Baylor Bear Luxury

Hotel-Storefront Management System
Vision (Small Project)

Version <1.0>

Revision History

Date	Version	Description	Author
31/1/23	1.0	An initial iteration of a business vision document	Will Clore, Nicholas Nolen, Derek Martinez, Joseph Zuniga, Alan Villagrand, Harrison Hassler

Table of Contents

1. Introduction 4
   - 1.1 References 4
2. Positioning 4
   - 2.1 Problem Statement 4
   - 2.2 Product Position Statement 4
3. Stakeholder and User Descriptions 5
   - 3.1 Stakeholder Summary 5
   - 3.2 User Summary 5
   - 3.3 User Environment 6
   - 3.4 Summary of Key Stakeholder or User Needs 6
   - 3.5 Alternatives and Competition 6
4. Product Overview 6
   - 4.1 Product Perspective 6
   - 4.2 Assumptions and Dependencies 6
5. Product Features 6
6. Other Product Requirements 7

Vision (Small Project)

1. Introduction

The purpose of this document is to highlight the needs and features of the Baylor Bear Luxury. It focuses on the capabilities needed by the stakeholder and the target user. The details of how the Baylor Bear Luxury Hotel is implemented are crucial in defining well-written use cases and specifications of the project highlighted in this document.

1.1 References

1. Vision Document, UT Dallas, 30/11/10

2. Group Project Overview, Professor Quevedo Caballero, 16/01/24

2. Positioning

2.1 Problem Statement

The problem of	having a manual hotel reservation and storefront system
affects	hotel management
the impact of which is	it limits business efficiency and provides a frustrating experience for customers
a successful solution would be	to automate the processes with online reservations and an online storefront.

2.2 Product Position Statement

For	Hotel businesses
Who	have a poor or non-existent room reservation system
The Hotel-Storefront Management system	is a software application
That	provides the ability to efficiently check in guests to their assigned rooms and check out when needed
Unlike	currently available systems that have complicated user interfaces with slow and limited functionality
Our product	provides a smooth experience with a simple user interface that is easy to learn while also having plenty of functionality to make sure that guests are properly assigned to their rooms and are ready to be checked in/out

3. Stakeholder and User Descriptions

3.1 Stakeholder Summary

Name	Description	Responsibilities
Requirement Engineers	Receives requirements for the project and communicates these requirements to the other team members.	Communicates the requirements that the team needs to build the hotel software Translates the client's needs into plausible software requirements for the engineers
Project Librarian	keeps all meeting logs and makes all design artifacts available for the team.	ensures that engineers are well informed so they can make an error-free hotel management system monitors project progress through record keeping
Quality Assurance Engineer	Develops test cases and tests software features.	maintains hotel and store software stability through testing develops test cases for features added onto the system
Design Engineer	designs the software features.	develops the actual code for the hotel and storefront management system designs the general design for the software system
Project Manager	Communicates with the team and coordinates meetings as well as team roles.	Organizes the team meetings Communicates ideas from the client to the rest of the team Maintains a safe, conflict-free working environment

3.2 User Summary

Name	Description	Responsibilities	Stakeholder
Hotel Guest	Primary End use of the system	Uses application to reserve or cancel rooms and purchase items through the hotel and shop.	Customer
Hotel Clerks	Primary End user	Uses application to modify reservations and send billing information	Employees
Admin User	Primary End user	Uses application to create staff accounts	Owner

3.3 User Environment

A single user should be able to book a reservation for themselves and their companions (families, company peers, etc.) through their user account. The user or a companion of said user, should be able to use their user account details (username and password) to log into this account from different access points such as a computer terminal or a cellular device. Once logged in the current user needs to be able to update their booking information from said access point. This product will need to be compatible with the iOS, Android, and Windows operating systems and the devices that work off of those systems.

3.4 Summary of Key Stakeholder or User Needs

Manual room reservation

- The problems with a manual room reservation system is unreliable as it creates room for human error, causes a slow check in time for guests, and requires more labor.
- The stakeholder is currently using a manual reservation system to deal with this issue
- The stakeholder wants an automated reservation system to solve this issue

Need	Priority	Concerns	Current Solution	Proposed Solutions
Automated room reservation system	An efficient reservation system	A slow, tedious, inefficient reservation system	Manual reservation system	Automated reservation system

3.5 Alternatives and Competition

- Stakeholder buying a competitor's product
- Using a paper manual system

4. Product Overview

4.1 Product Perspective

The product is independent as far as the software goes. The user and the staff both use the system independently of each other. The actual product is entirely built on one platform so it doesn’t require any outside design ideas.

4.2 Assumptions and Dependencies

It is assumed the user is literate and can type

It is assumed the software is downloaded on the correct hardware

It is assumed that the user speaks U.S english

5. Product Features

Login/Logout

Creating Staff Accounts

Shopping

Reservations

Billing

6. Other Product Requirements

The product must be able to handle setting, canceling and modifying reservations. The product will require a database to store the appropriate information a customer would need for their reservation (i.e. billing, calendar date, login information). The product must be able to set a reservation in a reasonable time frame and provide feedback to the customer on the status of a reservation.

Will Clore Use Cases:

ID UC User Login
Scope Hotel Reservation System
Level User Goal
Stakeholders and interests
Customer
- person that is interested in hotel room and hotel services
Staff
- person that deals with customer directly and uses system to assist the customer
Manager
- person that deals with customer complaints and problems
System Maintainer
- person that maintains the system
Precondition: System is properly running

Postcondition: the customer now has access to the system

Main Success Scenario:

1. Customer wants to log into the system
2. Customer will boot the system
3. Customer will provide the system with a username and password
4. The system will identify the user's information and complete the login of the customer
5. The customer will have full access to the system's features

Alternate paths:

I. Anytime system doesn't respond
   A. Customer will restart the system

II. If the Customer is unable to login and has an account
   A. Customer will notify the hotel manager
   B. Any requests by the Customer will be noted by the manager

III. If the Customer doesn't have a login
   A. The Customer will sign up with a new account

IV. If the Customer has invalid login credentials
   A. The customer will try to login again
   B. The customer will request a password reset by the manager

ID UC Apply Discount
Scope Hotel Reservation System
Level User Goal
Stakeholders and interests
Customer
- person that is interested in hotel room and hotel services
Staff
- person that deals with customer directly and uses system to assist the customer
Manager
- person that deals with customer complaints and problems
System Maintainer
- person that maintains the system
Precondition: The customer has a discount to use

Postcondition: the customer's bill is discounted

Main Success Scenario:

1. Customer wants to book a room using a discount
2. The discount code will be asked by the clerk
3. Clerk will login into the system
4. Clerk will find and reserve the guest's desired room, if it is available
5. The discount will be applied to the guest's account by the clerk, via a discount code
6. The system will generate a bill upon check out
7. the clerk will give the bill to the customer with the applied discount
8. The customer will pay the bill

Alternate paths:

I. Anytime system doesn't respond
   A. Clerk will restart the system

II. If the Clerk is unable to login
   A. Clerk will notify the hotel manager
   B. Any requests by the customer will be noted by the clerk including reservation information

III. If the customer gives invalid discount code
   A. The Clerk will inform the customer that the guest's stay will be at full price

IV. If the Customer's discount information is valid but doesn't work in the system
   A. The clerk will try to enter it again
   B. The clerk will notify the manager

C. The clerk will make sure the guest gets his reservation and his discount applied

V. If the customer disputes or has complaints about the bill
   A. The clerk will inform the manager of these complaints
   B. If the complaints are valid, the manager will resolve the issue with the bill and update the bill
   C. The customer will pay the bill

ID UC Check Room Availability
Scope Hotel Reservation System
Level User Goal
Stakeholders and interests
Customer
- person that is interested in hotel room and hotel services
Staff
- person that deals with customer directly and uses system to assist the customer
Manager
- person that deals with customer complaints and problems
System Maintainer
- person that maintains the system
Precondition: The customer is looking for rooms

Postcondition: The customer finds available rooms

Main Success Scenario:
1. The Customer wants to find rooms that are available
2. The Customer will login in to the system with a username and password
3. The customer will navigate to the part of the system Where room availability is specified
4. The customer will find their desired room type and check if it is available to reserve for the desired dates
5. If the guest wants to reserve the room, they can but if not they can continue looking at other available rooms

Alternate Paths:

I. Anytime system does not respond
   A. The customer will restart their system and try again

II. If the Customer is unable to login and has an account
   A. Customer will notify the hotel manager
   B. Any requests by the Customer will be noted by

the manager

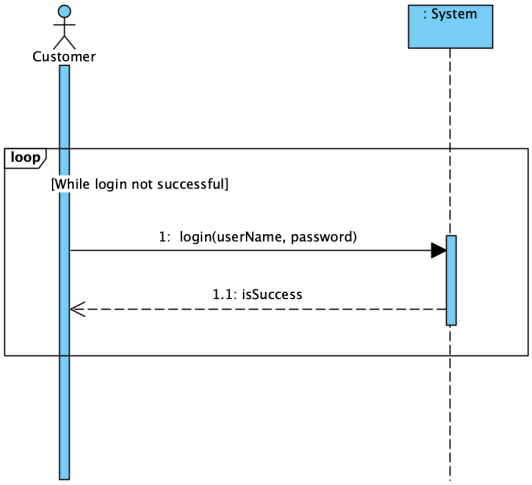
III. If the Customer doesn't have a login
   A. The Customer will sign up with a new account

IV. If the Customer has invalid login credentials
   A. The customer will try to login again
   B. The customer will request a password reset by the manager

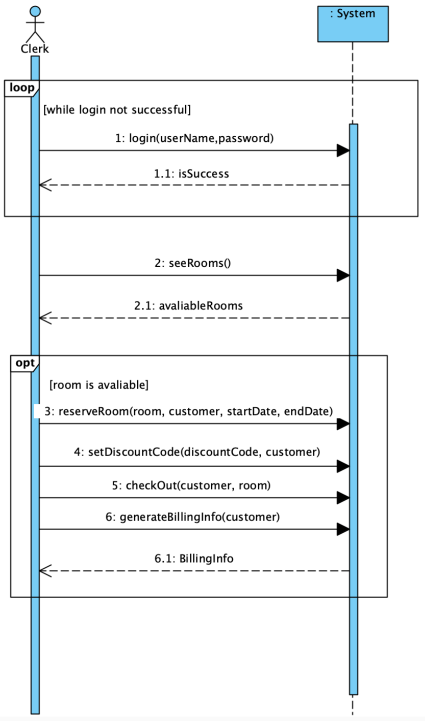
V. If there are no available rooms in the time frame
   A. The customer can look for other rooms at different dates

Will Clore System Sequence Diagrams:

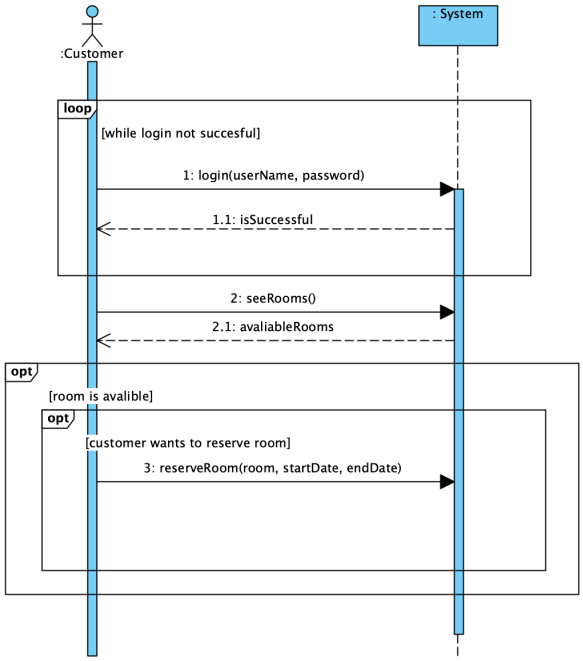
Use Case: Login



Use Case: Apply Discount



U



Will Clore Operation Contracts:

Contract C01: login
Operation: login(userName, password)
Cross References: Use Case: User Login, Use Case: Apply Discount,
Use Case: check Room Availability
Pre Condition: The user wants to access their account
Post Condition:

- userName is associated with an existing userAccount, based on userName
- password is compared against existing userAccount's password and is associated if they match

Contract C02: seeRooms
Operation: seeRooms()
Cross References: Use Case: Apply Discount, Use Case: check Room Availability
Pre Conditions: the clerk wants to see available rooms
Post Conditions:

- An instance of availableRooms rms is created
- rms is associated with a predefined roomsList, based on whether each room in roomsList is reserved or not

Contract C03: reserveRooms

Operation: reserveRoom(room, customer, startDate, endDate)
Cross References: Use Case: Apply Discount, Use Case: check Room Availability
Pre Conditions: a reservation is underway
Post Conditions:

- An instance of a new reservation rev is created
- rev is associated with a room number
- rev is associated with a customer
- rev is associated with a startDate
- rev is associated with an endDate

Contract C04: setDiscountCode
Operation: setDiscountCode(discountCode, customer)
Cross References: Use Case: Apply Discount
Pre Conditions: an account needs a discount applied
Post Conditions:

- An instance of a new discount disc is created
- The discount code is associated with disc
- disc is associated with a customer

Contract C05: checkOut
Operation: checkOut(customer, room)
Cross References: Use Case: Apply Discount
Pre Conditions: the system needs to check someone out
Post Conditions:

- the customer is disassociated with the room

Contract C06: generateBillingInfo
Operation: generateBillingInfo(customer)
Cross References: Use Case: Apply Discount
Pre Conditions: a bill needs to be generated by the system
Post Conditions:

- A new instance of billingInfo val is created
- val is associated with the customer

Derek Martinez Use Cases:

ID UC Reset Password
Scope Management System
Level user goal
Stakeholders and interests

Customer
– guest that needs to change their password.

