EECE 419 - Pod 1

Hotel Bookings Requirements Specification

November 6, 2009

Sean Clark	36538056
Yang Gao	52588050
Shen Li	65962060
Neil Gentleman	62973029
Arash Malekzadeh	33685058
Michael Tando	79529061
Wei-Chen Wang	64341043

Table of Contents

1.0 Introduction 1
- 1.1 Purpose 1
- 1.2 System Overview 1
- 1.3 Project Scope 1

2.0 Assumptions 2

3.0 Non-Functional Requirements 3
- 3.1 Usability 3
- 3.2 Scalability 3
- 3.3 Security 3
- 3.4 Portability 4
- 3.5 Performance 4

4.0 Functional Requirements 5
- 4.1 Use Cases 6
  - Create an account 8
  - Make a reservation 9
  - Cancel a reservation 10
  - Edit a reservation 11
  - Create rooms 12
  - Edit room 13
  - Delete rooms 14
  - Create room type 15
  - Edit room type 16
  - Delete room type 17
  - Add chargeable item(s) [elided: edit chargeable item, delete chargeable item] 18
  - Charge chargeable item(s) 19
  - Check-in 20
  - Check-out 21
  - Check Dirty Room 22
  - Mark Clean Room 23
  - View Statistic Report 24

Appendix A: User Interface Mockups 25

Table of Figures

Figure 1: Class Diagram....4
Figure 2: Use Case Diagram....5
Figure 3: Create an Account (Sequence Diagram)....6
Figure 4: Make a Reservation (Sequence Diagram)....7
Figure 5: Cancel a Reservation (Sequence Diagram)....8
Figure 6: Edit a Reservation (Sequence Diagram)....9
Figure 7: Create Rooms (Sequence Diagram)....10
Figure 8: Edit Room (Sequence Diagram)....11
Figure 9: Delete Rooms (Sequence Diagram)....12
Figure 10: Create Room Type (Sequence Diagram)....13
Figure 11: Edit Room Type (Sequence Diagram)....14
Figure 12: Delete Room Type (Sequence Diagram)....15
Figure 13: Add Chargeable Items (Sequence Diagram)....16
Figure 14: Charge Chargeable Item (Sequence Diagram)....17
Figure 15: Check-in (Sequence Diagram)....18
Figure 16: Check-out (Sequence Diagram)....19
Figure 17: Check Dirty Room (Sequence Diagram)....20
Figure 18: Mark Clean Room (Sequence Diagram)....21
Figure 19: View Statistic Report....22
Figure 20: UC1 Mockup - Create Account....23
Figure 21: UC2 Mockup - Make a Reservation (Step 1)....24
Figure 22: UC2 Mockup - Make a Reservation (Step 2)....25
Figure 23: UC2 Mockup - Make a Reservation (Step 3)....26
Figure 24: UC2 Mockup - Make a Reservation (Step 4)....27
Figure 25: UC2 Mockup - Make a Reservation (Step 5)....28

1.0 Introduction

1.1 Purpose

The purpose of this document is to explain different aspects of software requirements regarding a hotel reservation software. The scope of this document is limited to the main functional and non-functional requirements of the system. These functions are defined as those required to have a stable and usable hotel reservation system. Extra features designed for different hotels will not be mentioned in this document.

1.2 System Overview

X-Reserve is a hotel reservation system that allows hotels to provide guests with a web-based reservation interface. There are a number of features available to both clients and the hotel management or staff. A client can create an account and use it to view available rooms for different dates. The room search function provides users the ability to find a room based on the price, date, room layout, capacity and other options. As for the management side, there are features designed for managers, staff and maids. Hotel managers can view reports and graphs on reservation trends, and be able to maximize their profits. They can add customized room layouts to the database and have automatic seasonal price changes. The hotel staff can make reservations for guests, add chargeable items to rooms, set special prices rates and have notes associates with each guest's account. Rooms will automatically be marked so that maids can keep track of which room is to be occupied and needs cleaning.

1.3 Project Scope

X-Reserve is to be designed as a web-based application which will only operate with one specific database. This software can be deployed for different hotels, and It will allow users around the world to make reservations. The goal of this software is to provide a reliable and fully customizable reservation system for any type of hotel within Canada and United States.

2.0 Assumptions

In order to simplify development and testing, we have decided to only support Firefox version 3.5 and up. Users of our software should be familliar with the navigation of websites and simple web applications, such as Google Maps or Gmail. Our interface will be designed around the current UI norms and should be intuitive enough to use without any instruction. Installation of X-Reserve is too technical for normal users, but we will provide enough documentation to make the process easy for people familiar with installing server-side software. To ease installation, our software will useable without any external database or servlet container. It will also be able to take advantage of pre-installed servlet containers and database servers, such as Apache Tomcat and MySQL, to enhance scalability. We will do our best to minimize resource usage of X-Reserve, and it will be usable on limited hardware with a small number of users.

3.0 Non-Functional Requirements

The following section will describe our non-functional requirements in detail. They are broken down into five sections -- usabillity, scalability, security, portability and performance. Our key constraints are usability, security and performance.

3.1 Usability

The target clients of the software are hotel managers, receptionists, maids, and hotel customers. Most of the clients may know little about computers and software systems. As a result, the software interface takes a very important role. The software has to provide flexible and easy-to-use interfaces so that our clients can achieve reasonable success with little training. The software is a web-based application running on Mozilla Firefox 3.5, and the interface will be designed and constructed following the Nielsen's Ten Usability Heuristics. The software will expose different functionality depending on different roles of users. For example, if the login user is identified as manager, the manager will be seeing an interface with the viewing report option. The details of these interfaces will be discussed further in the functional requirements. Moreover, all these different interfaces will be consistent and will follow the same standards. The goal of these interfaces is to minimize user confusion and increase the usability of our software. Certain reports, like the maid's room occupancy view, are likely to be used in printed form, so all views will have a print-friendly mode.

3.2 Scalability

X-Reserve is only intended to manage one hotel, not a chain, so our ultimate scalability goals are limited. The system will, however scale to the largest of single hotels, with hundreds of of rooms. In this process, customer volumes are expected to scale to several thousands customers. The objective for this requirement is to cater for future growth in the business. The system also needs to provide access for a number of terminals that can be used by the hotel personnel. Eventually, there could be over twenty terminals in a hotel and thus the system should be able to handle over twenty simultaneous requests. This requirement is to allow the system to be used across the whole hotel by the hotel personnel at any time. The potential for database sharding or application server clustering will not be explored.

3.3 Security

Since our software will be storing sensitive personal information, like credit card information and addresses, security is essential. We are building on top of the Spring framework, which provides a secure base to work from. It provides authentication that will be used throughout the software to hide information from the wrong users. We will also be actively looking for bugs that could cause a security hole or information leak.

3.4 Portability

To ease installation and broaden our customer base, our software will be portable across a variety of operating systems, including Linux, OS X, and Windows. To simplify development, we will only be testing on Linux and Windows; however, our software should run without problems on any system with Java installed.

3.5 Performance

Performence is an important requirement for X-Reserve, especially due to its scalability. A system with our minimum requirements (1Ghz processor, 1GB of RAM) will be able to support up to 10 simultaneous users. The actual performance varies depending on a number of factors, including network latency and the capabilities of our user's computers. On an internal network, with relatively new computers (at least 600MHz with 256MB of RAM), X-Reserve will perform quite well. Normal operations and page refreshes will complete inside of 1 second, and more complicated operations (like statistical reports) will take at most 5 seconds.

4.0 Functional Requirements

This section contains descriptions and diagrams of our man functional requirements. The class diagram, shown in Figure 1, shows the relationship between all of the classes we will be using to implement our software. There are also detailed use cases and sequence diagrams, which describe our user interactions and the sequence of messages needed to complete them.

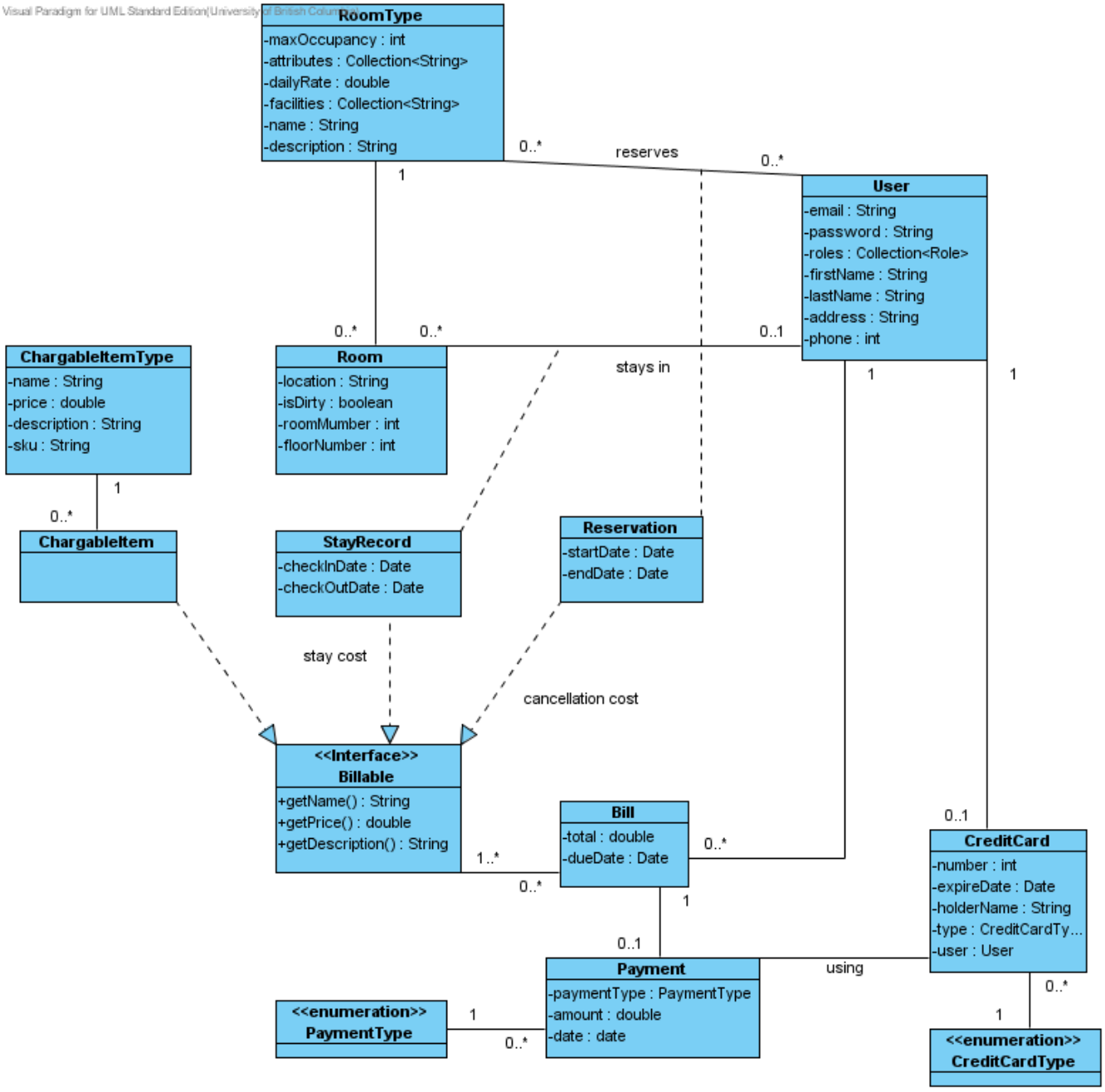

Figure 1: Class Diagram

4.1 Use Cases

This section describes the use cases for our project. The use case diagram in Figure 1 shows the use cases and their relation to the different types of users of our project. The rest of this section explains the use cases in detail, with sequence diagrams explaining the flow of messages.

