HO CHI MINH NATIONAL UNIVERSITY

University of Information Technology, Ho Chi Minh

Building a hotel service managements

Logo

Description automatically generated

Class:

SE104.L23.PMCL

Instructor:

Nguyen Thi Thanh Truc

UNIVERSITY OF INFORMATION TECHNOLOGY

Team members:

Ha Thuc Dang Khoa

Huynh Lu Anh Khoa

Le Van Tri

# Table of content

[**Table of content**](#_8prikj7mh0fs)1

[**Table of figure**](#_bop274n63vi0)2

[**Introduction**](#_sk5svb1fl648)2

[Purpose](#_aqw7tcjvt2pl) 2

[Scope](#_wewar3bamsk3) 2

[Definitions, Acronyms, Abbreviations](#_dz6z9nhhtwgj) 2

[References](#_jxwaqr42toym) 3

[Overview](#_297h2rcvbtd7) 3

[**Overall description**](#_6xdnq0qbtl16)3

[Product perspective](#_ikphclgjdmqs) 3

[Product feature](#_rbiqtld3e1ez) 4

[User characteristics](#_thzlhrntldxa) 5

[Design constraints](#_svwpkebtqgmr) 5

[Assumption and dependencies](#_m538wreeap13) 5

[**Functional requirements**](#_11f1e39g6wk)5

[**Non-functional requirements**](#_k24v3h6yhmpp)11

[Performance](#_ntcfajows6t7) 11

[Interfaces](#_a72h7gl16bjo) 11

[User interfaces](#_s4jwcpbjgici) 11

[Hardware interfaces](#_4s9qe6aagorh) 11

[Software interfaces](#_yfxymp3vuyu) 11

# Table of figure

[1. Service's information form. 3](#_Toc76403638)

[2. Deposit information 4](#_Toc76403639)

[3. Invoices' detail 4](#_Toc76403640)

[4. Use case diagram 6](#_Toc76403641)

# Introduction

## Purpose

This documentation is the software requirements specification for building a hotel service management system for the final project of the Introduction to Software engineering course for Ho Chi Minh University of Information Technology. Course code: SE104.L23.PMCL.

## Scope

The purpose of the hotel service management system is to provide a way to administer hotel’s service usage and automatically invoicing. The software provides an easy way for hotel receptionists to monitor and add a new record whenever a service is used. When the customers check out, the software will be invoicing and determine the required fee.

## Definitions, Acronyms, Abbreviations

|  |  |  |
| --- | --- | --- |
| Abbreviation | DDescription | Comment |
| GPLv2+CE | GNU General Public License, version 2 with Classpath Exception | License for producing software. |

|  |  |  |
| --- | --- | --- |
| Definition | Description | Comment |
| Hotel service | Services provided by the hotel, excluding booking service. |  |

## Overview

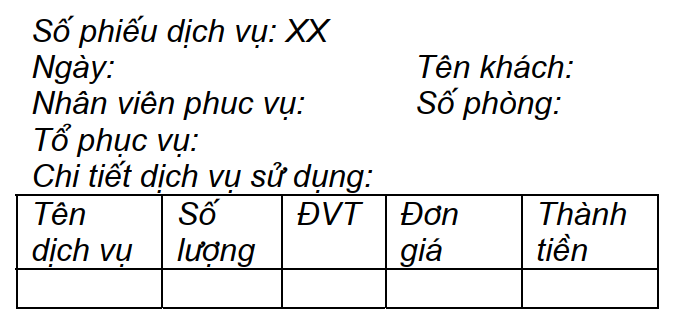
This document from this point onward will provide the following information with the according order:

* Overall description: This section will describe general factors that affect the product and requirements’ overview.
* Functional requirements: As the title said, we will be giving you information on the functions required by the stakeholders through actors and use cases.
* Non-functional requirements: Non-functional requirements will be listed and given a detailed description.

