Team: Akshit Shah
Gaurav Gandhi
Sarang Kulkarni

Title: Buff-Beds

Summary: The Project focuses on implementing an online interface to provide users a platform to make their holiday preparation an easy experience. Our project aims to provide a portal to book rooms for their favourite destination. The system will contain many of the functions like searching for hotels and checking out the rooms. There are three Access levels Admin, Customer and Hosts.

All the actors will have to login to the system. Depending on their accounts they will have corresponding functions they can access. Once login in as a customer, it can search for hotels, view rooms and proceed to checkout or forfeit. The host can approve or decline the booking request. The host can also post ads for its property. There is also an additional option for hosts to pay more to admins for sponsored ads. The admin can approve hosts only after background check and it is responsible for smooth checkout and refunds. Lastly, customer and hosts can give feedbacks.

Data Storage: We will be using MySQL as our data storage tool. There will be separate tables for users, hotels and orders

Project Requirements:

Business Requirements: There are no business requirements

User Requirements:

ID	**Requirement**	**Topic Area**	**Actor**	**Priority**
UR-01	Customer can sign up.	Login	Customer, Host	Critical
UR-02	Host can Apply for hotel/SignUp	Login	Host, Admin	Critical
UR - 03	End Users can log in.	Login	Customer, Hosts,Admin	Critical
UR – 04	Admin can block both hosts and Customers	Login	Admin	High
UR – 05	Customer can search and sort hotels or rooms	Database	Customer	High
UR – 06	Customer can check out the hotel/room	Orders	Customer	High

UR – 07	Customer cancel the booking	Orders	Customer	High
UR – 08	Customer can request addons to their orders	Orders	Customer, Host	Low
UR – 09	Hosts can upgrade their account	Database	Hosts	High
UR - 10	Host can update the info	Profile	Host	Medium
UR - 11	End User can rate and give feedbacks	Database	Customer, Host	Medium
UR – 12	Admin can approve request for new hosts/request for update information	Database	Admin	High
UR - 13	Admin can approve a refund	Orders	Admin	Medium

Non-Functional Requirements:

ID	Requirement	Topic Area	Priority
NR-01	The system must be backed up daily and database must be timely updated	Reliability	Critical
NR-02	Customer and Hosts can access functions corresponding to their access levels	Security	Critical
NR-03	The system must be up and running 24*7	Availability	Critical
NR-04	Only verified hosts should be able to add account	Legal	Critical
NR-05	The system should respond within 4 seconds	Responsiveness	High

Use Case Documents

Use case ID	UR - 01
Use Case Name	Customer can sign up.
Description	Customers can create an account

<table>
<tr><td><b>Actors</b></td><td>Customer</td></tr>
<tr><td><b>Pre Condition</b></td><td>System must be up and running</td></tr>
<tr><td><b>Post Condition</b></td><td>Database will be updated with new details of customer.</td></tr>
<tr><td><b>Frequency of Use</b></td><td>Once for every new customer</td></tr>
<tr><td><b>Flow of Events</b></td><td>
<table>
<tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Click on Sign up button</td><td>Show signup interface</td></tr>
<tr><td>2</td><td>Fill in all the details</td><td></td></tr>
<tr><td>3</td><td>Check on ‘Agree terms of use’</td><td></td></tr>
<tr><td>4</td><td>Click on Signup button</td><td>Submit an email to verify the email address</td></tr>
</table>
</td></tr>
<tr><td><b>Variations</b></td><td></td></tr>
<tr><td><b>Exceptions</b></td><td>2.a Email already exists
2.b User did not fill all the details
3.a User did not check ‘Agree terms of use’</td></tr>
<tr><td><b>Developer Notes</b></td><td></td></tr>
</table>

Use case ID	UR - 02
Use Case Name	Host can Apply for hotel/SignUp
Description	Host can create an account and Admin will approve the host after background check
Actors	Host, Admin
Precondition	System must be up and running
Post Condition	Database will be updated with new details of Hotel info and Host.
Frequency of Use	Once for every new hotel

