

**Hotels Management System Online**

**Student’s Name**

Abdulrahman Mohammed Shaiban

Emad Abdulkareem 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:** Abdulrahman Shaiban , Emad Alareefi

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

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

**Abstract**

This document has been prepared by the students of Computer and Information Technology as a graduation project. The document aims at providing a full documentation for building an information system. All the chapters inside this document provide all the necessary information for developing an information system.

The system that will be developed based on this document is a Hotel`s Management System.

Table of Contents

[List of figures 3](#_Toc506274401)

[Chapter 1 4](#_Toc506274402)

[1.1 Project Title 5](#_Toc506274403)

[1.2 Introduction 5](#_Toc506274404)

[1.3 Problem Statement 5](#_Toc506274405)

[1.4 Project Objectives 5](#_Toc506274406)

[1.5 Project Scope 6](#_Toc506274407)

[1.6 Project Methodology 6](#_Toc506274408)

[1.7 Tools 7](#_Toc506274409)

[1.8 Limitation 7](#_Toc506274410)

[Chapter 2 8](#_Toc506274411)

[2.1 Introduction 9](#_Toc506274412)

[2.2 Nazeel Management System 9](#_Toc506274413)

[2.2.1 Leasing 9](#_Toc506274414)

[2.2.2 Furniture 9](#_Toc506274415)

[2.2.3 Customers 9](#_Toc506274416)

[2.2.4 Services 9](#_Toc506274417)

[2.2.5 Bills 9](#_Toc506274418)

[2.2.6 SMS 10](#_Toc506274419)

[2.2.7 Reports 10](#_Toc506274420)

[2.2.8 Configuration 10](#_Toc506274421)

[2.3 Conclusion 11](#_Toc506274422)

[Chapter 3 12](#_Toc506274423)

[3.1 User Requirements 13](#_Toc506274424)

[1.1.1 Functional Requirements 13](#_Toc506274425)

[1.1.2 Non-Functional Requirements 13](#_Toc506274426)

[3.2 System Requirements 14](#_Toc506274427)

[1.2.1 Functional Requirements 14](#_Toc506274428)

[1.2.2 Non-Functional Requirements: 15](#_Toc506274429)

[3.3 Use Case 16](#_Toc506274430)

[Use Case Description 17](#_Toc506274431)

[Add hotel use. 17](#_Toc506274432)

[Request add user. 18](#_Toc506274433)

[System configuration. 19](#_Toc506274434)

[Purchases management. 20](#_Toc506274435)

[Accounts management. 21](#_Toc506274436)

[Rooms reservation. 22](#_Toc506274437)

[Generate reports 23](#_Toc506274438)

[3.4 Class Diagram 24](#_Toc506274439)

List of figures

[Figure 1.1 - Methodology Planning Loops 5](#_Toc506237484)

# 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

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 in the future.

## 1.4 Project Objectives

* This project aims at developing a standard Hotel`s System with an accepted price.
* 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 (Agile XP 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.

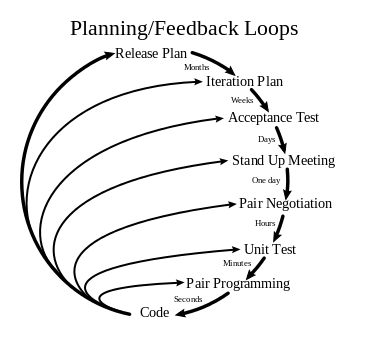


Figure 1.1 - Methodology Planning Loops

## 1.7 Tools

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

## 1.8 Limitation

The system won’t have an accounting module.

# 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. All the system`s features as described below.

### 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**

## 3.1 User Requirements

### 1.1.1 Functional Requirements

1. The system must allow the admin to configure hotel`s rooms.
2. The system must allow the admin to manage hotel`s Purchases.
3. The system must allow the admin to add hotel`s services.
4. The system must allow the admin to add hotel`s reports and send it to the hotel manager on demand.
5. The system must allow the admin to manage hotel`s accounts.
6. The system must allow the admin to add Reservation.
7. The system must allow the admin to send SMS messages to customer.

