**Table Of Contents**

**1.Introduction**

**2.Overview**

**3.Requirement Specification**

**3.1. Functional Requirements**

**3.2. Non-Functional Requirements**

**4. System Model**

**4.1. Use Case Model**

**4.2. Dynamic Models**

**4.3. Object and Class Model**

**5. User Interface**

**6.Important Decisions In overall Analysis**

**7.Conclusion**

**8.References**

**1.INTRODUCTION**

In today’s intense world, sometimes it may be hard to understand that the hotel you have been interested in is whether fully suitable for you or not. The most important factors while adjusting your hotel is related to the hotel’s joy and demand satisfying facilities. Our project has make possible to stay customers from every budget possible. In the way that our program do is after choosing number of people who are going to accomodate can increase the price for a specific room. If you choose just to stay in the room with basic form of it you pay less, otherwise with additional features like minibar, restaurant, housekeeping, laundry etc. you can stay in a luxury form. This property makes our program better, because for a good and peaceful holiday you should not be regret from the money that you have paid. In many booking system you can only choose your room with fix features, it becomes hard to change the money that you are going to pay, if it is not friendly with your budget you even have to change the hotel you choose. In addition, because we believe that the more simple of the program is more easy to understand, we specially focus on the plainness of the program. Therefore, we want to create a hotel reservation system that allows user to make their booking with pleasure and and easy way.

The report structure is the following: Section 2 describes the case description. Section 3 includes the functional, non-functional requirements and constraints of the system. Section 4 covers the Use Case models and their descriptions. User interface is shown in Section 5. Section 6 shows class diagrams, dynamic models of the system which are state chart and sequence diagram with scenarios.

**2.OVERVIEW**

The system has three basic operation, which are make booking, specification of room features and cancel booking. These operations can be used by customer, who wants to stay in the hotel and also receptionist working for the hotel. In the hotel, there are 7 floors and at the each floor there are 20 rooms. Single rooms are in the first floor, double rooms are in the second floor so on. And also, odd numbered rooms have sea view and even numbered rooms have forest view.

Make Booking Part: When the user makes booking, first of all selects the check in and out dates and numbers of rooms before checking the number of people in the room/rooms. Therefore, user can make booking for multiple rooms in the same date range.

If there is an available room that has specified features, program passes the next step and can continue to pay the price or make the reservations for specified rooms.

If there is no available room, the user see a warning message that states desired room is not possible to make the reservation and direct his/her to the main page.

Specifications of Room Features Part: As mentioned before, users can add some room features or other type of services by themselves and according to this process, new price is going to be computed. However, just user’s demand is not going to be enough to make changes in room features, the receptionists should also give an approval to that request.

Cancel Booking Part: Customer has a right to cancel his/her booking. Additionally, receptionist can cancel a booking instead of customer or by the way of moving all the reservations for a specific room to another one, the receptionist can cancel all the reservations for a specific room. And lastly, the receptionist can cancel all the reservations in the hotel.

**3. REQUIREMENT ANALYSIS**

**3.1. Functional Requirements**

In this system, there two types of users, which are customer and receptionist. They have different roles and authorisations on this system.

***For Customer:***

•   The customer should be able to select check in/out dates and number of people in rooms.

•   The customer should be able to make booking at most three rooms for the same date range.

•   The customer should be able to informed about rooms in hotel and select a room or rooms.

•   The customer should be able to select features to his/her room.

•   The customer should be informed about his/her reservation.

•   The customer should be able to cancel his/her reservation.

•   The customer should be able to make his/her payment.

•

***For Receptionist:***

•   The receptionist should be able to make booking for a customer.

•   The receptionist should be able to check and give approval to a room’s features that determined by customer.

•   The receptionist should be able to view information about a specific customer and     also about all customers in the rooms.

•   The receptionist should be able to cancel a reservation.

•   The receptionist should be able to view all rooms.

•   The receptionist should be able to view all reservations made for a specific room.

•   The receptionist should be able to move all reservations of any room to another room that fits with it, and delete the old room’s all reservations.

•   The receptionist should be able to make payment for a reservation of a customer.

•   The receptionist should be able to move a certain reservation for a room to another room.

**3.2. Non-Functional Requirements**

•   The system should response quickly at each step (1 second), because the main purpose is to make reservation quickly and efficiently.

•   Customer should be informed about other choices.

For example, if there is no available double room with sea view, other alternatives can be shown to the customer like double room with forest view instead of a sea view.

•   Program should be learned quickly by customers and receptionist, and used easily.

•   Extendibility : In software engineering, reusability and extendibility are two importance concepts. We plan to make Hotel Reservation System suitable to be extended and re-used for future works.

**3.3 Constraints**

* The program must be a desktop application.
* The program must be implemented within six weeks.

**3.4 Scenarios**

**Scenario Name**: Make booking

**Scenario**: Ahmet wants to make a booking for himself and his friends for multiple rooms. He starts to use the system first of all, he selects the check in/out dates and number of rooms and types of rooms to determine the specifications. After all selections have been done, if there is not any available room, he will see a warning message related to its unavailability. Otherwise, Ahmet can make his reservations if there is an available room his desired type. And also, if Ahmet does not want to make a reservation after seeing the price he can delete all the choices he made and can turn back to the main page. However, if he is comfortable with the price he can make the payment. After all the operations, he is going to be informed about his reservations and rooms’ features, and make booking is finished.

**Scenario Name:** Make payment

**Scenario:** Mehmet came from the work very tired and decided to go on a vacation. After making the reservation via the system, he also wants to the make the payment. However, he is shocked with the insufficient credit card limit warning.

**Scenario Name:** Add specifications

**Scenario:** Murat wants to make booking for a double room. He selects check in/out dates and number of people in the room, and then he chooses an available room from the hotel. However, he decides to add new feature to his room. He selects the mini-bar option and cancel the housekeeping service, then his reservation is approved by the receptionist, and booking is finished, and Murat is informed about his reservation.

**Scenario Name**: Cancel booking

**Scenario:** Eda was made booking in hotel, but she cannot go holiday in specified dates because of her job. Therefore, she enters the system and cancels her current booking and add new booking for herself.