High Level Design

for

Milton Hotel Website

Version 1.0

Prepared by

Waleed Moursy

Parijat Bandyopadhyay

Vinay Govindan

Sreeraj Punnoli

Rishab Gupta

Simon Achkar

November 14, 2018

Table of Contents

[1. INTRODUCTION 3](#_Toc530321519)

[1.1 Purpose 3](#_Toc530321520)

[1.2 Scope 3](#_Toc530321521)

[1.3 Overview 3](#_Toc530321522)

[1.4 Reference Material 3](#_Toc530321523)

[1.5 Definitions and Acronyms 4](#_Toc530321524)

[2. SOLUTION ARCHITECTURE 4](#_Toc530321525)

[2.1 Architectural Goals and Constraints 4](#_Toc530321526)

[2.2 Customers Dashboard 4](#_Toc530321527)

[2.3 Employees Dashboard 5](#_Toc530321528)

[3. DATA DESIGN 6](#_Toc530321529)

[3.1 Data Description 6](#_Toc530321530)

[3.2 Data Dictionary 6](#_Toc530321531)

[3.1 Entity Relationship Diagram 7](#_Toc530321532)

[4. Component Design 8](#_Toc530321533)

[4.1 Overview of Component Design 8](#_Toc530321534)

[4.2 Use Case 8](#_Toc530321535)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Refer to cover page | 11/14/2018 | Initial version | 1.0 |
|  |  |  |  |

# INTRODUCTION

## 1.1 Purpose

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

## 1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

## 1.3 Overview

The project is a hotel website for the customers to view and book the hotel services online. A login is required to make the bookings. If the customer is not an existing user, he will have to sign up using email or one of his social networks. After successful login/signup, the customer will be redirected to his dashboard and can make a search to see the list of available rooms and halls and other events happening in Milton Hotel. The customer can pay online to avail the services via a secure payment gateway. An email will be sent on customer’s email address sending a booking and payment confirmation. An airport pickup can also be scheduled. The hotel also has a restaurant and the menu can be found on the hotel website. The website provides the flexibility to modify/cancel the booking. The customer can view his previous booking and based on those, a series of deals will be shown to the customer. Incorporated in the website is a 24/7 customer support service to assist the customers for queries and inquiries.

The hotel employees have been given different privileges to access the hotel website. From the employee dashboard, they can make entries to the database to update the room availability or change some event details. The employees can additionally access customer’s previous bookings and view their current status to assist them better.

## 1.4 Reference Material

People.ok.ubc.ca. (2018). [online] Available at: <https://people.ok.ubc.ca/rlawrenc/research/Students/CJ_05_Design.pdf>

|  |  |  |
| --- | --- | --- |
| S. No. | Document Name | Version |
| 1 | SRS – Milton Hotel Website | 1.0 |

## 1.5 Definitions and Acronyms

* MHW: Milton Hotel Website
* Employees Dashboard: The part of the system that is used by the employees to manage customers and room bookings.
* Customers Dashboard: The part of the system that is used by the hotel customers to book rooms, view bookings and pay online.

# SOLUTION ARCHITECTURE

## Architectural Goals and Constraints

The MHW architecture has been designed with the following objectives in mind:

1. To allow customers to book rooms, and to edit and cancel their bookings.
2. To enable customers to pay online for hotel services and room booking.
3. To facilitate user assistance for the hotel's employees by allowing the employees to manage customers' room booking.