Staff
– clerk that needs to change their password.

Manager
– admin is contacted to change passwords with his access.

System maintainer
– person responsible for application run.

Precondition: Staff is authenticated.

Postcondition: Account has remained unchanged besides new password. Able to access account a change in passwords

Main success scenario:

1. The customer or staff wants to change their password.
2. Contact admin.
3. Admin will log on to their account.
4. Customer/staff will confirm identity with admin
5. Admin will ask customer/staff for their account information.
6. Admin will find their account.
7. Admin will change the password to requested change.
8. Customer/staff will sign in to account to ensure the change worked.

Alternate paths:

1. anytime system does not respond
    I. staff will restart application

2a. If system does not allow admin to login,
    I. Admin will make a note of info and password change to do when the system reboots.

2b. If system cannot connect to network
    I. Admin will mark down info and password change to do when connected.

3a. Customer cannot identify
    I. Admin will refuse to change password.

ID UC Clerk Account
Scope Management System
Level user goal
Stakeholders and interests

Staff

– clerk that needs an account created.

Manager

– admin creates account for clerk.

System maintainer
– person responsible for application run.

Precondition: Staff is hired.

Postcondition: Clerk has specialized access and account is created with default name and password.

Main success scenario:

1. The staff is hired.
2. Contact admin for account creation.
3. Staff will confirm identity with admin.
4. Admin gets information from the clerk.
5. Admin creates their account with a default password.
6. Clerk logs in to ensure access.
7. Clerk is given access to the hotel management system.

Alternate paths:

1. anytime system does not respond
    I. staff will restart application

2a. If system does not allow admin to login,
    I. Admin will make a note of info and create the account when system reboots.

2b. If system cannot connect to network
    I. Admin will mark down info and create the account when connected..

3a. Clerk cannot identify
    I. Admin will refuse to create the account.

ID UC Billing Information
Scope Management System
Level user goal
Stakeholders and interests

Customer
– customer who receives their billing information for their stay.

Staff
– clerk that generates the billing information.

System maintainer
– person responsible for application run.

Precondition: Clerk has purchased a reservation or bought items.

Postcondition: Customer is given billing information of their charges.

Main success scenario:

1. Customer asks the staff for their billing information
2. The staff confirms the identity of the customer
3. The staff receives the information from the customer's account.
4. The staff locates the users information for their billing information
5. The staff generates the billing information
6. The customer must then provide accepted payment.
7. The clerk fulfills the payment amount with the customer’s payment method.
8. The clerk checks out the customer.

Alternate paths:

1. anytime system does not respond
    I. staff will restart application

2a. If system does not allow admin to login,
    I. Staff will manually generate the billing information.

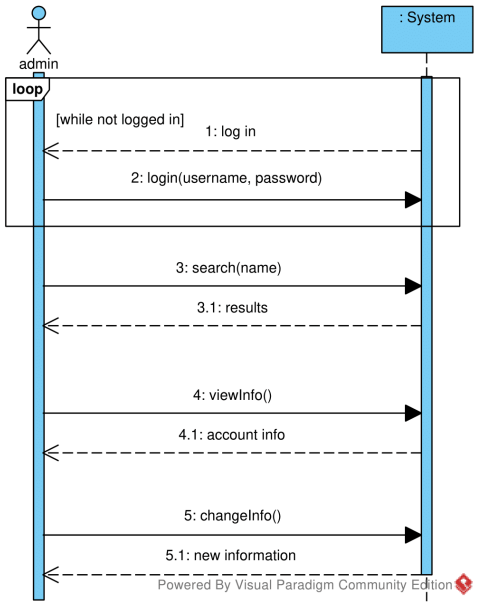
2b. If system cannot connect to network
    I. Staff will manually generate billing info.
    II. Staff will wait for the system to connect and generate it through the system.

3a. Customer cannot identify
    I. Staff will refuse to generate and accept payment.

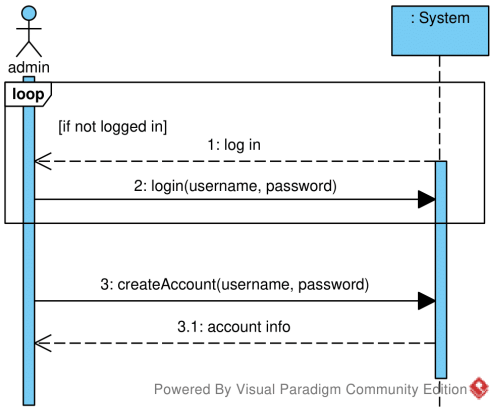
4. Customer does not provide payment.
    I. Staff will hold their card and personal information till they can pay.

Derek Martinez System Sequence Diagrams:

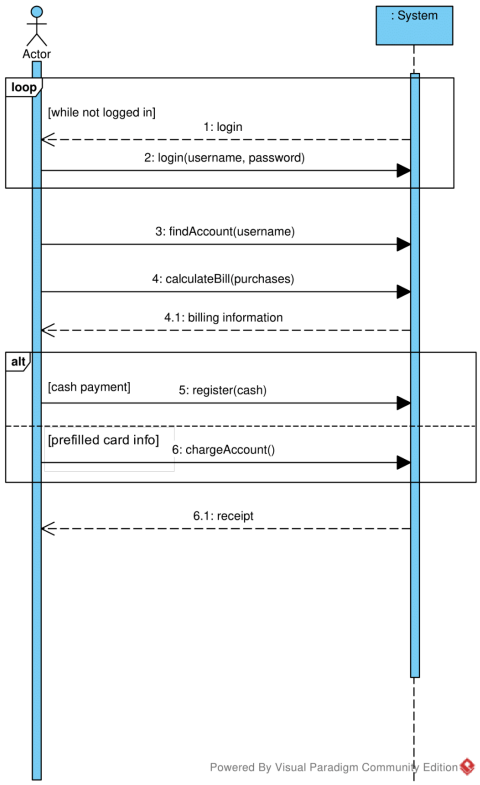
Use Case: Reset Password



Use Case: Create Clerk Account



Use Case: Generate Billing Info



Derek Martinez Operation Contracts:

Contract C01: login
Operation: login(userName, password)
Cross References: Use Case: Generate Billing Info, Use Case:Create Clerk Account, Use Case: Reset Password
Pre Condition: The user wants to access their account
Post Condition:

- userName is associated with an existing userAccount, based on userName
- password is compared against existing userAccount's password and is associated if they match

Contract C02: find account
Operation: findAccount(userName)/search(userName)
Cross References: Use Case: Generate Billing Info, Use Case: Reset Password
Pre Condition: the user wants to find account info
Post Condition:

- userName is associated with an existing userAccount, based on userName

Contract C03: change info

Operation: changeInfo()
Cross References: Use Case: Reset Password
Pre Condition: The user wants to change their info
Post Condition:

- userName is associated with an existing userAccount, based on userName
- password is compared against existing userAccount's password and is associated if it is a different password

Contract C04: create account
Operation: createAccount(userName, password, paymentMethod, clearance)
Cross References: Use Case: Create Clerk Account
Pre Condition: The user wants to create a account for clerks
Post Condition:

- userName is associated with a newly created userName
- Password is associated with a newly created password
- Payment method is associated if user inputs one, otherwise default value
- Clearance is associated with type of account being made, otherwise default value

Contract C04: calculate bill
Operation: calculateBill(userName, purchases)
Cross References: Use Case: Generate Billing Information
Pre Condition: The user wants to get billing info
Post Condition:

- userName is associated with a newly created userName
- Password is associated with a newly created password
- Payment method is associated if user inputs one, otherwise default value
- Clearance is associated with type of account being made, otherwise default value

Contract C04: charge Account
Operation: chargeAccount()
Cross References: Use Case: Generate Billing Information
Pre Condition: user wants to charge account bills
Post Condition:

- Billing Information is provided to account associated with userName
- Purchases are cleared based on the payment amount

Alan Villagrand Use Cases:

ID UC Make a Reservation (Clerk)
Scope Hotel reservation system
Level user goal
Stakeholders and interests
Customer
– guest that is interested in a hotel room reservation.

Staff
– clerk responsible for hotel room reservation work.
Manager
– admin directing staff and solving unusual problems.
System maintainer
– person responsible for application run.
Precondition: Staff is authenticated.

Postcondition: Transaction is saved, room is assigned.

Main success scenario:

1. the customer wants to make a reservation.
2. ask a staff for serving
3. staff will login to the system
4. staff will ask customer about reservation details
5. staff will check availability of rooms
6. customer will choose room
7. staff will ask customer for identification
8. staff will make sure customer is over 18
9. staff will fill out received information to the system
10. staff will advise customer about hotel rules
11. customer confirm reservation
12. staff confirms reservation in system

Alternate paths:

I. Anytime system does not respond
   A. staff will restart application
II. If system does not allow staff to login
   A. staff will call manager
   B. staff will make manual reservation
III. If system cannot connect to network
   A. staff will call manager
   B. staff will make manual reservation
IV. If room is not available for customer
   A. staff will inform customer
V. Customer cannot identify
   A. staff will discard reservation
VI. If customer does not agree with hotel rules
   A. reservation is not confirmed and discarded

ID UC Make a Reservation (Guest)
Scope Hotel reservation system
Level user goal
Stakeholders and interests

Customer
– person that is interested in a hotel room reservation and hotel services
Staff
– person responsible for hotel room reservation work.
Manager
– person directing staff and solving unusual problems.
System maintainer
– person responsible for application run.
Precondition: Customer has account

Postcondition: Transaction is saved, room is assigned.

Main success scenario:

1. The customer wants to make a reservation.
2. Customer will log in to the system
3. Customer will check availability of rooms
4. Customer will choose room
5. Customer will prove identification
6. System will make sure customer is over 18
7. Customer will fill out information to the system
8. Customer will agree to the hotel rules
9. Customer confirms reservation

Alternate paths:

I. Anytime system does not respond
   A. Customer will restart application
II. If system does not allow customer to login
   A. Customer will contact staff
   B. Staff will make reservation
III. If system cannot connect to network
   A. Customer will contact staff
   B. Staff will make reservation
IV. If room is not available for customer
   A. Customer will have to wait for next open room
V. If customer does not agree with hotel rules or is not of age
   A. Reservation is not confirmed and discarded

ID UC Cancel a Reservation (Guest)
Scope Hotel reservation system
Level user goal
Stakeholders and interests
Customer
– person that is interested in a hotel room reservation and hotel services
Staff
– person responsible for hotel room reservation work.
Manager
– person directing staff and solving unusual problems.

System maintainer
– person responsible for application run.
Precondition: Customer has reservation

Postcondition: Reservation is canceled, room opens up.

