

**Hotels Management System Online**

**Student’s Name**

Abdulrahman Mohammed Shaiban

Emad Abdulkarim Alareefi

**Supervisor**

Dr. Asma’a Al-Sharjabi

**Declaration**

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Bachelor of Science in *(Information Technology)* is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

**Signed:** Emad Alareefi

**Registration No.:** \_\_\_\_\_\_\_\_\_\_\_

**Date:** Day, xx Feb 2018.

**Abstract**

Table of Contents

[Chapter 1 2](#_Toc506099485)

[1.1 Project Title 3](#_Toc506099486)

[1.2 Introduction 3](#_Toc506099487)

[1.3 Problem Statement 3](#_Toc506099488)

[1.4 Project Objectives 3](#_Toc506099489)

[1.5 Project Scope 4](#_Toc506099490)

[1.6 Project Methodology 4](#_Toc506099491)

[1.7 Tools 5](#_Toc506099492)

[1.8 Limitation 5](#_Toc506099493)

[Chapter 2 6](#_Toc506099494)

[2.1 Introduction 7](#_Toc506099495)

[2.2 Nazeel Management System 7](#_Toc506099496)

[**2.2.1 Leasing** 7](#_Toc506099497)

[**2.2.2 Furniture** 7](#_Toc506099498)

[**2.2.3 Customers** 7](#_Toc506099499)

[**2.2.4 Services** 7](#_Toc506099500)

[**2.2.5 Bills** 7](#_Toc506099501)

[**2.2.6 SMS** 8](#_Toc506099502)

[**2.2.7 Reports** 8](#_Toc506099503)

[**2.2.8 Configuration** 8](#_Toc506099504)

[2.3 Conclusion 9](#_Toc506099505)

# Chapter 1

**Introduction**

## 1.1 Project Title

Hotels Management System Online.

## 1.2 Introduction

Technology has spread so fast in the last 20 years. Due to this revelation in technology, organizations nowadays tend to move its functions towered this revolution. Most organizations today have huge systems to manage their business processes.

Due to the huge growth in terrorism and the huge needs for hotels to be around, Hotels Management System is a huge case to be considered. We see hotels get belt so fast every day and Management systems are not easy to get or access.

We thought of building a Hotel`s management system that can be easy to access and doesn’t cost a fortune.

## 1.3 Problem Statement

Most hotels management systems are hard to get and too expensive.

If we see hotels as users, we might build an easy access management system that provide all functions required inside a hotel.

The system will be much cheaper than the regular systems because it will be accessed online, and it will be easy to grow it in the future.

## 1.4 Project Objectives

* The system will be accessed over the internet.
* Lower the cost of such systems.
* The system will provide all the functions that the hotel needs effectively.

## 1.5 Project Scope

**1.5.1 Geographical Scope:**

* This system can be used in any hotel across the world.

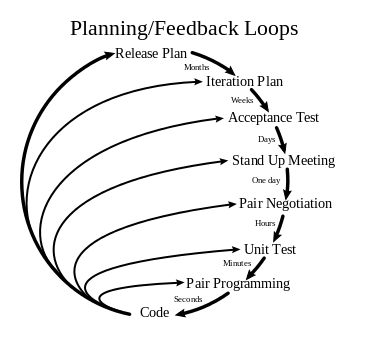
**1.5.1 Functional Scope:**

* The system will manage and monitor rooms.
* The system will be accessed online.
* The system will provide all the reports needed.
* Every hotel will have his own database in the server.

## 1.6 Project Methodology

The methodology that will be used in this project is an extreme programming methodology.

One of the reasons for choosing this methodology is that the development team doesn’t has an expert analyst and must solve the problems programmatically.



## 1.7 Tools

* Visual Studio 2015
* SQL Server 2016
* Git
* GitHub
* Google Chrome
* Notepad++

## 1.8 Limitation

# Chapter 2

**literature review**

## 2.1 Introduction

Since this project is based on building a new system we will be examining an existing system and try to find its draw backs. The reason behind this review is to find a better and faster solutions for problems to apply it to the new system.

## 2.2 Nazeel Management System

Nazeel is an online management system for hotels. Hotels must buy the system in order to use it – it does not have a trail.

### **2.2.1 Leasing**

This system allows hotels to:

* Show the user the available rooms in order to check in.
* Show the user the rented rooms in order to check out.
* Allows user to change the states of the rooms.