Figure 2: Use Case Diagram

Create an account

Use case number: UC1

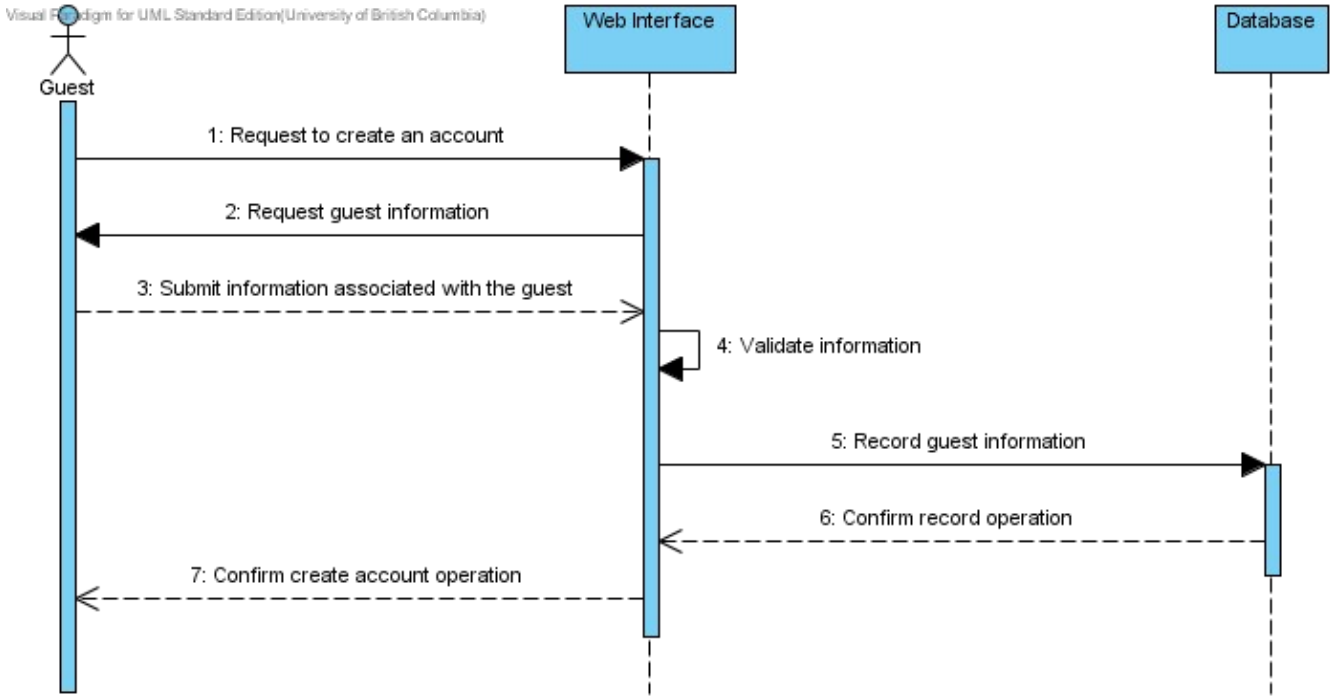

Figure 3: Create an Account (Sequence Diagram)

1. Guest enters his email address, password, credit card information and other relevant information
2. System validates all the information is correct
3. System records all the information and logins the guest

Make a reservation

Use case Number: UC2

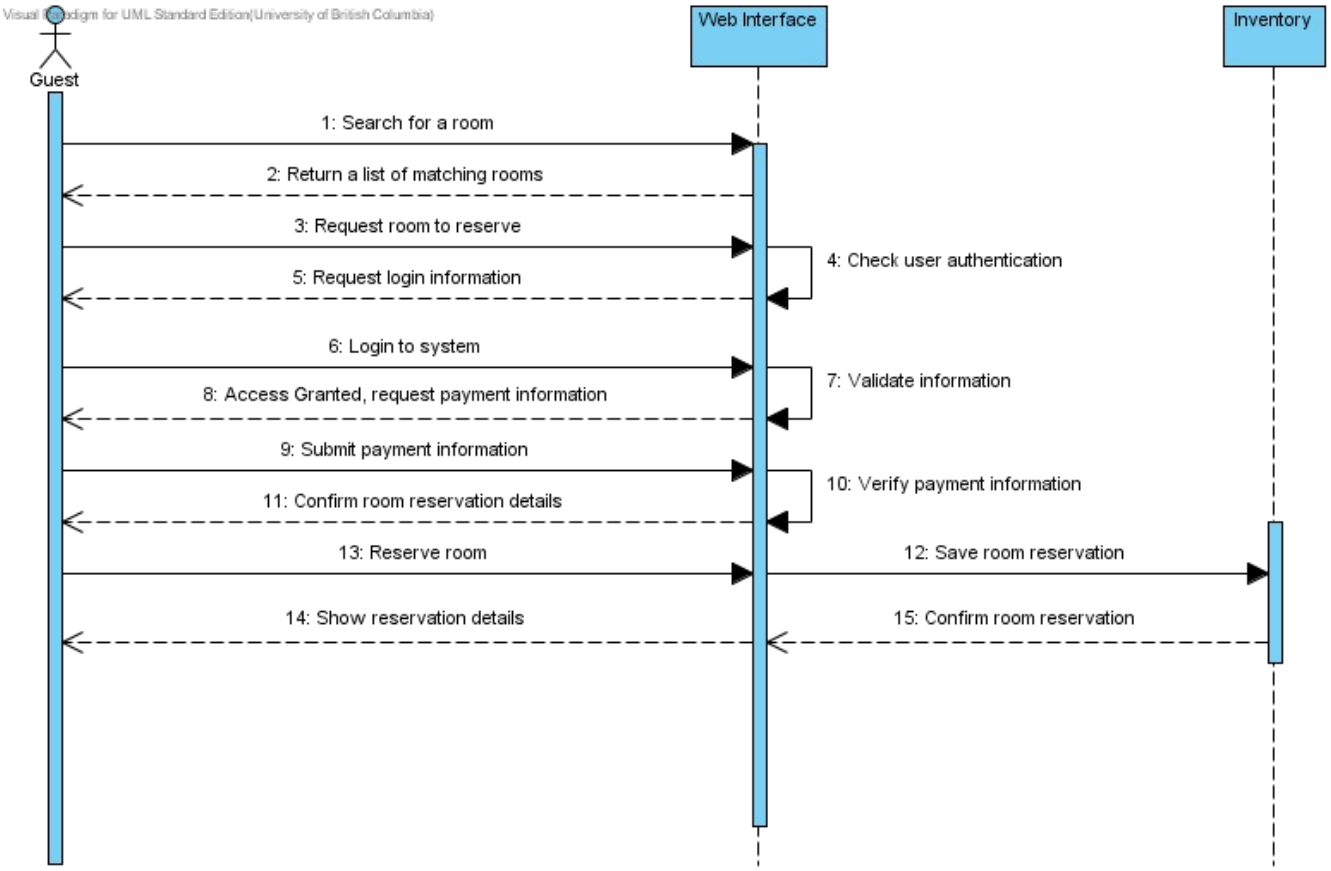

Figure 4: Make a Reservation (Sequence Diagram)

1. Guest or Staff enters the check-in and check-out dates, room capacity, and price range to the system.
2. The system will return a list of rooms.
3. User selects a room from this list to reserve.
4. If guest, guest logs on to retrieve her already stored customer information.
   4.1. If guest does not have an user account, include UC1
5. User selects the payment type (Visa or MasterCard)
6. User confirms the reservation
7. The system generates a confirmation number and a reservation summary

Cancel a reservation

Use case Number: UC3

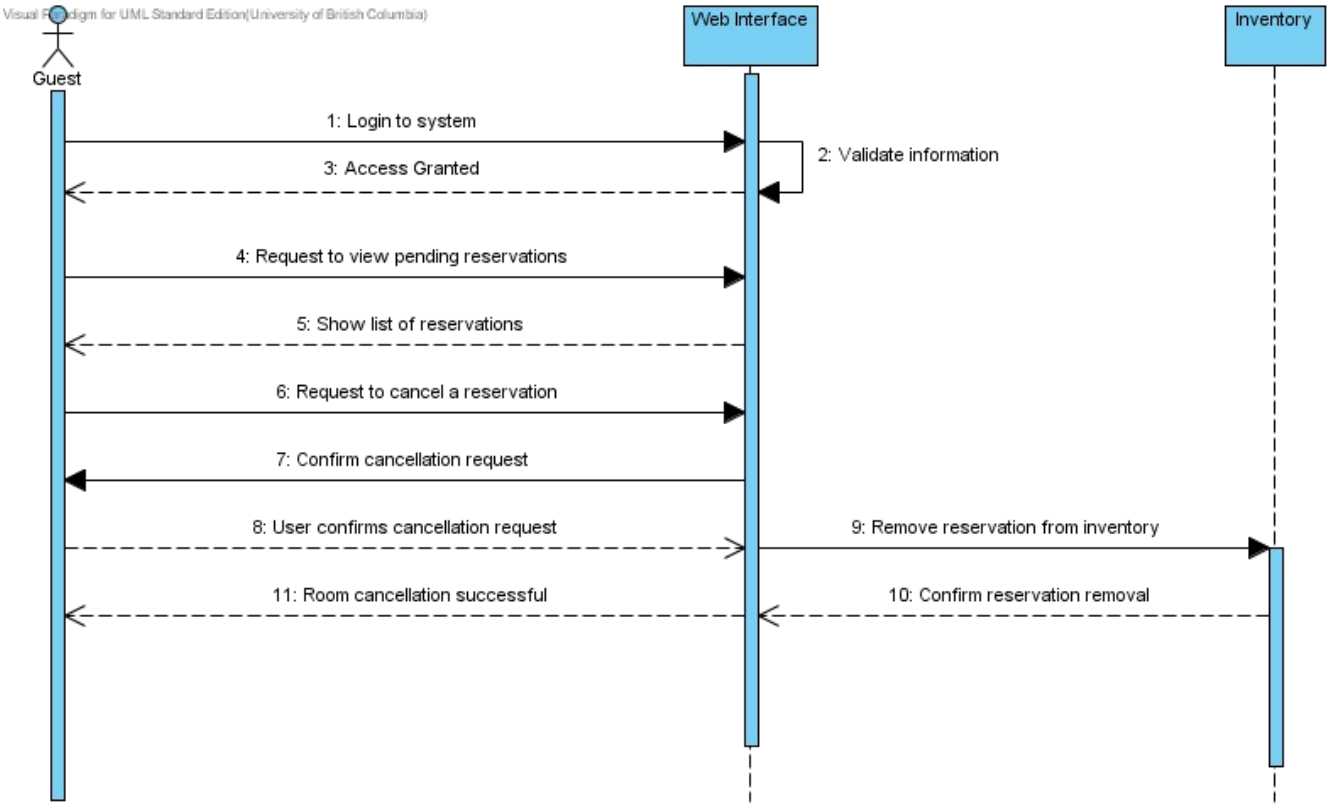

Figure 5: Cancel a Reservation (Sequence Diagram)

1. Guest logins to his account
2. Guest requests to view all the reservations
3. System responds with a list of reservations
4. Guest requests to cancel one of listed reservation
5. System validates that reservation can be cancelled
   5.1. System will display an error if the reservation cannot be cancelled
6. System displays any cancellation fees to the guest
7. Guest confirms to proceed with the cancellation
8. System confirms the reservation has been cancelled

