SOFTWARE ENGINEERING

Assignment: Requirements Specification

Student: S244679

Contents

1 Introduction 3
2 System Users 3
3 Constraints 3
4 User Stories 4
5 Use Cases 7
6 Functional Requirements 9
7 Conceptual Class Diagram 11
8 Sequence and Activity Diagrams 12
7 References 27

1 Introduction

A local hotel business wishes to replace its paper-based reservation process with a computer-based reservation system.

Their immediate goals from the system are:

- To eliminate the double booking of rooms.
- To have a record of guests that have stayed at the hotel (for future marketing purposes)
- To make room bookings more efficient to provide better customer service

Although not to be implemented as part of this project, the hotel owner has future goals for the system that should be considered.

- To analyse the data collected to help improve the business.
- To manage other aspects of the hotel via the system, such as room maintenance and cleaning.
- To allow guests to reserve rooms themselves via the internet.

This document details the functional and non-functional requirements for this Hotel Management system. It focuses on the areas to be implemented during this project but will capture high-level requirements for future consideration as they arise. It is expected that the system will be developed and delivered using an incremental strategy.

2 System Users

There are three types of users (actors) of the system

- *Receptionists* – these customer facing staff will use the system to reserve rooms for guests, process check-in/check-out and answer queries during a guests stay at the hotel.
- *Managers* – will use the system to define and modify the rooms on the system and set prices and discounts. Once future goals are implemented, they will also use the system to prepare statistical reports and manage maintenance and cleaning rotas.
- *IT Administrators* – will maintain the system and its hardware. An external company will likely be used to fulfil this role rather than employing administrators as staff.

In the future, *Guests,* could be users of the system to reserve their own rooms via the internet.

3 Constraints

The hotel owner wishes to keep the cost of the system at a minimum. The hotel owns two Windows PCs (4 years old, i3 processor, 8GB RAM) and wants to use them as client PCs for system usage.

The first increment of the system must be tested and delivered by 11th March 2025. Increment 2 can be planned and delivered early Summer 2025 with exact details to be discussed after successful delivery of Increment 1. If necessary, lower priority requirements in this document can be delayed beyond Increment 1 to ensure delivery timescales of Increment 1 are met.

4 User Stories

ID	Story	Acceptance Criteria
U1	As any user of the system, I want to **login** with a username and password and be shown a menu of the system features that are available to my role, so that I can perform my daily tasks. When I've finished working I will **logout**.	• The system restricts access until valid login details have been provided. • Following successful login the system only shows and allows access to the system features the user's role permits. • The system provides a logout facility.
U2	As a manager, I want to **configure** the hotel **room types** on the system so that I can define the name, price, room quality, bathroom features and maximum number of guests.	• The system provides a list of room types currently defined. • The system allows a room type to be added/edited/deleted. • The room type editor allows the name, price, room quality, bathroom features and maximum number of guests to be entered and all entries are validated before saving. • Validation errors are displayed to the user. • When room type details are amended, the list display is updated.
U3	As a manager, I want to **manage** a list of hotel **rooms** on the system, by adding and removing them as necessary. Any room added to the list will be considered available for reservation when not already reserved. When adding a room, I will set the room number and choose the room type which will set its price and features.	• The system provides a list of rooms currently defined along with important details from its associated room type. • The system allows a room to be added/edited/deleted. • The room editor allows the room number and the room type to be entered and all entries are validated before saving. • Validation errors are displayed to the user. • When room details are amended, the list display is updated.
U4	As a manager, I want to **add/delete promotional discounts** to the system, so they can be used to validate promotional codes used by guests and used by the system to calculate the correct price.	• The system provides a list of promotional discounts currently defined. • The system allows a promotional discount to be added/edited/deleted. • The promotional discount editor allows the promotion code and the percentage discount to be entered and all entries are validated before saving. • Validation errors are displayed to the user. • When promotional discounts are amended, the list display is updated.
U5	As a receptionist, I want to **find available rooms** based on my search criteria so that I can offer suitable rooms to satisfy a guest's booking request.	• The system provides a list of rooms that are not currently reserved for the

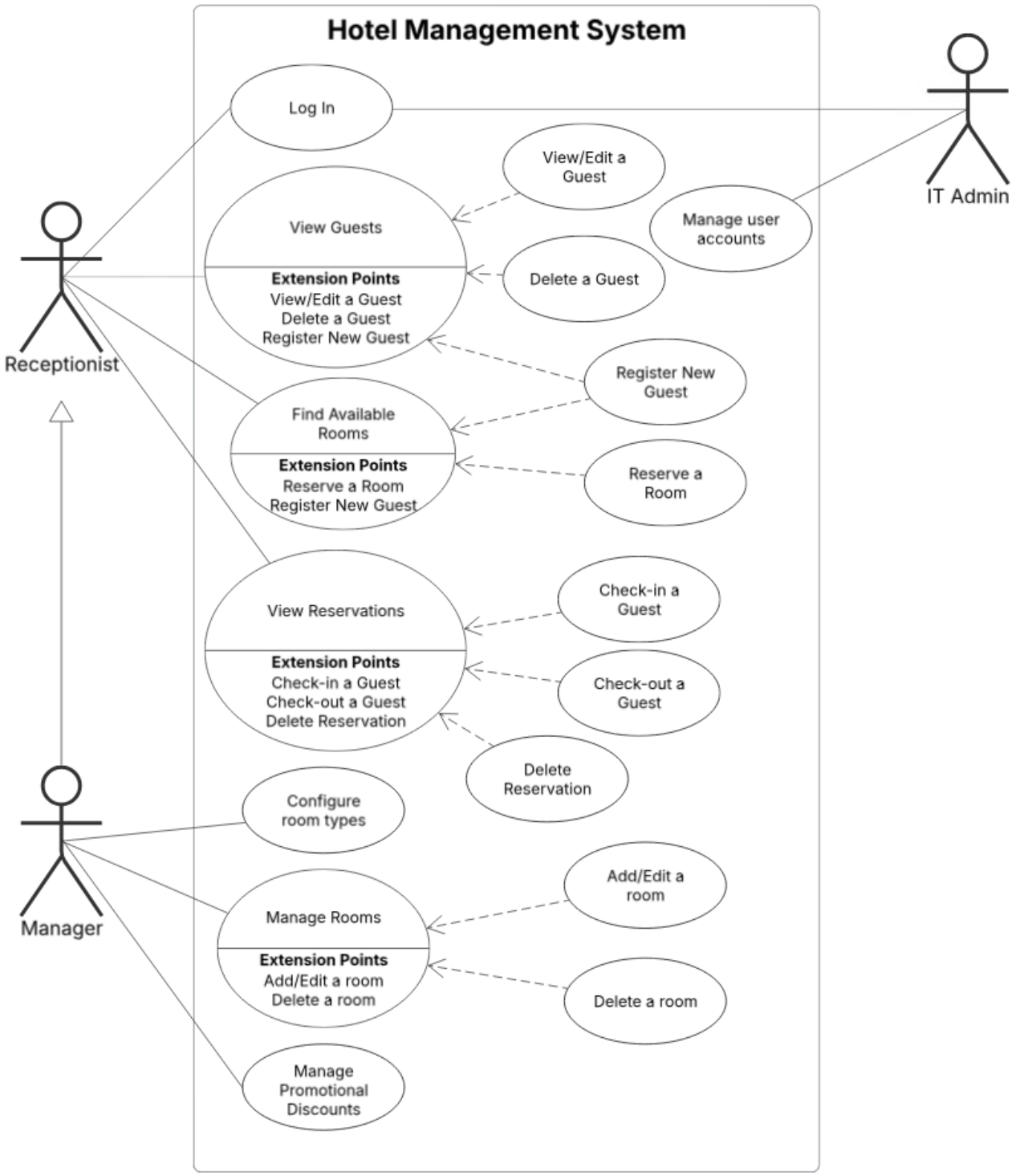
	I will provide the start date for the search and the required length of stay. Optionally, I may wish to filter the results by the type of room.	date range provided by the receptionist. • The system provides the ability to filter the results by room type. • The system allows a room to be selected for use in the reservation process.
U6	As a receptionist, I want to **register a new guest** on the system, so that I have record of their full name, their address, telephone number, and e-mail address.	• The system provides an editor to capture guest details including full name, address, telephone number, and e-mail address. • All entries are validated before saving. • Validation errors are displayed to the user. • When guest details are amended, the guest list display is updated.
U7	As a receptionist, I want to **view guests** registered on the system so that I can **edit** their details or assign them to a room reservation. Optionally, I may wish to search for a guest by name or by postcode.	• The system provides a list of guests currently registered on the system along with details to aid identification (such as name, first line of the address and postcode). • The system allows a guest to be added/edited/deleted. • The guest editor allows the guest's name, address, telephone number, and e-mail address to be amended. • All entries are validated before saving. • Validation errors are displayed to the user. • When guest details are amended, the guest list display is updated. • The system provides the ability to filter the results by name or postcode.
U8	As a receptionist, I want to **delete a guest** from the system in case they were registered incorrectly or request to be fully removed from our records.	• The system provides the ability to delete a guest from the database. • The system asks the user for confirmation before the deletion is actioned.
U9	As a receptionist, I want to **reserve a room** for a guest. I will provide the start date, the length of stay, and the number of guests staying in the room. During reservation the guest may use a promotional discount which will affect the price. The system must remember the price quoted at the point of reservation. The guest may pay a deposit or may provide additional information that I must be able to note on the system. I will process any payments using our existing methods, so the system should not do this.	• Following selection of an available room, the system displays an editor to capture the start date, the length of stay, and the number of guests staying in the room. • The system provides an option to register a new guest or select an existing guest. • The system allows a promotional discount to be used. • The system calculates the correct price based on the room type, the length of stay and any discounts used. • The system allows any payment made to be recorded along with any notes from the receptionist.

