Use Cases

Author: Wesley Goyette

Search for Available Rooms:

Primary Actor: Hotel Clerk

Goal: The hotel clerk wants to find available rooms to make a reservation.

Precondition: The hotel clerk is logged in to the system.

Trigger: The hotel clerk selects the “Find available rooms" option.

Scenario: The hotel clerk enters the desired arrival date and departure date.

The system queries the database for available rooms within the specified for dates and returns each available room type (types are broken down into a certain number and type of beds, a smoking/non-smoking status, and a quality level).

The room can be reserved directly from the results page, or the hotel clerk can decide to further filter the results page by number and type of beds, a smoking/non-smoking status, quality level (executive level, business level, comfort level, and economy level) and even room number or floor.

Author: Aaron Sierra

Cancel Room Reservation

Primary Actor:

Guest Scope: Hotel Reservation System

Level: User goal

Brief:

- The guest would like to cancel a hotel reservation system for a prior registration made.

Stakeholders

- Guest who has a prior reservation.

Post Conditions

- Minimal Guarantees: none - Success Guarantees: A guest receives compensation for Room Reserved

Pre-Conditions

- The Cancel Reservation system is presented to the guest

Triggers

- the Guest invokes a cancellation request

Basic Flow:

1. Guest access the hotel reservation system

2. Guest invokes reservation cancellation

3. Guest enters reservation details (e.g. name, reservation number, date)

4. System displays information regarding reservation such as compensation returned and prompts guest to confirm cancellation

5. System cancels the reservation and updates the availability of the room

6. System displays confirmation message: "Reservation successfully cancelled."

Alternate Flow: Incorrect Reservation Details

1. Guest enters incorrect information

2. System displays error message: "Reservation not found."

3. Guest returns to step 2

Alternate Flow 2: Non-Cancellable Reservation

1. Guest attempts to cancel a non-cancellable reservation

2. System displays an error message: "This reservation cannot be cancelled as per policy..."

3. Guest returns to step 1

Assumptions: * Guest has access to the reservation system

Notes: * Cancellation policy may vary depending on the type of reservation and the time of cancellation * the System may impose fees or penalties for late cancellations or no-shows

Author: Chris Chenoweth

Casual: Guest modifies their reservation

Main success scenario: A Customer needs to modify their reservation, so the Customer will login to their account using the credentials used by the Customer to make the original reservation. The Customer will then edit the desired criteria relating to the reservation and submit them to the system for updating. The changes are then accepted given that they are before the cutoff date for the given reservation.

Alternate scenario: A Customer wants to modify their current reservation, so the Customer attempts to login to their account but are denied access. So, the Customer should contact a Staff member to reset the account information. The Staff member should work to reset the account and ensure that the Customer has regained access to the account.

Alternate scenario: A Customer wants to modify their reservation, so the Customer will login to their account using the credentials that the reservation is under. The Customer will then find that the reservation cannot be modified due to it being after the cutoff for their reservation. The

Customer will then contact a Staff member to make the modifications but incur "last minute" fees for being within the cutoff window.

Author: Luke Smith

Enter Credit Credentials (Casual):

Main Success: A customer enters their credit card information, and the system checks to see if the card is a valid number (In our case, we will assume normal card information: name, numbers, expiration, etc.). Once the card is validated through the system the customer will be notified of the successful payment.

Alternate Scenario: If the customer's credit card is not valid, then the system will prompt the user to re-enter their information. The system will prompt three times total before cancelling the transaction.

Author: David Day

Creating New Users

Primary Actor: Admin User

Main Success Scenario: A new clerk has joined the staff and needs access to the system. The admin will be able to create his username and default password which will be able to be changed later.

Author: Shepard Berry

Guest makes a reservation.

Primary Actor: Guest

Main Success: A guest makes a reservation in the hotel system and is sent a confirmation email with room details and check-in information for the room. The guest is given options on the user interface for room options and pricing.

Alternate Scenario: A guest wants to make a reservation, but there are no spots in the hotel available. The hotel will then recommend other hotel options that have spots available for the guest to stay. The guest will also be provided with contact information about other hotels, as well as a way to check for a room in the future.

Author: Shepard Berry

User, clerk, or admin login (casual)

Primary Actor: User, Clerk, Admin

Main Success: A user, clerk, or admin logs into the hotel system and is given access to the functionality assigned to their permission level. Once logged in, all access to an actor's permission level is granted and they are free to execute functionality throughout the hotel according to their permission.