Edit a reservation

Use case Number: UC4

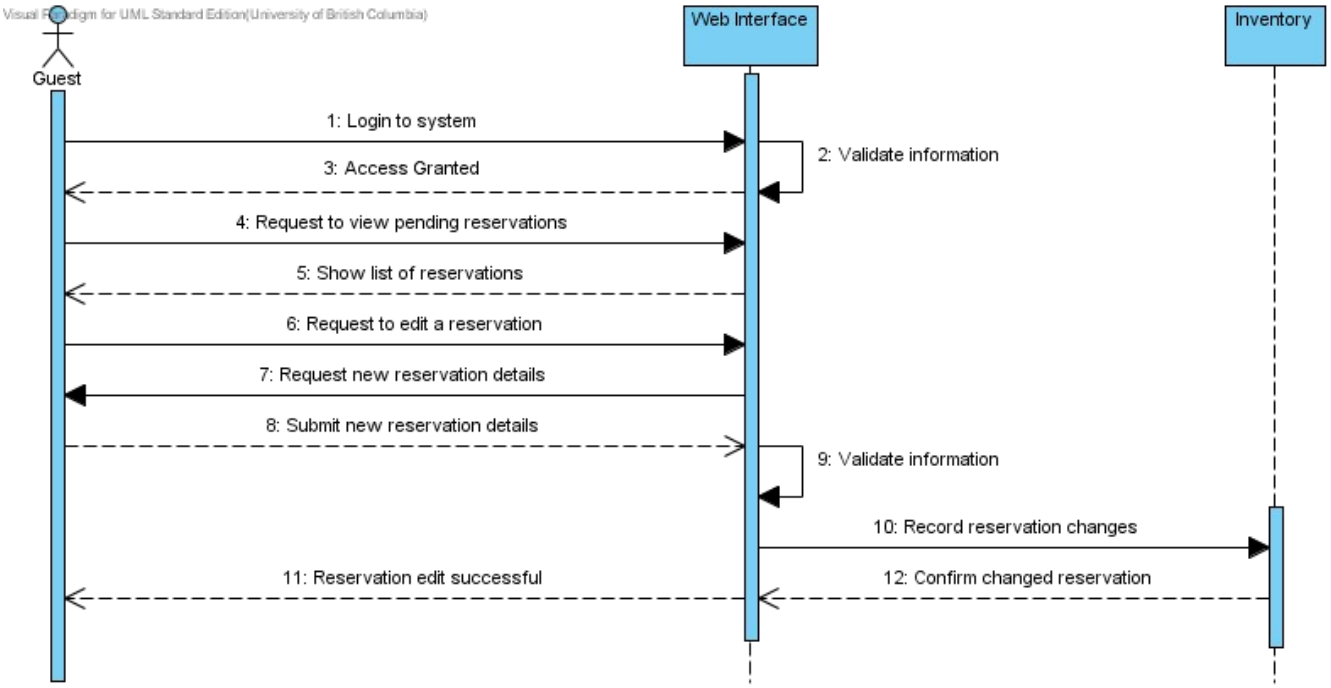

Figure 6: Edit a Reservation (Sequence Diagram)

1. Guest logins to his account
2. Guest requests to view all the reservations
3. System responds with a list of reservations
4. Guest requests to edit one of listed reservations
5. System prompts the user to enter new reservation details
6. Guests enter new reservation details
7. System validates the new input
8. System validates that changes can be accomidated
   8.1. System displays an error if a matching room cannot be found
9. The system generates a confirmation number and a reservation summary

Create rooms

Use case Number: UC5

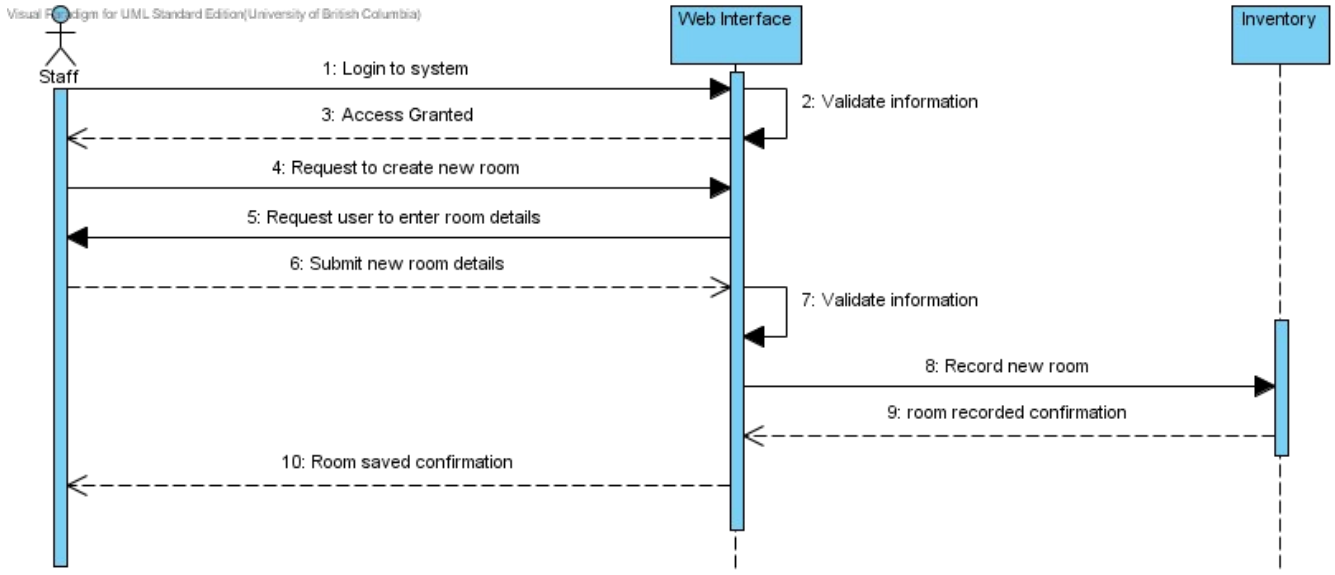

Figure 7: Create Rooms (Sequence Diagram)

1. Manager logs into the system using his or her account
2. Manager requests to create a new room
3. Manager selects an existing room type
4. Manager enters a list of room numbers and any information that should overwrite the default
5. System validates all the information entered is correct
   5.1. System displays an error message if the information is incorrect
6. System confirms a list of new rooms are added

Edit room

Use case Number: UC6

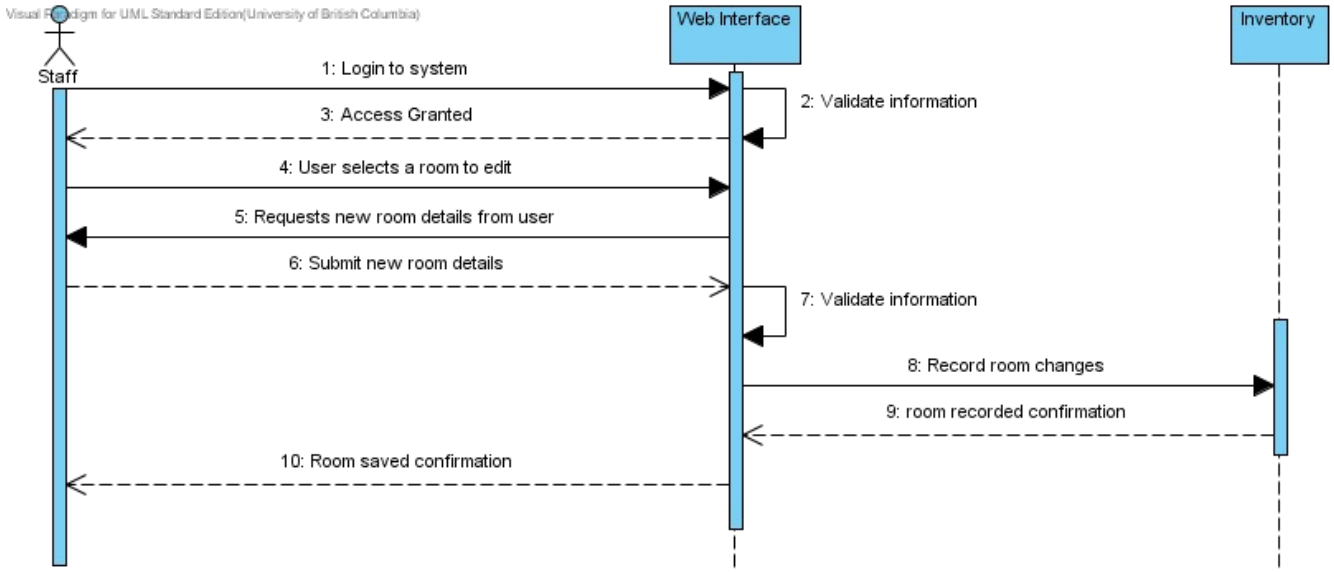

Figure 8: Edit Room (Sequence Diagram)

1. Manager logs into the system using his or her account
2. Manager selects a room to edit
3. Manager selects an existing room type
4. Manager can change room type, room number and any information that should overwrite the type's defaults
5. System validates all the information entered is correct
   5.1. System displays an error message if the information is incorrect
6. System confirms the changes

Delete rooms

Use case Number: UC7

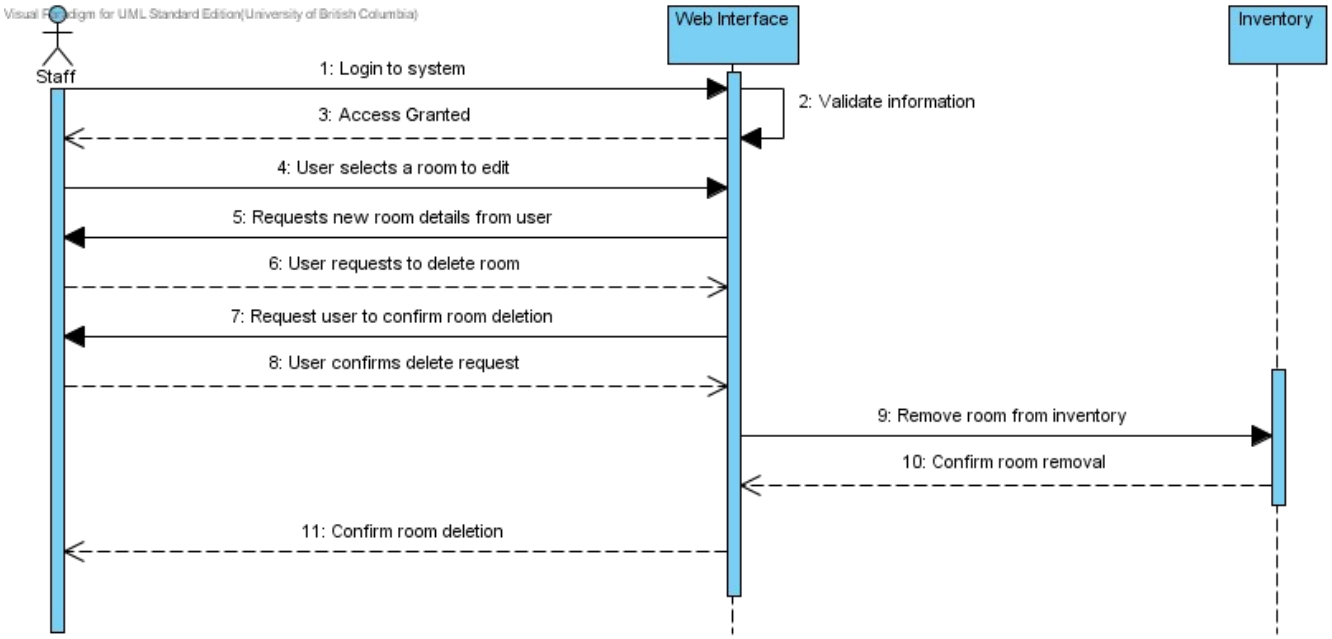

Figure 9: Delete Rooms (Sequence Diagram)

1. Manager logs into the system using his or her account
2. Manager selects a list of rooms to delete
3. System asks for confirmation
4. Manager confirms deletion
5. System confirms the deletion