<table>
<tr><td><b>Flow of Events</b></td><td>
<table>
<tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Click on Sign up button</td><td>Show signup interface</td></tr>
<tr><td>2</td><td>Fill in all the Hotel details</td><td></td></tr>
<tr><td>3</td><td>Check on ‘Agree terms of use’</td><td></td></tr>
<tr><td>4</td><td>Click on Signup button</td><td>Submit an email to verify the email address</td></tr>
<tr><td>5</td><td></td><td>The System will generate a request to admin for approval of Host</td></tr>
<tr><td>6</td><td></td><td>Admin will approve or deny the request and notify the host.</td></tr>
</table>
</td></tr>
<tr><td><b>Variations</b></td><td></td></tr>
<tr><td><b>Exceptions</b></td><td>2.a Email already exists
2.b User did not fill all the details
2.c Duplicate Addresses
3.a User did not check ‘Agree terms of use’
6.a Background check failed</td></tr>
<tr><td><b>Developer Notes</b></td><td></td></tr>
</table>

Use case ID	UR - 03
Use Case Name	End Users can log in
Description	Customer and Host can login using their registered email ID
Actors	Customer, Hosts
Pre-Condition	User should have signed up using a valid email ID
Post Condition	User is logged in into the system with default account view
Frequency of Use	Often

<table>
<tr><td><b>Flow of Events</b></td><td><table>
<tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>User enters registered email, password</td><td></td></tr>
<tr><td>2</td><td>User Clicks on Login</td><td>System checks if correct email-id is entered from database</td></tr>
<tr><td>3</td><td></td><td>System checks if email id for admin/host/customer</td></tr>
<tr><td>4</td><td></td><td>Default account Homepage is displayed based on account type</td></tr>
</table></td></tr>
<tr><td><b>Variations</b></td><td></td></tr>
<tr><td><b>Exceptions</b></td><td>1.a User enters wrong email ID
1.b User enters wrong password</td></tr>
<tr><td><b>Developer Notes</b></td><td></td></tr>
</table>

Use case ID	UR - 04
Use Case Name	Admin can block both hosts and Customers
Description	Admin can block the customer or host account, if he/she receives 3 red flags about customer/host.
Actors	Admin
Pre condition	Admin has received red flags about a host/customer
Post Condition	Customer/Host cannot access its account
Frequency of Use	Whenever admin receives 3 flags about customer/host

<table>
<tr><td><b>Flow of Events</b></td><td>
<table>
<tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Admin logs into the system</td><td>System displays flagged users on the homescreen</td></tr>
<tr><td>2</td><td>Admin clicks on the user from the list</td><td></td></tr>
<tr><td>3</td><td>Admin clicks on the ‘Block’ button</td><td>System Blocks the login of the user and displays success message on the screen</td></tr>
</table>
</td></tr>
<tr><td><b>Variations</b></td><td></td></tr>
<tr><td><b>Exceptions</b></td><td></td></tr>
<tr><td><b>Developer Notes</b></td><td></td></tr>
</table>

<table>
<tr><td><b>Use case ID</b></td><td>UR - 05</td></tr>
<tr><td><b>Use Case Name</b></td><td>Customer can search and sort hotels or rooms</td></tr>
<tr><td><b>Description</b></td><td>Customer will search for rooms and enter preferences and sort from database</td></tr>
<tr><td><b>Actors</b></td><td>Customer</td></tr>
<tr><td><b>Pre Condition</b></td><td>User should have a valid account</td></tr>
<tr><td><b>Post Condition</b></td><td>Display search results</td></tr>
<tr><td><b>Frequency of Use</b></td><td>Frequently used by users</td></tr>
<tr><td><b>Flow of Events</b></td><td>
<table>
<tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Click on Login button</td><td>Display Search page</td></tr>
<tr><td>2</td><td>Enter the preferences</td><td>Display the search results</td></tr>
<tr><td>3</td><td>Click on the hotel you liked</td><td>Display the properties of the hotel</td></tr>
</table>
</td></tr>
<tr><td><b>Variations</b></td><td>2.a Suggest the preferences based on previous history of bookings
2.b Use drop down menu</td></tr>
</table>