Alternative Scenario: If an actor fails to enter their login information correctly, they will be prompted a total of 3 times, until they are locked out for a set period of time, or they reset their password.

Author: Shepard Berry

User, clerk, admin reset password

Primary Actor: User, Clerk, Admin

Main Success: An actor resets their password. The new data is logged in the database and the actor is now able to login using the same username with their newly set password.

Alternative Scenario: The user enters the same password as their previously set one. This will register an error and will either prompt the user to continue using their old password or produce a unique one.

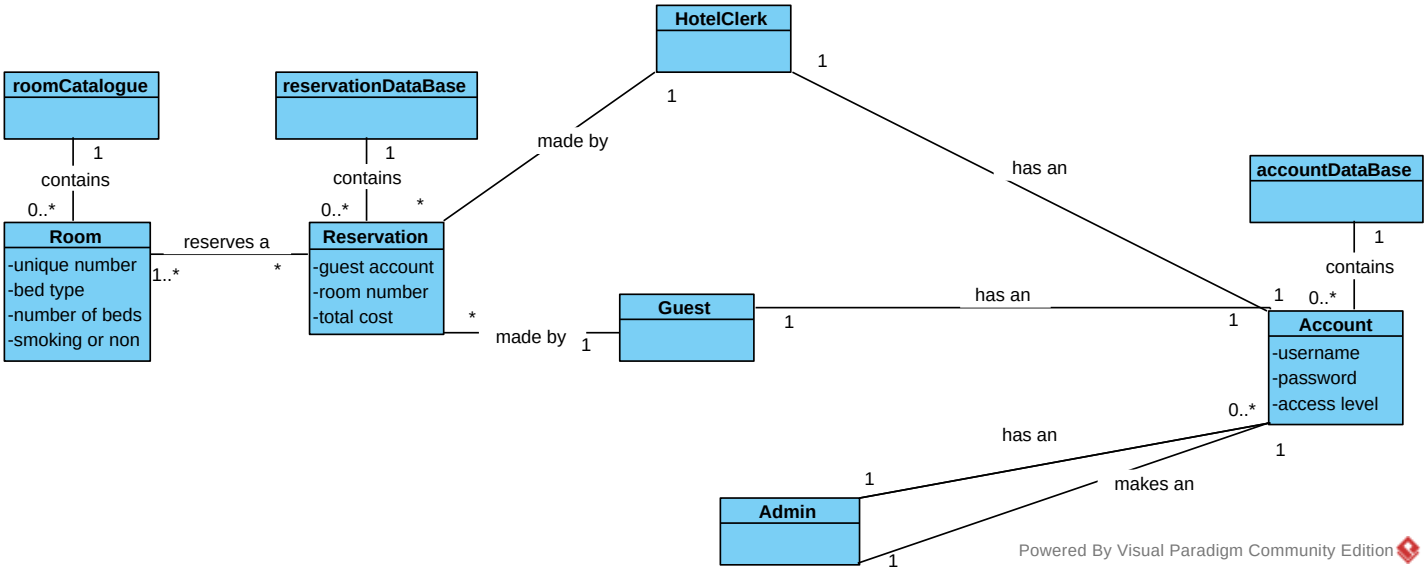
Author: Shepard Berry