### **2.2.2 Furniture**

The system allows furniture tracking for each room inside the hotel. The user can add, modify or remove any peace of a furniture.

### **2.2.3 Customers**

The system has a customer’s module to manage new or existing customers.

### **2.2.4 Services**

The system allows residents to order any kinds of services and the system will automatically add the charge to the bill of a certain resident.

### **2.2.5 Bills**

The system has an entire billing module:

* Exchange Bonds
* Banknotes
* Services
* Receipts

### **2.2.6 SMS**

The system is integrated with a third-party SMS provider to allow hotels to use the service for their residents. The charges for each message is explained in the front page of the system.

### **2.2.7 Reports**

The system provides the following reports:

* Report the movement of apartments
* Total monthly report
* Fund Movement Report
* Report receivable bonds
* Bills Report
* Report on exchange items
* Report of service bonds
* Report of staff statistics
* Customer Reservation Report
* Change the apartments report
* Message summary report
* Open Reservations Report
* Annual report of months

### **2.2.8 Configuration**

The system has a gate for all possible configuration:

* Prices of apartments
* Manage users
* Messaging settings
* Peak time settings
* Booking settings
* Company data settings
* Conditions of Contract
* Settings of exchange items
* Accounts
* Services Management
* Names settings for Windows
* Change the price by type
* Customer rating

## 2.3 Conclusion

The system is well designed and has a lot of features, although it lacks the globalization. The system is located on the cloud, yet, it is not integrated with other systems like laundries and restaurants. The system also lacks a parking module since some hotels have a large parking floors.

The goal here is to build a larger system that can also integrate with other systems that interact with the hotel.

# Chapter 3

**Analysis**

* **User requirements** 
  + **Functional Requirements**
  + **Non-Functional Requirements**
* **System Requirements** 
  + **Functional Requirements**
  + **Non-Functional Requirements**
* **Use Case**
* **Use Cases Description**
* **Entity Relationship Diagram**
* **Class Diagram**

## User Requirements

### Functional Requirements

1. The system must manage all the data in the system.
2. The system must allow the admin to add and show:

* hotel`s rooms.
* hotel`s services.
* hotel`s Purchases.
* hotel`s reports.
* hotel`s bonds.

1. The system must allow the admin to add food in hotel`s restaurant.
2. The system must allow the admin to show room`s cases.
3. The system must allow the admin to add Reservation.
4. The system must allow the admin to send SMS messages to customer.
5. The system must send the report to the hotel manager on demand.

### Non-Functional Requirements

1. The system must be fast.
2. Usability.
3. High security.
4. Maintainability.
5. Efficiency.

## System Requirements

### Functional Requirements

1. The system must manage all the data in the system.
2. The system consists of several interfaces every interface has specific task.
3. The system must allow the admin to add and show:

* hotel`s rooms.
* In this part the system will provide interface to add and show hotel`s rooms to and from database.
* hotel`s services.
* In this part the system will provide interface to add and show hotel`s services to and from database.
* hotel`s Purchases.
* In this part the system will provide interface to add and show hotel`s Purchases to and from database.
* hotel`s reports.
* In this part the system will provide interface to add and show hotel`s reports to and from database.
* hotel`s bonds.
* In this part the system will provide interface to add and show hotel`s bonds to and from database.

1. The system must allow the admin to add food in hotel`s restaurant.
2. There is interface in the system about the hotel`s restaurant in this interface you can add the restaurant`s food.
3. The system must allow the admin to show room`s cases.
4. There is a specific interface to show the room`s cases, there is three types of cases:
5. Busy.
6. Empty.
7. Need maintain or clean.
8. The system must allow the admin to add Reservation.
9. The admin can add Reservation in the empty rooms after show room cases.
10. The system must allow the admin to send SMS messages to customers.
11. The admin can send SMS messages to customers to tell them about the important things.
12. The system must send the report to the hotel manager on demand.
    * + 1. if the hotel manager requests any type of report the admin will send it to him.

### Non-Functional Requirements:

1. Speed:

* The system must be fast in work.

1. Usability:

* The system must be easy to use.

1. High security:

* The system must have high security.