Create room type

Use case Number: UC8

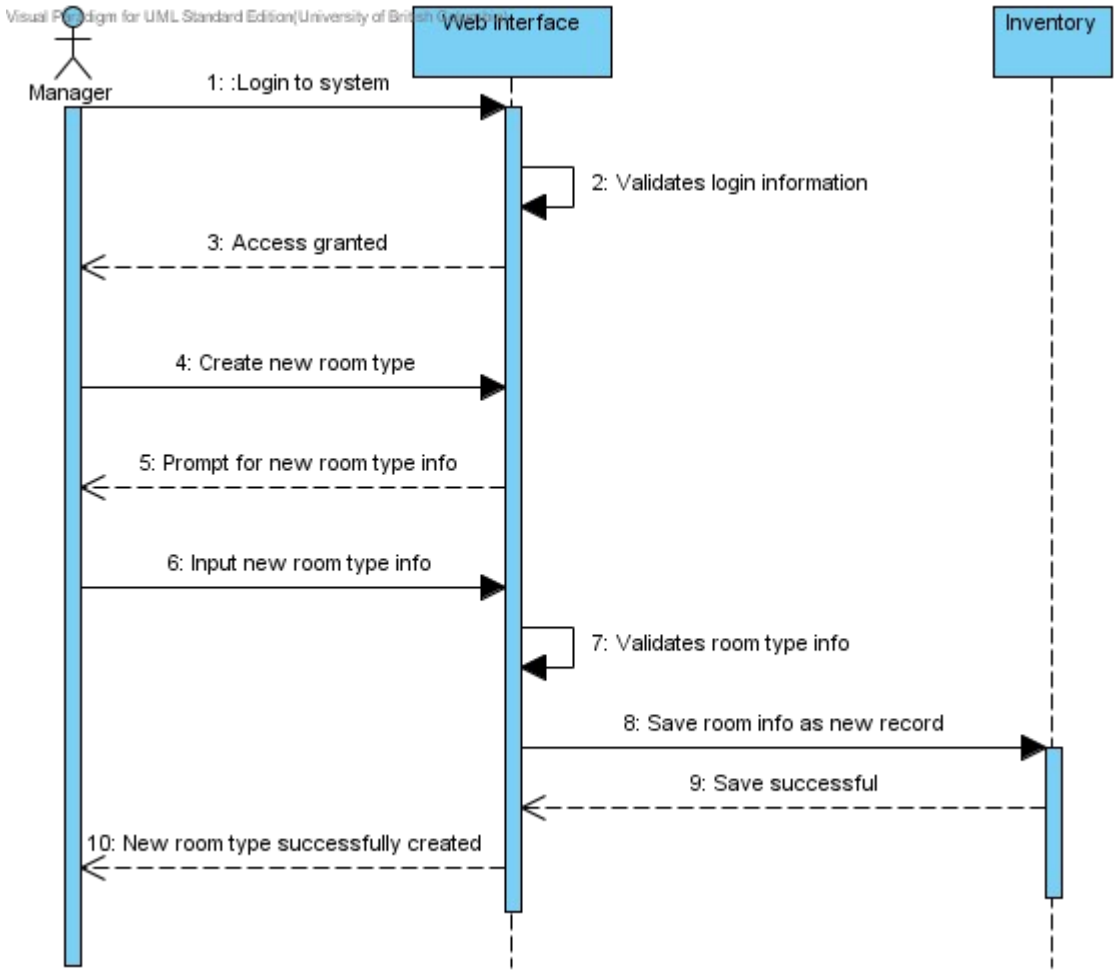

Figure 10: Create Room Type (Sequence Diagram)

1. Manager logs into the system using his or her account
2. Manager selects to create a new room type
3. Manager inputs name of type, beds, kitchenette, etc.
4. System validates all the information entered is correct

   4.1. System displays an error message if the information is incorrect
5. System confirms the changes

Edit room type

Use case Number: UC9

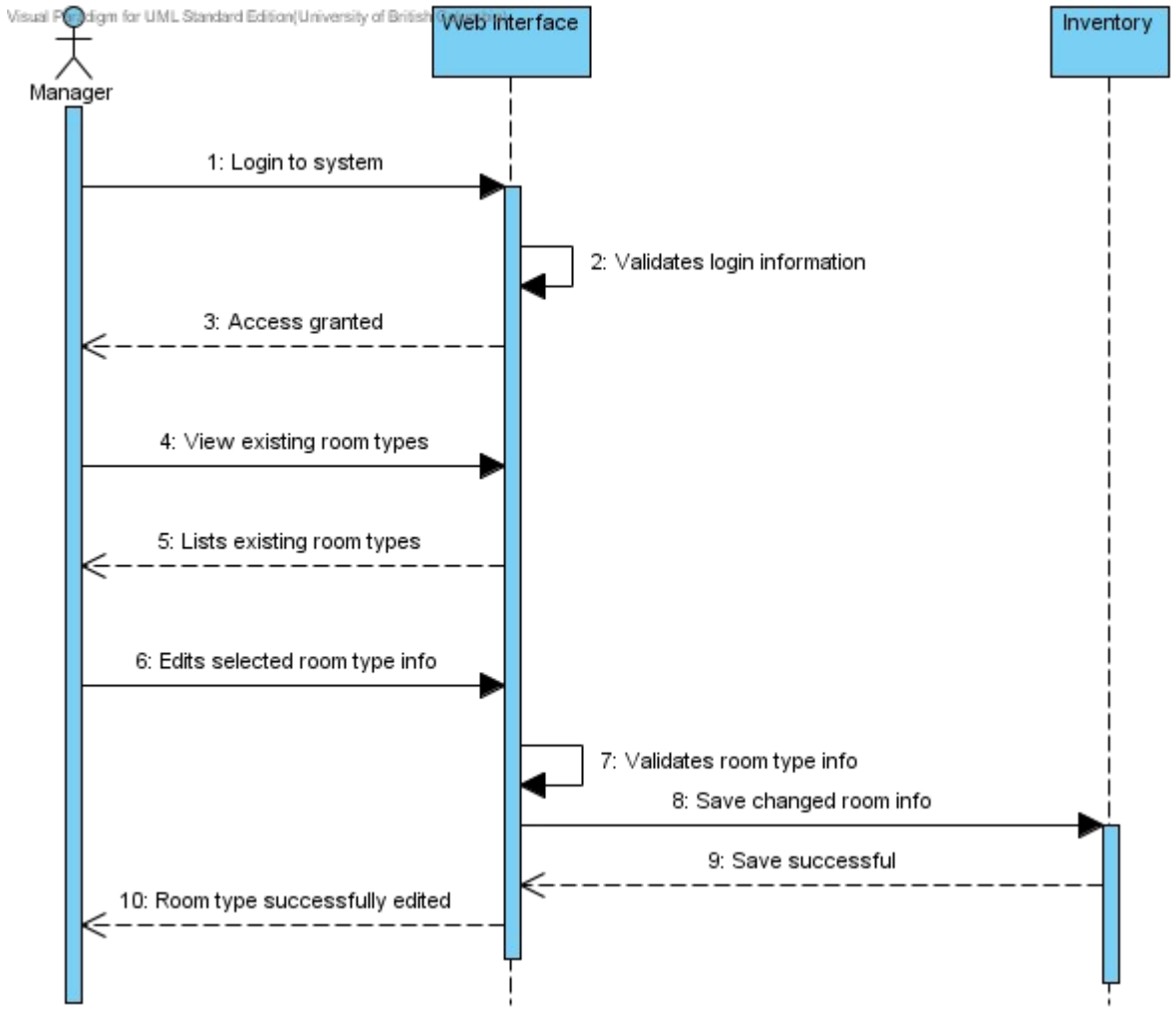

Figure 11: Edit Room Type (Sequence Diagram)

1. Manager logs into the system using his or her account
2. Manager selects room type to edit from list of types
3. Manager changes name of type, beds, kitchenette, etc.
4. System validates all the information entered is correct
   4.1. System displays an error message if the information is incorrect
5. System confirms the changes

Delete room type

Use case Number: UC10

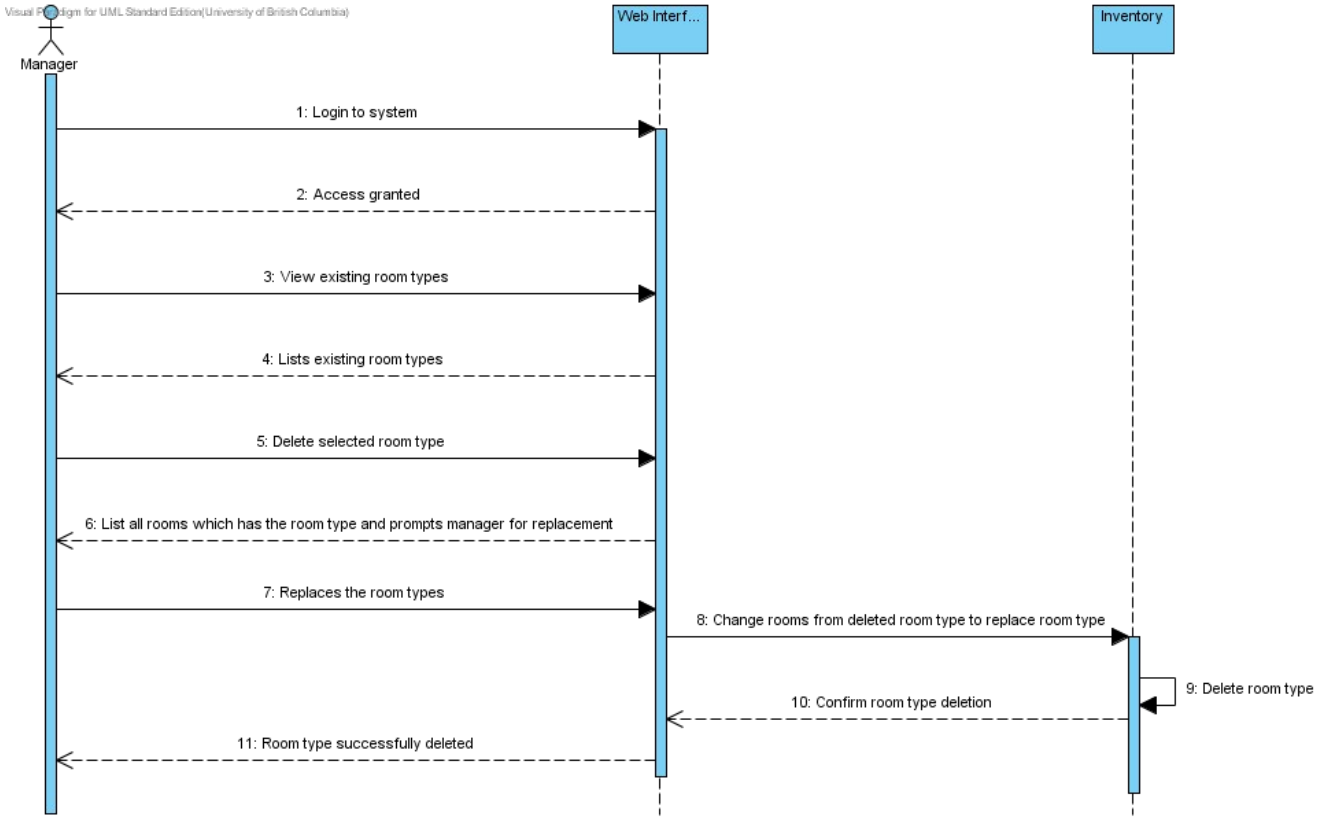

Figure 12: Delete Room Type (Sequence Diagram)

1. Manager logs into the system using his or her account
2. Manager selects room type to delete from a list of types
3. System checks that no rooms use that type

   3.1. If rooms use this type, system prompts for a type to replace this type with
4. System confirms the changes

Add chargeable item(s) [elided: edit chargeable item, delete chargeable item]