Main success scenario:

1. The customer wants to cancel a reservation.
2. Customer will log in to the system
3. Customer will look for reservation
4. Customer will choose to cancel reservation
5. Customer confirms cancellation
6. Customer will receive refund
7. Customer will be charged if cancellation is made after 2 days of reservation date

Alternate paths:

I. Anytime system does not respond
   A. Customer will restart application

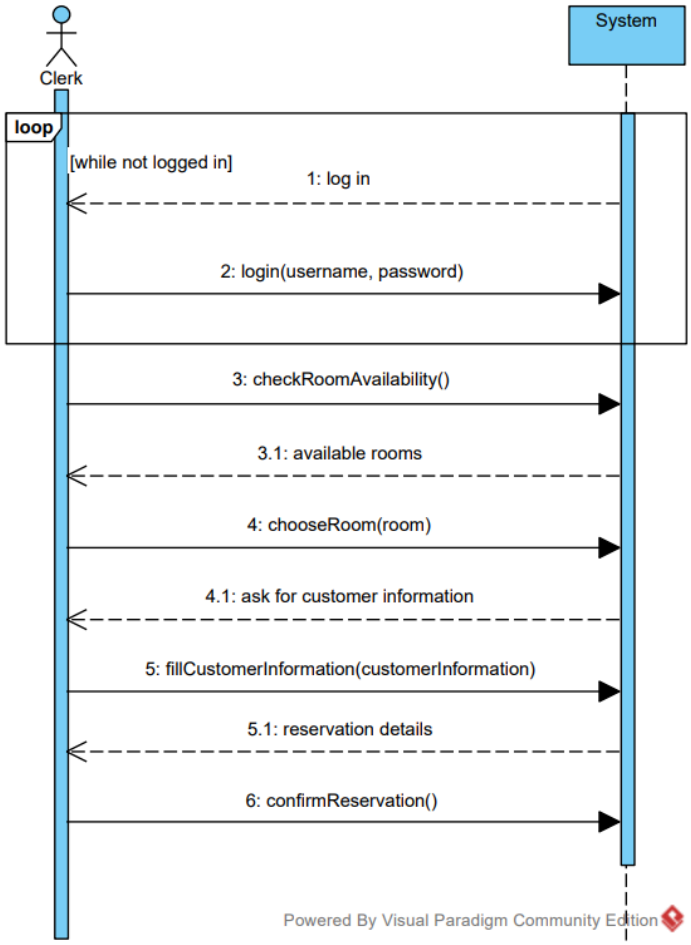
II. If system does not allow customer to login
   A. Customer will contact staff
   B. Staff will cancel reservation

III. If system cannot connect to network
   A. Customer will contact staff
   B. Staff will cancel reservation

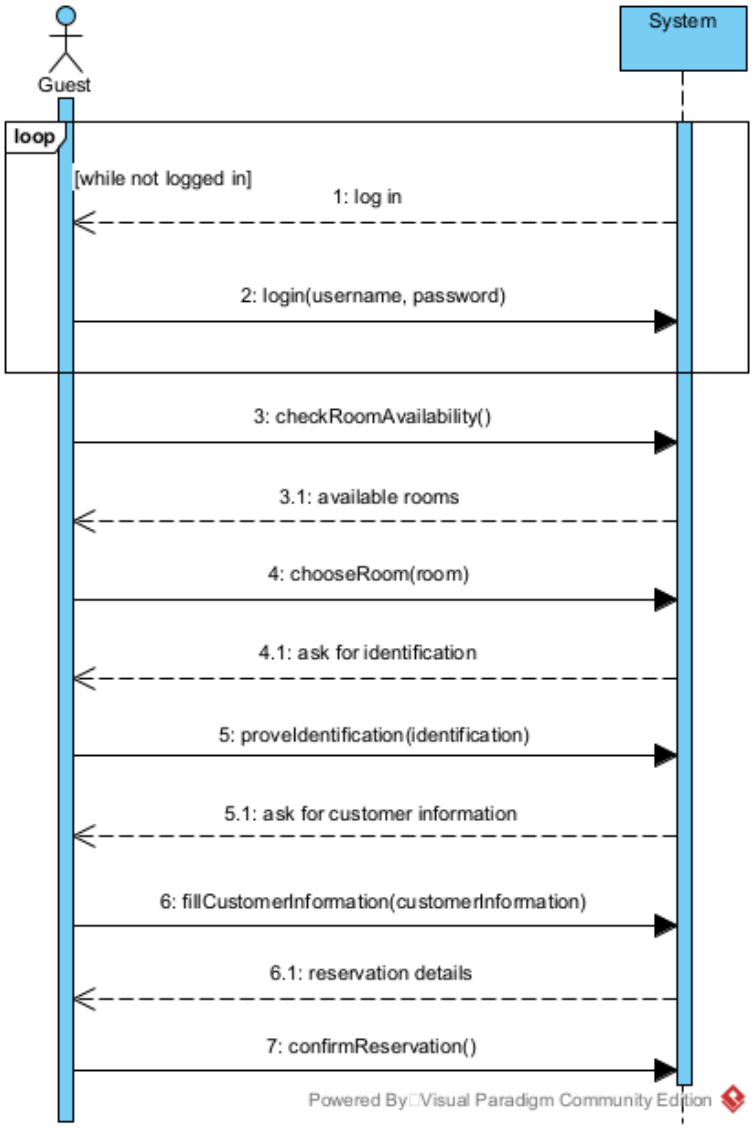
IV. If Customer cannot find reservation
   A. There is no reservation to be canceled

Alan Villagrand System Sequence Diagrams:

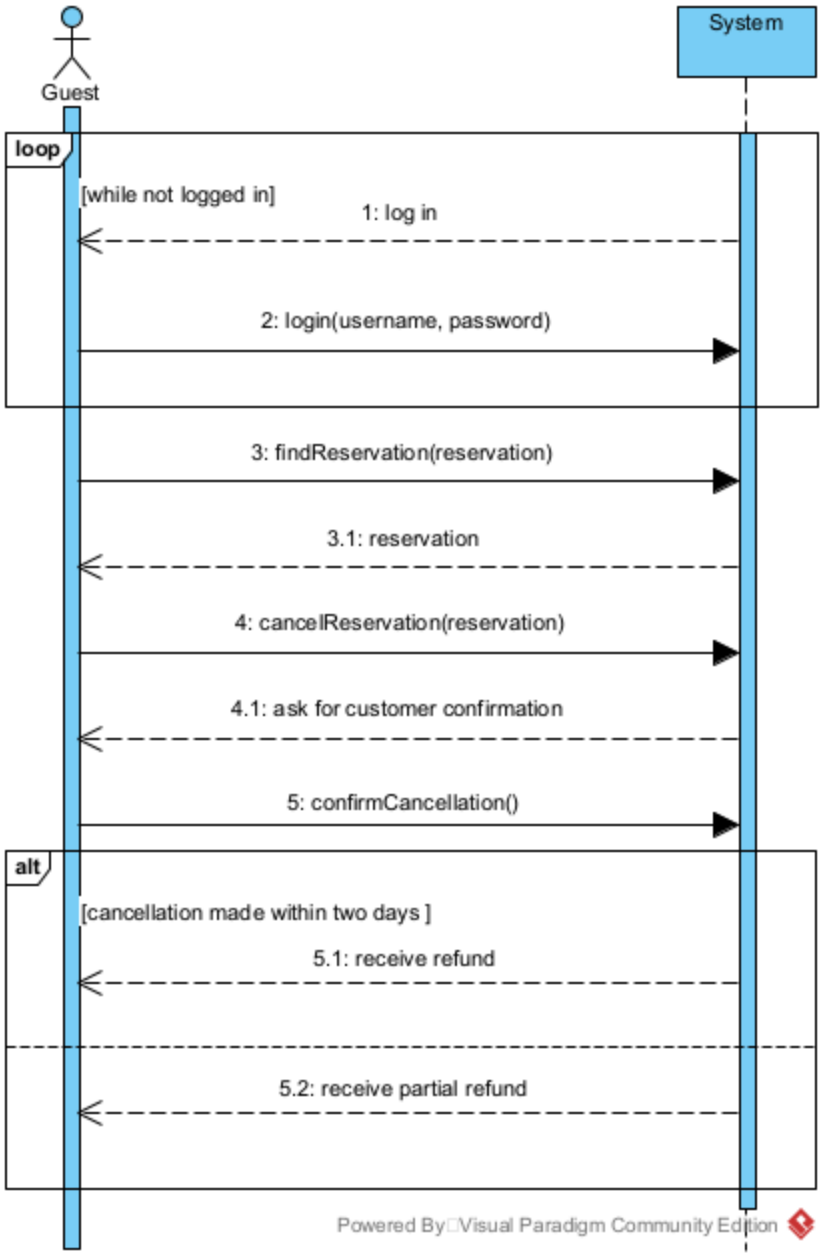
Use Case: Make a Reservation (clerk)



Use Case: Make a Reservation (guest)



Use Case: Cancel a Reservation (guest)



Alan Villagrand Operation Contracts:

Contract CO1: login
Operation: login(username, password)
Cross References: Use Case: Make reservation (clerk), Use Case: Make reservation (guest), Use Case: Cancel reservation
Pre-conditions: The user has a username and password
Post-conditions:

- The user is logged into the system

Contract CO2: checkRoomAvailability
Operation: checkRoomAvailability()

Cross References: Use Case: Make reservation (clerk), Use Case: Make reservation (guest)
Pre-conditions: The user wants to see room availability
Post-conditions:

- A list of rooms available is given to the user

Contract CO3: chooseRoom
Operation: chooseRoom(room)
Cross References: Use Case: Make reservation (clerk), Use Case: Make reservation (guest)
Pre-conditions: The user has an available room
Post-conditions:

- The user is assigned to the chosen room
- The user is asked to fill out customer information

Contract CO4: fillCustomerInformation
Operation: fillCustomerInformation(customerInformation)
Cross References: Use Case: Make reservation (clerk), Use Case: Make reservation (guest)
Pre-conditions: The user has the customer information
Post-conditions:

- If information is valid, user is given reservation details

Contract CO5: confirmReservation
Operation: confirmReservation()
Cross References: Use Case: Make reservation (clerk), Use Case: Make reservation (guest)
Pre-conditions: The user has read over the reservation details
Post-conditions:

- The reservation is confirmed
- The room is assigned to the customer

Contract CO6: proveIdentification
Operation: proveIdentification(identification)
Cross References: Use Case: Make reservation (clerk), Use Case: Make reservation (guest)
Pre-conditions: The user is of age and has identification
Post-conditions:

- The system proves user is of age
- The system proceeds to room reservation process

Contract CO7: findReservation
Operation: findReservation(reservation)
Cross References: Use Case: Cancel reservation
Pre-conditions: The user has a reservation
Post-conditions:

- The system finds the reservation
- The user can modify the reservation

Contract CO6: cancelReservation

Operation: cancelReservation(reservation)
Cross References: Use Case: Cancel reservation
Pre-conditions: The user has a reservation to be canceled
Post-conditions:

- The system asks the user to confirm cancellation

Contract CO6: confirmCancellation
Operation: cancelCancellation()
Cross References: Use Case: Cancel reservation
Pre-conditions: The user has requested to cancel reservation
Post-conditions:

- The system cancels the reservation
- The user will receive a refund depending on when the cancellation was made

Nicholas Nolen (Use Cases):

Nicholas (Making Reservation- Restaurant):
ID UC Make a reservation for the hotel restaurant
Scope Restaurant reservation system
Level user goal
Stakeholders and interest
Customer
-person trying to use restaurant services
-person responsible for making a reservation through the system
Manager
-person who manually solves problems and makes system decisions
System maintainer
-person who will manually fix the system if it breaks
-person responsible for application run.

Precondition: Restaurant reservation is during operational hours

Postcondition: Record is made of a customer's booking

Main success scenario:

1. Customer wants to make a reservation
2. Customer will login to the hotel website
3. Customer will navigate to the portion handling the restaurant
4. Customer will search their desired date
5. Customer will input their party size
6. System will present the customer with available times
7. Customer will choose from the times available
8. Customer will confirm their date and time

9. System will save the Customer's reservation
10. System will create a report of the reservations

Alternate paths:

a. If the online reservation system is down
    1. Customer will call the Staff and book their reservation
    2. Staff will manually process reservation in a physical ledger
b. if Customer makes a reservation but System fails to record it
    1. Customer will present staff with proof of reservation
    2. Staff will attempt to keep Customer's reservation
    3. Staff will create another reservation for Customer if unable to keep reservation
c. If Customer violates restaurant rules
    1. Staff will cancel reservation and ask Customer to leave
    2. Security will escort the Customer to the door if Customer fails to leave

Nicholas (Modify Password - Person):

ID UC Modify the password for a system user
Scope Hotel reservation system
Level user goal
Stakeholders and interest
Customer
-person who is interested in staying in a room and using the hotel services
-person who is interested in accessing their user account
Staff
-person who is responsible for the guest services provided by the hotel
System maintainer
-person who will manually fix the system if it breaks
-person responsible for application run.