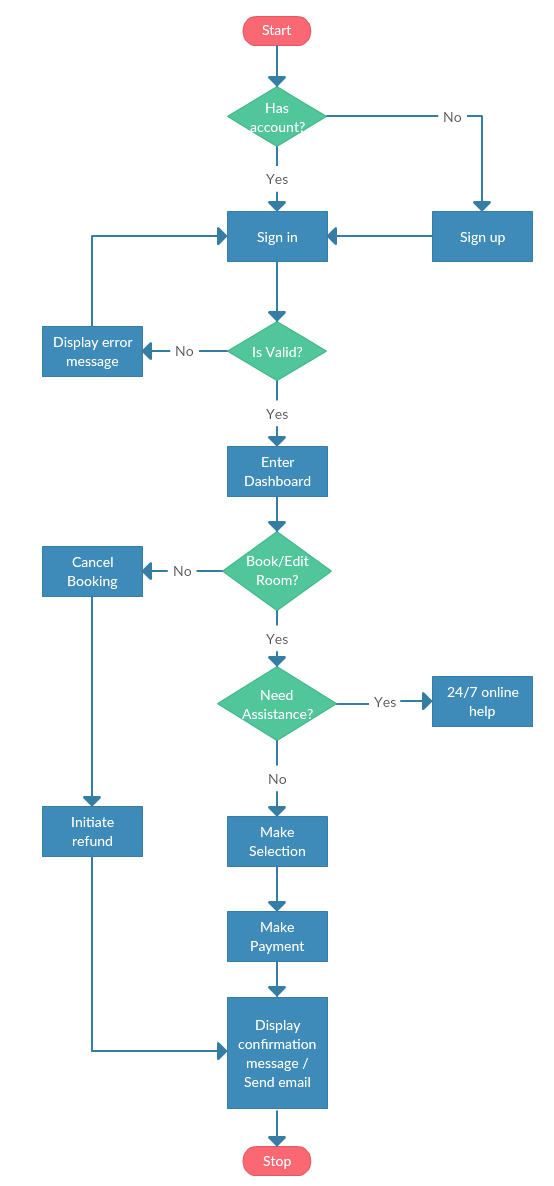
The major design and implementation constraints for the system are:

1. Simplicity and ease of use
2. Flexibility and Scalability

The following sections will explore more deeply how the system works from both the employees and the customers' point of view.

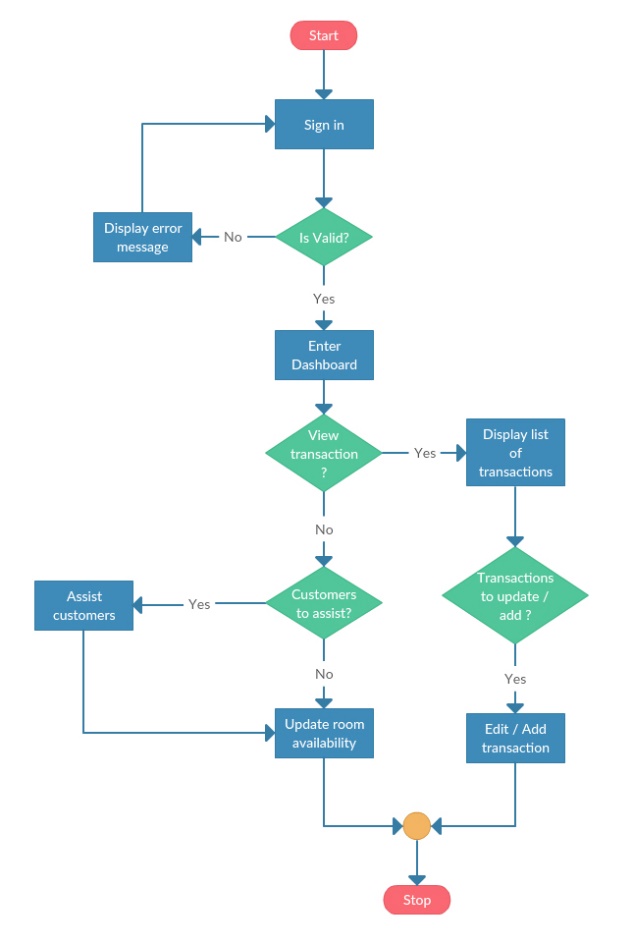
## 2.2 Customers Dashboard

This is where the customers book rooms, edit or cancel their booking and pay online for the hotel’s services. The customers must have an account that is connected to all their bookings and payment history. A button should be present allowing the customer to require assistance. The following is the flowchart that describe this section:



## 2.3 Employees Dashboard

This is where the employees manage and assist customers. This give them access to the customers’ booking information in order to help them update or delete a booking. The following is the flowchart that describe this section:



# DATA DESIGN

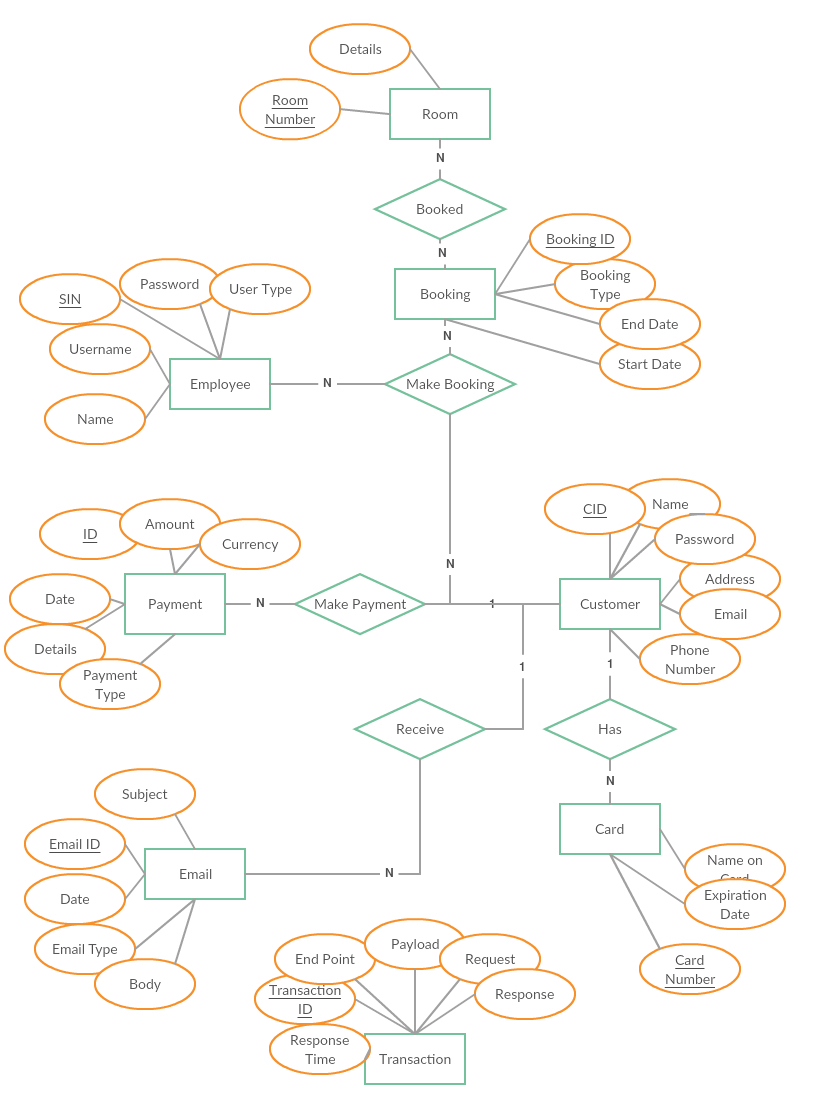
## 3.1 Data Description

MHW is trying to accomplish many tasks, that requires a large and solid database. Security and sustainability are a major concern relating to the database thus a relational model is used.

## 3.2 Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Column Type | Description | Value Restriction | Is Required |
| Table Name: **ROOM** | | | | |
| Room Number | int | The number assigned to a room within the hotel. | Should be unique | Yes |
| Details | varchar | Any additional detail concerning the room. E.g. Room Type (Single, Double, Suite) – View (Ocean view – No view) … | None | No |
| Table Name: **BOOKING** | | | | |
| Booking ID | varchar | A primary key for each column in table booking. | Should be unique | Yes |
| Booking Type | varchar | E.g. Online Booking, Employee Booking, on site Booking … | None | Yes |
| Start Date | date | Room check-in date. | None | Yes |
| End Date | date | Room check-out date. | Should be later than Start Date | Yes |
| Table Name: **CUSTOMER** | | | | |
| CID | varchar | Customer Login ID for each Customer | Should be unique | yes |
| Name | varchar | Name of the Customer | None | Yes |
| Password | varchar | Customer Password | unique |  |
| Address | varchar | Customer Address | None | No |
| Email | varchar | Customer email address | Unique | Yes |
| Phone Number | int | Phone Number of the Customer | Unique | Yes |
| Table Name: **CARD** | | | | |
| Name on Card | varchar | Customer Debit/Credit Card | Unique | Yes |
| Expiration Date | date | Expiration date in customer card | Unique | Yes |
| Card Number | int | Customer card no | Unique | yes |
| Table Name: **Payment** | | | | |
| Currency | varchar | Currency type used by customer for payment | None | Yes |
| Amount | float | Amount paid by customer for booking | None | yes |
| ID | varchar | Payment ID generated after payment | Unique | Yes |
| Date | date | Date of payment | Unique | Yes |
| Details | varchar | Payment mode | Unique | Yes |
| Payment Type | varchar | Credit card/Debit card | Unique | Yes |
| Table Name: **Email** | | | | |
| Subject | varchar | Payment/booking details to customer | Unique | Yes |
| Email ID | varchar | Email ID of the Hotel | Unique | Yes |
| Date | date | Date of email | Unique | Yes |
| Email Type | varchar | Booking/Cancellation/Invoice |  |  |
| Body | varchar | Details of customer booking and payment | Unique | Yes |
| Table Name: **Transaction** | | | | |
| Transaction ID | varchar | Transaction ID generated after payment | Unique | Yes |
| Response Time | time | Time taken for transaction | None | Yes |
| End Point | varchar | API End Point | Unique | Yes |
| Payload | varchar | API Payload | Unique | Yes |
| Request | varchar | API request | Unique | Yes |
| response | varchar | API response | Unique | Yes |