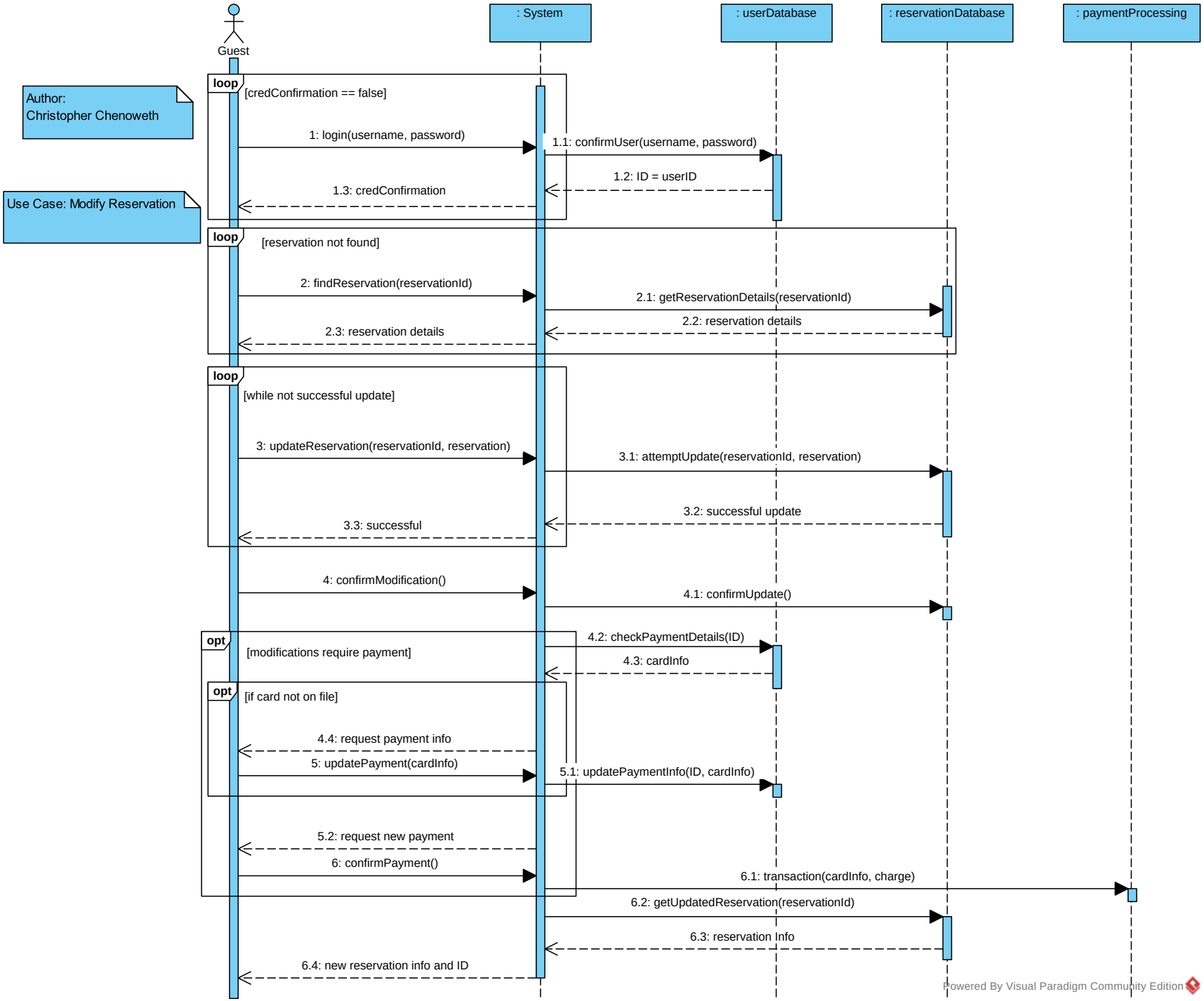
Admin modifies a room

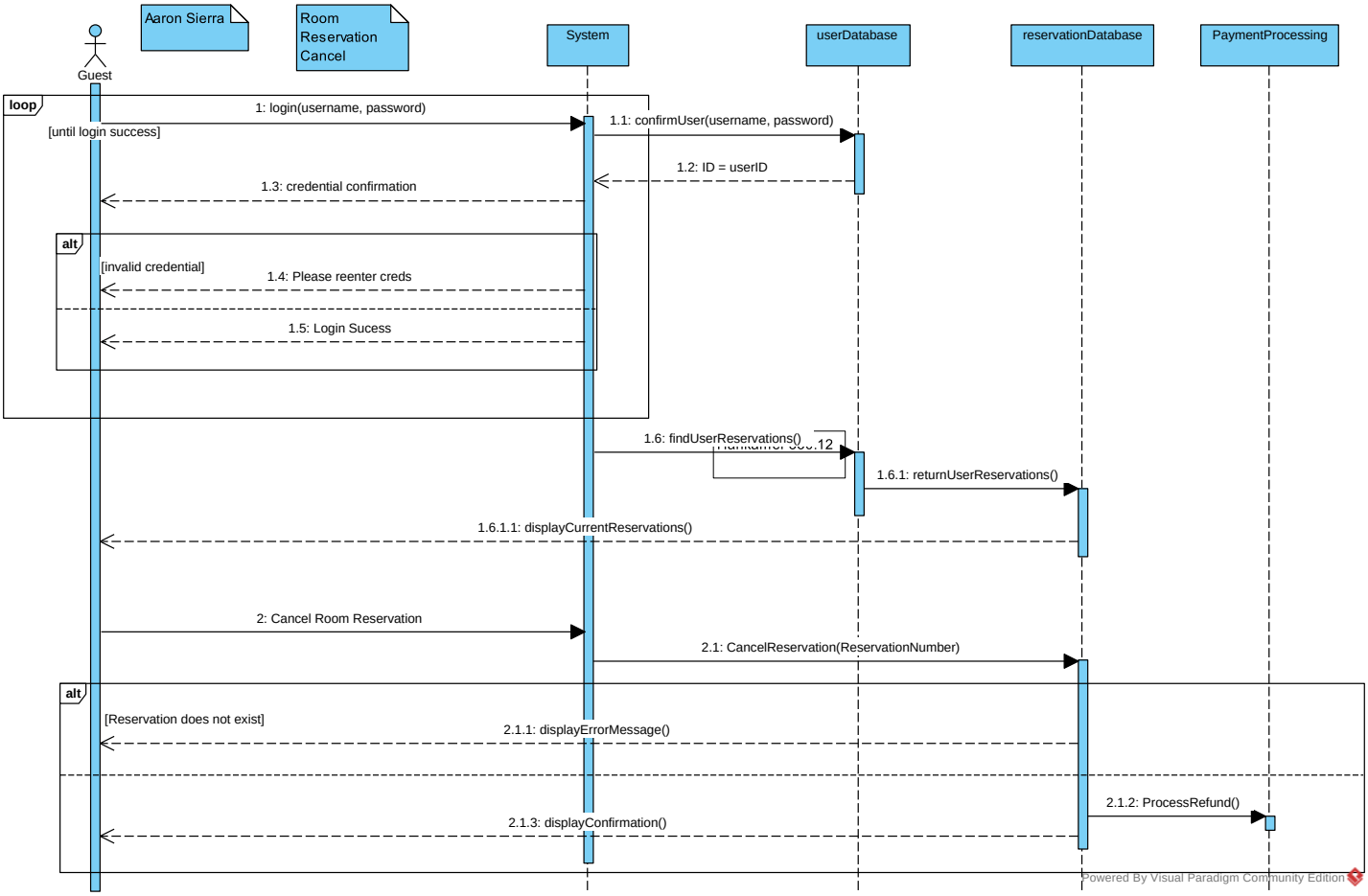
Primary Actor: Admin

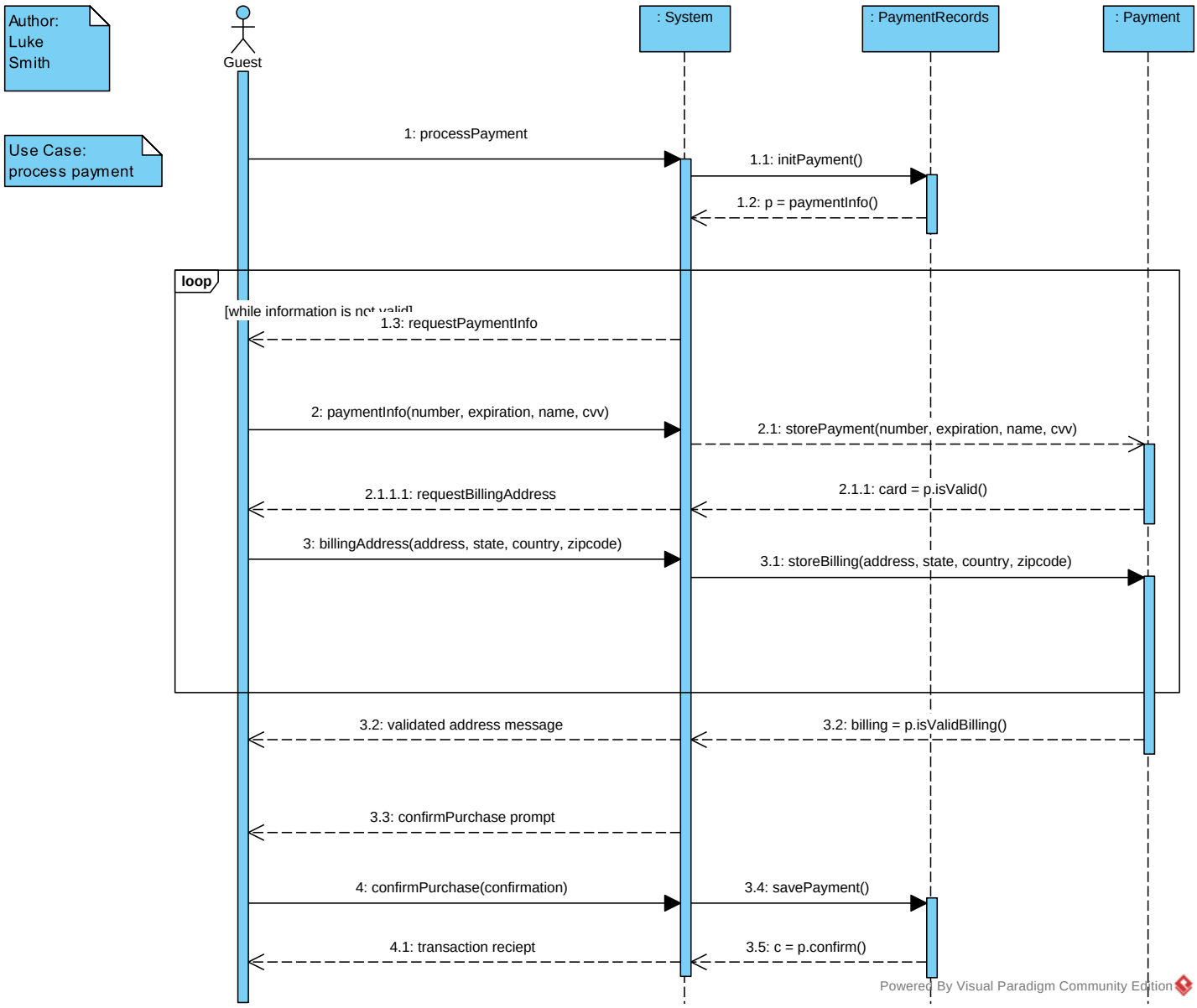
Main Success: An admin changes the contents of a room, for example from a queen-sized room to a suite. The admin has these privileges if the hotel changes in layout, or there are modifications made to its structure.