Exceptions	3.a No results based on added preferences
Developer Notes	

<table>
<tr><td><b>Use case ID</b></td><td>UR - 06</td></tr>
<tr><td><b>Use Case Name</b></td><td>Customer can checkout the hotel/room</td></tr>
<tr><td><b>Description</b></td><td>After searching for desired hotels through the filters customer can book the hotel/room that he/she wants</td></tr>
<tr><td><b>Actors</b></td><td>Customer</td></tr>
<tr><td><b>Precondition</b></td><td>Customer should have an account with the system</td></tr>
<tr><td><b>Post Condition</b></td><td>Customer gets the booking in the place that he/she wanted on a particular date</td></tr>
<tr><td><b>Frequency of Use</b></td><td>Often</td></tr>
<tr><td><b>Flow of Events</b></td><td>
<table>
<tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Customer clicks on Book Now button</td><td>System Displays amount due</td></tr>
<tr><td>2</td><td>Customer clicks on the pay now button</td><td>System navigates to the payment page</td></tr>
<tr><td>3</td><td>User Enters Payment Information</td><td>System accepts the payment and prompts user to enter emergency contact details</td></tr>
<tr><td>4</td><td>Customer enters Emergency contact details</td><td>System displays Success Message</td></tr>
</table>
</td></tr>
<tr><td><b>Variations</b></td><td></td></tr>
<tr><td><b>Exceptions</b></td><td>4.a Payment fails then system tells host to enter different payment information</td></tr>
<tr><td><b>Developer Notes</b></td><td></td></tr>
</table>

<table>
<tr><td><b>Use case ID</b></td><td>UR - 07</td></tr>
<tr><td><b>Use Case Name</b></td><td>Customer cancel the booking</td></tr>
<tr><td><b>Description</b></td><td>Booking cancellation process for user</td></tr>
<tr><td><b>Actors</b></td><td>Customer</td></tr>
<tr><td><b>Pre Condition</b></td><td>Customer should have a valid booking id</td></tr>
<tr><td><b>Post Condition</b></td><td>Booking will be cancelled and database will be update</td></tr>
<tr><td><b>Frequency of Use</b></td><td>Occasionally</td></tr>
<tr><td><b>Flow of Events</b></td><td>
<table>
<tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Click on my bookings</td><td>Display booking details</td></tr>
<tr><td>2</td><td>Click on cancel booking</td><td>Notifies admin and host about cancellation request</td></tr>
<tr><td>3</td><td></td><td>It awaits admin’s approval and update database</td></tr>
</table>
</td></tr>
<tr><td><b>Variations</b></td><td></td></tr>
<tr><td><b>Exceptions</b></td><td>3.a Check for user history if it repeats the cancellation</td></tr>
<tr><td><b>Developer Notes</b></td><td></td></tr>
</table>

Use case ID	UR - 08
Use Case Name	Customer can request addons to their orders
Description	Customer can add breakfast/dinner options, request extra bed or paid amenities
Actors	Customer
Pre Condition	User should have proceeded to hotel checkout
Post Condition	Display add on options

<table>
<tr><td><b>Frequency of Use</b></td><td>Frequently used by users</td></tr>
<tr><td><b>Flow of Events</b></td><td>
<table>
<tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Click on next button after entering rooms details for hotel</td><td>Display Add on options</td></tr>
<tr><td>2</td><td>Enter the add on customer wants to buy</td><td>Add it to final booking</td></tr>
<tr><td>3</td><td>Click on the hotel you liked</td><td>Display the properties of the hotel</td></tr>
</table>
</td></tr>
<tr><td><b>Variations</b></td><td></td></tr>
<tr><td><b>Exceptions</b></td><td></td></tr>
<tr><td><b>Developer Notes</b></td><td></td></tr>
</table>

Use case ID	UR - 09
Use Case Name	Hosts can upgrade their account
Description	Hosts can pay more and upgrade their account to advertise their rooms and hotels with more preference.
Actors	Hosts
Precondition	Host have opened their account with valid email address
Post Condition	Account will be upgraded and their hotel/room will be given preference in advertising
Frequency of Use	Once for every host

