

**Product Requirements Document**

**Submission 1 Specification**

**Alphecca**

SWEN90007 SM2 2021 Project

Haimo Lu, 1053789

Xiaotian Li 1141181

Yifei Wang 1025048



**Revision History**

*<****students*** *must use this table to track individuals’ contributions to this document.*

*Every time you change this document, add the date you changed it, a description of your performed task and your name. For the version, please adopt the following format:*

***01.00-D<number>*** *for draft versions related to Part 1 (any version before the final submission is considered draft). When your document is reviewed and finally ready to be submitted, change it to 01.00. For Part2, start with 02.00-D<number> and so on. This document should always be kept on GitHub>*

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 08-08-2022 | 01.00-D01 | Add the domain model and its descriptions | Haimo Lu |
| 09/08/2022 | 01.00-D02 | Update domain | Xiaotian Li |
| 10/08/2022 | 01.00-D02 | Add use case diagram v1.0 | Yifei Wang |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

[**1.**](#_heading=h.35nkun2) **Introduction 6**

[*1.1*](#_heading=h.1ksv4uv) *Proposal 6*

[*1.2*](#_heading=h.44sinio) *Target Users 6*

[*1.3*](#_heading=h.2jxsxqh) *Conventions, terms and abbreviations 6*

[**2.**](#_heading=h.z337ya) **Actors 6**

[**3.**](#_heading=h.3j2qqm3) **Use Cases 7**

[*3.1*](#_heading=h.4d34og8) *<students to add info here> Use Case 1: Student Undertakes Examination 7*

[**4.**](#_heading=h.2s8eyo1) **Domain Model 7**

[*4.1*](#_heading=h.17dp8vu) *Domain Model Diagram 7*

[*4.2*](#_heading=h.3rdcrjn) *Domain Model Description 8*

[4.2.1](#_heading=h.26in1rg) Object 8

[4.2.2](#_heading=h.lnxbz9) Inheritance 9

[*4.3*](#_heading=h.1y810tw) *Domain Model Diagram 11*

1. **Introduction**
   1. **Proposal**

This document specifies the SWEN90007 project use cases, actors to be implemented, and the system’s domain model.

* 1. **Target Users**

This document is mainly intended for SWEN90007 students and teaching team.

* 1. **Conventions, terms and abbreviations**

This section explains the concept of some important terms that will be used throughout this document. These terms are detailed alphabetically in the following table.

| **Term** | **Description** |
| --- | --- |
| HBS | Hotel Booking System |

1. **Actors**

| **Actor** | **Description** |
| --- | --- |
| Admin | Admin is a user role of the system, they are the HBS provider that aggregate all hotels created in the system, they can also view all users and stays users have booked. |
| Customer | Customer is a user role of the system, they can book hotels, view, and manage their bookings. |
| Hotelier | Hotelier is a user role of the system, they can register and manage their hotels with different configurations in the system and can manage bookings they received from customers. |