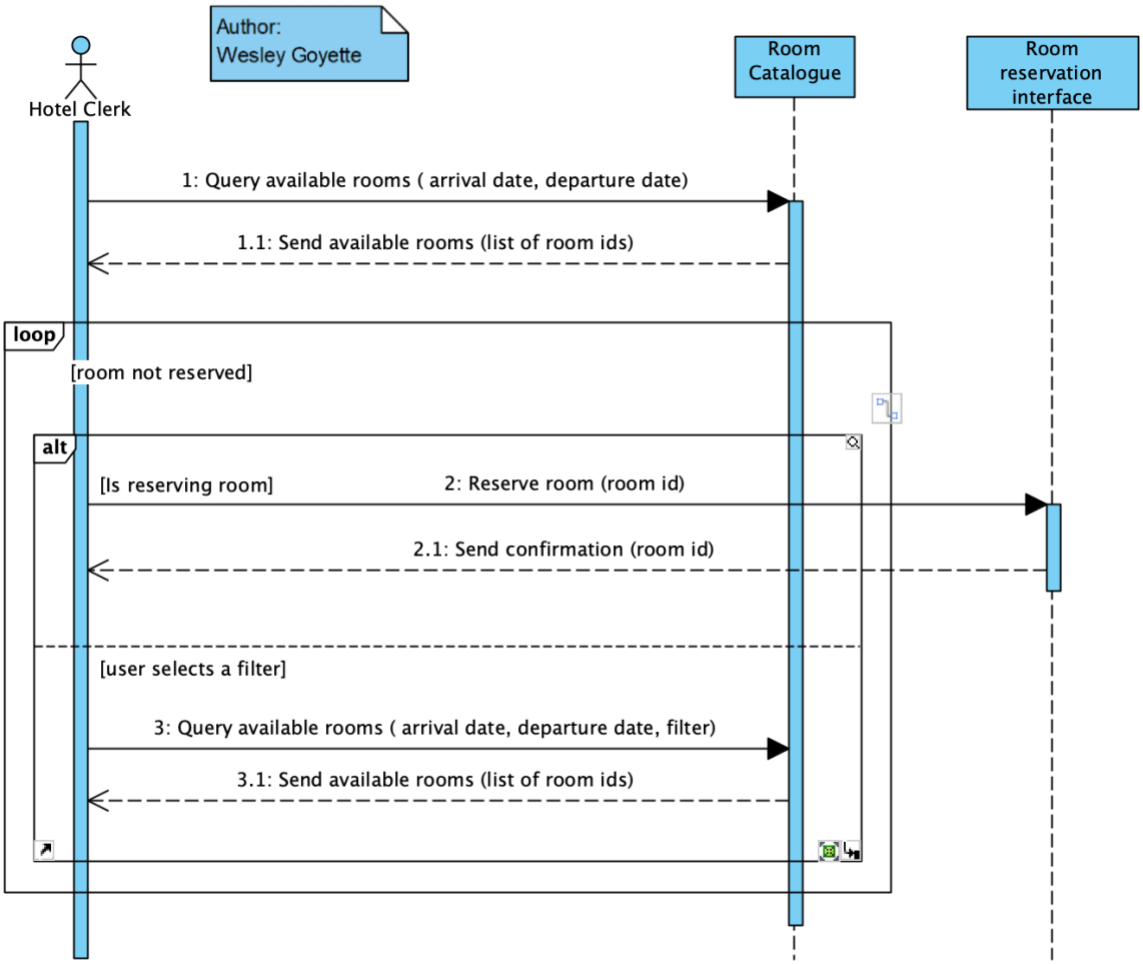
Alternative Scenario: The market price for a room of a certain quality and size has gone up so the price of the room needs to be evaluated in view of the current market.