<table>
<tr><td><b>Flow of Events</b></td><td>
<table>
<tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Host clicks on the “Upgrade now” button</td><td>System navigates the user to information button with payment button</td></tr>
<tr><td>2</td><td>Hosts clicks on the payment</td><td>System gives different payment timelines</td></tr>
<tr><td>3</td><td>Host selects the timeline for which it wants upgraded account</td><td>System gives the due amount</td></tr>
<tr><td>4</td><td>Host enters payment information and clicks on pay now</td><td>Host allocate premium account to the host</td></tr>
</table>
</td></tr>
<tr><td><b>Variations</b></td><td></td></tr>
<tr><td><b>Exceptions</b></td><td>4.a Payment fails then system tells host to enter different payment information</td></tr>
<tr><td><b>Developer Notes</b></td><td></td></tr>
</table>

Use case ID	UR - 10
Use Case Name	Host can update the information
Description	Host can update hotel information.
Actors	Host
Pre-Condition	Host have opened their account with valid email address
Post Condition	Hotel info will be updated in the database
Frequency of Use	Occasionally

<table>
<tr><td><strong>Flow of Events</strong></td><td>
<table>
<tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Host clicks on update info</td><td>Hotel info is displayed</td></tr>
<tr><td>2</td><td>Hotel enters all the information changes</td><td>Request is generated for admin to approve it</td></tr>
<tr><td>3</td><td></td><td>Once approved it is updated in the database and host is notified</td></tr>
</table>
</td></tr>
<tr><td><strong>Variations</strong></td><td></td></tr>
<tr><td><strong>Exceptions</strong></td><td></td></tr>
<tr><td><strong>Developer Notes</strong></td><td></td></tr>
</table>

Use case ID	UR - 11
Use Case Name	End User can rate and give feedbacks
Description	After the booking is complete and customer has used the booking then customer & host can post feedback about each other
Actors	Customer, Hosts, Admin
Pre-Condition	Customer has successfully completed and used the booking
Post Condition	Admin verifies the authenticity and posts the review
Frequency of Use	Once Every booking

<table>
<tr><td><b>Flow of Events</b></td><td>
<table>
<tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>End User clicks on his past booking</td><td>System Displays navigates user to past bookings page</td></tr>
<tr><td>2</td><td>End User clicks on the “feedback option”</td><td>System navigates the user to the feedback form</td></tr>
<tr><td>3</td><td>End User writes subject and details of the feedback</td><td></td></tr>
<tr><td>4</td><td>End User Gives rating to the host/customer and flag it if there were some serious problems</td><td></td></tr>
<tr><td>5</td><td>End User clicks on the Submit</td><td>System Takes the feedback and gives it to Admin</td></tr>
<tr><td>6</td><td></td><td>Admin approves the feedback</td></tr>
<tr><td>7</td><td></td><td>System stores the feedback in the database and posts the feedback on the website</td></tr>
</table>
</td></tr>
<tr><td><b>Variations</b></td><td>6.a Admin Rejects the feedback if it's not authentic or it contains foul language. Then system discards the feedback data</td></tr>
<tr><td><b>Exceptions</b></td><td>User doesn’t enter mandatory details in step 4 and 5</td></tr>
<tr><td><b>Developer Notes</b></td><td></td></tr>
</table>

Use case ID	UR - 12
Use Case Name	Admin can approve request for new hosts
Description	After host creates an account, admin can verify the credentials and wait for background check and approve it.
Actors	Admins
Pre Condition	Host completes the signup and admin gets a request
Post Condition	Account has been approved and added to the database or rejected
Frequency of Use	Once every new host account is created