### 1.1.2 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 has high security.

1. Maintainability:

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

1. Efficiency:

* The system must work without errors.

## 3.2 System Requirements

### 1.2.1 Functional Requirements

1. The system must allow the admin to configure hotel`s rooms.
2. The system must provide interface to add hotel`s rooms.
3. The system must provide interface to show hotel`s rooms.
4. The system must allow the admin to manage hotel`s Purchases.
5. The system must provide interface to add and show the different type of hotel`s Purchases.

* Furniture.
* Material.
* Others.

1. The system must allow the admin to add hotel`s services.
2. The system must provide interface to add and show the different type of hotel`s services.

* Clean services.
* Maintenance services.
* Food services.
* Other services.

1. The system must allow the admin to add hotel`s reports and send it to the hotel manager on demand.
2. The system must provide interface to add and show the different type of hotel`s reports.

* Reservations report.
* Services report.
* Purchases report.
* Accounts report.
* Users report.
* Restaurants report.
* Customers report.

1. The system must allow the admin to manage hotel`s accounts.
2. The system must allow the admin to add Reservation.
3. Before adding reservations, the admin must add customers data.
4. The system must allow the admin to send SMS messages to customer.
5. The system must provide a third-party SMS provider for the admin to send to their residents.

### 1.2.2 Non-Functional Requirements:

1. Speed:

The system must be fast in work where each process will take 3 seconds at most.

1. Usability:

The system must be easy to use where is the training time will be three days at most.

1. High security:

* The system must have high security to protect the website from any external attack.

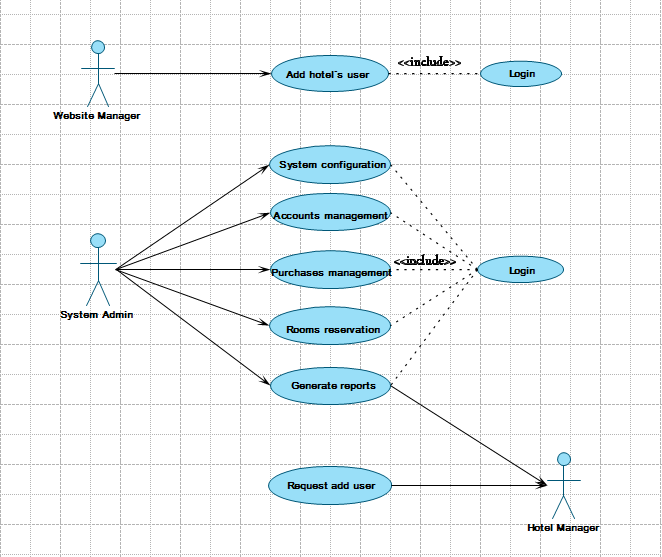
1. Maintainability:

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

1. Efficiency:

* The system must work 100% without any errors.

## 3.3 Use Case



## Use Case Description

### Add hotel use.

|  |  |  |  |
| --- | --- | --- | --- |
| 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. The user will open the website. 2. The user will request add user [E1]. 3. The website manager will receive the request. 4. The website manager will open a new user. | | | |
| Alternate/Exceptional Flows:  [E1] If the user want to see the system before get a user:  1. The website manager will give him an 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: Hotel 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. The user will open the website. 2. The user want to get a user [E1]. 3. The user will request add user. | | | |
| Alternate/Exceptional Flows:  [E1] If the user want to see the system before add a user:  1. The website manager will give him an experimental user to see the system. | | | |