		• The system validates entered details and rechecks the room availability before saving the reservation. • Validation errors are displayed to the user.
U10	As a receptionist, I want to **view reservations** on the system**,** based on date search criteria, so I can delete it if the customer cancels the booking, or process a guest check-in / check-out.	• The system provides a list of room reservations along with details to aid identification (such as room number, first date of stay, guest name). • The system allows a reservation to be deleted. • The system allows for a reservation to be selected for check-in or check-out. • The system provides the ability to filter the results by date criteria.
U11	As a receptionist, I want to **check-in a guest** on arrival, the reservation must be updated to remember the checked-in state and should allow the reservation notes to be updated.	• Following selection of a reservation and the choice of 'check-in', the system allows reservation notes to be amended, and offers the user a button to confirm the check-in process. • If the user confirms check-in, the system updates the reservation status to checked-in and saves. • The reservation list is updated to reflect the new status.
U12	As a receptionist, I want to **check-out a guest** on departure, the reservation must be updated to remember the checked-out state. The system should show the amount left to be paid. I will process payment using our existing methods, so the system should not do this, but I should be able to update the amount paid on the reservation record.	• Following selection of a reservation and the choice of 'check-out', the system displays the amount left to pay. • Following confirmation from the receptionist that the money has been paid, the system updates the reservation status to checked-out and saves. • The reservation list is updated to reflect the new status.
U13	As an IT Administrator, I want to **add/edit/delete users** on the system and assign them to the role of Receptionist or Manager.	• The system provides a list of users currently registered on the system and allow users to be added/edited/deleted. • The system allows each user to have a unique username and a secure password. • The system allows users to be given either Manager or Receptionist permissions to the system features.

5 Use Cases

ID	Name	Role	Description	User Story ID
UC1	**Log in / out**	All	Gain access to the system features for a user role.	U1
UC2	**Configure Room Types**	Manager	Define and amend room types.	U2
UC3	**Manage Rooms**	Manager	Add/remove rooms and assigns a room type.	U3
UC4	**Manage Promotional Discounts**	Manager	Add/delete promotional discount codes.	U4
UC5	**Find Available Rooms**	Receptionist	Search for a suitable room for a guest to reserve.	U5
UC6	**Register a new Guest**	Receptionist	Add a guest's details to the system.	U6
UC7	**View Guests**	Receptionist	View the guests on the system, filtering if necessary.	U7
UC8	**Edit a Guest**	Receptionist	Edit the guest's details on the system.	U7
UC9	**Delete a Guest**	Receptionist	Delete a guest's details from the system.	U8
UC10	**Reserve a Room**	Receptionist	Reserve a room for a guest.	U9
UC11	**View Reservations**	Receptionist	View a list of room reservations, filtering if necessary.	U10
UC12	**Delete a Reservation**	Receptionist	Select a reservation and delete it.	U10
UC13	**Check-in a Guest**	Receptionist	Select a reservation and process check-in	U11
UC14	**Check-out a Guest**	Receptionist	Select a reservation and process check-out.	U12
UC15	**Manage user accounts**	IT Administrator	Create, edit, delete user accounts setting password and role	U13

Conceptual Use Case Diagram of the Hotel Management System



6 Functional Requirements

The following table details the functional requirements based on the use cases. There are many ways to prioritise requirements, e.g. MoSCoW method, Kano, Value vs Complexity, Voting systems. For this project it is clear there are many dependencies in the system which will govern the priority of implementation. For instance, a receptionist cannot reserve a room if the room itself cannot be created on the system, and a room cannot be created until the room type is defined. It has been decided, for each requirement, to capture these vital **dependencies**, the **value** to the user, and the implementation **complexity**. Value is ranked between 1 (low) and 5 (high) and complexity is ranked between 1 (easy) and 5 (complex).

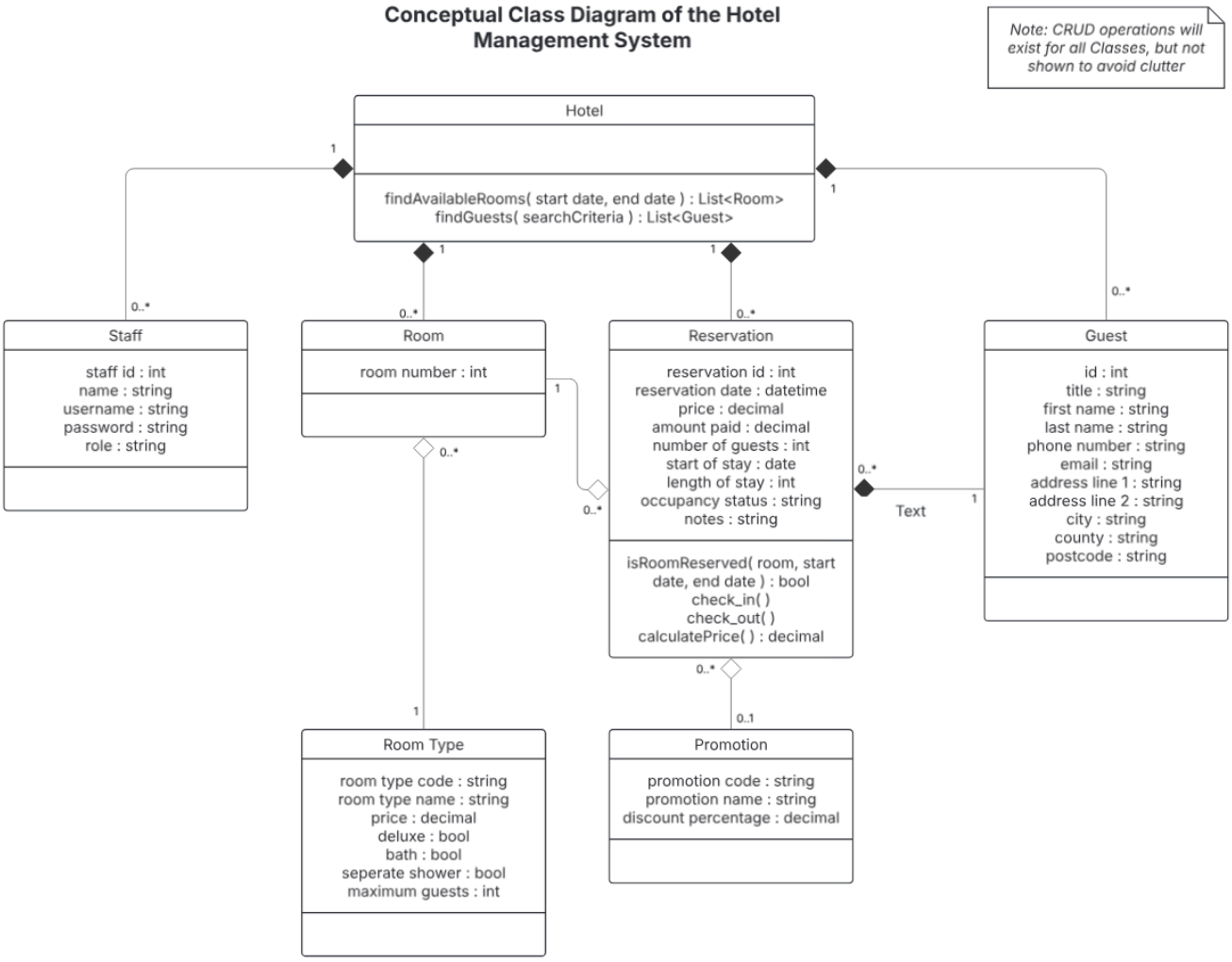
First a bubble sort algorithm will be applied so that the list is ranked from high to low **value**, and when the **value** of two requirements is equal, their rank will be decided in order of low to high **complexity**. Finally, the ranked table will be searched in order, checking that all requirements are listed at a lower priority than a requirement they depend on, if necessary that dependent requirement will be promoted above the current requirement.

The final prioritised requirements list is as follows:

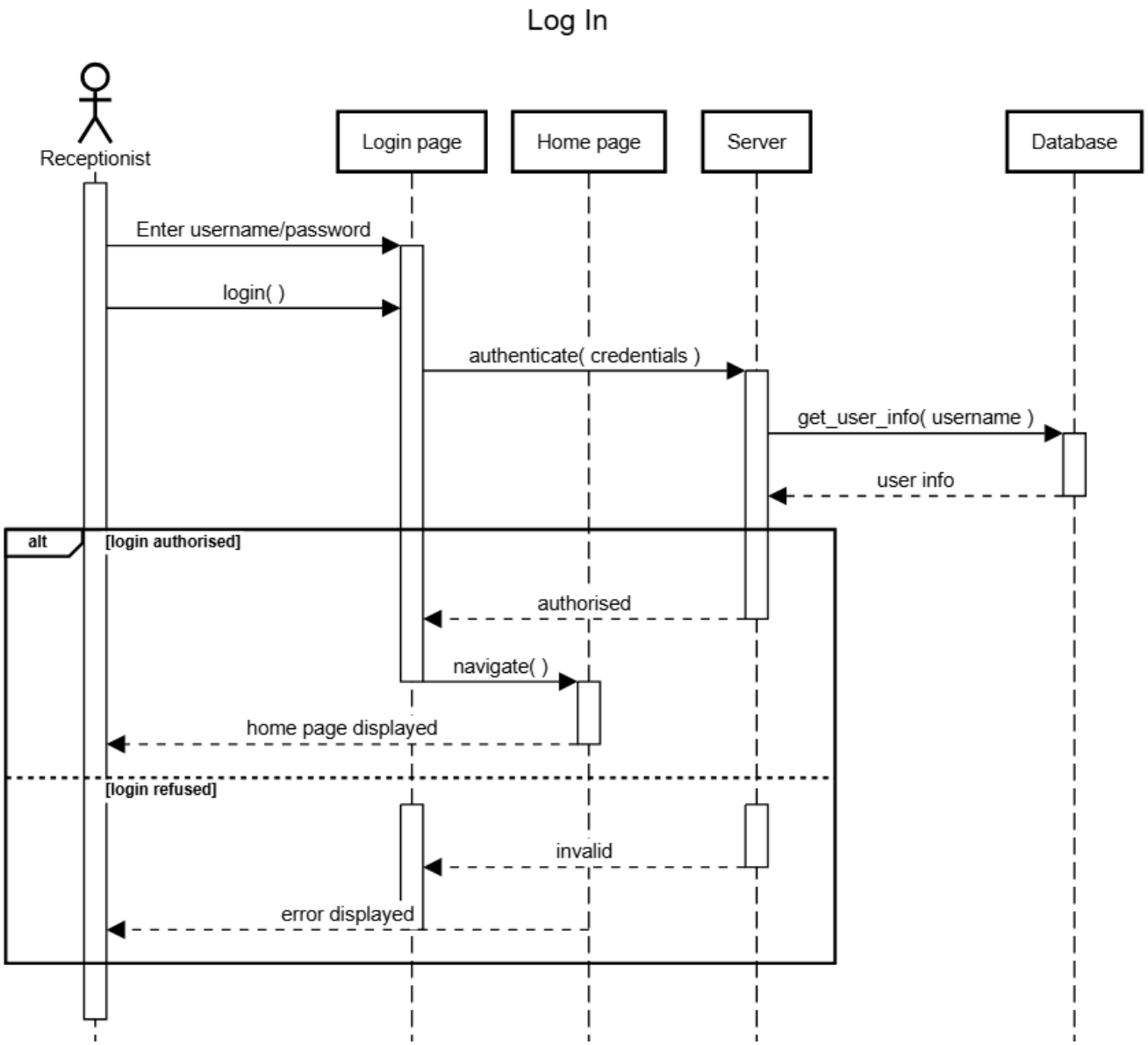
ID	Use case ID	Requirement	Dependency	Value	Complexity
F28	UC15	Provide the ability to create, edit, delete user accounts (setting password and role)		5	2
F1	UC1	Provide a screen for the user to login with a username and password and validate the login	F28	5	2
F2	UC1	Provide a home page offering the choice of features available to the user's role	F1	5	2
F16	UC7	Provide a display of registered guests on the system		5	3
F5	UC2	Provide a room type editor to support addition and editing of room types (validating the entered details)		4	4
F8	UC3	Provide a room editor to support addition and editing of rooms (validating the entered details)	F5	4	2
F13	UC5	Provide the ability to search for available rooms by start date and length of stay and display in a list	F8	5	4
F15	UC6 & UC8	Provide a guest editor so a new guest can be registered on the system or an existing		5	4

		one edited (validating the entered details)			
F22	UC11	Provide the ability to display a list of existing reservations		4	3
F24	UC12	Provide the ability to select a reservation and delete it	F22	4	1
F7	UC3	Provide a display of rooms on the system		4	2
F18	UC9	Provide the ability to delete a guest (with confirmation message)	F16	4	2
F25	UC13	Provide the ability to select a reservation and process a check-in	F22	4	2
F4	UC2	Provide a display of room types on the system		4	3
F19	UC10	Provide the ability to reserve an available room for a guest	F13, F15	4	3
F26	UC14	Provide the ability to select a reservation and process a check-out	F22	4	3
F3	UC1	Provide the ability to logout		3	1
F6	UC2	Provide the ability to delete a room type.	F4	3	1
F9	UC3	Provide the ability to delete a room.	F7	3	1
F17	UC7	Provide the ability to filter the guest list by name or postcode.	F16	3	2
F21	UC10	Provide the ability to record notes on the reservation	F19	3	2
F23	UC11	Provide the ability to filter the reservations list by date	F22	3	2
F27	UC14	Provide the ability during check-out to calculate the amount due to be paid	F26	2	1
F10	UC4	Provide a display of promotional discounts on the system		2	2
F14	UC5	Provide an option to filter available rooms by room type	F13	2	2
F20	UC10	Provide the ability to apply promotional discounts to room reservations	F19	2	2
F11	UC4	Provide a promotion editor to support addition and editing of promotional discounts (validating the entered details)		2	3
F12	UC4	Provide the ability to delete a promotion.	F10	1	1