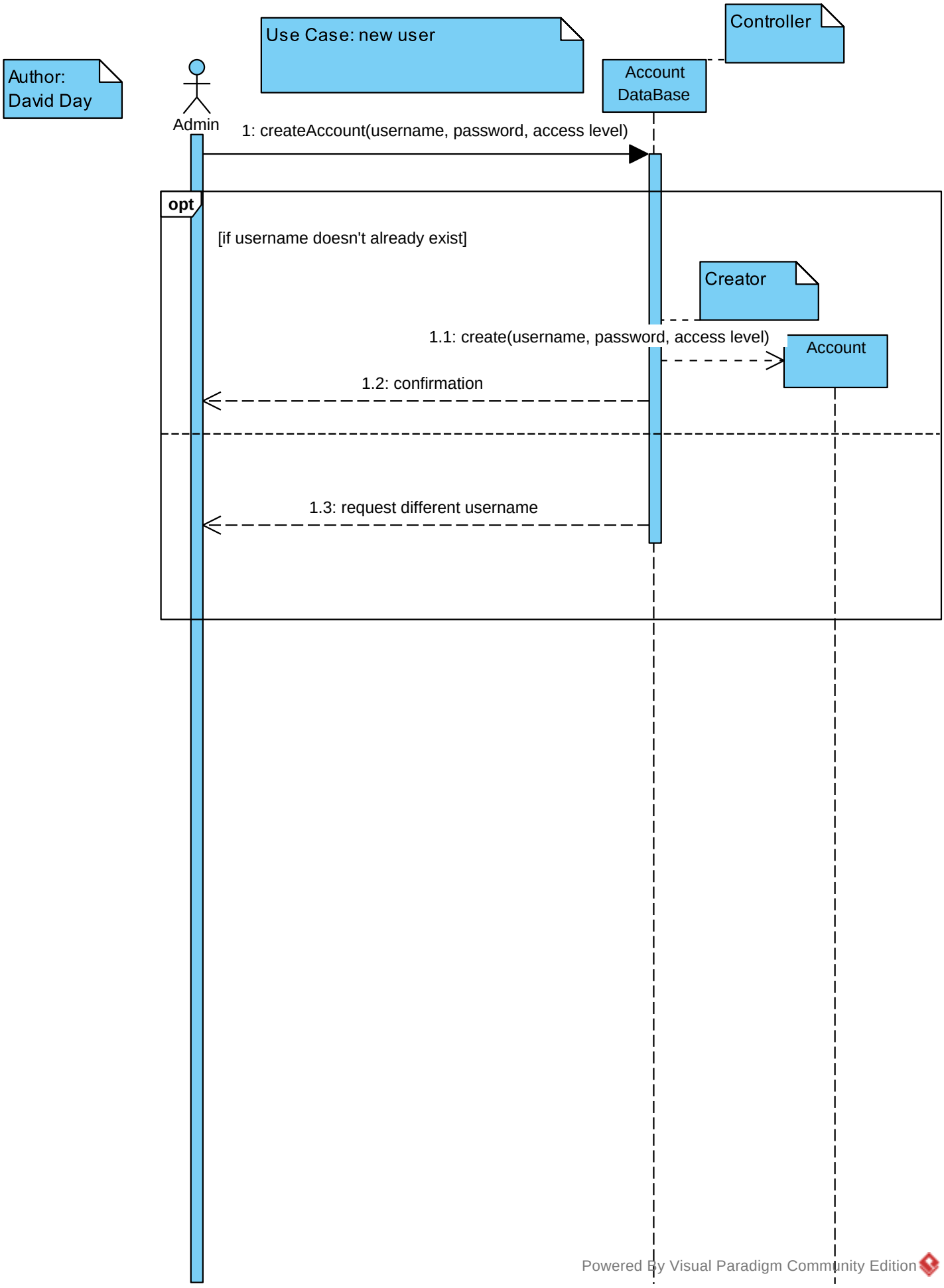

HotelClerk
roomCatalogue
reservationDataBase
accountDataBase
made by
has an
contains
reserves a
Room
-unique number
-bed type
-number of beds
-smoking or non
Reservation
-guest account
-room number
-total cost
Guest
Account
-username
-password
-access level
Admin
makes an
Powered By Visual Paradigm Community Edition


Guest
: System
: userDatabase
: reservationDatabase
: paymentProcessing
Author:
Christopher Chenoweth
Use Case: Modify Reservation
loop
[credConfirmation == false]
1: login(username, password)
1.1: confirmUser(username, password)
1.2: ID = userID
1.3: credConfirmation
loop
[reservation not found]
2: findReservation(reservationId)
2.1: getReservationDetails(reservationId)
2.2: reservation details
2.3: reservation details
loop
[while not successful update]
3: updateReservation(reservationId, reservation)
3.1: attemptUpdate(reservationId, reservation)
3.2: successful update
3.3: successful
4: confirmModification()
4.1: confirmUpdate()
opt
[modifications require payment]
4.2: checkPaymentDetails(ID)
4.3: cardInfo
opt
[if card not on file]
4.4: request payment info
5: updatePayment(cardInfo)
5.1: updatePaymentInfo(ID, cardInfo)
5.2: request new payment
6: confirmPayment()
6.1: transaction(cardInfo, charge)
6.2: getUpdatedReservation(reservationId)
6.3: reservation Info
6.4: new reservation info and ID
Powered By Visual Paradigm Community Edition