# Overall description

## Product perspective

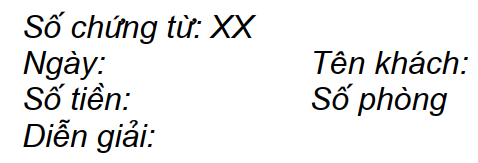
The product is used for managing service usage of Nhật Tân hotel. The hotel provides service for guests. The services include booking, food ordering, phone service,... etc. Aside from the booking service, we will call the rest “hotel service”. The receptionist will be responsible for receiving service orders. Orders’ information will be recorded using the following form:



Form 1: Services’ information

1. Service's information form.

The hotel provides two types of booking: hostel booking and full booking, in which, full booking means booking the whole room while hostel booking means booking a slot in the room. For long-staying customers, deposit is required. Deposit information is recorded using the following form:



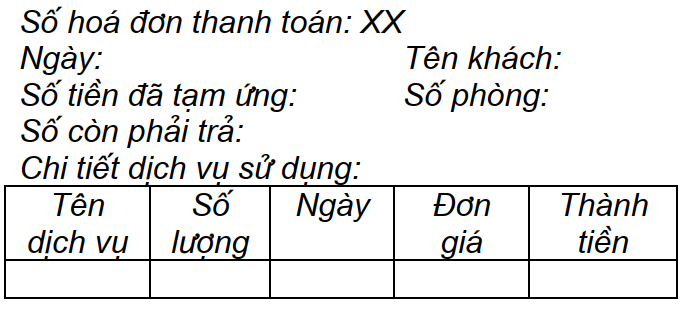
Form 2: Deposit information

2. Deposit information

The hotel provides multiple different payment methods, including:

* Paid by themselves.
* Paid by the company as business expenses.
  + Pay everything.
  + Pay for certain types of service.
* Paid as a tourist group.
* Paid as debt. (only for regular customers).

When the customer(s) checks out, the expense is calculated based on used service(s), minus the deposit, and an invoice is created using the following form:



Form 3: Invoices’ detail

3. Invoices' detail

## Product feature

The major features of this products are:

* Create a record of service usage.
* Booking an available room.
* Record deposit information under met constraints.
* Provides payment method when checking out
* Calculate the required expense and invoicing.

## User characteristics

The main users of the system are hotel’s receptionists, whose technical level may vary. Of whose required management functionalities are:

* User:
  + Add/Modify a service usage record
  + Add/modify a new booking record
  + Get all vacant rooms
  + Get all available services

Each room will have a different number of slots and the price may vary according to room quality.

## Design constraints

1. SQL commands to register data in the database.
2. Invoicing process must be automated.

## Assumption and dependencies

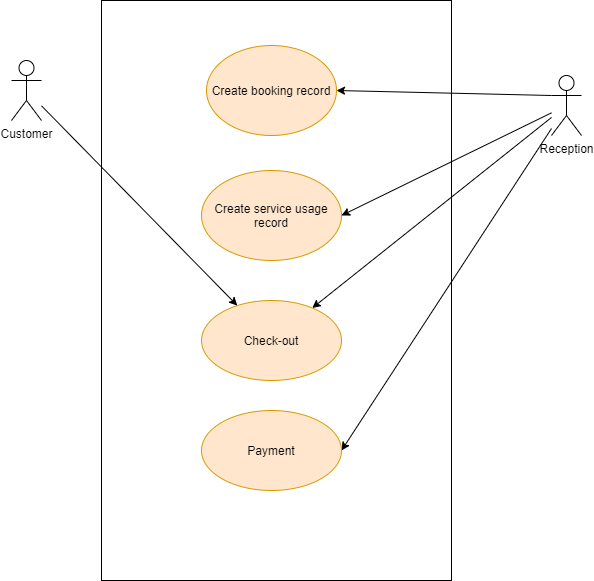
Let us assume that this is a hotel service management system and will be used for the following application:

* Creating record of hotel’ service usage and booking information.
* Creating record of deposit information.
* Calculating service fee and automatic invoicing when the hotel’s customers check out.
* Room is picked automatically from the top of the query result.

Assuming the use is locally within the hotel only, the system will be using a local database that runs in the same computer as the system.