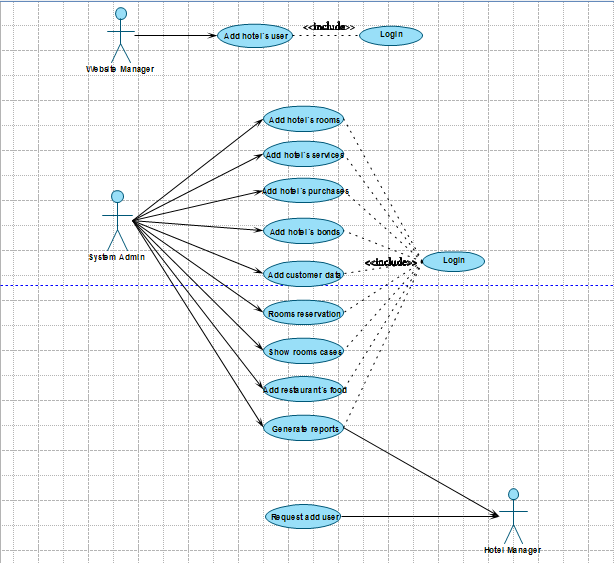
1. Maintainability:

* The system must be Maintainable to develop the system in the future.

1. Efficiency:

* The system must work 100% without any errors.

## Use Case



## Use Case Description

### Add hotel user.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 1 | Use Case Name: Add hotel user | |
| Use Case Type: Essential | | | Primary Actor: Website manager |
| Stakeholders:  Website manager | | | |
| Brief Description: This use case describes how the system add hotel user. | | | |
| Pre-condition: user open the website and request add user. | | | |
| Post-condition : ----- | | | |
| Trigger: user open the website and request add user.  Type: External | | | |
| Relationships:  Include: Login.  Extend: | | | |
| Normal Flow of Events:   1. User will open the website and request add user [E1]. 2. The website manager will receive the request and open a new user. | | | |
| Alternate/Exceptional Flows:  [E1] the user want to see the system before add a user:  1. Website manager will give him experimental user to see the system. | | | |

### Request add user.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 2 | Use Case Name: Request add user | |
| Use Case Type: Essential | | | Primary Actor: Website manager |
| Stakeholders:  Hotel manager | | | |
| Brief Description: This use case describes how the user Request add user. | | | |
| Pre-condition: user open the website and want to get a user. | | | |
| Post-condition : ----- | | | |
| Trigger: user open the website and request a user.  Type: External | | | |
| Relationships:  Include:  Extend: | | | |
| Normal Flow of Events:   1. User will open the website and want to get a user[E1]. 2. The user will request add user. | | | |
| Alternate/Exceptional Flows:  [E1] the user want to see the system before add a user:  1. Website manager will give him experimental user to see the system. | | | |

### Add hotel`s rooms.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 3 | Use Case Name: Add hotel`s rooms | |
| Use Case Type: Essential | | | Primary Actor: System admin |
| Stakeholders:  System admin | | | |
| Brief Description: This use case describes how the admin add hotel`s rooms. | | | |
| Pre-condition: admin login to the system | | | |
| Post-condition : ----- | | | |
| Trigger: admin open the interface of add hotel`s rooms.  Type: internal | | | |
| Relationships:  Include: login.  Extend: | | | |
| Normal Flow of Events:   1. The admin will login to the system. 2. The admin will open the interface of add hotel`s rooms. 3. The admin will add hotel`s rooms . | | | |

### Add hotel`s purchases.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 4 | Use Case Name: Add hotel`s purchases. | |
| Use Case Type: Essential | | | Primary Actor: System admin |
| Stakeholders:  System admin | | | |
| Brief Description: This use case describes how the admin add hotel`s purchases. | | | |
| Pre-condition: admin login to the system | | | |
| Post-condition : ----- | | | |
| Trigger: admin open the interface of add hotel`s purchases.  Type: internal | | | |
| Relationships:  Include: login.  Extend: | | | |
| Normal Flow of Events:   1. The admin will login to the system. 2. The admin will open the interface of add hotel`s purchases. 3. The admin will add hotel`s purchases. | | | |

### Add hotel`s services.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 5 | Use Case Name: Add hotel`s services. | |
| Use Case Type: Essential | | | Primary Actor: System admin |
| Stakeholders:  System admin | | | |
| Brief Description: This use case describes how the admin add hotel`s services . | | | |
| Pre-condition: admin login to the system | | | |
| Post-condition : ----- | | | |
| Trigger: admin open the interface of add hotel`s services .  Type: internal | | | |
| Relationships:  Include: login.  Extend: | | | |
| Normal Flow of Events:   1. The admin will login to the system. 2. The admin will open the interface of add hotel`s services. 3. The admin will add hotel`s services . | | | |