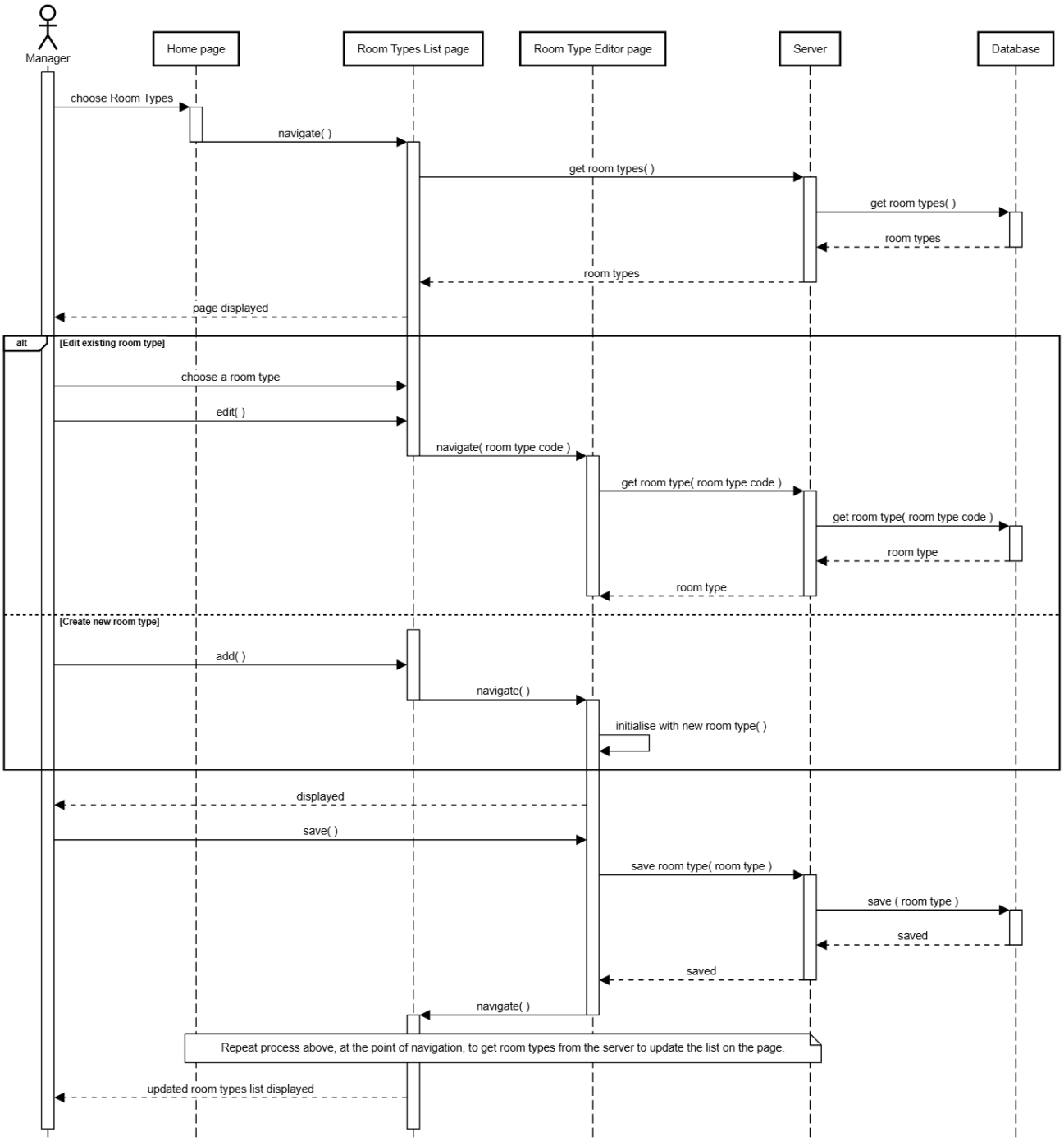
7 Conceptual Class Diagram



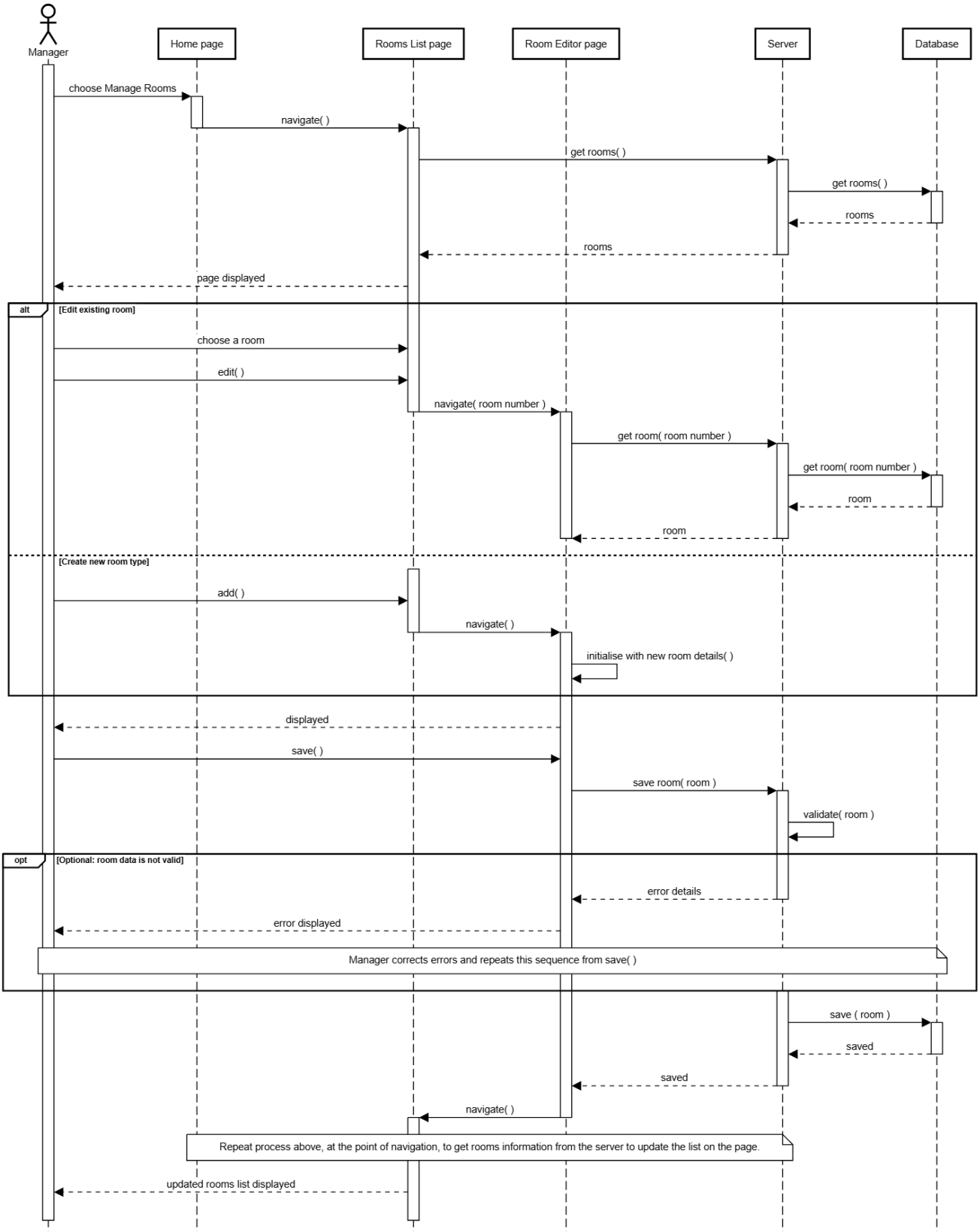
8 Sequence and Activity Diagrams

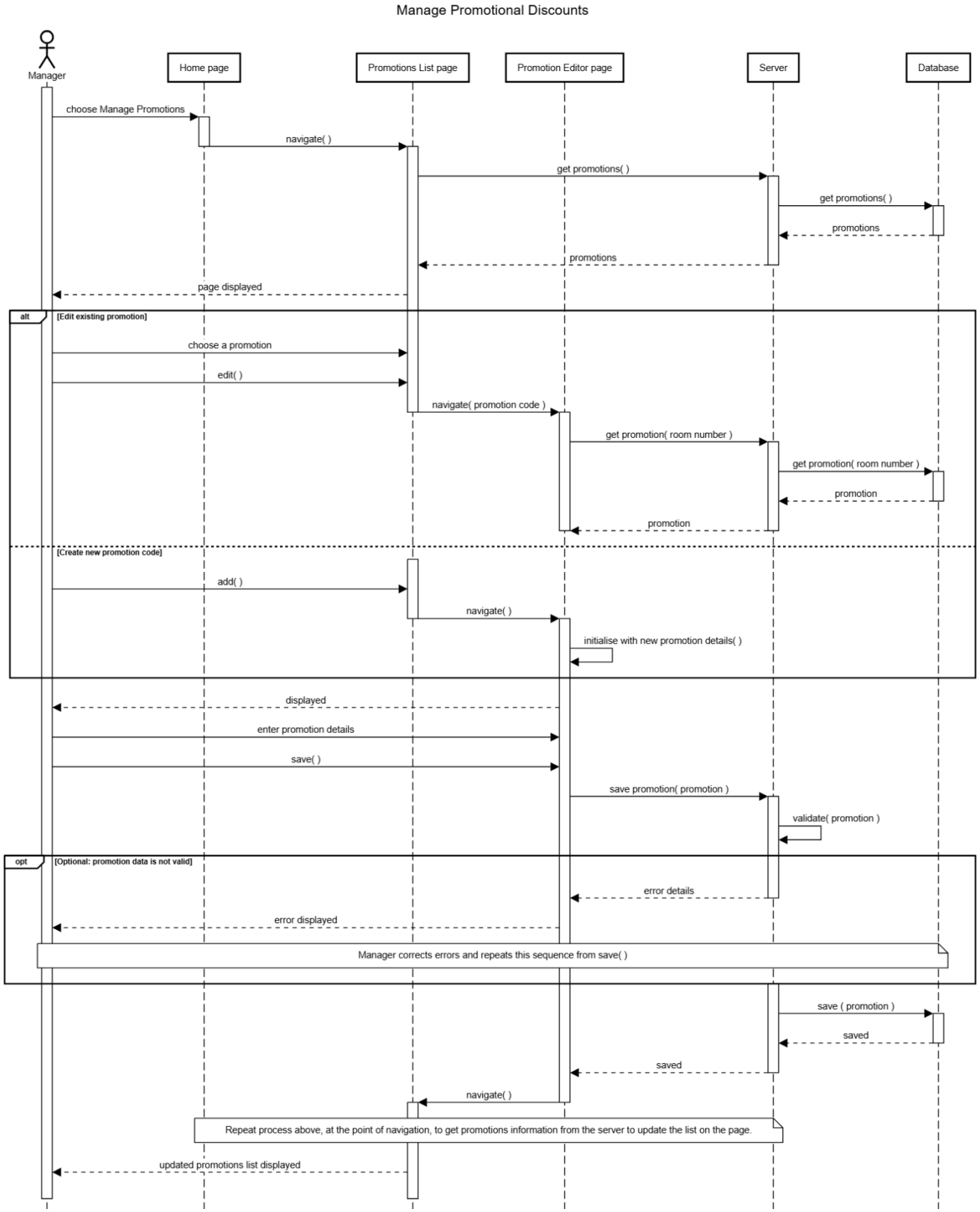


Configure Room Types

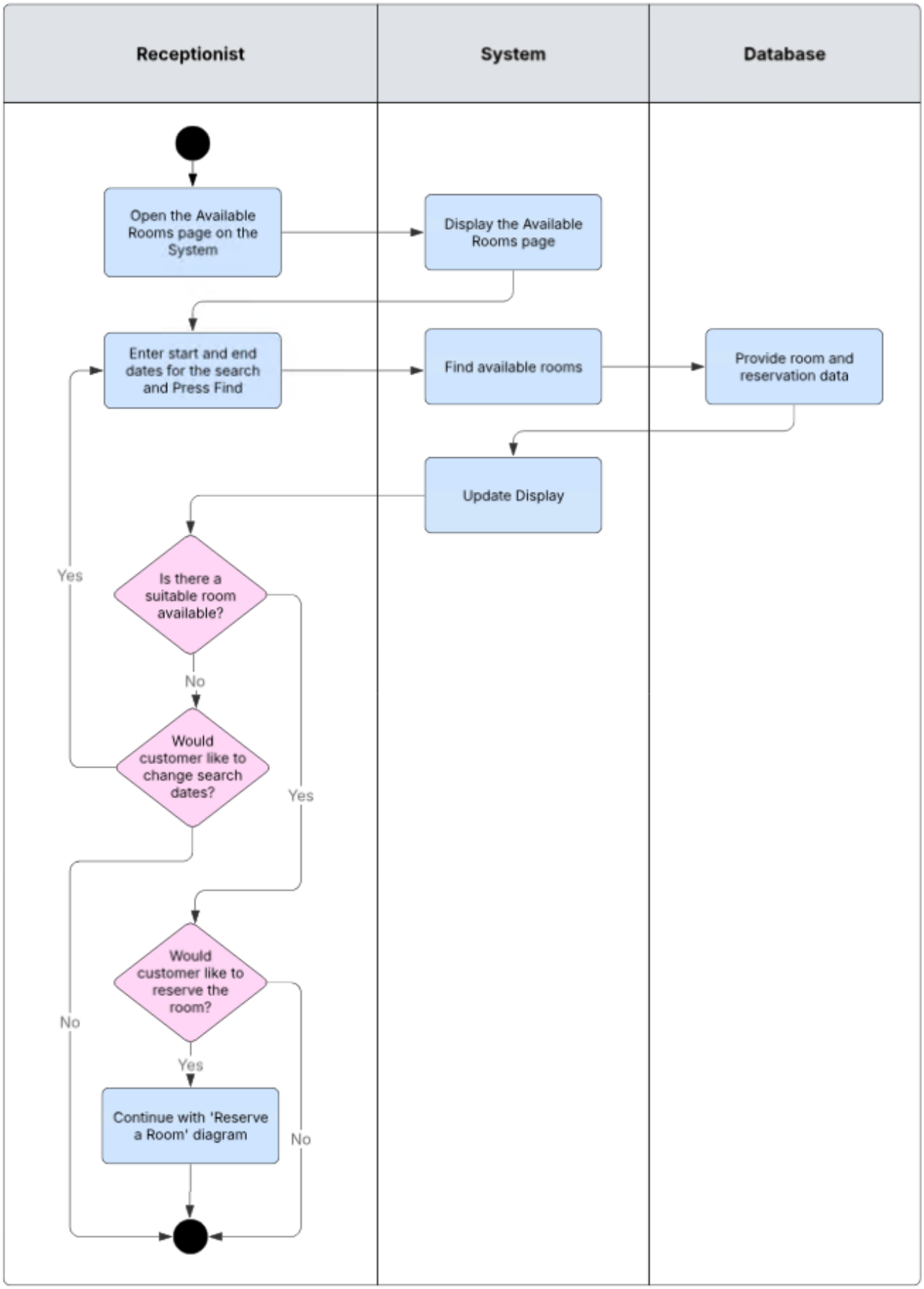


Manage Rooms

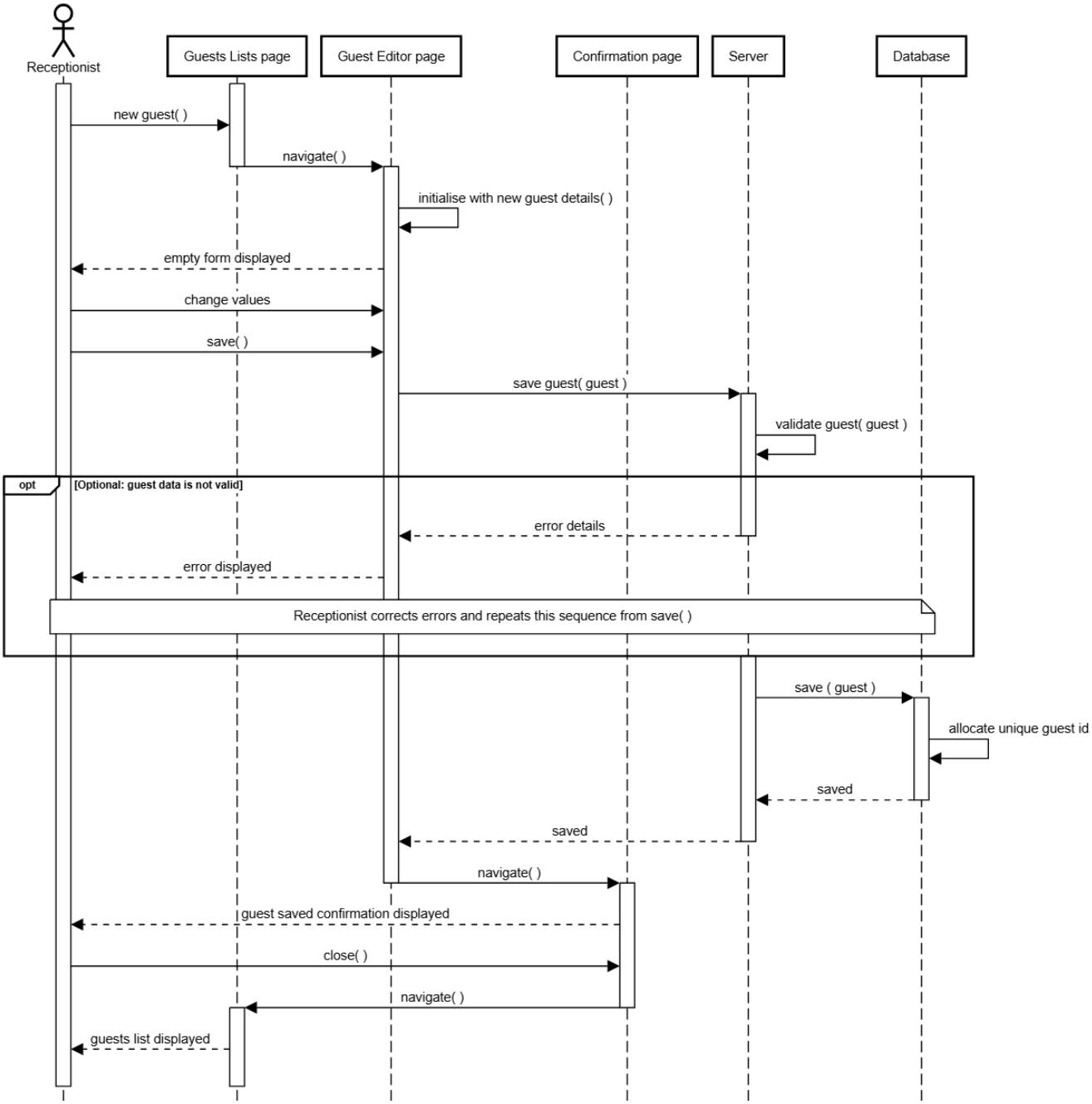



Manage Promotional Discounts
Manager
Home page
Promotions List page
Promotion Editor page
Server
Database
choose Manage Promotions
navigate( )
get promotions( )
get promotions( )
promotions
promotions
page displayed
alt
[Edit existing promotion]
choose a promotion
edit( )
navigate(promotion code)
get promotion(room number)
get promotion(room number)
promotion
promotion
[Create new promotion code]
add( )
navigate( )
initialise with new promotion details( )
displayed
enter promotion details
save( )
save promotion(promotion)
validate(promotion)
opt
[Optional: promotion data is not valid]
error details
error displayed
Manager corrects errors and repeats this sequence from save( )
save (promotion)
saved
saved
navigate( )
Repeat process above, at the point of navigation, to get promotions information from the server to update the list on the page.
updated promotions list displayed

