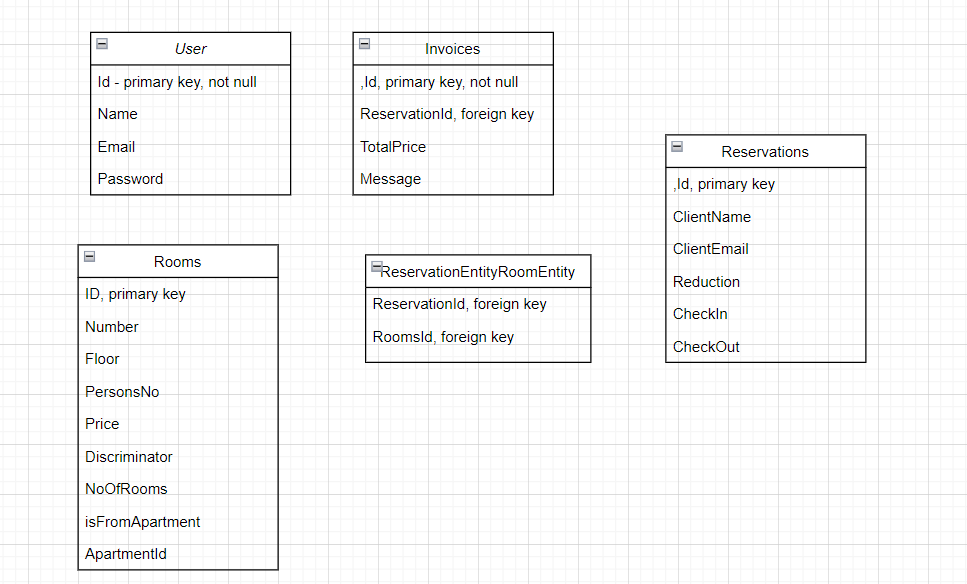
* Get estimate price. If a client has reserved a room several times in the past automatically apply a reduction for loyalty.
* Book rooms for clients.
* Cancel room bookings.
* Update room bookings
* Issue invoices
* Send mail to client with invoices
* See all rooms.
* See reservations for rooms

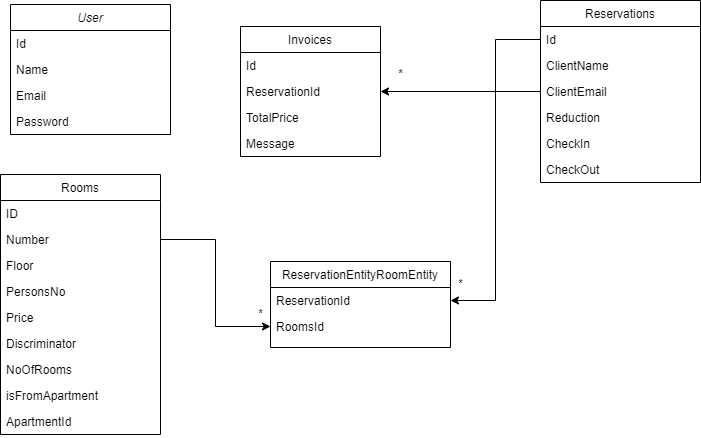
The administrator can perform the following operations:

* CRUD operations on user accounts.
* Add rooms/apartments
* Update price for rooms/apartments
* Issue reports in different formats (*xml*, *txt*) with the room bookings from a chosen date interval, the clients that have the largest number of reservations, invoices from current month etc.