Use case Number: UC11

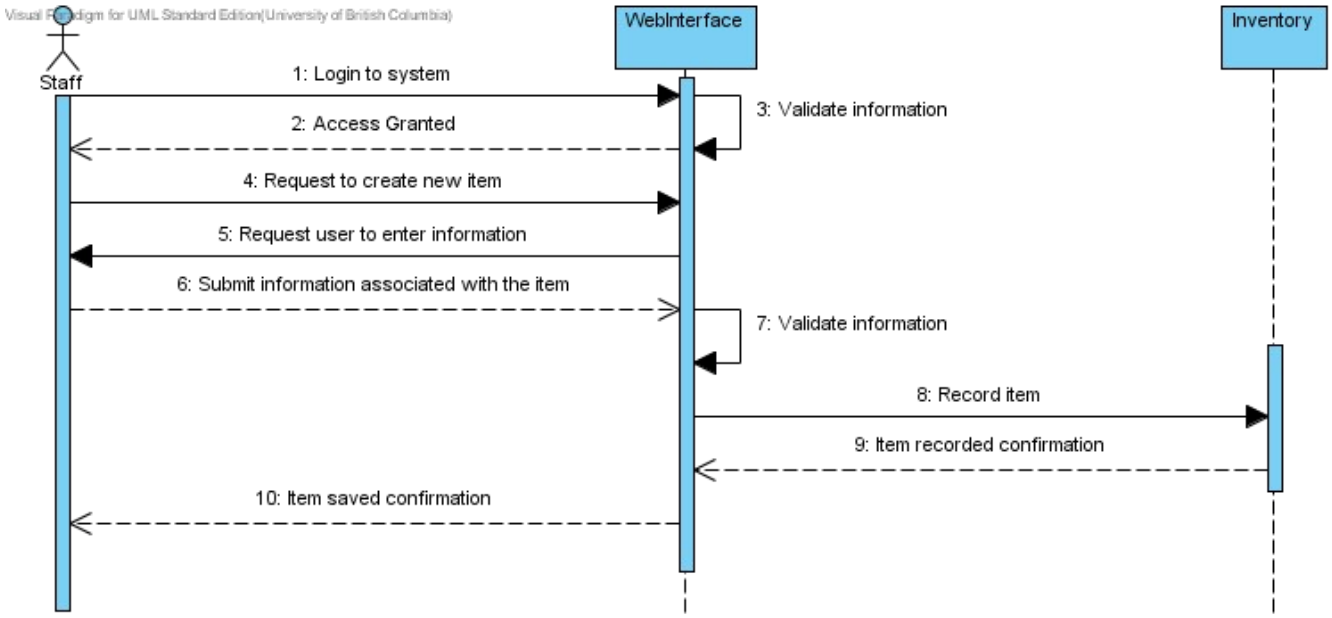

Figure 13: Add Chargeable Items (Sequence Diagram)

1. Hotel staff logs into the system using their account
2. Staff requests to create a new item
3. Staff enters all the information about the new item such as name and price
4. System validates all the information entered is correct
   4.1. System displays an error message if the information is incorrect
5. System confirms a new item is added

Charge chargeable item(s)

Use case Number: UC12

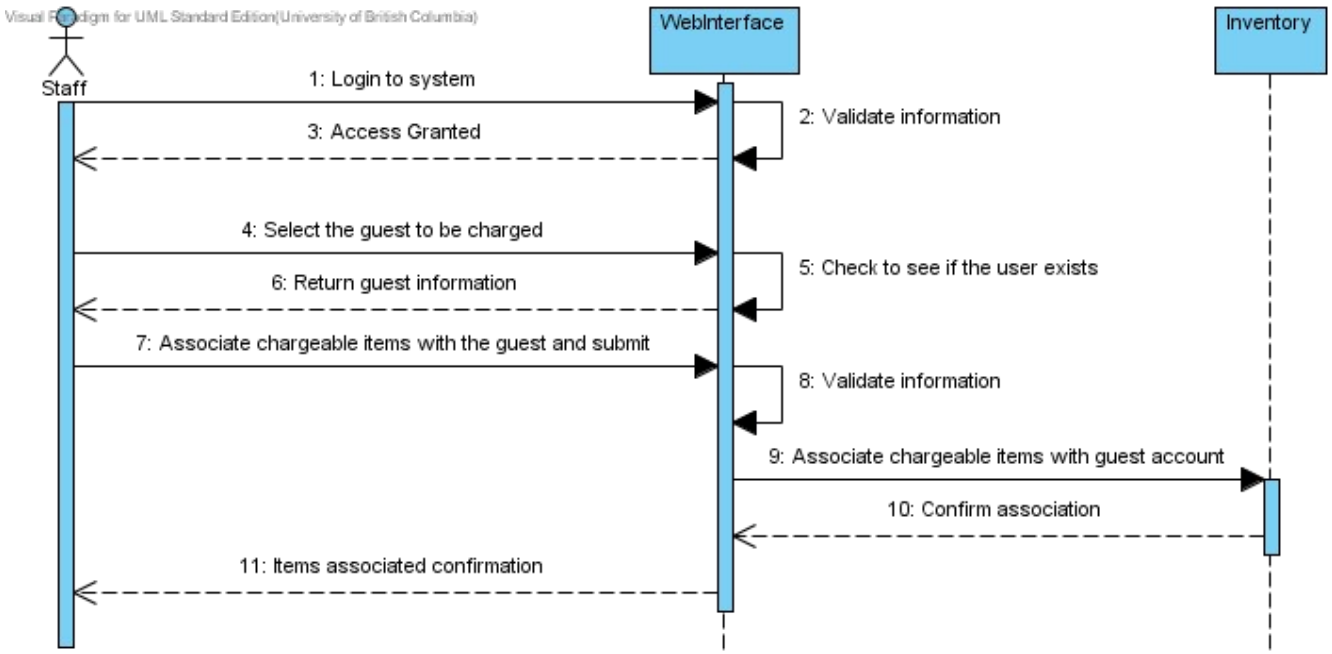

Figure 14: Charge Chargeable Item (Sequence Diagram)

1. Hotel staff logs into the system using their account
2. Staff selects (by room #, customer name, reservation #) guest to bill
3. Staff select chargeable items and quantity
4. System validates all the information entered is correct
   4.1. System displays an error message if the information is incorrect
5. System confirms a new item is added

Check-in

Use Case Number: UC13

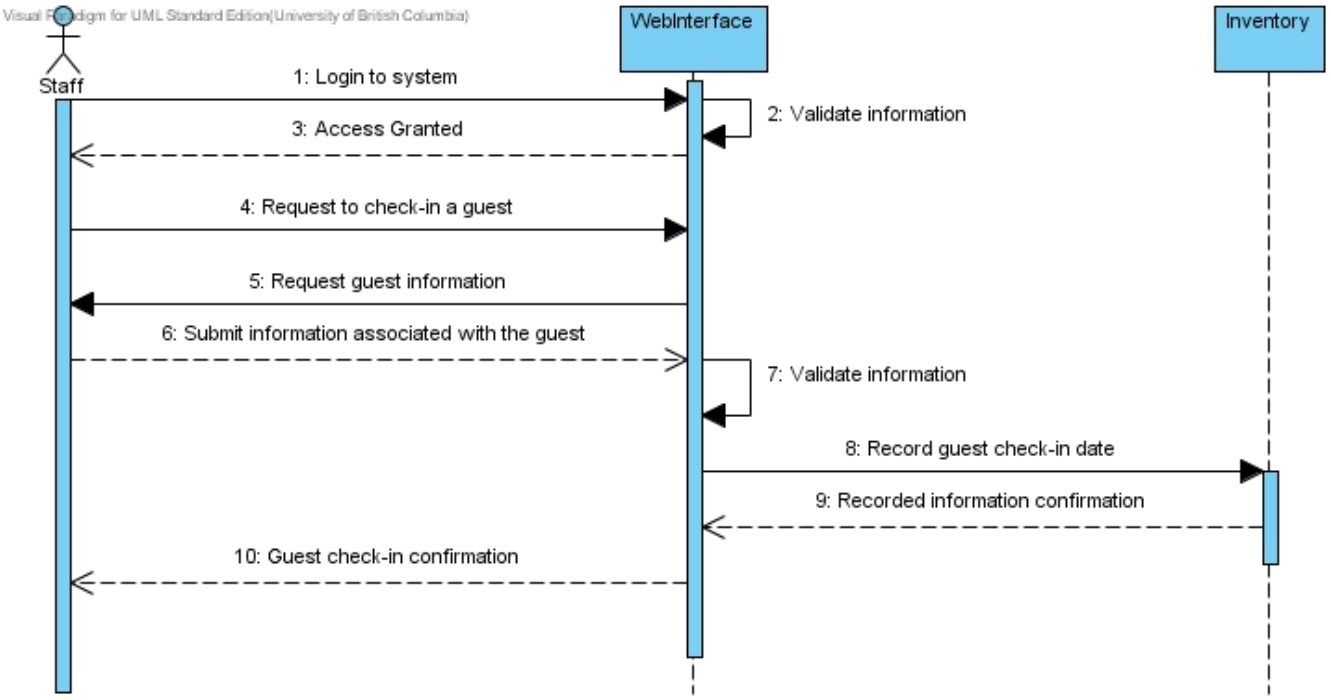

Figure 15: Check-in (Sequence Diagram)

1. Hotel staff logs into his or her account
2. Hotel staff requests check-in
3. Receptionist enters corresponding information (customer name, ID number) to retrieve customer data
   3.1. System displays an error message if the corresponding customer is not found
4. System displays a detail customer information summary including customer name, ID number, telephone number, address, room type, room number and stay duration
5. Guest or receptionist validates the correctness of the summary
6. System confirms check-in

Check-out

Use Case Number: UC14

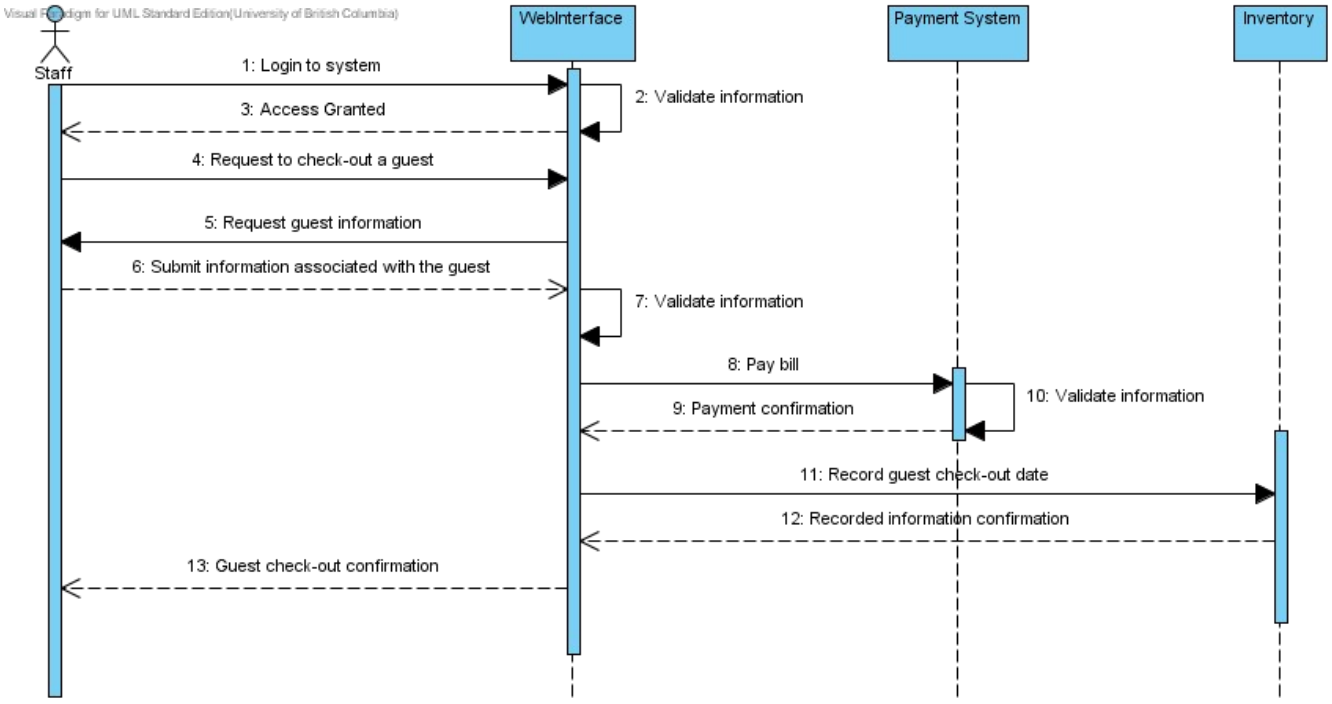

Figure 16: Check-out (Sequence Diagram)