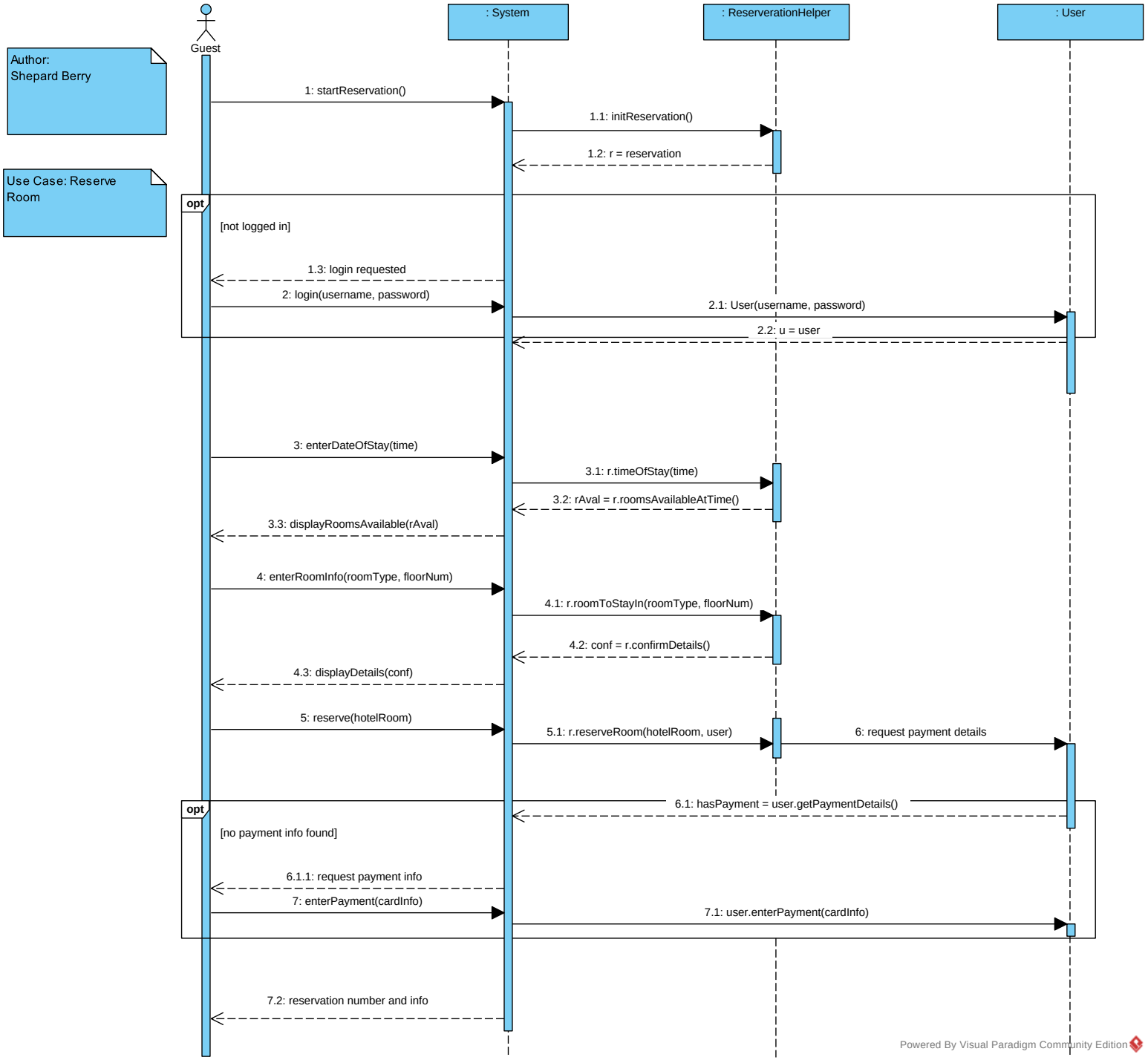

Aaron Sierra
Room Reservation Cancel
Guest
System
userDatabase
reservationDatabase
PaymentProcessing
loop
1: login(username, password)
[until login success]
1.1: confirmUser(username, password)
1.2: ID = userID
1.3: credential confirmation
alt
[invalid credential]
1.4: Please reenter creds
1.5: Login Sucess
1.6: findUserReservations()
1.6.1: returnUserReservations()
1.6.1.1: displayCurrentReservations()
2: Cancel Room Reservation
2.1: CancelReservation(ReservationNumber)
alt
[Reservation does not exist]
2.1.1: displayErrorMessage()
2.1.2: ProcessRefund()
2.1.3: displayConfirmation()
Powered By Visual Paradigm Community Edition


Author:
Luke
Smith
Use Case:
process payment
Guest
: System
: PaymentRecords
: Payment
1: processPayment
1.1: initPayment()
1.2: p = paymentInfo()
loop
[while information is not valid]
1.3: requestPaymentInfo
2: paymentInfo(number, expiration, name, cvv)
2.1: storePayment(number, expiration, name, cvv)
2.1.1.1: requestBillingAddress
2.1.1: card = p.isValid()
3: billingAddress(address, state, country, zipcode)
3.1: storeBilling(address, state, country, zipcode)
3.2: validated address message
3.2: billing = p.isValidBilling()
3.3: confirmPurchase prompt
4: confirmPurchase(confirmation)
3.4: savePayment()
4.1: transaction reciept
3.5: c = p.confirm()