## Entity Relationship Diagram



# Component Design

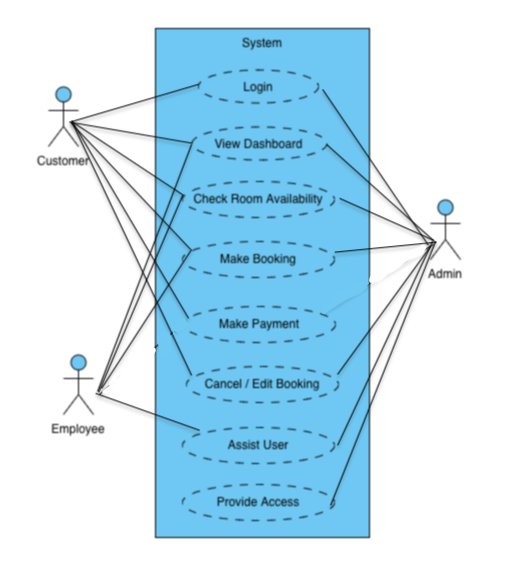
## 4.1 Overview of Component Design

Describe the functionality of the system from the user’s perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

## 4.2 Use Case

The following is the System’s use case. The three main players are the customers, employees and admin of the system.

The following is the System’s use case. The three main players are the customers, employees and admin of the system.



**4.3 Overview**

The following is a detailed explanation of each functionality of the system:

* + **Login** –
    - Once signed up they have to login in order to access their Dashboard and thereby can view their bookings and payment history.
    - Employees’ need an account to access their own Dashboard that allows them to assist the customers.
    - An admin login is also required for authorization.
  + **Signup –**
    - A customer must sign up if they do not have an account.
    - Signup can be done by filling up a signup form or through preferred social media account.
  + **View Dashboard** –
    - For the customers a Dashboard is an interface where all the booking information will be shown so that the customers could make changes to their bookings or start making new bookings. The customer is also capable of seeing all the payments made.
    - For the employees, the dashboard allows them to assist their customers by managing their bookings.
    - The admin’s dashboard is an interface where the admin (super user) can view and monitor everything that is happening in the system. That is why the admin has all the privileges of the other users besides making an online payment which is only available for the customers.
  + **Check Room Availability** –
    - A customer needs to see if a room is available for booking or not.
    - An employee needs to access this information in case they need to assist a customer. They are also responsible to maintain the room availability.
    - Admin provides room availability editing privileges to the employees according to their employment type.
  + **Make Payment** –
    - Enter payment information and make payment using the payment API.
    - Receive an acknowledgement on the payment on the screen and as an email
  + **Cancel/Edit Booking** –
    - Every future dated booking should be editable and cancelable for the customers.
    - Admin has the major control so that the employees can approach them if there is anything goes wrong.
  + **Assist Customer** –
    - This will open a chat box with the customers.
    - This feature allows the employees to assist the customer via a live chat.
  + **Provide Access** –
    - An admin can create a user of type employee and can give them specific access and privileges.