1. Hotel Staff logs into to his or her account
2. Hotel Staff enters corresponding information (customer name, ID number) to retrieve customer data
   2.1. System displays an error message if the corresponding customer is not found
3. System displays a detail customer information summary including customer name, ID number, telephone number, address, room type, room number and stay duration
4. System displays a summary of the stay and a detailed bill
5. Guest and hotel staff validate the correctness of the summary and bill
6. System charges the guest's credit card
7. System confirms check-out

Check Dirty Room

Use Case Number: UC15

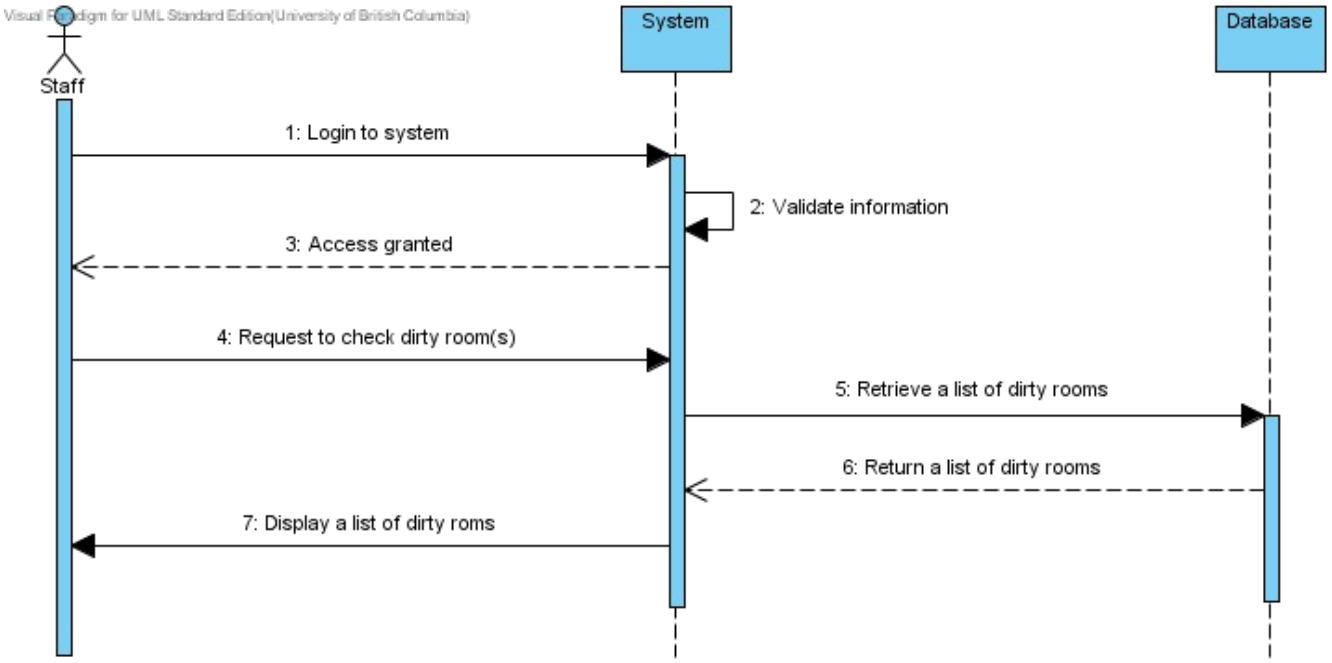

Figure 17: Check Dirty Room (Sequence Diagram)

1. Hotel staff logs into his or her account
2. Hotel requests to check dirty rooms
3. System displays a list of rooms that is dirty

Mark Clean Room

Use Case Number: UC16

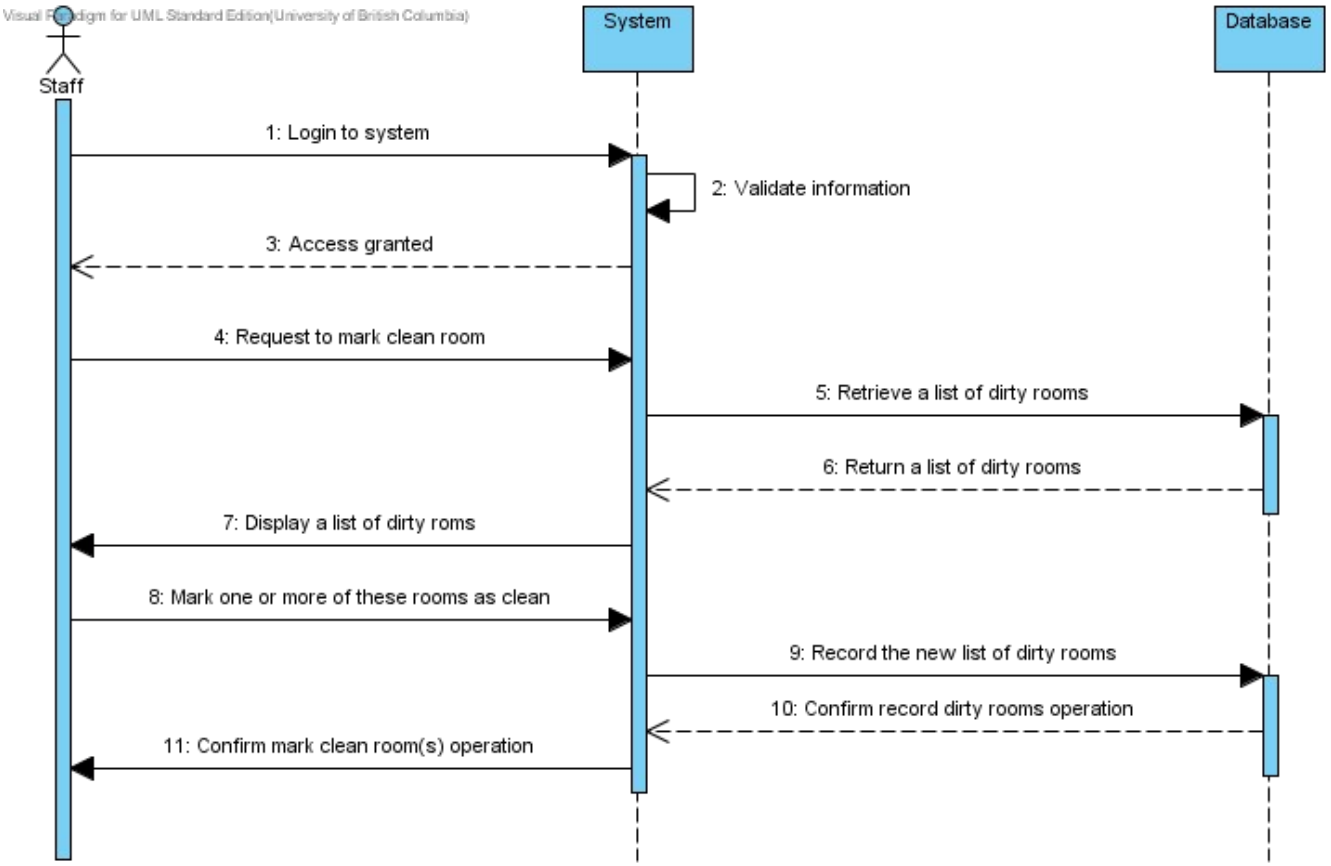

Figure 18: Mark Clean Room (Sequence Diagram)

1. Hotel staff logs into his or her account
2. Hotel staff requests to mark a room clean
3. System displays a list of rooms that are dirty
4. Hotel marks one or more of these rooms
5. System confirms the rooms being marked are indicated as “clean”

View Statistic Report

Use Case Number: UC17

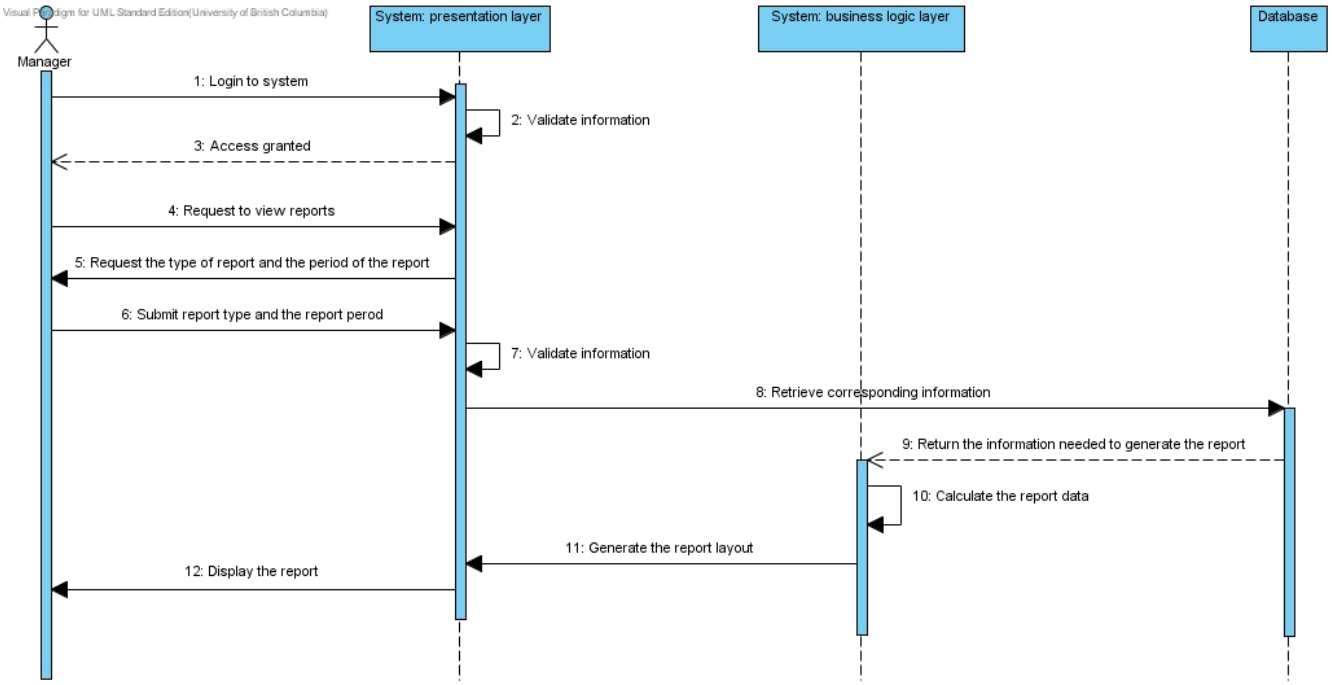

Figure 19: View Statistic Report

1. Manager logins his account
2. Manager requests to view reports
3. Manager enters the type of report (statistical analysis room occupancy, most popular type of room reservation, real-time view of occupied/free status) and the period of the report
4. System displays the corresponding report