Precondition: Guest has already created a user account

Postcondition: Password has been updated

Main success scenario:

1. Customer wants to update thier password
2. Customer will login to the website
3. Customer will navigate to the user information modification section
4. Customer will choose the option to modify their password
5. System will prompt customer to input their old password
6. Customer will attempt to input their password until it is input correctly
7. System will prompt the user to input a new password
8. Customer attempt to input a password until their new password meets requirements

9. Customer will confirm their new password
10. System will save the new password

Alternate paths:

a. If the customer fails to input their password over and over
    1. System will email the user a temporary password
    2. User will input temporary password
    3. System will prompt the user to update their password
    4. Customer will update their password according to system requirements

b. If the customer chooses the forgot password option at login
    1. System will email the user a temporary password
    2. User will input temporary password
    3. System will prompt the user to update their password
    4. Customer will update their password according to system requirements

Nicholas (Ordering Food):

ID UC Guest Orders food from hotel resturaunt
Scope Restaurant system
Level user goal
Stakeholders and interest
Customer
-person trying to use restaurant services
-person responsible for ordering the food that they want
Staff
-person who takes customer's order and inputs it into system
System maintainer
-person who will manually fix the system if it breaks
-person responsible for application run.

Precondition: Restaurant is open and customer has a table

Postcondition: Customer's order is input into the system for processing

Main success scenario:

1. Customer wants to order food
2. Customer will look through the menu
3. Customer will communicate to staff when ready to order
4. Staff will take customer's order
5. Staff will input order into System
6. System will create order tickets to give to the kitchen staff
7. Staff will bring out customer's food
8. System will calculate the customer's bill
9. System will print the bill for the customer

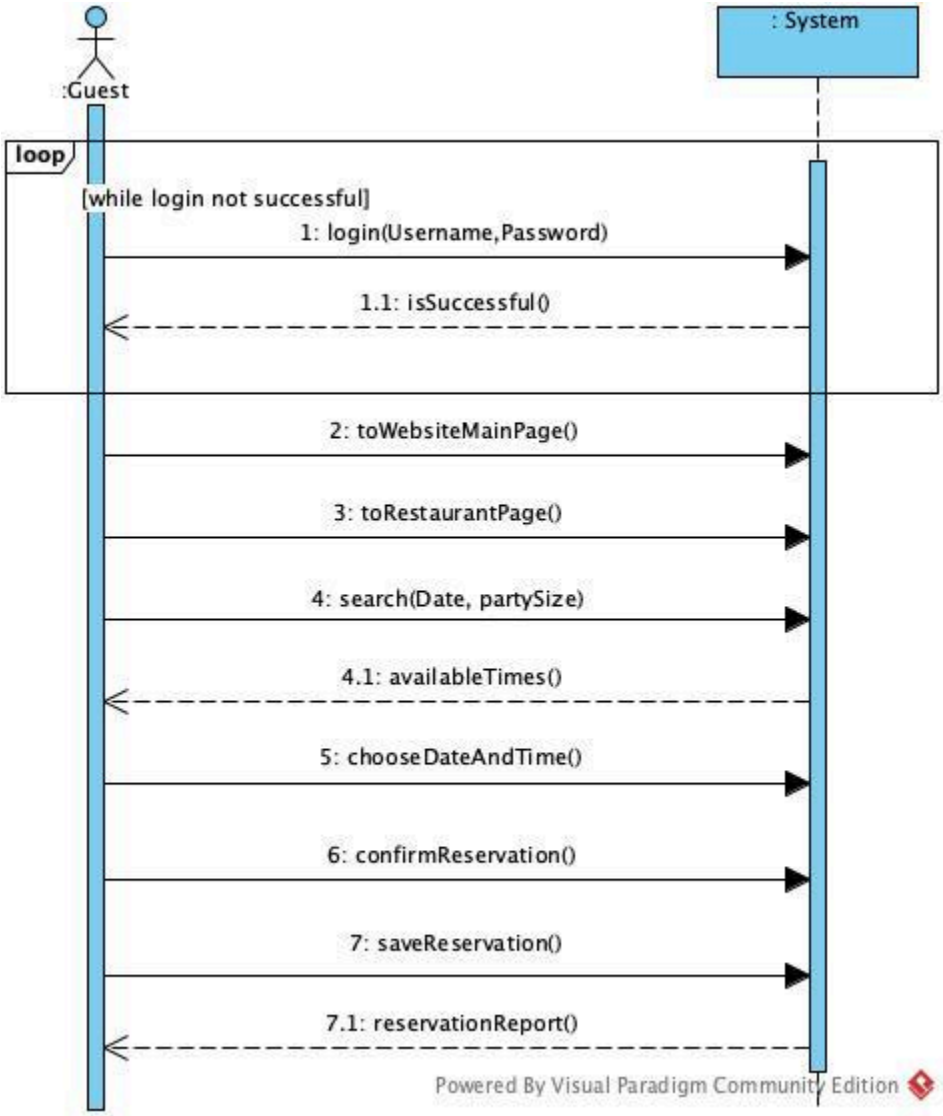
Alternate paths:

b. If the food ordering system is down
    1. Staff will take the customer's order
    4. Staff will manually write order tickets to give to the kitchen staff

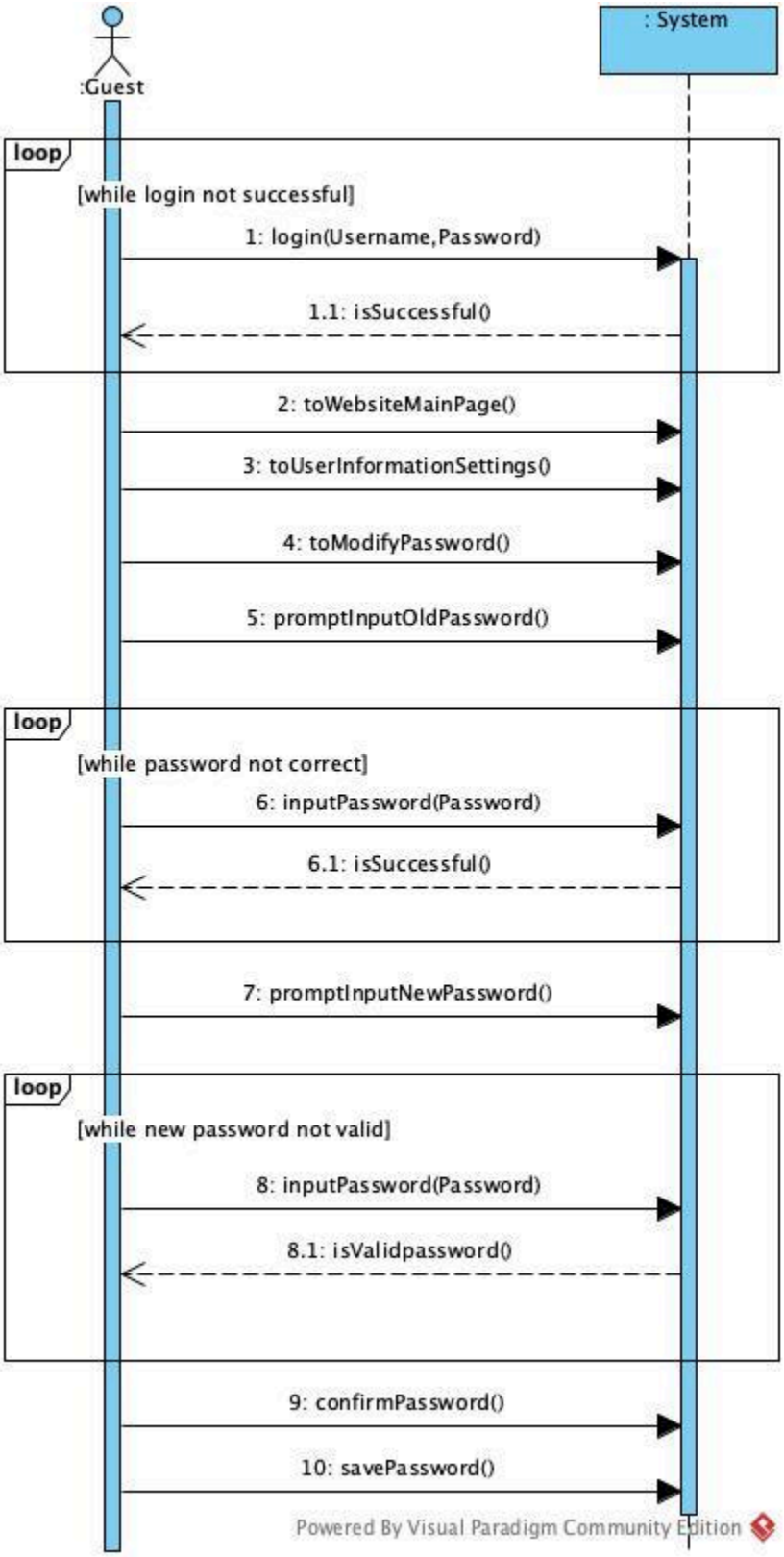
b. if Staff inputs order incorrectly
    1. Staff will remove incorrect item from the system
    2. Staff will add correct item to system
    3. Staff will bring correct item to customer
    4. Up to Staff's discretion to take item off of final bill

Nicholas Nolen (System Sequence Diagrams):

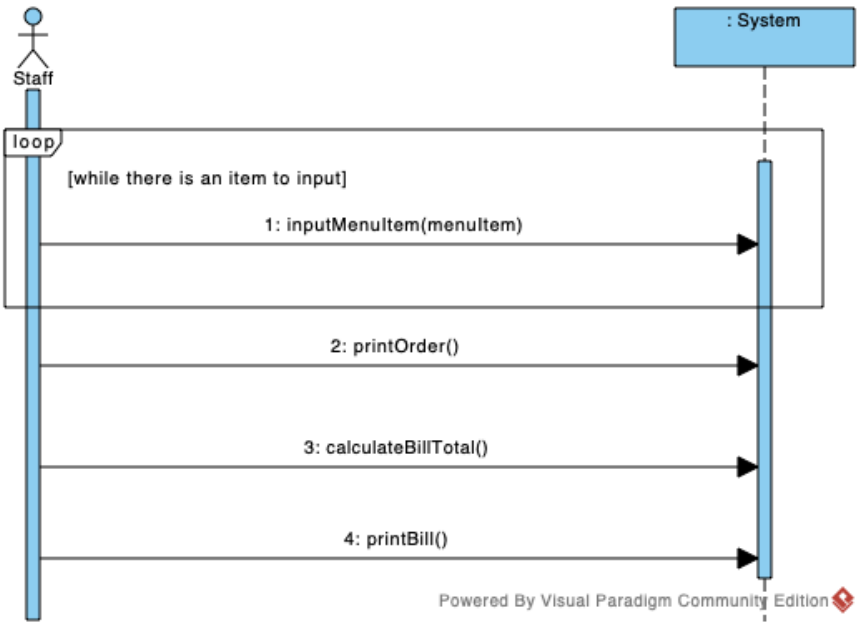
Make Reservation- Restaurant:



Nicholas Nolen(Modify Password):



Nicholas Nolen(Order Food):



Nicholas Nolen Operation Contract(Making Reservation- Restaurant):

Contract C01: login

Operation: login(Username,Password)
Cross References: Use Case: Making Reservation- Restaurant, Modify Password
Pre-conditions: The user is not logged in to the system
Post-conditions: The user attempts to log in to the system

Contract C02: isSuccesful

Operation: isSuccesful(reservation)
Cross References: Use Case: Making Reservation- Restaurant, Modify Password
Pre-conditions: The user has attempted to log in to the system
Post-conditions: True or False if the user is able to log into the system

Contract C03: toWebsiteMainPage

Operation: toWebsiteMainPage()
Cross References: Use Case: Making Reservation- Restaurant
Pre-conditions: The user has successfully logged in to the system
Post-conditions: The user is navigated to the main page of the website

Contract C04: toRestaurantMainPage

Operation: toRestaurantMainPage()
Cross References: Use Case: Making Reservation- Restaurant
Pre-conditions: The user has successfully navigated to the main website page
Post-conditions: The user is navigated to the Restaurant reservation page of the website

Contract C05: search

Operation: search(Date, partySize)
Cross References: Use Case: Making Reservation- Restaurant
Pre-conditions: The user has successfully navigated to the Resturaunt reservation page of the website
Post-conditions: The user is provided with the available times on the date and with the party size they provided.