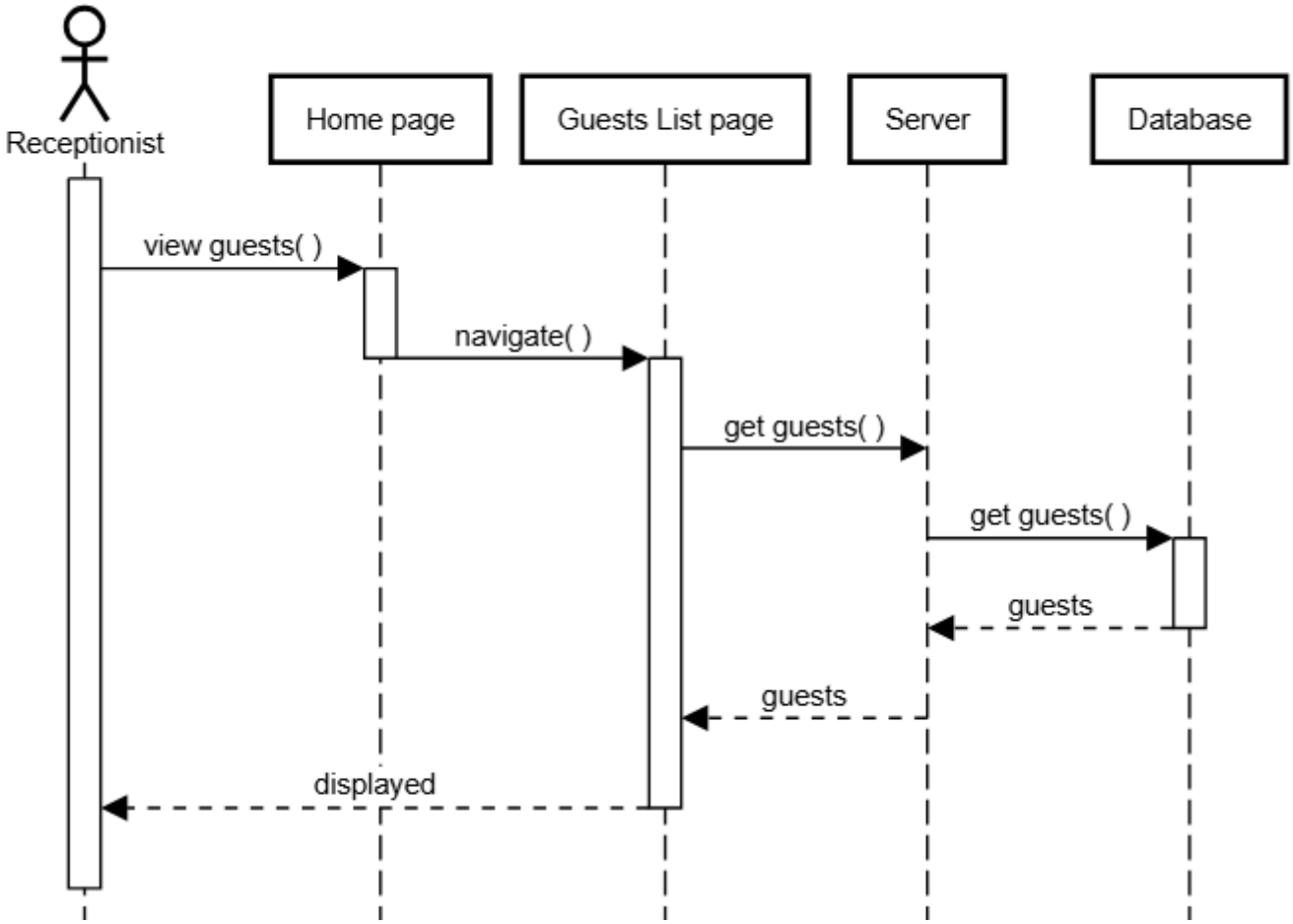
Find Available Rooms



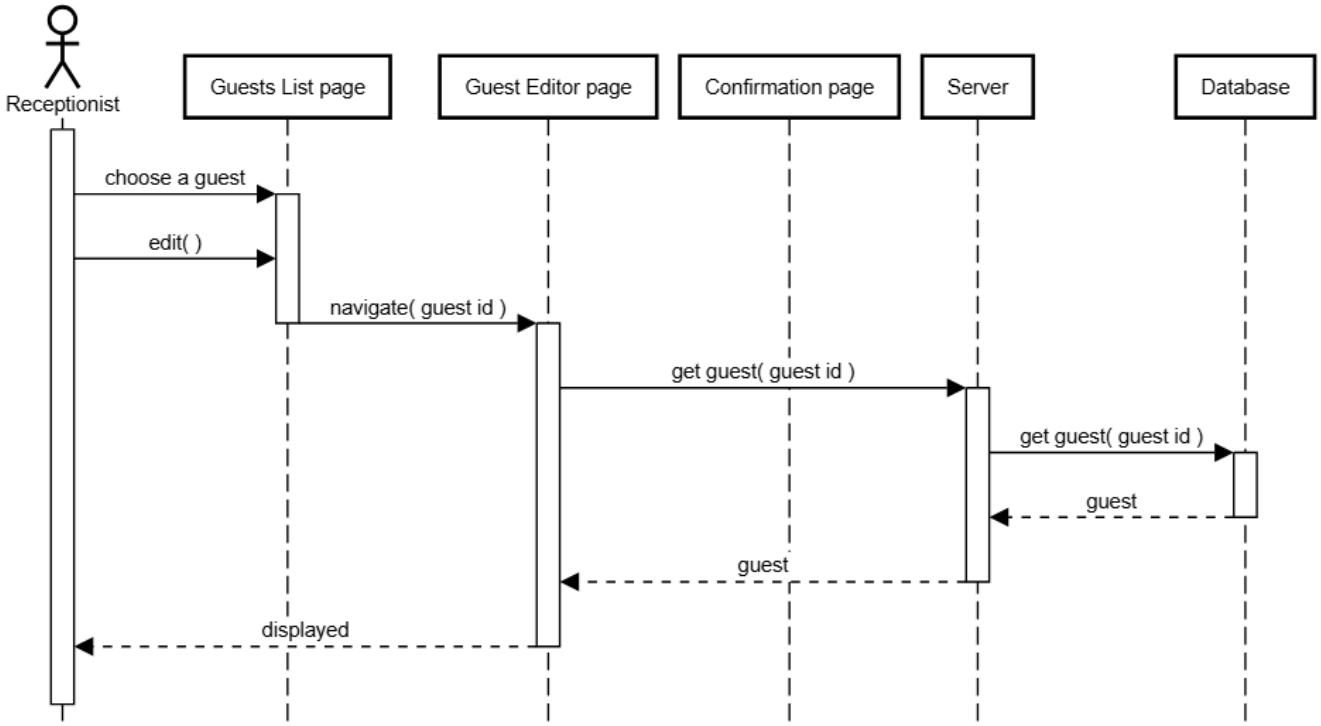
Register a new Guest



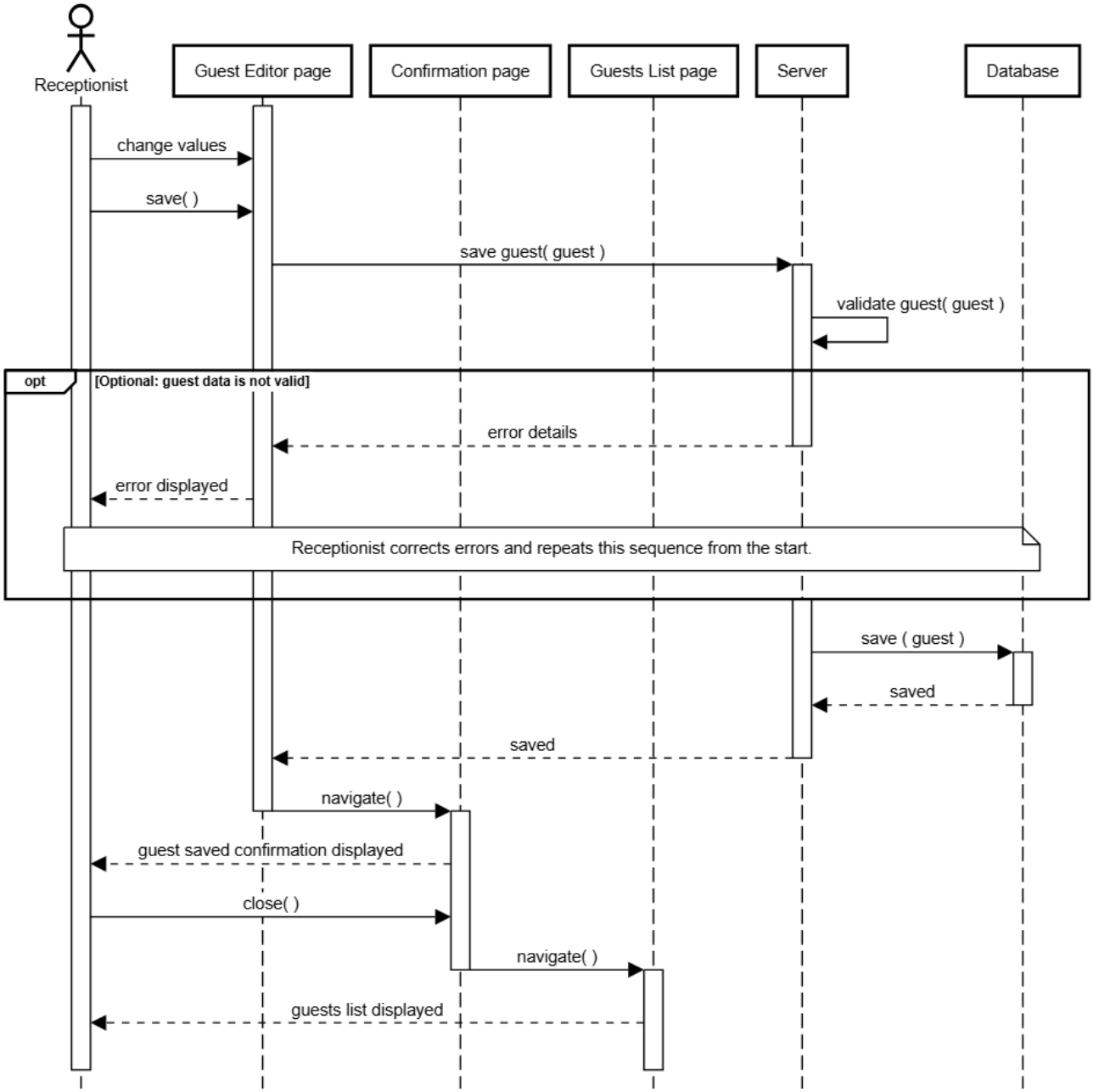
View Guests


Receptionist
Home page
Guests List page
Server
Database
view guests( )
navigate( )
get guests( )
get guests( )
guests
guests
displayed

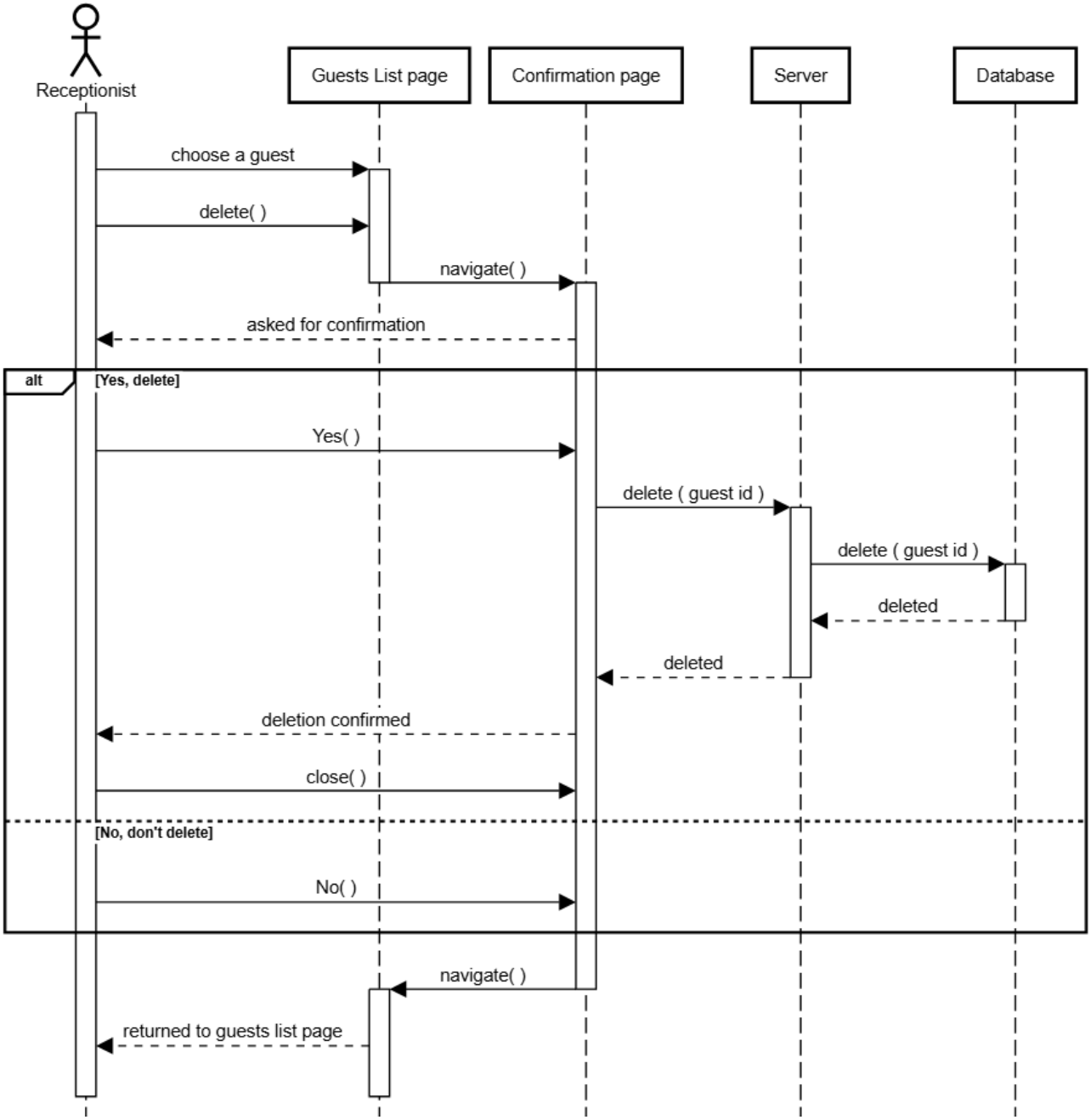
View a Guest (extends Views Guests)