Powered By Visual Paradigm Community Edition


Author:
Wesley Goyette
Hotel Clerk
Room Catalogue
Room reservation interface
1: Query available rooms (arrival date, departure date)
1.1: Send available rooms (list of room ids)
loop
[room not reserved]
alt
[Is reserving room]
2: Reserve room (room id)
2.1: Send confirmation (room id)
[user selects a filter]
3: Query available rooms (arrival date, departure date, filter)
3.1: Send available rooms (list of room ids)



Powered By Visual Paradigm Community Edition


Author:
Shepard Berry
Use Case: Reserve Room
Guest
: System
: ReserverationHelper
: User
1: startReservation()
1.1: initReservation()
1.2: r = reservation
opt
[not logged in]
1.3: login requested
2: login(username, password)
2.1: User(username, password)
2.2: u = user
3: enterDateOfStay(time)
3.1: r.timeOfStay(time)
3.2: rAval = r.roomsAvailableAtTime()
3.3: displayRoomsAvailable(rAval)
4: enterRoomInfo(roomType, floorNum)
4.1: r.roomToStayIn(roomType, floorNum)
4.2: conf = r.confirmDetails()
4.3: displayDetails(conf)
5: reserve(hotelRoom)
5.1: r.reserveRoom(hotelRoom, user)
6: request payment details
6.1: hasPayment = user.getPaymentDetails()
opt
[no payment info found]
6.1.1: request payment info
7: enterPayment(cardInfo)
7.1: user.enterPayment(cardInfo)
7.2: reservation number and info
Powered By Visual Paradigm Community Edition