Contract C06: availableTimes

Operation: availableTimes()
Cross References: Use Case: Making Reservation- Restaurant
Pre-conditions: The user has queried the website search function with a date and a party size
Post-conditions: The available times on the user's requested date with the capacity to accommodate the user's requested party size.

Contract C07: chooseDateAndTime

Operation: chooseDateAndTime()
Cross References: Use Case: Making Reservation- Restaurant
Pre-conditions: The user has received the available times for their reservation
Post-conditions: The user chooses their reservation time

Contract C08: confirmReservation
Operation: confirmReservation()
Cross References: Use Case: Making Reservation- Restaurant
Pre-conditions: The user has chosen their reservation time
Post-conditions: The user confirms their desire to reserve the specific time they have chosen

Contract C09: saveReservation
Operation: saveReservation()
Cross References: Use Case: Making Reservation- Restaurant
Pre-conditions: The user has confirmed their reservation
Post-conditions: The system will save their reservation and limit future bookings of that date and time.

Contract C10: reservationReport
Operation: reservationReport()
Cross References: Use Case: Making Reservation- Restaurant
Pre-conditions: The System has saved the user's reservation
Post-conditions: The System returns the reservation details to the user.

Nicholas Nolen Operation Contract(Modify Password):

Contract C01: login
Operation: login(Username,Password)
Cross References: Use Case: Making Reservation- Restaurant, Modify Password
Pre-conditions: The user is not logged in to the system
Post-conditions: The user attempts to log in to the system

Contract C02: isSuccesful
Operation: isSuccesful(reservation)
Cross References: Use Case: Making Reservation- Restaurant, Modify Password
Pre-conditions: The user has attempted to log in to the system
Post-conditions: True or False if the user is able to log into the system

Contract C03: toWebsiteMainPage
Operation: toWebsiteMainPage()
Cross References: Use Case: Making Reservation- Restaurant, Modify Password
Pre-conditions: The user has successfully logged in to the system
Post-conditions: The user is navigated to the main page of the website

Contract C04: toUserInformationSettings
Operation: toUserInformationSettings()
Cross References: Use Case: Making Reservation- Modify Password
Pre-conditions: The user has successfully navigated to the main website page

Post-conditions: The user is navigated to the information settings page of the website

Contract C05: toModifyPassword
Operation: toModifyPassword()
Cross References: Use Case: Making Reservation- Modify Password
Pre-conditions: The user has successfully navigated to the information settings page
Post-conditions: The user is navigated to the password settings page

Contract C06: promptInputOldPassword
Operation: promptInputOldPassword()
Cross References: Use Case: Making Reservation- Modify Password
Pre-conditions: The user has successfully navigated to the password settings page
Post-conditions: The user is prompted to input their old password

Contract C07: inputPassword
Operation: inputPassword(Password)
Cross References: Use Case: Making Reservation- Modify Password
Pre-conditions: The user has successfully navigated to the password settings page
Post-conditions: The user has attempted a password

Contract C08: promptInputNewPassword
Operation: promptInputNewPassword()
Cross References: Use Case: Making Reservation- Modify Password
Pre-conditions: The user has successfully input their old password
Post-conditions: The user is prompted to input a new password

Contract C09: isValidPassword
Operation: isValidPassword()
Cross References: Use Case: Making Reservation- Modify Password
Pre-conditions: The user has attempted a new password
Post-conditions: The System returns true of false if the user's new password is valid

Contract C10: confirmPassword
Operation: confirmPassword()
Cross References: Use Case: Making Reservation- Modify Password
Pre-conditions: The user has successfully input a new password
Post-conditions: The user confirms their new password

Contract C11: savePassword
Operation: savePassword()
Cross References: Use Case: Making Reservation- Modify Password
Pre-conditions: The user has confirmed their new password

Post-conditions: The System saves their new password.

Nicholas Nolen Operation Contract(Order Food):

Contract C01: inputMenuItem
Operation: inputMenuItem(menuItem)
Cross References: Use Case: Order Food
Pre-conditions: The staff has taken a customer's order
Post-conditions: The an item of the customer's order is input into the system

Contract C02: printOrder
Operation: printOrder()
Cross References: Use Case: Order Food
Pre-conditions: All of the customer's selected menu item's have been input into the system
Post-conditions: Tickets for all of the menu items are printed for the kitchen staff

Contract C03: calculateBillTotal
Operation: calculateBillTotal()
Cross References: Use Case: Order Food
Pre-conditions: All of the customer's selected menu item's have been input into the system
Post-conditions: The system calculates the bill for the customer

Contract C04: printBill
Operation: printTotal()
Cross References: Use Case: Order Food
Pre-conditions: The customer's bill has been calculated
Post-conditions: The customer's bill is printed for them

Joseph Zuniga

ID UC Make a Purchase from Hotel Store Using Online Shopping Cart
Scope Point of Sale
Level user goal
Stakeholders and interest
Customer
-person trying to make a purchase from online shopping cart
Staff
-person responsible for completing order
Manager
-person directing staff and solving unusual problems.
System maintainer
-person responsible for application run.

Precondition: Store is in operational hours

Postcondition: Transaction is complete

Main success scenario:

1. Customer wants to make an online purchase from store
2. Customer will login to the hotel website
3. Customer will access online store from hotel website
4. Customer will add desired items to cart
5. Customer will then proceed to checkout
6. Customer will then enter credit/debit card information or can add to reservation tab
7. Customer can then choose to have the items delivered to room for a fee or pick up items
8. Point of Sale System will generate an itemized receipt for customer
9. Staff will confirm the order in the store
10. Staff will gather items to complete order
11. Staff will then deliver the items to the room if requested
12. If customer picks up, staff will confirm with the customer via name or reservation ID

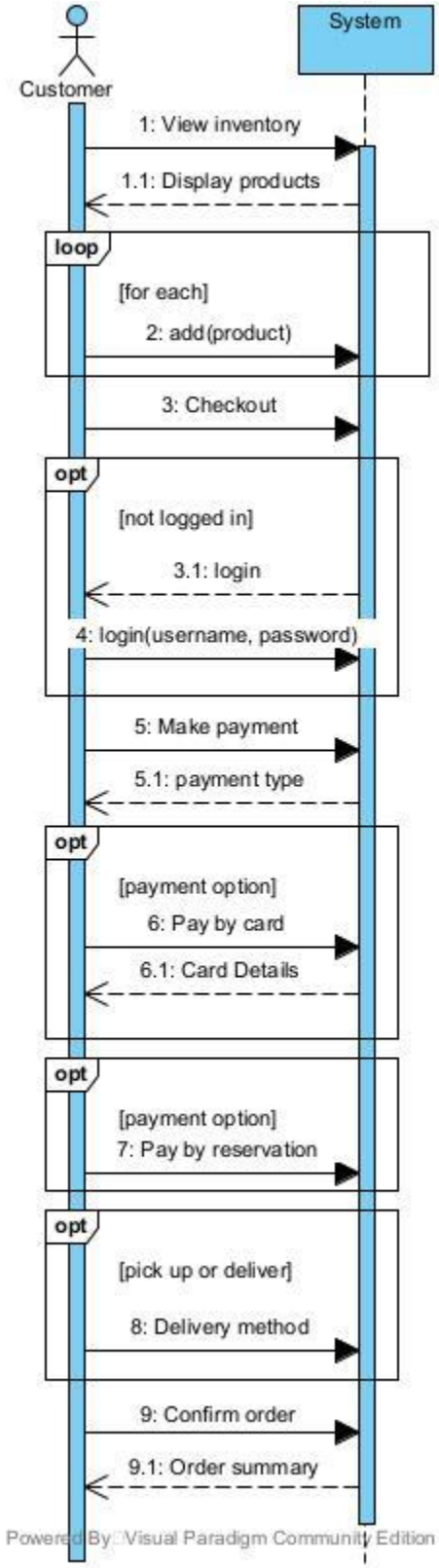
Alternate paths:

a. If the online shopping cart is down
    1. Staff will attempt to restart the application
    2. Staff will receive orders via phone

3.a if customer cannot access the shopping cart via the hotel website
    1. Customer can call the store and make an order

6.a if the customer cannot make payment online
    1. Customer can make physical payments at the store
    2. Staff at the store will complete the transaction

ID UC Make a Purchase from Hotel Store Using Online Shopping Cart
System Sequence Diagram (SSD):



ID UC Make a Purchase from Hotel Store Using Online Shopping Cart
Operation Contracts:

Contract C01: Inventory
Operation: displayInventory()
Cross References: Use Cases: Check Inventory
Pre Condition: User wants to view store items
Post Condition:

- Store Items are displayed

Contract C02: Add product to cart
Operation: addProduct(Product)
Cross References: Use Cases: Check Inventory, Make cart
Pre Condition: User wants to add a product to cart
Post Condition:

- Cart is created
- Product is added to cart

Contract C03: Proceed to checkout
Operation: checkout(Cart)
Cross References: Use Cases: Check Inventory, Make cart, Create bill
Pre Condition: User wants to checkout with items in cart
Post Condition:

- Bill is generated
- Itemized list generated with cost and tax

Contract C04: Generate a bill
Operation: generateBill(checkout)
Cross References: Use Cases: Check Inventory, Make cart, proceed to checkout
Pre Condition: User is attempting to pay for cart
Post Condition:

- Payment method is requested
- Items are held

Contract C05: Hold items for customer
Operation: holdCart(Cart)
Cross References: Use Cases: Make cart, proceed to checkout, generate bill
Pre Condition: User is going to buy items in cart
Post Condition:

- Items in cart are not available for sale online

Contract C06: login
Operation: login(userName, password)
Cross References: Use Case: Generate Billing Info, Use Case:Create Clerk Account, Use Case: Reset Password
Pre Condition: The user wants to access their account
Post Condition:

- userName is associated with an existing userAccount, based on userName
- password is compared against existing userAccount's password and is associated if they match

Contract C07: Payment
Operation: payment(Card info, total)
Cross References: Use Case: Generate Billing Info
Pre Condition: The user is paying for items
Post Condition:

- Items removed from inventory
- Receipt is generated

Contract C08: Delivery
Operation: deliverMethod(to room)
Cross References: Use Case: customer account, has a room, bill statement
Pre Condition: The user wants items delivered to room
Post Condition:

- Additional charge added to bill

Contract C09: Order Summary
Operation: orderSummary()
Cross References: Use Case: Generate Billing Info, checkout items, payment
Pre Condition: The user wants a summary of order
Post Condition:

- Summary of order is generated

ID Cancel reservation (Clerk)
Scope Hotel Reservation System
Level Clerk
Stakeholders and interest
Clerk
-Person trying to cancel reservation
Staff
-Person who oversees reservations
Manager
-person directing staff and solving unusual problems.
System maintainer
-person responsible for application run.