### Add hotel`s bonds.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 6 | Use Case Name: Add hotel`s bonds. | |
| Use Case Type: Essential | | | Primary Actor: System admin |
| Stakeholders:  System admin | | | |
| Brief Description: This use case describes how the admin add hotel`s bonds. | | | |
| Pre-condition: admin login to the system. | | | |
| Post-condition : ----- | | | |
| Trigger: admin open the interface of add hotel`s bonds.  Type: internal | | | |
| Relationships:  Include: login.  Extend: | | | |
| Normal Flow of Events:   1. The admin will login to the system. 2. The admin will open the interface of add hotel`s bonds. 3. The admin will select the customer to add bonds. 4. The admin will add bonds to the customer. | | | |

### Add food to hotel`s restaurant.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 7 | Use Case Name: Add food to hotel`s restaurant. | |
| Use Case Type: Essential | | | Primary Actor: System admin |
| Stakeholders:  System admin | | | |
| Brief Description: This use case describes how the admin add restaurant `s food. | | | |
| Pre-condition: admin login to the system. | | | |
| Post-condition : ----- | | | |
| Trigger: admin open the interface of add restaurant `s food .  Type: internal | | | |
| Relationships:  Include: login.  Extend: | | | |
| Normal Flow of Events:   1. The admin will login to the system. 2. The admin will open the interface of add restaurant `s food. 3. The admin will add restaurant `s food. | | | |

### Add customer data.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 7 | Use Case Name: Add customer data | |
| Use Case Type: Essential | | | Primary Actor: System admin |
| Stakeholders:  System admin | | | |
| Brief Description: This use case describes how the admin add customer user. | | | |
| Pre-condition: admin login to the system. | | | |
| Post-condition : ----- | | | |
| Trigger: the user will open the interface of adding data.  Type: internal. | | | |
| Relationships:  Include: login.  Extend: | | | |
| Normal Flow of Events:   1. The admin will login to the system. 2. The admin will open the interface of adding customer data. 3. The admin will add customer data. | | | |

### Rooms reservation.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 8 | Use Case Name: Rooms reservation. | |
| Use Case Type: Essential | | | Primary Actor: System admin |
| Stakeholders:  System admin | | | |
| Brief Description: This use case describes how the admin add reservation to rooms. | | | |
| Pre-condition: admin must add customer data. | | | |
| Post-condition : ----- | | | |
| Trigger: admin open the interface of add reservation.  Type: internal. | | | |
| Relationships:  Include: Login, add customer data.  Extend: | | | |
| Normal Flow of Events:   1. The admin will login to the system 2. The admin will open the interface of add rooms reservation. 3. The admin will show the rooms to knew the empty rooms. 4. The admin will add reservation to the rooms. | | | |

### Show room`s cases.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 9 | Use Case Name: Show room`s cases. | |
| Use Case Type: Details. | | | Primary Actor: System admin |
| Stakeholders:  System admin | | | |
| Brief Description: This use case describes how the admin show room`s cases. | | | |
| Pre-condition: admin must add hotel`s rooms. | | | |
| Post-condition : ----- | | | |
| Trigger: admin open the interface of room`s cases.  Type: internal. | | | |
| Relationships:  Include: Login.  Extend: | | | |
| Normal Flow of Events:   1. The admin will login to the system 2. The admin will open the interface of room's cases. 3. The admin will show the room`s cases. | | | |

### Generate reports.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 10 | Use Case Name: Generate reports. | |
| Use Case Type: Details. | | | Primary Actor: System admin |
| Stakeholders:  System admin, hotel manager. | | | |
| Brief Description: This use case describes how the admin send reports to hotel manager. | | | |
| Pre-condition: hotel manager request reports. | | | |
| Post-condition : ----- | | | |
| Trigger: admin open the interface of add reports.  Type: internal. | | | |
| Relationships:  Include: Login.  Extend: | | | |
| Normal Flow of Events:   1. The admin will login to the system 2. The admin will open the interface of add reports. 3. The admin will add new reports. 4. The admin will send the reports to hotel manager. | | | |

## Class Diagram