Appendix A: User Interface Mockups

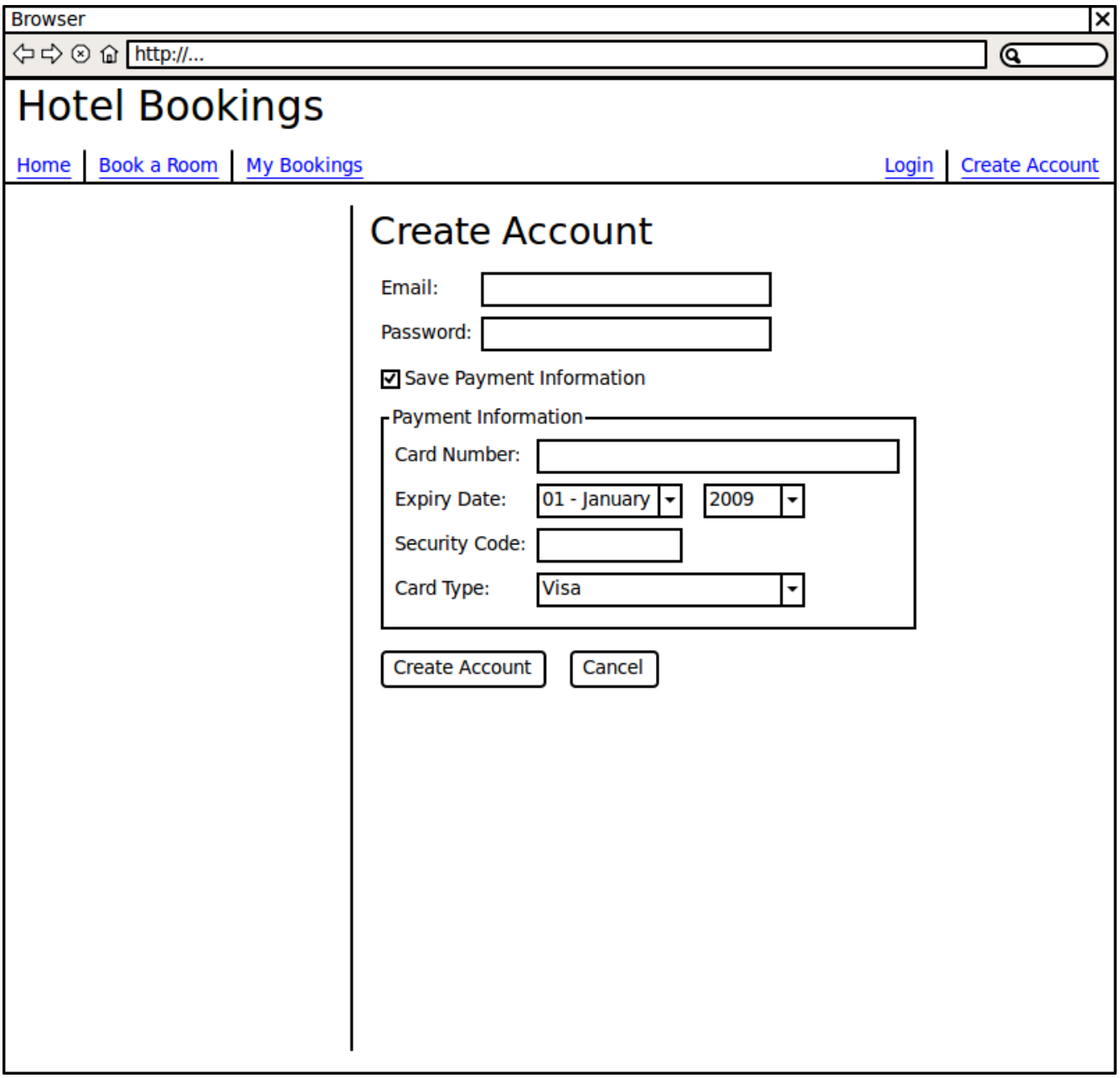

Figure 20: UC1 Mockup - Create Account

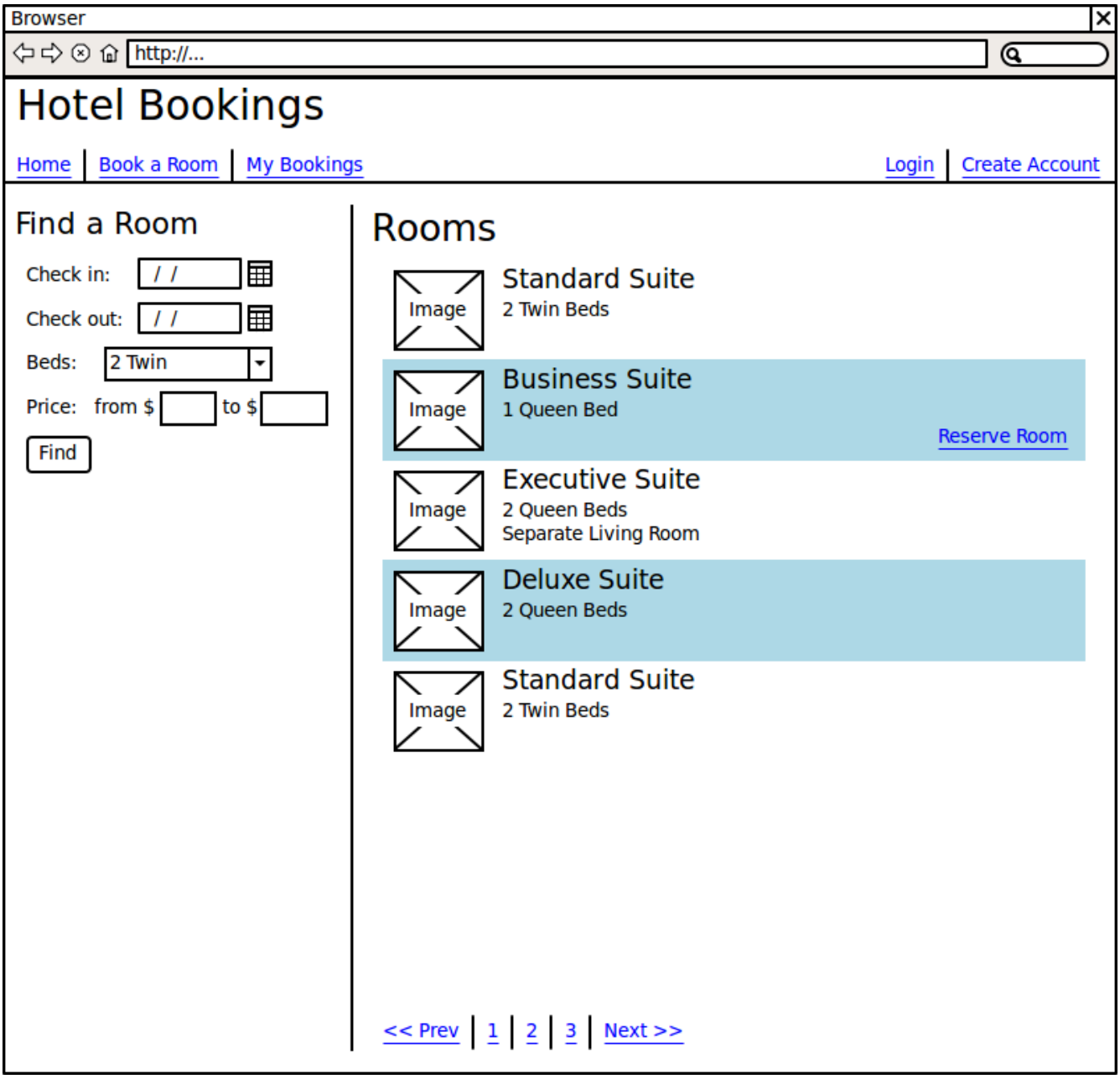

Figure 21: UC2 Mockup - Make a Reservation (Step 1)

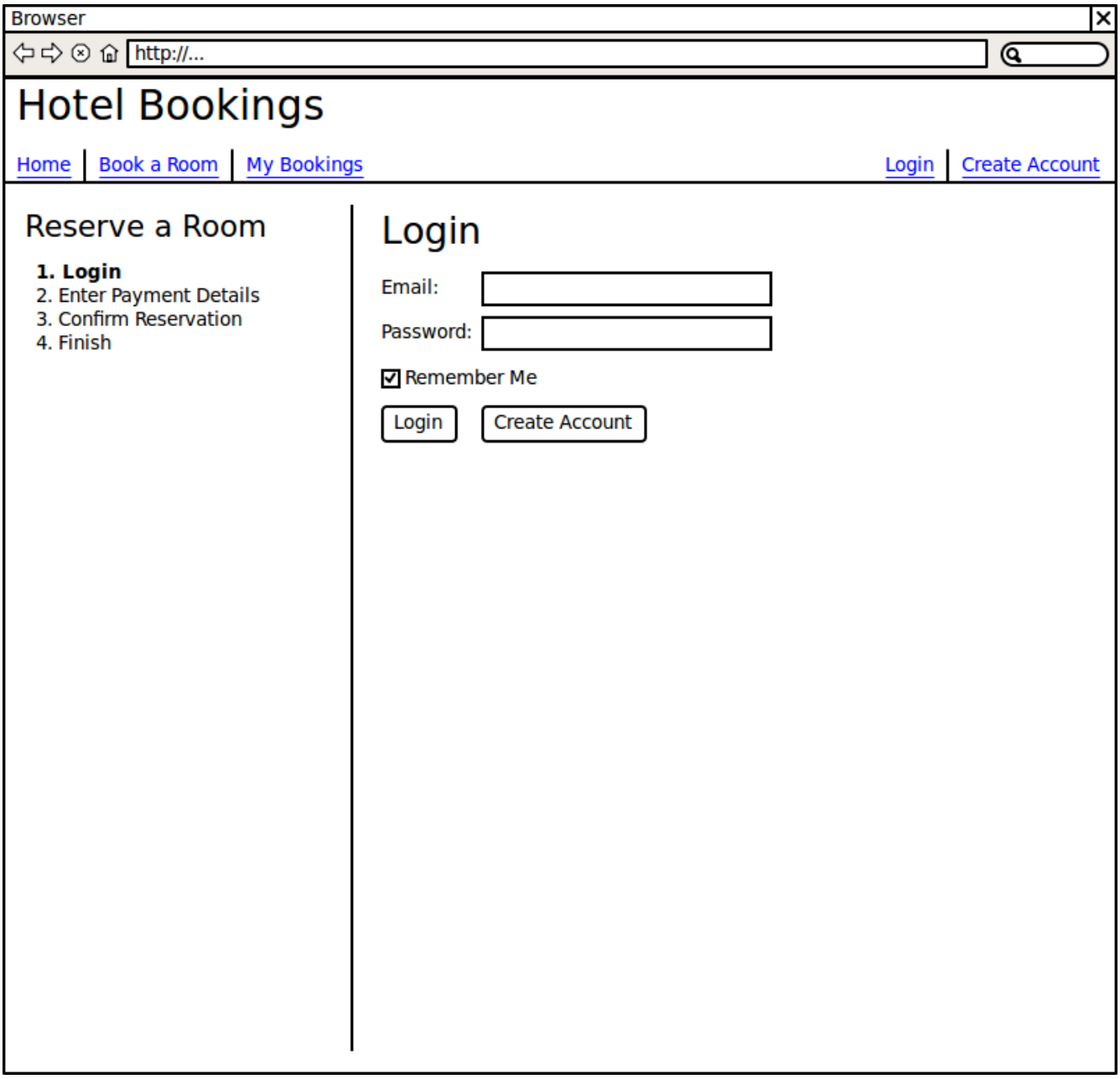

Figure 22: UC2 Mockup - Make a Reservation (Step 2)