Precondition: Reservation to cancel

Postcondition: Reservation is canceled and room is available

Main success scenario:

1. Clerk needs to cancel a reservation
2. Clerk will login to Reservation Management System
3. Clerk will access current reservations
4. System will generate and display current reservations
5. Clerk will select reservation to cancel
6. System will check if reservation is canceled within two days of reservation made
7. Clerk will notify guest if there's any fees with cancellation
8. Guest will confirm the charges
9. Clerk will continue with cancellation
10. If paying fees, Clerk will take payment from card on reservation or ask for alternate payment
11. Refund is initiated and guest is refunded for stay
12. Receipt will be generated and sent to guest
13. Guest will receive notice of canceled reservation
14. System will make room available

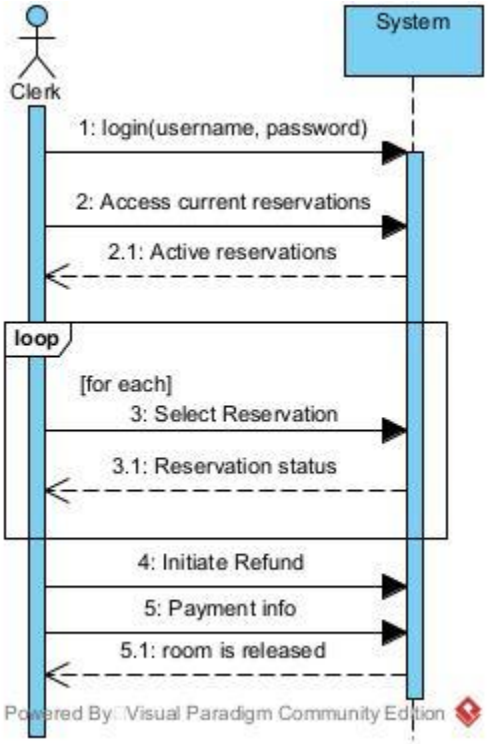
Alternate paths:

2. Hotel Reservation System is down
    - 2.1. Clerk will notify system manager of problem
    - 2.2. System manager will restart the system
    - 2.3. Clerk will track reservations manually until system is back up
3. Clerk can’t find reservation
    - 3.1. Clerk will ask guest for reservation number
    - 3.2. Clerk will access guest account to check reservation status
    - 3.3. Clerk will check if guest has been charged for reservation
        - 3.3.1. If guest has been charged, Clerk will initiate a refund

3.3.2. Notify System Manager of missing reservation

9. Guest card not working for fees payment
    9.1. Clerk will ask for an alternative payment
    9.2. Clerk will withhold refund until fees are paid
10. Guest card not working for refund
    10.1 Clerk will ask for alternate refund method
    10.2 Guest will be accredited with the amount of the stay on alternate card

ID Cancel reservation (Clerk)
System Sequence Diagram:



ID Cancel reservation (Clerk)
Operation Contracts:

Contract C01: login
Operation: login(userName, password)
Cross References: Use Case: Generate Billing Info, Use Case:Create Clerk Account, Use Case: Reset Password
Pre Condition: The user wants to access their account
Post Condition:

- userName is associated with an existing userAccount, based on userName
- password is compared against existing userAccount's password and is associated if they match

Contract C02: View Reservations
Operation: currentReservations()
Cross References: Use Case: User Login, Set Reservation
Pre Condition: The user wants to view all reservations
Post Condition:

- A list of all reservations is generated

Contract C03: Select Reservation
Operation: addReservation(Reservation)
Cross References: Use Case: View Reservations, Make Reservation
Pre Condition: The user wants to select a reservation to modify
Post Condition:

- Reservations are stored in a list

Contract C04: Reservation Status
Operation: reservationStatus()
Cross References: Use Case: View Reservations, Make Reservation, Select Reservation
Pre Condition: The user wants info on the reservation
Post Condition:

- The information of a reservation is generated

Contract C05: Initiate Refund
Operation: paymentRefund()
Cross References: Use Case: View Reservations, Make Reservation, Payment, Generate Bill
Pre Condition: The user wants to start a refund
Post Condition:

- Payment information is generated
- Room status is changed to not occupied

Contract C06: Payment
Operation: payment(Card info, total)
Cross References: Use Case: Generate Billing Info
Pre Condition: The user is paying for items
Post Condition:

- Items removed from inventory
- Receipt is generated

Contract C07: Release Room
Operation: releaseRoom(Room)
Cross References: Use Case: Hold Room, Generate Refund
Pre Condition: The user want to make room available
Post Condition:

- room's status is updated and is available for reservation

ID Generate Billing Information (Clerk)
Scope Hotel Reservation System
Level Clerk
Stakeholders and interest
Clerk
-Person trying to generate billing information
Staff
-Person who accommodates guest needs
Manager
-person directing staff and solving unusual problems
System maintainer
-person responsible for application run

Precondition: Stay is complete, Clerk is logged into Reservation System

Postcondition: Billing information is generated for guest to pay

Main success scenario:

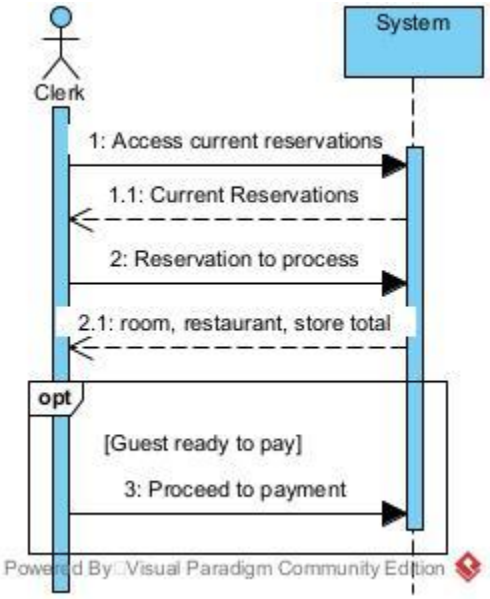
1. Clerk needs to generate billing information
2. Clerk will login to Reservation Management System
3. Clerk will access current reservations
4. System will return current reservations
5. Clerk will choose the reservation to generate billing info
6. System will generate billing info by accessing necessary fields
7. System will access the room rate and the number of nights
8. System will access the running total of store and restaurant purchases
9. System will then calculate the cost with tax
10. System will then return an itemized receipt
11. Clerk will present the billing information to the guest
12. Clerk will proceed to payment

Alternate paths:

2. Hotel Reservation System is down
    2.1. Clerk will notify system manager of problem
    2.2. System manager will restart the system
3. Clerk can't find reservation
    3.1. Clerk will ask guest for reservation number
    3.2. Clerk will access guest account to check reservation status
    3.3. Clerk will check to total manually and will notify guest
7. System cannot generate total for room
    7.1. Clerk will notify system manager of problem
    7.2. Clerk will manually generate the total by checking the rate when the reservation was made

8. Guest card not working for fees payment
    8.1. Clerk will ask for an alternative payment
    8.2. Clerk will notify manager

ID Generate Billing Information (Clerk)
System Sequence Diagram:



ID Generate Billing Information (Clerk)
Operation Contracts:

Contract C01: View Reservations
Operation: currentReservations()
Cross References: Use Cases: User Login, Set Reservation
Pre Condition: The user wants to view all reservations
Post Condition:

- A list of all reservations is generated

Contract C02: Select Reservation
Operation: addReservation(Reservation)
Cross References: Use Cases: View Reservations, Make Reservation
Pre Condition: The user wants to select a reservation to modify
Post Condition:

- Reservations are stored in a list

Contract C03: Get total
Operation: calculateTotal(Reservation)
Cross References: Use Cases: View Reservations, Make Reservation, Select Reservation
Pre Condition: The user wants to select a reservation to calculate total
Post Condition:

- The total of the stay including tax is generated

Contract C04: Proceed to payment
Operation: addPayment(Total)
Cross References: Use Cases: Pay Bill, Release room
Pre Condition: The user wants to pay for stay
Post Condition:

- The room is released

Harrison Hassler Fully Dressed Use Cases

ID: Check in a customer
Scope: Hotel Reservation System
Level: Clerk
Stakeholders and Interests:
Customer:
-guest person that wants to check into the hotel
Clerk:
-person responsible for checking hotel guest in
using the system
System maintainer:
-person responsible for system working

Precondition: Clerk is registered in to the system
PostCondition: Hotel guest is checked into the system

Main success scenario:
1. The hotel clerk is logged into the system and it is working
2. Hotel guest asks to check into the hotel
3. The hotel clerk takes their id and searches their name in the system
4. The hotel clerk asks the guest if what they requested on the reservation is correct
5. The guest confirms that what the hotel clerk said is what they requested
6. The clerk checks them into the system and matches them with their room

Alternate paths:
1. a. the clerk cannot log into the system
    1. clerk will call manager
    2. clerk will make a manual check in

1. b. the system is not responding
    1. clerk will call manager
    2. clerk will make a manual reservation
3 a. the guest is not in the system
    1. the clerk confirms the booking was made
    2. clerk calls the manager
    3. clerk checks the guest into the hotel again and makes new reservation
5 a. the guest wants to change what they requested/check in early
    1. the clerk sees if the request can be made based on room availability
5 b. the guest claims that they didn't request what is in the system
    1. clerk calls manager

ID: Check out a customer
Scope: Hotel Reservation System
Level: Clerk
Stakeholders and Interests:
Customer:
-guest person that wants to check out of the hotel
Clerk:
-person responsible for checking hotel guest out and
using the system
System maintainer:
-person responsible for system working

Precondition: Clerk is registered into the system
Guest is registered into the system
PostCondition: Hotel guest is checked out of the system

Main success scenario:
1. The hotel clerk is logged into the system and it is working
2. Hotel guest asks to check out of the hotel
3. The hotel clerk takes their id and searches their name in the system
4. The hotel clerk asks the guest if their checkout information is correct
5. The guest confirms that the information is correct
6. The clerk checks them out of the hotel

Alternate paths:
1. a. the clerk cannot log into the system
    1. clerk will call manager
    2. clerk will make a manual check out
1. b. the system is not responding

1. clerk will call manager
2. clerk will make a manual check out

5 a. the guest does not agree with the checkout information
1. Manager is called over
2. clerk modifies information

b. the guest wants to extend their stay
1. clerk looks at available rooms to see if the request can be made

c. guest wants to checkout early
1. guest checkout info is updated

6. a. the system is not responding
1. manager is called over
2. guest is manually billed their checkout information

ID: set room status
Scope: Hotel Reservation System
Level: Clerk
Stakeholders and Interests:
Customer:
-guest that wants their room set
Clerk:
-person responsible for setting room status
using the system
System maintainer:
-person responsible for system working

Precondition: Clerk is registered into the system
PostCondition: Hotel guest is matched with a room

Main success scenario:
1. The hotel clerk is logged into the system and it is working
2. Hotel guest asks to check into the hotel
3. The hotel clerk takes their id and searches their name in the system
4. The hotel clerk sets the room that they want

Alternate paths:
1. a. the clerk cannot log into the system
1. clerk will call manager
2. clerk will make a manual room set

1. b. the system is not responding
    1. clerk will call manager
    2. clerk will make a manual room set
3. a. guest is not in system
    1. guest information is entered into system by clerk and a new reservation is made
4 a. the set system is not working
    1. manager is called over

Harrison Hassler Operation Contracts

Contract C00: CheckIn
Operation: CheckIn(guest)
Cross References: Use Cases: Check In
Pre-Conditions: The guest has a reservation
Post Conditions:
A guest was associated with a room
The guest has a check-out time

Contract C01: login
Operation: login(userName, password)
Cross References: Use Cases: Check In, Check Out
Pre-Conditions: The clerk has a login account already made
Post Conditions:
The clerk is logged into the system

Contract C02: searchGuest
Operation: searchGuest(name)
Cross References: Use Cases: Check In, Check Out
Pre-Conditions: The guest has made a reservation
The clerk is logged into the system
Post Conditions:
The guests reservation information is returned to the clerk

Contract C03: newReservation
Operation: newReservation(guestInfo)
Cross References: Use Cases: Check In, Check Out
Pre-Conditions: The clerk is logged into the system
Post Conditions:

A guest reservation is made and the availability of the
request is returned to the clerk

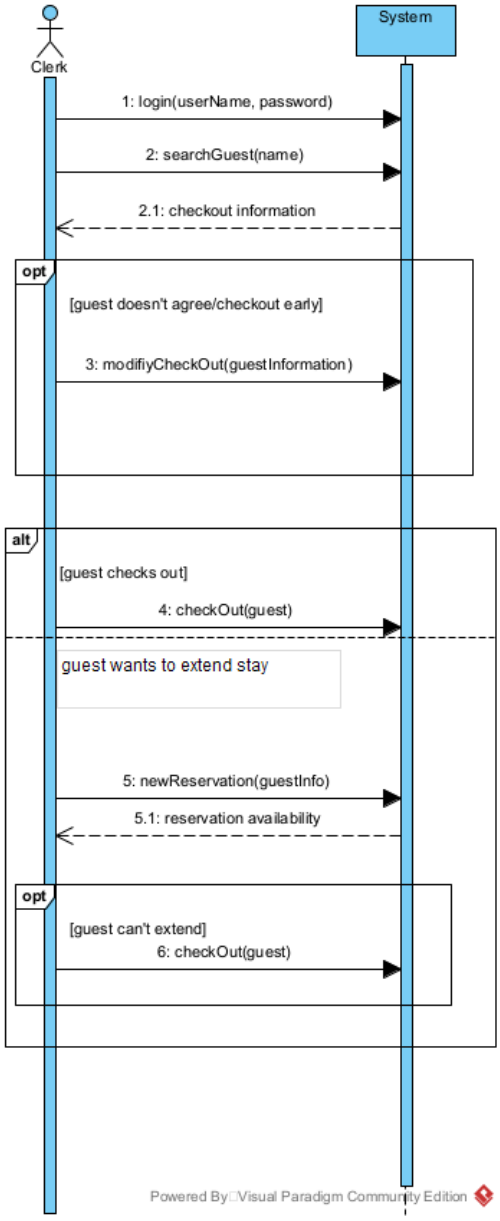
Contract C04: CheckOut
Operation CheckOut(guest)
Cross References: Use Cases: Check Out
Pre-Conditions: The guest was checked into a room at the hotel
Post Conditions:
A guest is disassociated with a room
The room is freed up
The guest's check-out time is complete