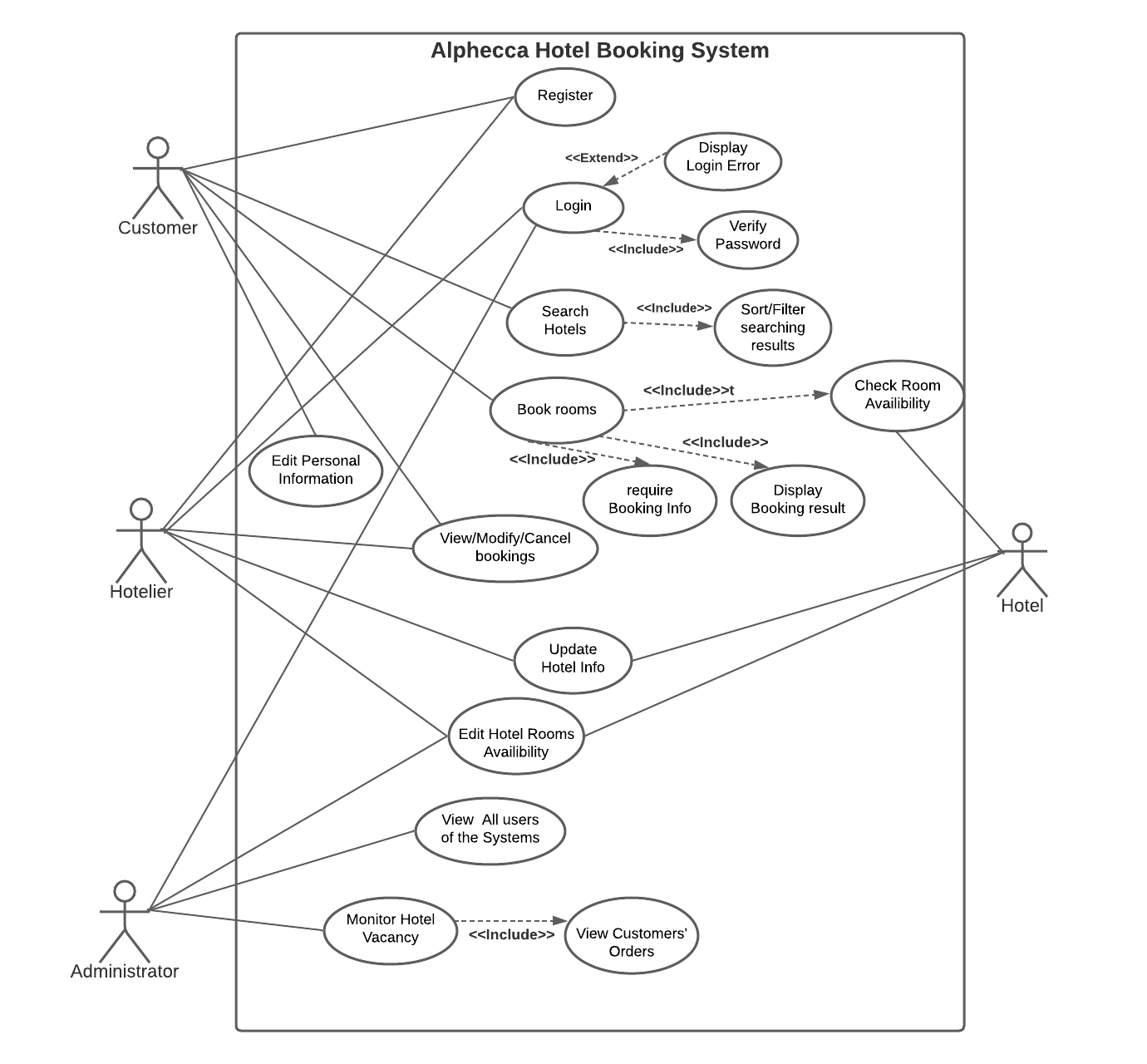
1. **Use Cases**
   1. ***<students to add info here>* Use Case 1: Student Undertakes Examination**