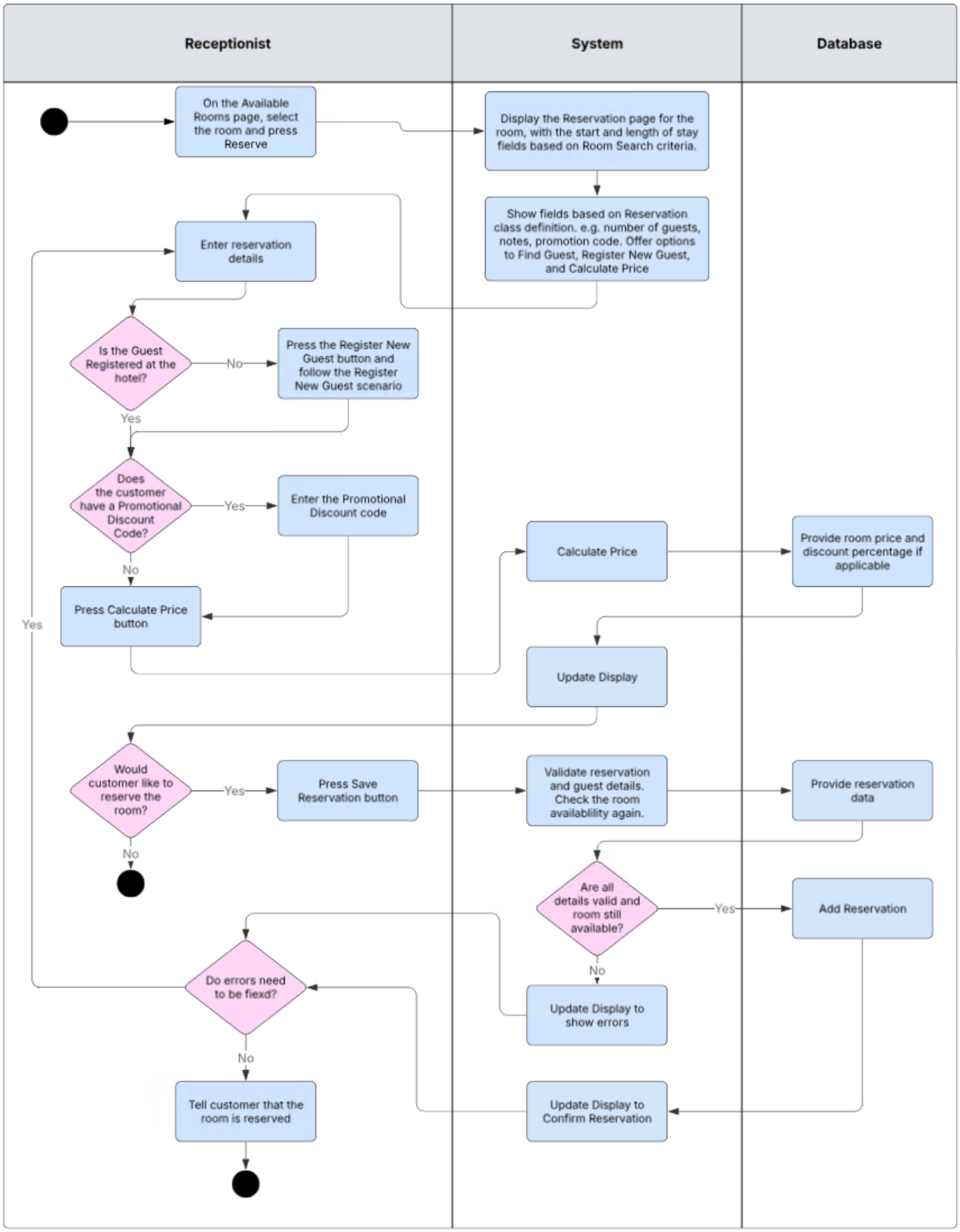
Edit a Guest (extends View a Guest)



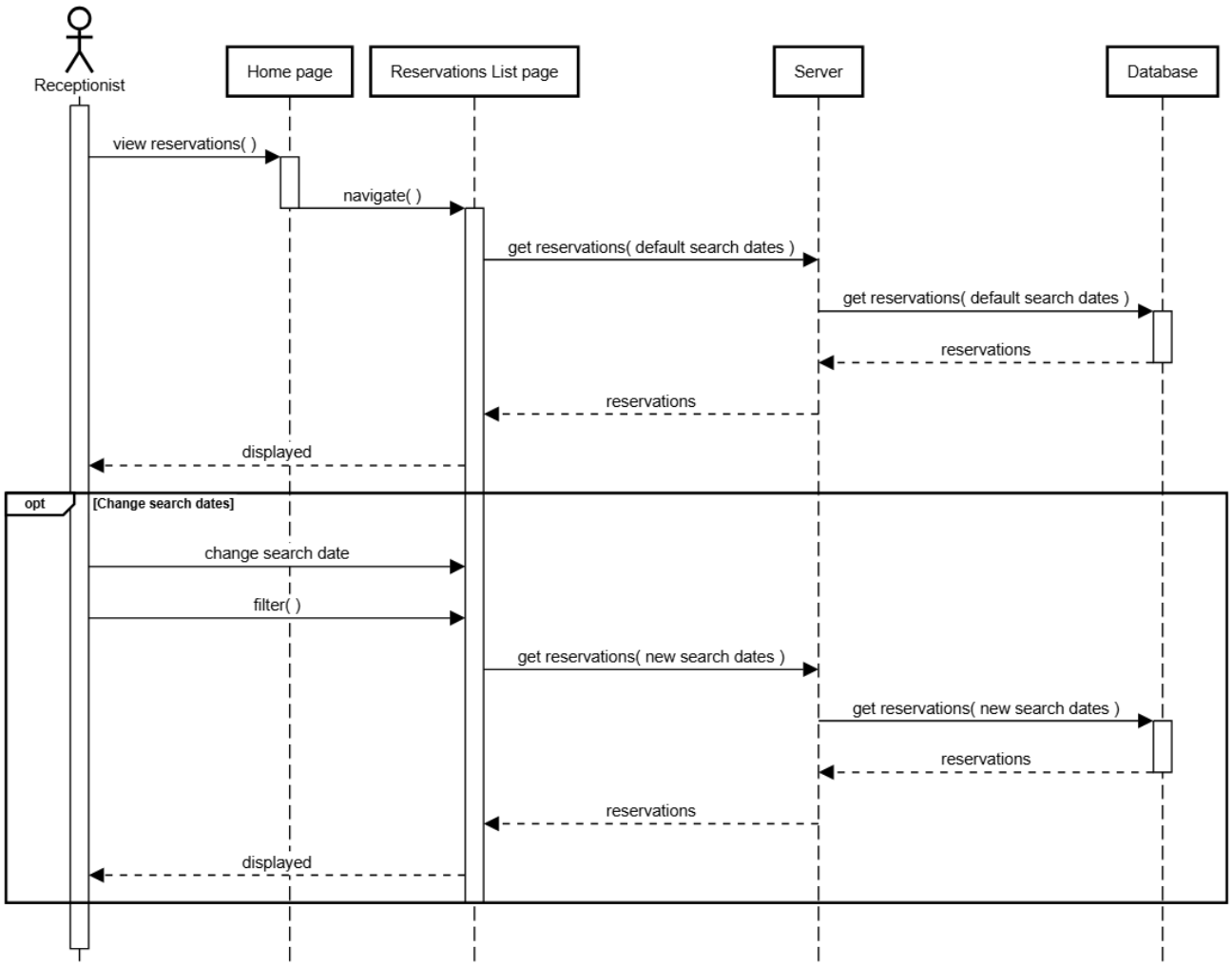
Delete a Guest (extends View Guests)