Contract C05: modifyCheckOut
Operation modifyCheckOut(guest information)
Cross References: Use Cases: Check Out
Pre-Conditions: The guest was checked into a room at the hotel
Post Conditions:
The guests checkout information is changed

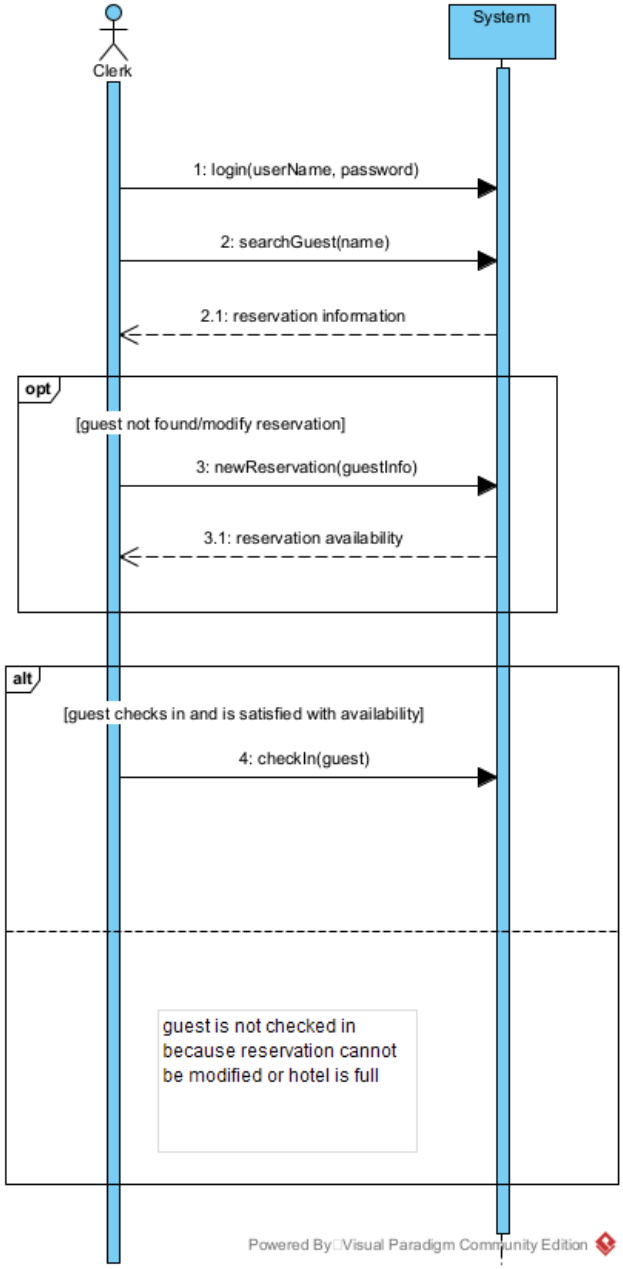
Contract C06: setRoom
Operation setRoom(name)
Cross References: Use Cases: set room status
Pre-Conditions: The guest has a reservation
Post Conditions:
The guests room is set based on their needs

Harrison Hassler System Sequence Diagrams:

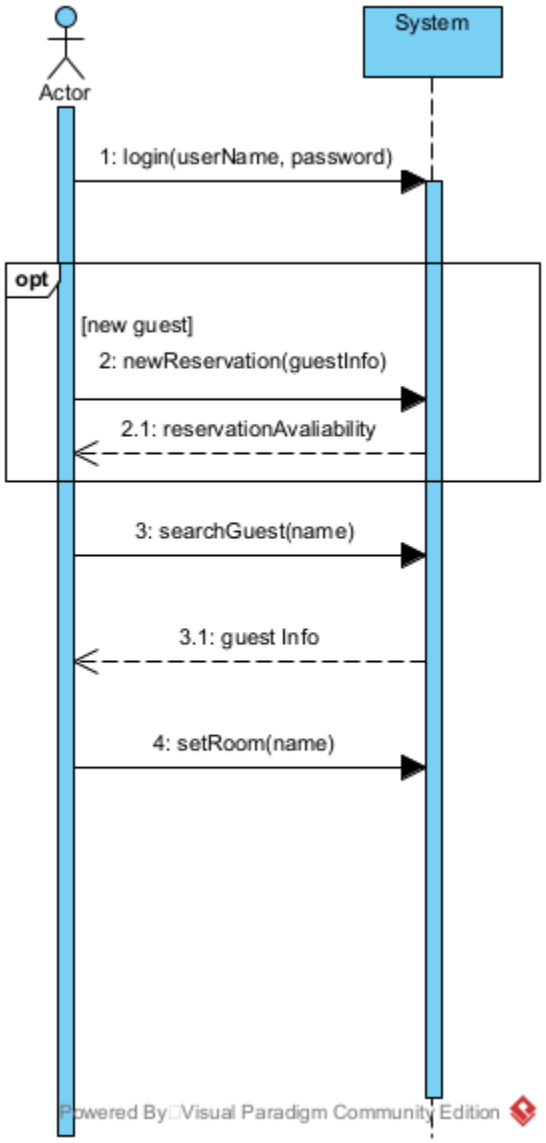
Use Case: Check-Out

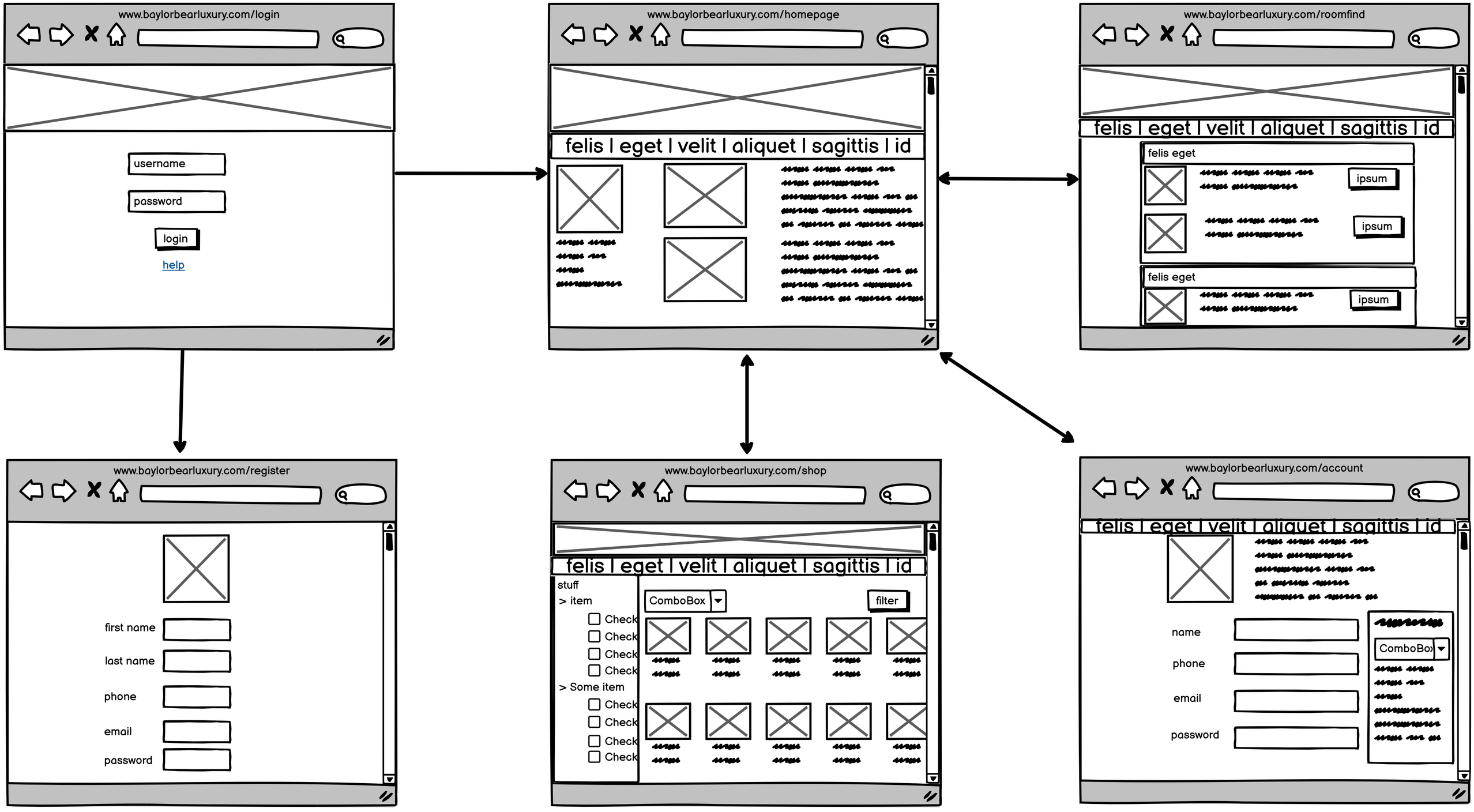

Clerk
System
1: login(userName, password)
2: searchGuest(name)
2.1: checkout information
opt
[guest doesn't agree/checkout early]
3: modifiyCheckOut(guestInformation)
alt
[guest checks out]
4: checkOut(guest)
guest wants to extend stay
5: newReservation(guestInfo)
5.1: reservation availability
opt
[guest can't extend]
6: checkOut(guest)
Powered By Visual Paradigm Community Edition