# Elaboration – Iteration 1.1

# Domain Model



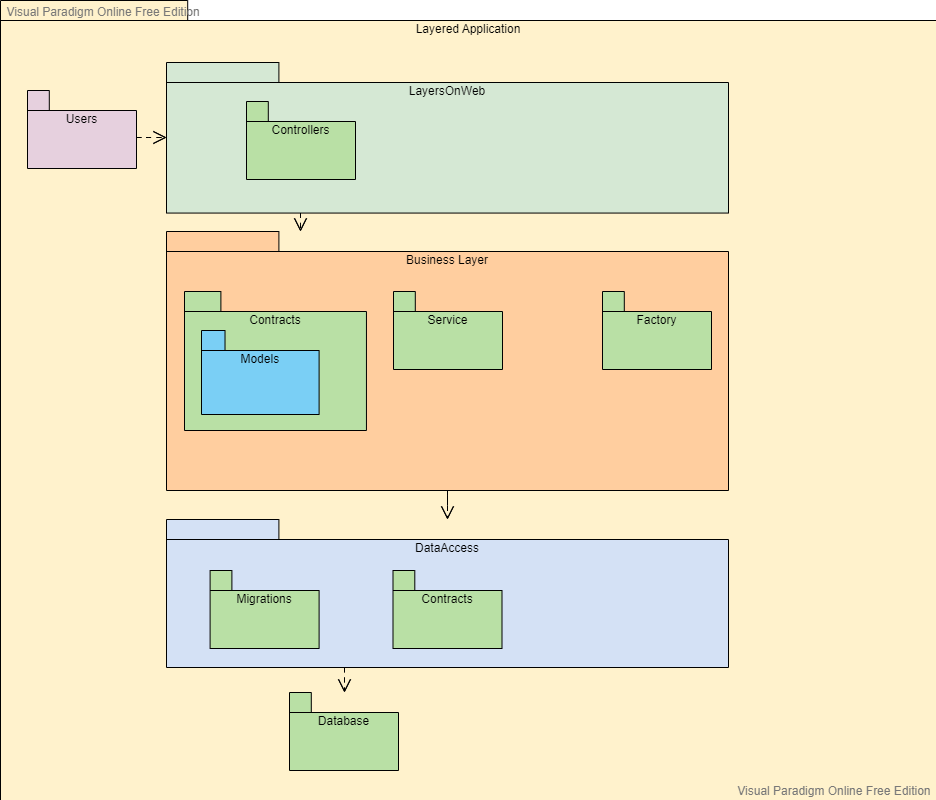
**

# Architectural Design

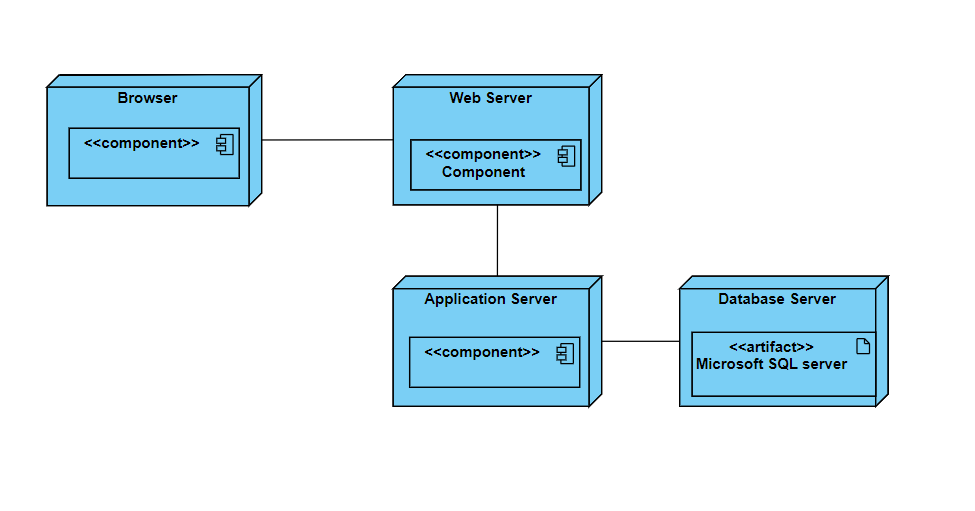
## Conceptual Architecture

## Blog API Layers, with Article components

## Package Design



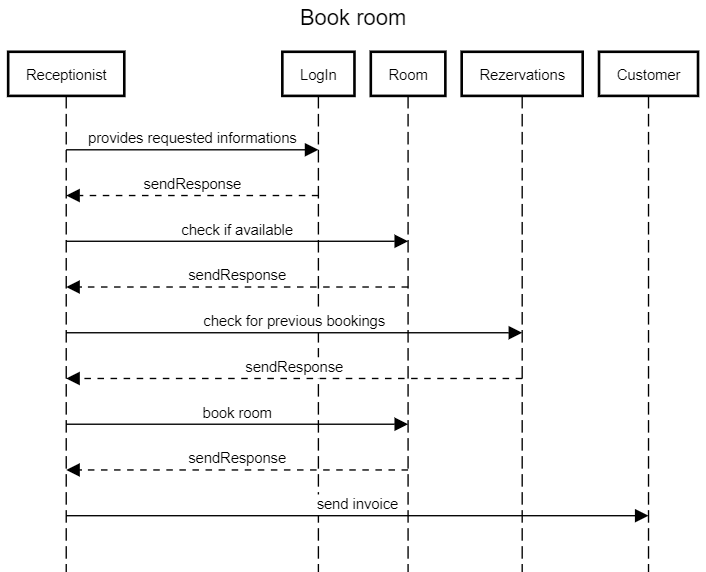
## Component and Deployment Diagrams

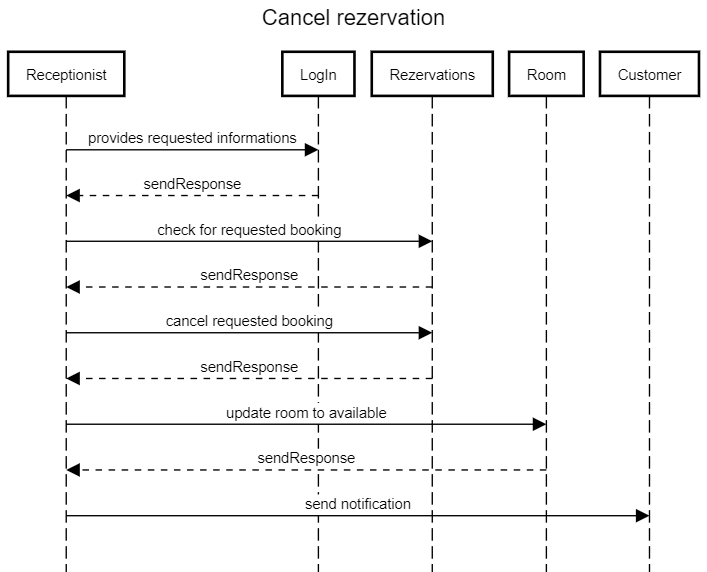


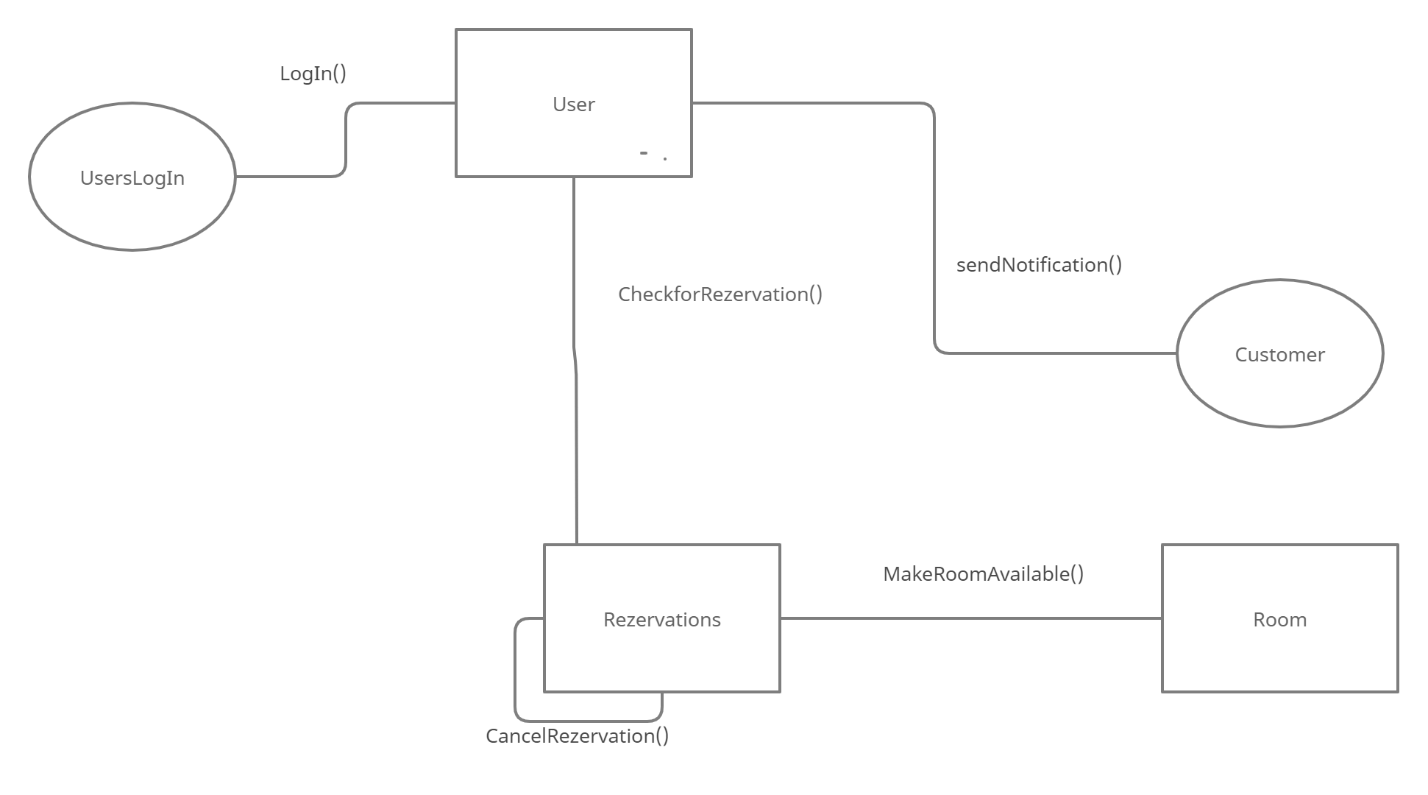
# Elaboration – Iteration 1.2

# Design Model

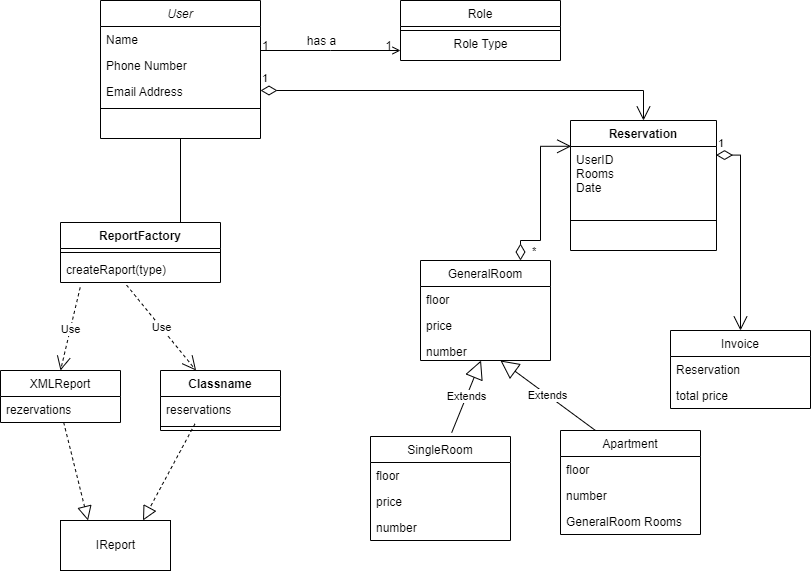
## Dynamic Behavior

**

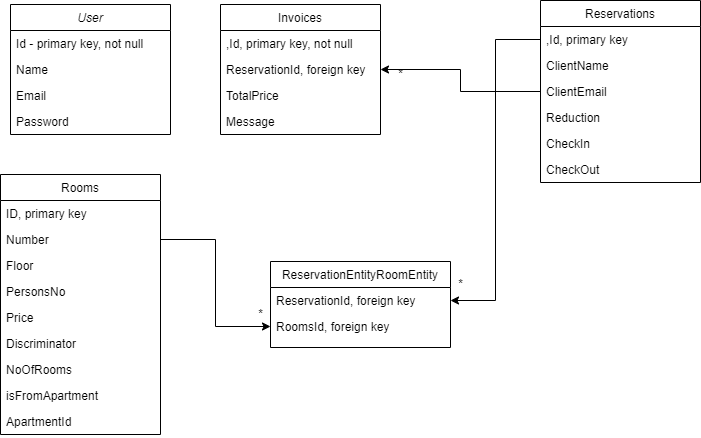




## Class Design

**

# Data Model

**

# Unit Testing

Test Scenario 1: Check the Login Functionality

Check system behavior when valid email and password is entered.

Check system behavior when invalid email id and valid password is entered.

Check system behavior when valid email id and invalid password is entered.

Check system behavior when invalid email id and invalid password is entered.

Check system behavior when email id and password are left blank.

Test Scenario 2: Check the Booking Functionality

Check system behavior when room is already booked.

Check system behavior when customer has reduction.

Check system behavior when customer doesn’t have reduction.

Check system behavior when customer books more rooms.

Test Scenario 3: Check the Issue Reports Functionality

Check system behavior when exporting xml document.

Check system behavior when exporting csv document.

Check system behavior when exporting report for costumers.

Check system behavior when exporting report with invalid date.

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

*[Present future improvements for the system]*

# Bibliography