### System configuration.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 3 | Use Case Name: System configuration | |
| Use Case Type: Essential | | | **Primary Actor: System admin** |
| Stakeholders:  System admin | | | |
| Brief Description: This use case describes how the admin configure the system. | | | |
| Pre-condition: admin login to the system | | | |
| Trigger: admin open the configuration interface.  Type: internal | | | |
| Relationships:  Include: login. | | | |
| Normal Flow of Events:   1. The admin will login to the system. 2. The admin will open the interface of configuration. 3. The system will show the different kind of configuration. 4. The admin will choose the kind of configuration. 5. if he choose Add hotel`s rooms [A1] , Add hotel`s services [A2] , Add restaurant`s food [A3] . | | | |
| Alternate Flows:  [A1] Add hotel`s rooms :   * The system will show the interface of Add hotel`s room. * The admin will add hotel`s rooms.   [A2] Add hotel`s services :   * The system will show the interface of Add hotel`s services. * The admin will choose the kind of services. * Clean services. * The admin will add clean services. * Maintenance services. * The admin will add Maintenance services. * Other services. * The admin will add the Other services.   [A3] Add restaurant`s food :   * The system will show the interface of Add restaurant`s food. * The admin will Add restaurant`s food. | | | |

### Purchases management.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 4 | Use Case Name: purchases management. | |
| Use Case Type: Essential | | | **Primary Actor: System admin** |
| Stakeholders:  System admin | | | |
| Brief Description: This use case describes how the admin manage the purchases. | | | |
| Pre-condition: admin login to the system | | | |
| Post-condition : ----- | | | |
| Trigger: admin open the interface of 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 purchases. 3. The admin will choose the kind of purchases. 4. If he choose furniture [A1] , material [A2] , others [A3]. | | | |
| Alternate Flows:  [A1] furniture :   * The admin will add the purchase bonds of furniture.   [A2] material :   * The admin will add the purchase bonds of material.   [A3] others:   * The admin will add the purchase bonds of things that they buy. | | | |

### Accounts management.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 5 | Use Case Name: Accounts management. | |
| Use Case Type: Essential | | | **Primary Actor: System admin** |
| Stakeholders:  System admin | | | |
| Brief Description: This use case describes how the admin manage the accounts. | | | |
| Pre-condition: admin login to the system. | | | |
| Post-condition : ----- | | | |
| Trigger: admin open the interface of accounts.  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 accounts. 3. The admin will choose the customer to add bonds. 4. The admin will add bonds to the customer. | | | |

### Rooms reservation.

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 6 | 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.  Extend: | | | |
| Normal Flow of Events:   1. The admin will login to the system 2. The admin will open the interface of show rooms cases. 3. The system will show the rooms cases to know the empty rooms. 4. The admin will search about the specific room. 5. The admin will add customers data. 6. The admin will add reservation to the room. | | | |

### Generate reports

|  |  |  |  |
| --- | --- | --- | --- |
| Importance Level: High | ID: 7 | 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 add and send reports to hotel manager. | | | |
| Pre-condition: hotel manager request reports. | | | |
| Trigger: admin open the interface of add and send reports.  Type: internal. | | | |
| Relationships:  Include: Login. | | | |
| Normal Flow of Events:   1. The admin will login to the system. 2. The admin will open the interface of add reports. 3. The system will show different types of reports. 4. The admin will choose the kind of report. 5. if he choose Reservations report [A1] , Services report [A2] , Purchases report [A3] , Accounts report [A4] , Users report [A5] , Restaurants report [A6] , Customers report [A7] . | | | |
| Alternate Flows:  [A1] Reservations report:   * The admin will add the Reservations report and send it to the hotel manager.   [A2] Services report:   * The admin will add the Services report and send it to the hotel manager.   [A3] Purchases report:   * The admin will add the Purchases report and send it to the hotel manager.   [A4] Accounts report:   * The admin will add the Accounts report and send it to the hotel manager.   [A5] Users report:   * The admin will add the Users report and send it to the hotel manager.   [A6] Restaurants report:   * The admin will add the Restaurants report and send it to the hotel manager.   [A7] Customers report:   * The admin will add the Customers report and send it to the hotel manager. | | | |

## 3.4 Class Diagram