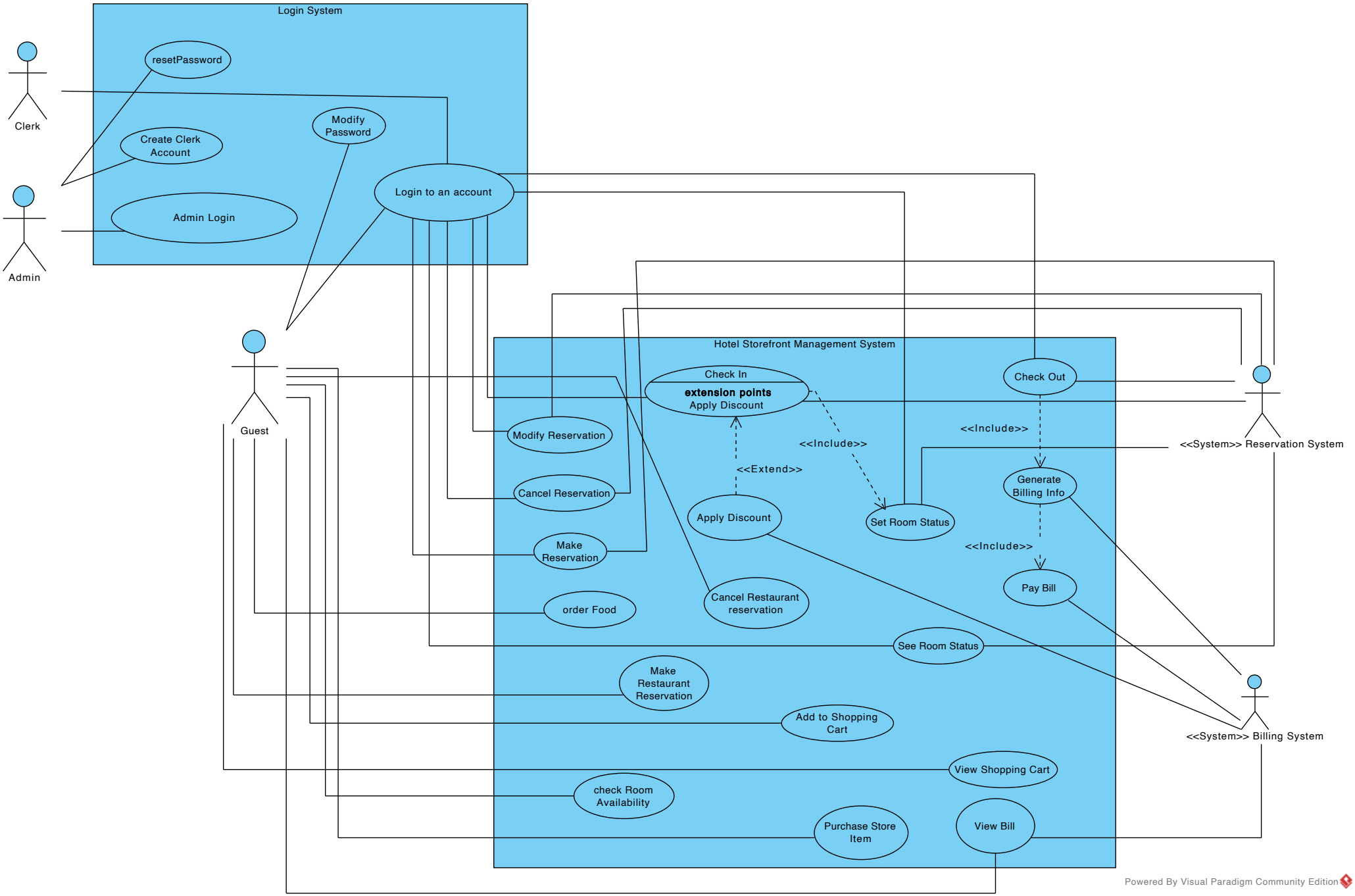
Use Case: Check-In

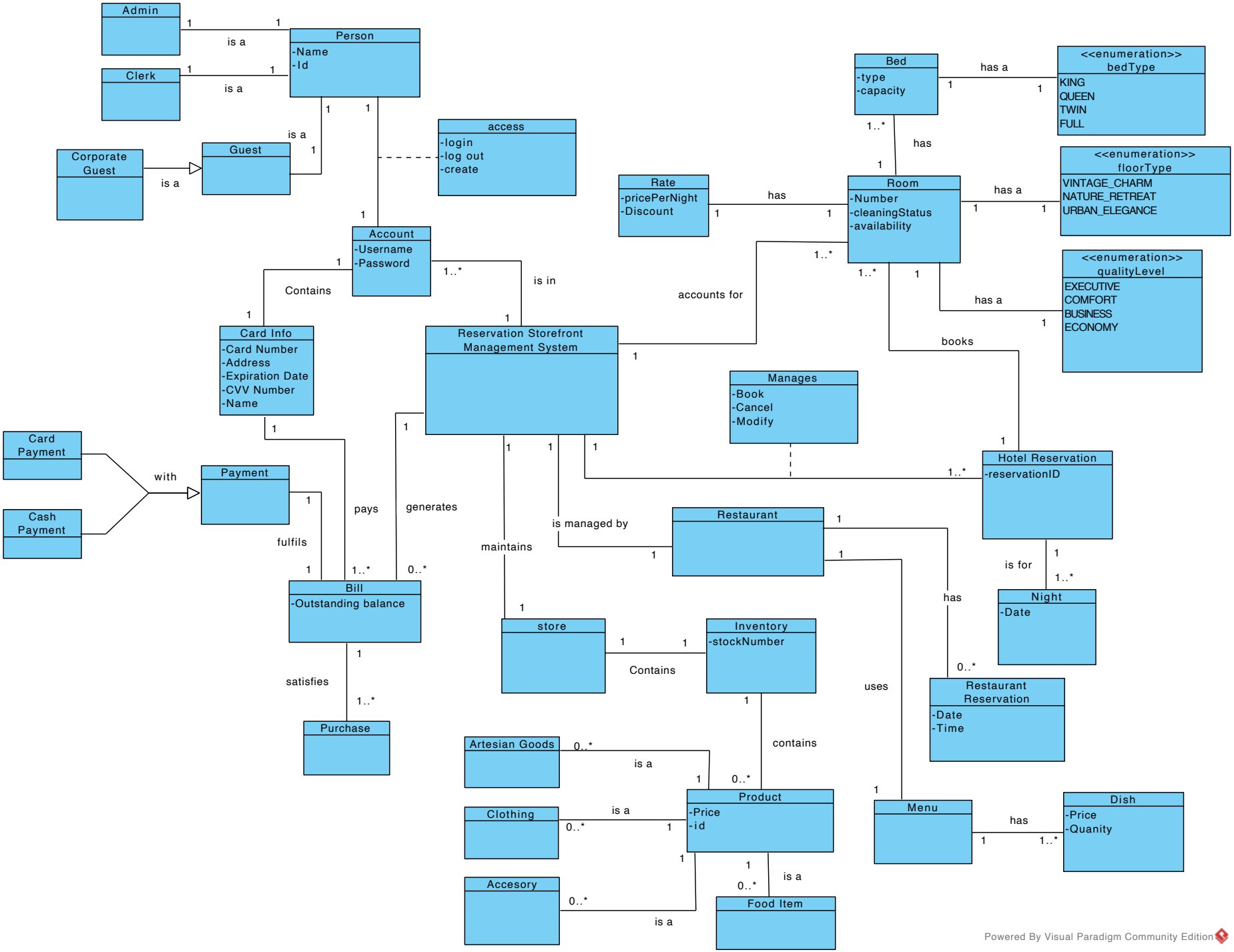


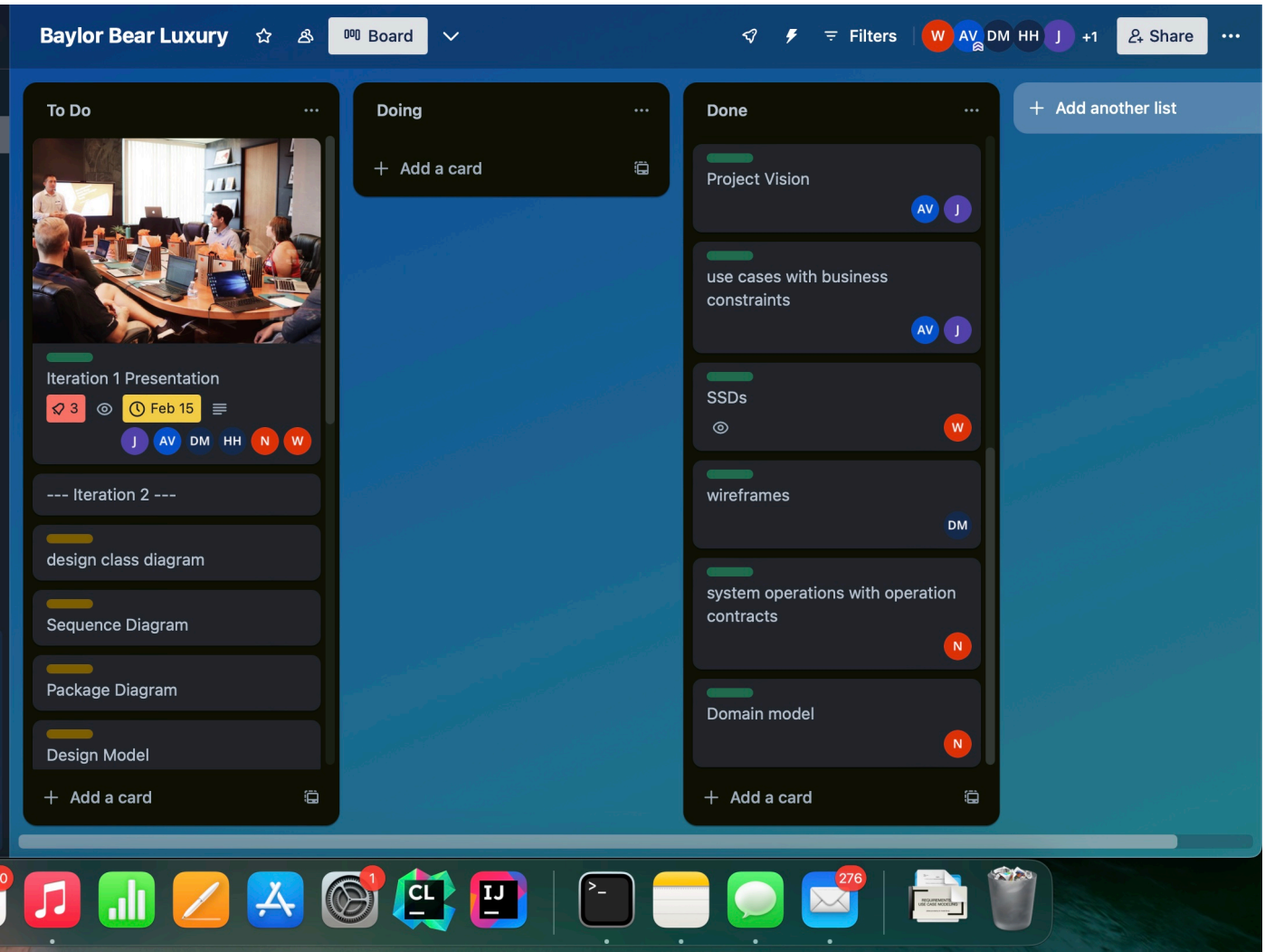
Use Case: set room status




www.baylorbearluxury.com/login
username
password
login
help
www.baylorbearluxury.com/homepage
felis | eget | velit | aliquet | sagittis | id
www.baylorbearluxury.com/roomfind
felis | eget | velit | aliquet | sagittis | id
felis eget
ipsum
ipsum
felis eget
ipsum
www.baylorbearluxury.com/register
first name
last name
phone
email
password
www.baylorbearluxury.com/shop
felis | eget | velit | aliquet | sagittis | id
stuff
> item
Check
Check
Check
Check
> Some item
Check
Check
Check
Check
ComboBox
filter
www.baylorbearluxury.com/account
felis | eget | velit | aliquet | sagittis | id
name
phone
email
password
ComboBox


Login System
resetPassword
Modify Password
Create Clerk Account
Login to an account
Admin Login
Clerk
Admin
Guest
Hotel Storefront Management System
Check In
extension points
Apply Discount
Check Out
Modify Reservation
Cancel Reservation
Make Reservation
order Food
Cancel Restaurant reservation
Apply Discount
Set Room Status
Generate Billing Info
Pay Bill
See Room Status
Make Restaurant Reservation
Add to Shopping Cart
View Shopping Cart
check Room Availability
Purchase Store Item
View Bill
<<Include>>
<<Extend>>
<<Include>>
<<Include>>
<<System>> Reservation System
<<System>> Billing System
Powered By Visual Paradigm Community Edition


Admin
Person
-Name
-Id
Clerk
is a
Corporate Guest
Guest
access
-login
-log out
-create
Account
-Username
-Password
Contains
is in
Card Info
-Card Number
-Address
-Expiration Date
-CVV Number
-Name
Reservation Storefront Management System
Bed
-type
-capacity
has a
<<enumeration>> bedType
KING
QUEEN
TWIN
FULL
has
Rate
-pricePerNight
-Discount
Room
-Number
-cleaningStatus
-availability
<<enumeration>> floorType
VINTAGE_CHARM
NATURE_RETREAT
URBAN_ELEGANCE
accounts for
<<enumeration>> qualityLevel
EXECUTIVE
COMFORT
BUSINESS
ECONOMY
books
Manages
-Book
-Cancel
-Modify
Hotel Reservation
-reservationID
Card Payment
Cash Payment
with
Payment
pays
generates
fulfils
Bill
-Outstanding balance
maintains
is managed by
Restaurant
is for
Night
-Date
has
Restaurant Reservation
-Date
-Time
uses
store
Inventory
-stockNumber
Contains
satisfies
Purchase
Artesian Goods
Clothing
Accesory
Product
-Price
-id
contains
Food Item
Menu
Dish
-Price
-Quanity
Powered By Visual Paradigm Community Edition


Baylor Bear Luxury
Board
Filters
W AV DM HH J +1
Share
To Do
Doing
Done
Add another list
Add a card
Iteration 1 Presentation
3
Feb 15
J AV DM HH N W
--- Iteration 2 ---
design class diagram
Sequence Diagram
Package Diagram
Design Model
Add a card
Project Vision
AV J
use cases with business constraints
AV J
SSDs
W
wireframes
DM
system operations with operation contracts
N
Domain model
N
Add a card