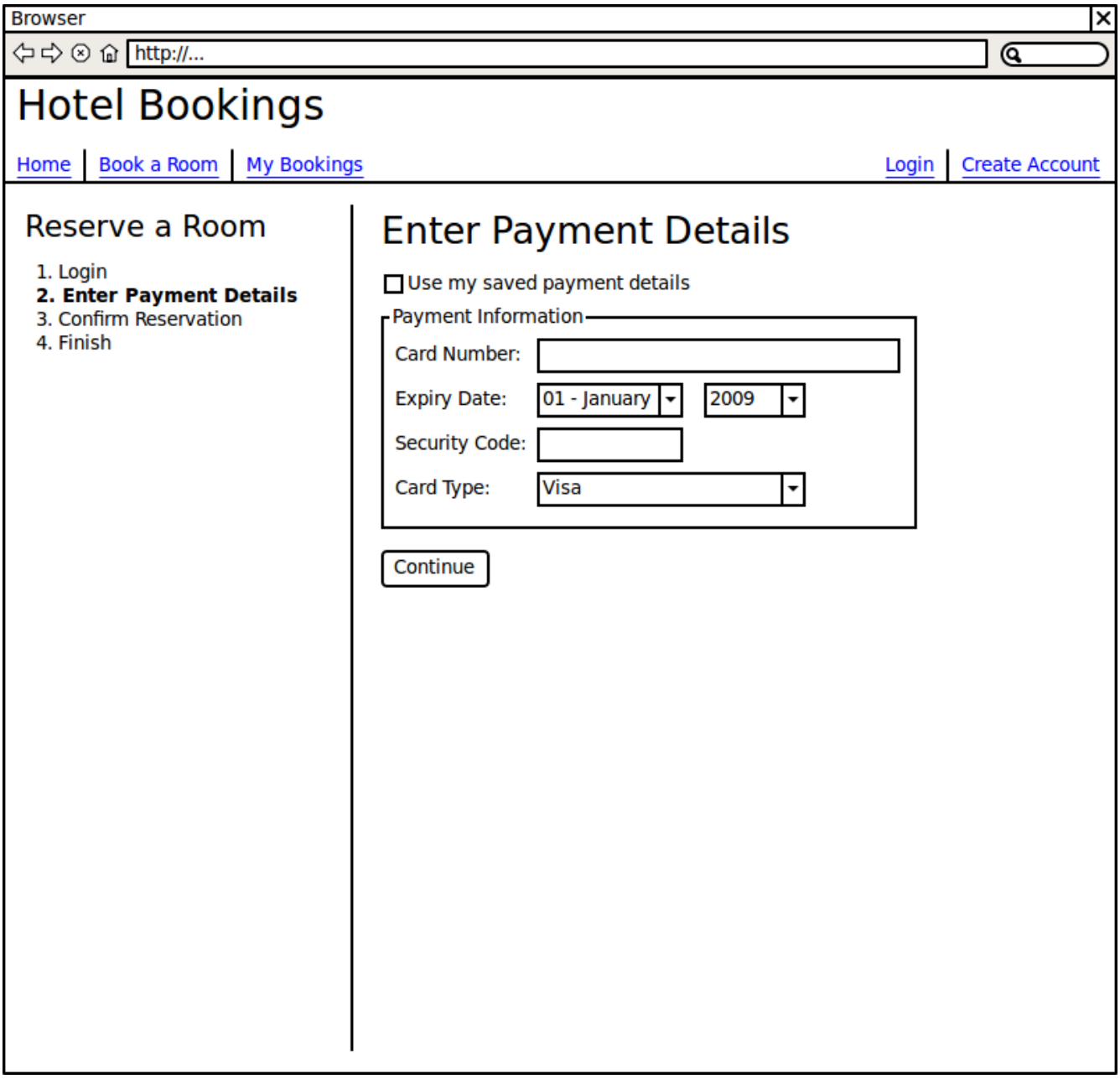

Figure 23: UC2 Mockup - Make a Reservation (Step 3)

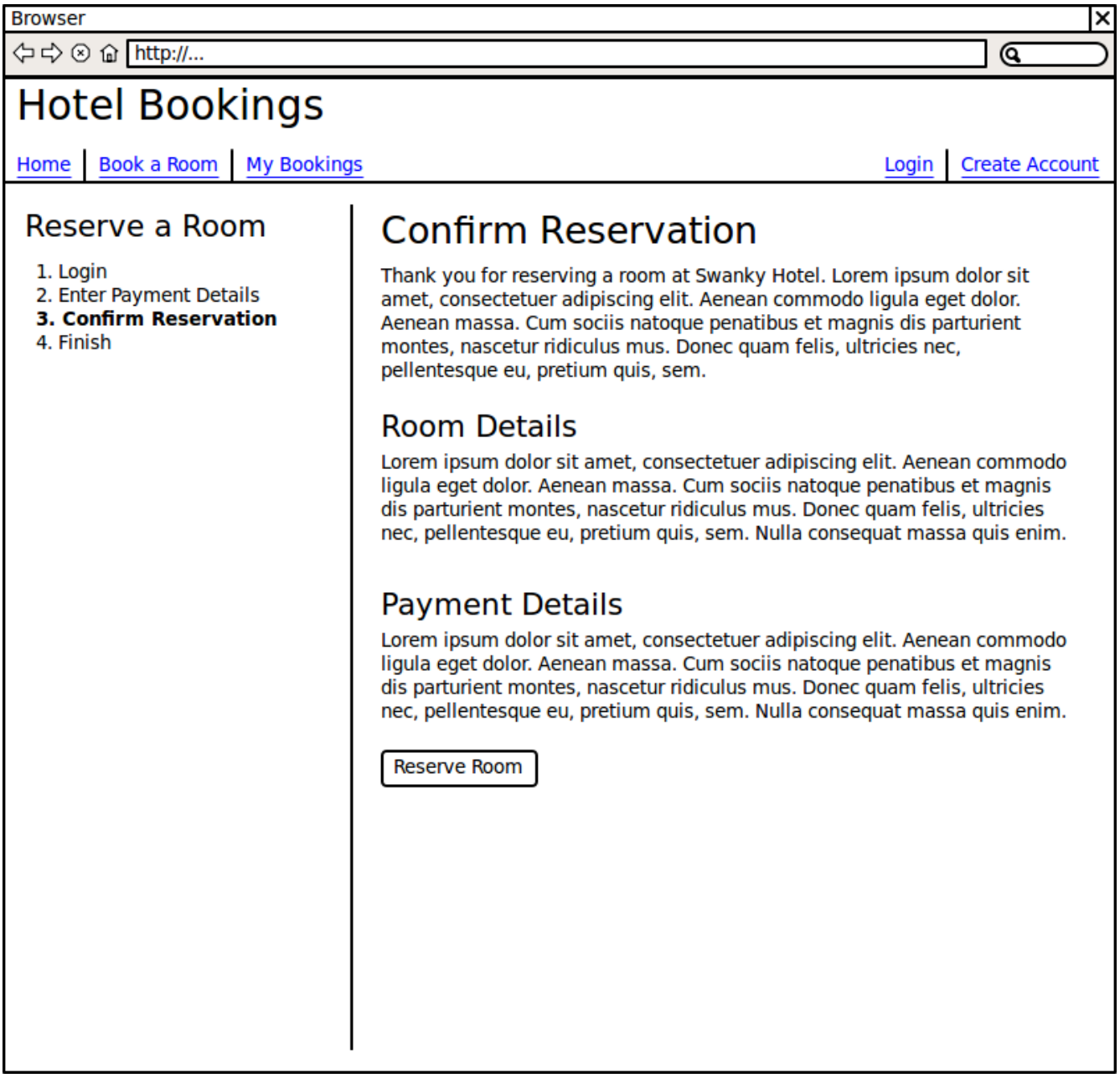

Figure 24: UC2 Mockup - Make a Reservation (Step 4)

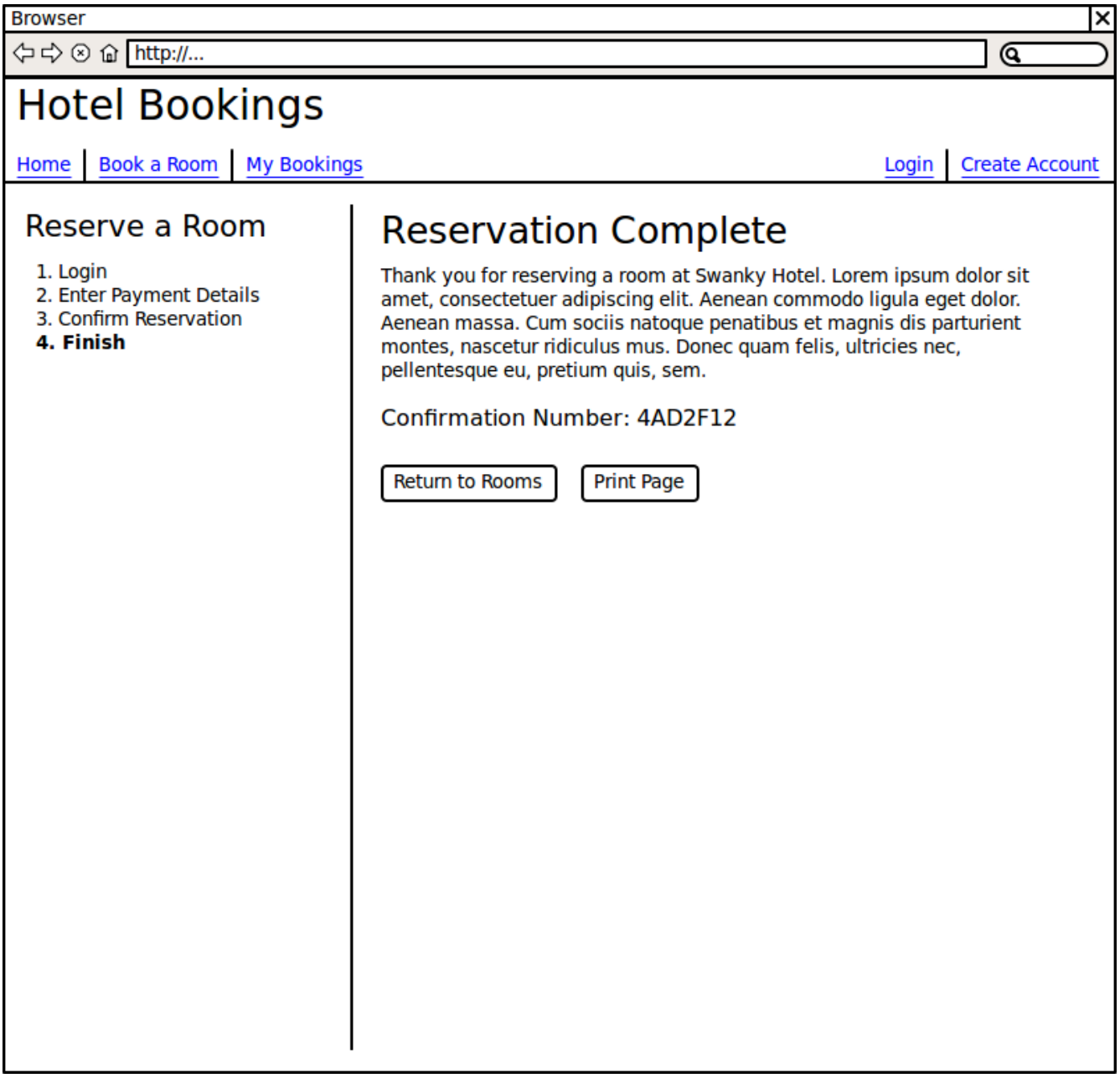

Figure 25: UC2 Mockup - Make a Reservation (Step 5)