|  |
| --- |
| N/A |
| Hotel Management System Application |
| For Peach |

|  |
| --- |
| Phuc Le Dinh  7-30-2017 |

Revision history

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Note** |
| Draft 1 | 9th July 2017 | Creation |
| Draft 2 | 20th July 2017 | Add a requirement for automated confirmation email. |
|  |  |  |

# Introduction

This document covers the system level and high-level requirements for the Hotel Management System software. This software will be referred as ‘Peach’ elsewhere in this document.

# Requirements

## System requirements

S1 Peach shall use input information (e.g. check in/out date, room type, etc.) to search for vacant rooms.

S2 Peach shall identify each booking with a unique number.

S3 Peach shall store booking information for later retrieval.

S4 Peach shall allow the update of room status (i.e. vacant, booked, upcoming check-in/check-out, cleaning).

S5 Peach shall notify upcoming check-in/check-out events for a room.

S6 Peach shall be a Java application.

S7 Peach shall initiate an automated booking confirmation email.

## High Level Requirements

S1H1 The following information shall be obtained before every booking:

* Customer information: first and last name, nationality, DOB, gender, ID type and ID number.
* Contact details: Mobile phone, address, email address.
* Booking information: Check-in/out date, room type, number of customers.

S1H2 The room information shall be available from a database.

S1H3 The room information shall include the following:

* Room type.
* Telephone extension.
* Room status: vacant, booked, upcoming check-in/check-out, cleaning.
* Check-in/out date.
* Relevant booking reference.

S1H4 The vacant rooms shall be listed in the form of a table.

S2H1 Peach shall provide a mean to generate a unique booking number for each booking.

S3H1 Each booking shall have the information mentioned in S1H1 and S1H3.

S3H2 Each confirmed booking shall be saved into a database.

S3H3 The database shall have a table for booking confirmations.

S3H4 The database shall have a table for room status.

S3H5 The database shall have a table for pricing.

S4H1 Peach shall allow changing room status to ‘booked’ upon agreeing the booking with the customer.

S4H2 Peach shall allow changing the room status to ‘vacant’ in the event of:

* Cancelation.
* Check-out complete.

S4H3 Peach shall allow changing room status to ‘checked-in’ upon the customer’s arrival.

S5H1 Peach shall generate a list of check-ins on a particular date on request.

S5H2 Peach shall generate a list of check-outs on a particular date on request.

S6H1 Peach shall be an offline Java application.

S6H2 Peach shall have a MySQL database.

S7H1 Peach shall use an automated email service to send a booking confirmation to the customer.

S7H2 Peach shall include a tourist information pack with the booking confirmation email.

# Future Work

## Web application

Although this proof of concept application provides basic functionality for the process of hotel management, it is confined in an offline environment. Hence functions such as booking or updating room status can only be on a PC running the application.

In order to provide a more flexible way of operating the same functionalities on other platform, a web interface would be considered in the next iteration of the project. It would eliminate the OS dependent nature of a normal Java application. It can also be operated on mobile platform.

## Database encryption

Customer information and privacy is of the utmost important for a business. A security management scheme of the database will need to be implemented in order to satisfied this need.

## Authentication

The proof of concept Java application does not have any requirement for user authentication. Hence malicious activity cannot be monitored. A multi-privilege scheme to limit the ability of a user to the database has to be implemented.

## Expansion to other services

Other services such as coffee house and restaurant can also use the same database to get a more personalised experienced for the customer.

# Use Cases

## Booking

Alice calls the hotel and provides her check-in/out dates. She wants a double room for the holiday. The receptionist uses the information to find out if there is any double room for the mentioned dates. The receptionist finds an available room and reports the total price. Alice provides her contact details. The receptionist then confirms the booking by clicking a button to reserve the room for Alice. A unique booking confirmation number is also generated and sent to Alice’s email address.

## Check-in

When Alice arrives at the hotel, she provides her unique booking number to the receptionist. The receptionist enters the booking number into the Peach application to find the booking information. Alice booking also appears in the current check-in list for the current date. The receptionist then asks Alice for her ID and more information about her stay and enter them into the Peach application. All of this are saved into the booking information under Alice’s unique booking number. The receptionist then marks the room as ‘checked-in’.

## Check-out

When Alice wants to check out, she hands back the key. The receptionist finds Alice’s booking has ended in the current check-out list for the current date. After everything in the room has been check, Alice’s booking now becomes finished and the room’s status become checked out.

After the room has been cleaned, its status is changed to vacant.

## Cancelation

Bob has booked a room and give a unique booking number. He calls the hotel the next day and wants to cancel the room booking. The receptionist asks for his unique booking number and search for his booking. The receptionist then marks his booking as cancelled. The room that has been assigned to Bob now becomes vacant.