# Functional requirements

1. Summary Use Case Diagram



4. Use case diagram

1. Use Case specification
   1. Booking

|  |  |
| --- | --- |
| Use case name | Booking |
| Actor(s): | Customer(Primary), Reception(Secondary) |
| Summary Description: | Allow customer to book a room if available |
| Priority: | Must have |
| Status: |  |
| Pre-condition: | Booking service is available |
| Post-condition(s): | - Customer have a room  - Booking information is recorded in the database |
| Basic path: | 1. The customer chooses the type of accommodation.  2. The reception check for room availability.  3. The customer chooses the date of departure.  4. The reception assigns them a room and puts it on the record. |
| Alternative path: | 2b. Room unavailable  3b. The reception asks for a deposit and records it. |
| Business rules: | There are two types of accommodation: full booking and hostel. Full booking mean an entire room is booked, while hostel means the a slot in the room is booked |
| Non-functional requirements: | NF1: Time for adding new customers.  NF2: Time for checking room availability |

* 1. Checking out

|  |  |
| --- | --- |
| Use case name | Checking out |
| Actor(s): | Customer(Primary), Reception(Secondary) |
| Summary Description: | Customer check-out |
| Priority: | Must have |
| Status: |  |
| Pre-condition: | Customer must have a room |
| Post-condition(s): | Reception marks the customer’s room available. |
| Basic path: | 1. Customer confirms checking out. 2. Reception awaits room checking status. 3. Reception finishes the check out procedure and asks for payment. |
| Alternative path: |  |
| Business rules: | An invoice will be created based on form 3. One will be printed for the customer to use. |
| Non-functional requirements: |  |

* 1. Payment

|  |  |
| --- | --- |
| Use case name | Payment |
| Actor(s): | Customer(Primary), Reception(Secondary) |
| Summary Description: | Customers pay for the services they use through various payment methods. |
| Priority: | Must have |
| Status: |  |
| Pre-condition: | Checking out procedures is done. |
| Post-condition(s): | The customer pays the required fee and it’s recorded in the system. |
| Basic path: | 1. Customer finishes checking out. 2. The room is in acceptable condition 3. Customer chooses between payment methods. 4. Calculate required fee. 5. Payment confirmation. 6. Record the payment into the system. 7. Create an invoice for the customer. |
| Alternative path: | 2b. The room is in unacceptable condition and for which an extra fee is added.  4b. If deposit is present, minus the amount of deposit from the fee. |
| Business rules: | Various payment methods are provided. |
| Non-functional requirements: | NF1: Time required for calculation is acceptable.  NF2: Fee calculation must be correct. |

* 1. Order hotel service

|  |  |
| --- | --- |
| Use case name | Order a hotel service |
| Actor(s): | Customer(Primary), Reception(Secondary) |
| Summary Description: | Customer orders room service from the reception. |
| Priority: | Must have |
| Status: |  |
| Pre-condition: | Customer currently having a room. |
| Post-condition(s): | The service is served and recorded into the system. |
| Basic path: | 1. The customer calls the reception to order a service. 2. The reception checks if the service is available for use. 3. Reception noted down the required information into the system. |
| Alternative path: | 2b. The service is unavailable and cannot be provided to the customer. |
| Business rules: | Each service is handled by different departments. Later on |
| Non-functional requirements: | NF1: Time required for checking service availability.  NF2: Time required for recording the service information into the system. |

# Non-functional requirements

## Performance

The performance of the software heavily relies on data accessing speed. As a result, we design the database using the Entity-Relationship model and normalize it.

## Interfaces

### User interfaces

* Front-end software: Java Swing
* Back-end software: MySQL

### Hardware interfaces

* OS: Windows 7, 8, 10
* At least 2GB RAM
* Intel i3 processor or equivalence

### Software interfaces

|  |  |
| --- | --- |
| Software used | Description |
| Operating system | The system will be built to run on Windows operating system for its user-friendliness. |
| Database | For recording service usage as well as booking record, we have chosen MySQL databases |
| Java | To implement the project, we have chosen Java because it’s our strongest suit. |