**Actors**

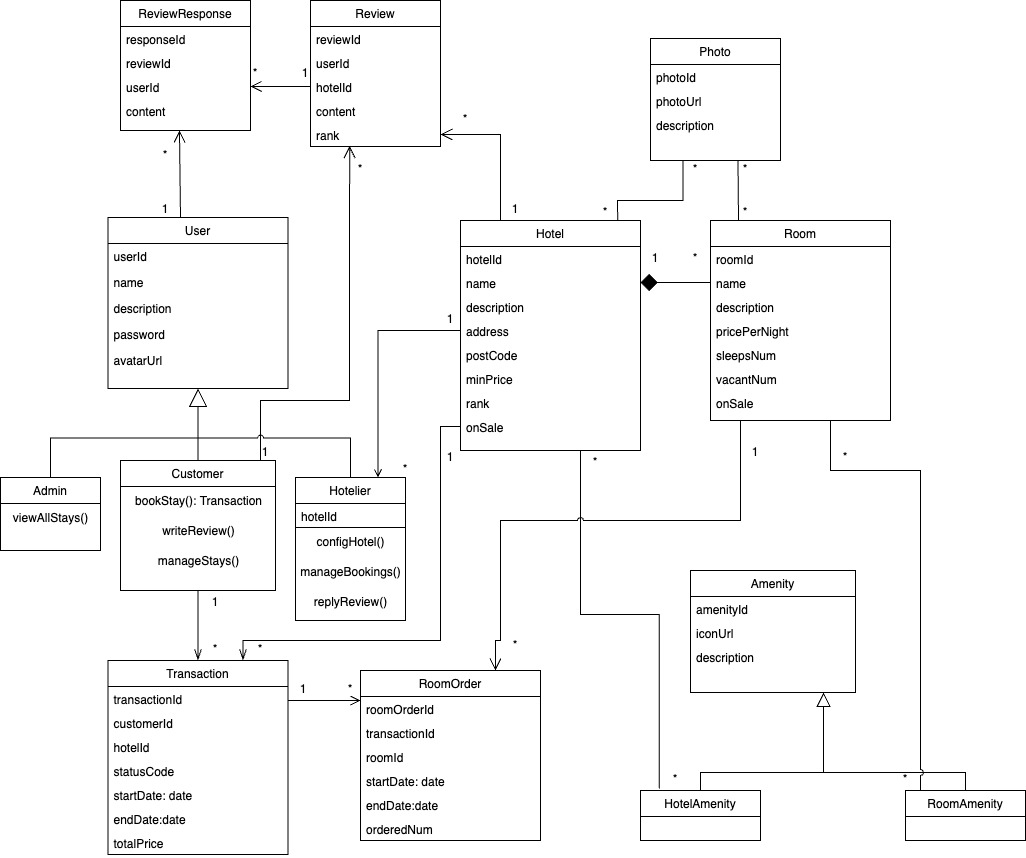
1. Student

**Basic Flow**

On Wednesdays, students of SWEN900XX log into the Online Examination Application to undertake a written examination. After logging in, they select the subject from their list of enrolled subjects. Then they select the examination corresponding to that Wednesday. They commence the exam, entering an answer or skipping each question in the examination. Once they have provided a response to each question, they submit the examination for marking.



1. **Domain Model**
   1. **Domain Model Diagram**



* 1. **Domain Model Description**
     1. **Domain**

| **Object** | **Description** |
| --- | --- |
| Review | Review is the feedback for the hotel from Customer. |
| ReviewResponse | It is the response for one review, it can be created by all users. |
| User | This represents the general user. Users then can be further categorized as three types. |
| Admin | Admin is one role of the system, they are the HBS provider that aggregate all hotels created in the system, they can also view all users and stays users have booked. |
| Customer | Customer is a user role of the system, they can book hotels, view, and manage their bookings. |
| Hotelier | Hotelier is a user role of the system; they can register and manage their hotels with different configurations in the system and can manage bookings they received from customers. |
| Transaction | Transactions are created when customers book stays, transactions are owned by the customers and also the respective hotel. One Transaction can hold multiple room orders. |
| RoomOrder | It is the domain for room orders in one transaction. |
| Photo | It is the domain for photo resources, it records photo url and the entity it is attached to. The same photo entity can also be reused by different hotels and rooms. |
| Hotel | It is the domain for hotel item. |
| Room | It is the domain for room item. It is the necessary component of the hotel domin. |
| Amenity | This represents the general amenity, amenities can be further categorized as two types. |
| HotelAmenity | HotelAmenity describes facilities or amenities a hotel has in public area. It may contain customized fields in the future extension. |
| RoomAmenity | RoomAmenity describes facilities or amenities a room has. It may contain customized fields in the future extension. |

* + 1. **From General type to Sub types**

| **General type** | **Sub type** | **Description** |
| --- | --- | --- |
| User | **Admin** | User is an abstraction of all roles in the HBS, users share fields among all roles in the system, while each sub type may have their own attributes or behivours for future extension. |
| **Customer** |
| **Hotelier** |
| Amenity | **HotelAmenity** | Amenity is an abstraction of all amenities in the HBS, of which they share attributes among all amenities. ub type may have their own attributes for future extension. |
| **RoomAmenity** |