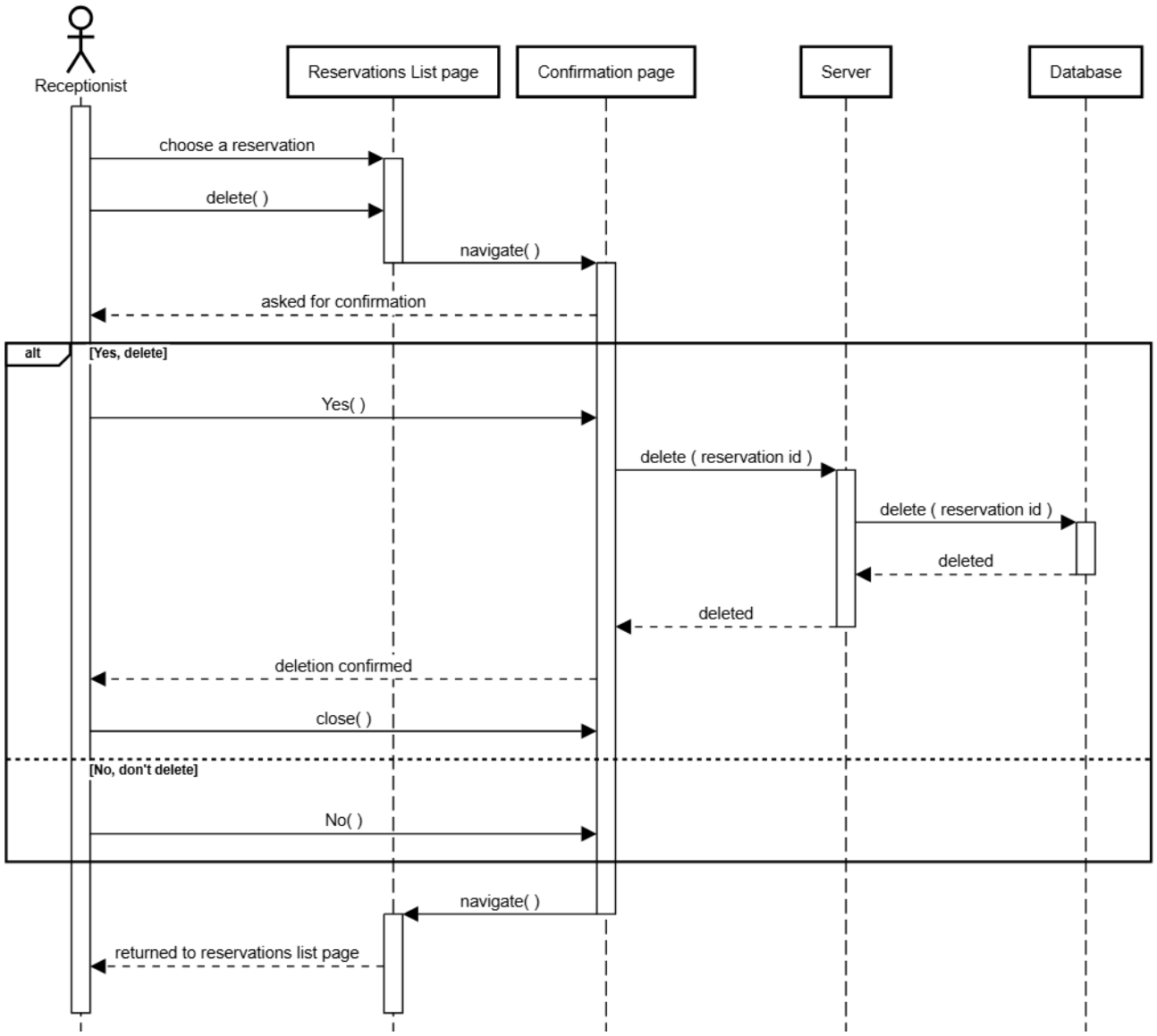
Reserve a Room



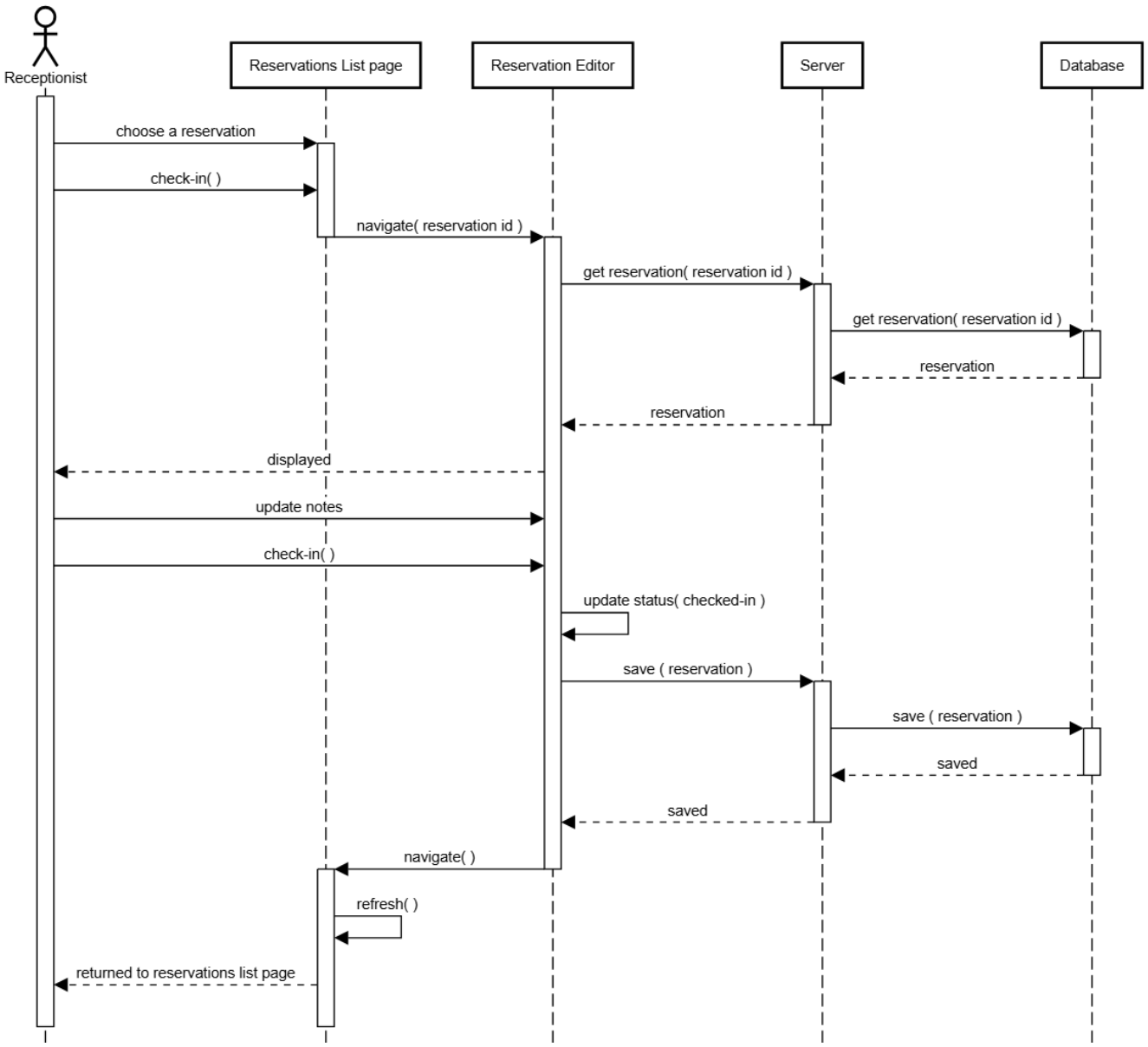
View Reservations



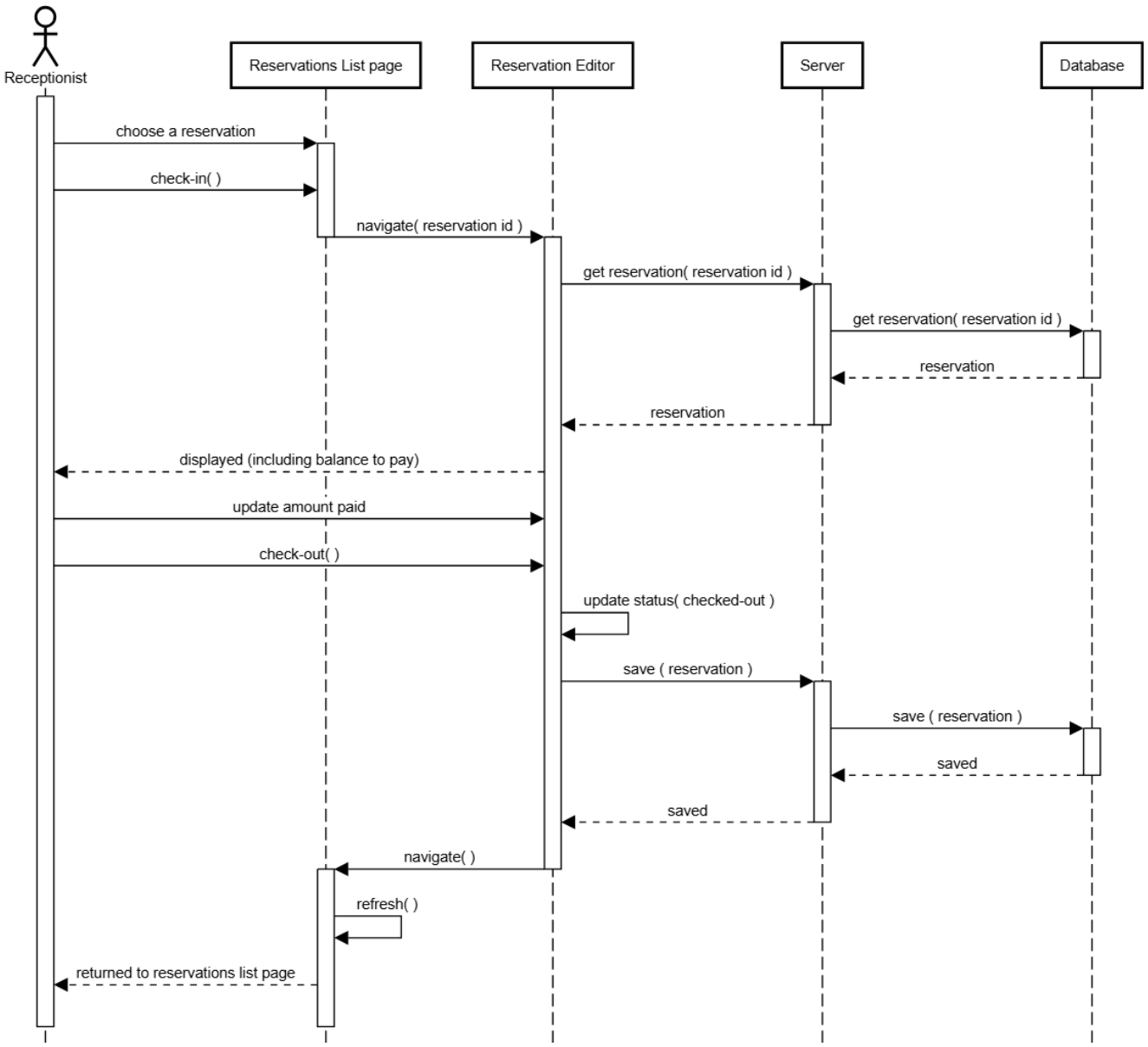
Delete a Reservation (extends View Reservations)



Check-in a Guest (extends View Reservations)



Check-out a Guest (extends View Reservations)



7 References

table format for copy/paste

Figure 1 *– Image*

Table 1 *– table*