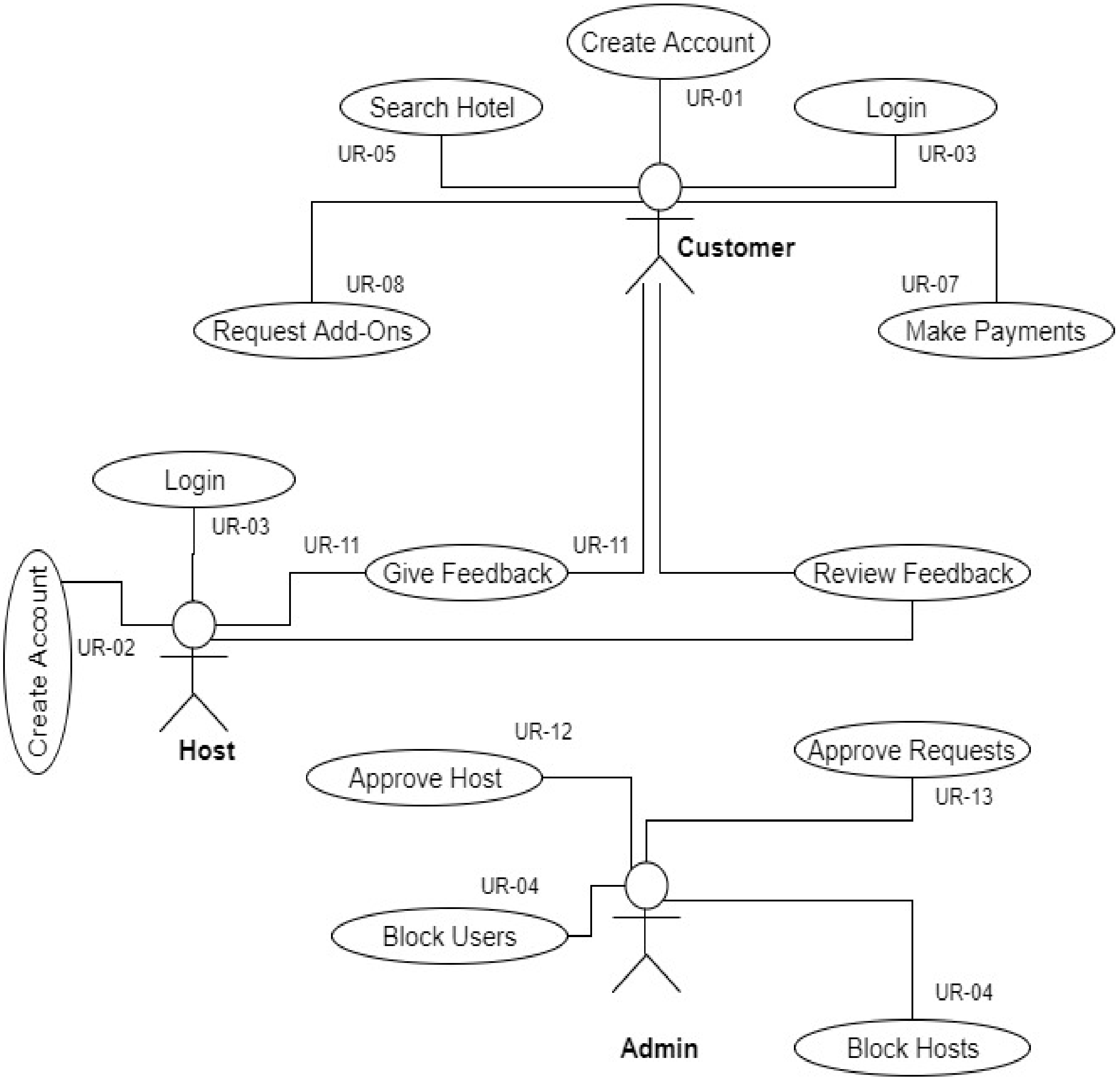
<table>
<tr><td><b>Flow of Events</b></td><td>
<table>
<tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Click on new request</td><td>Display new request details</td></tr>
<tr><td>2</td><td>Verify details</td><td></td></tr>
<tr><td>3</td><td>Click on approve or reject buttons</td><td>If approved add to database, if rejected sent a mail with reject info</td></tr>
</table>
</td></tr>
<tr><td><b>Variations</b></td><td></td></tr>
<tr><td><b>Exceptions</b></td><td>2.a Incomplete info, send host email requesting for more documents</td></tr>
<tr><td><b>Developer Notes</b></td><td></td></tr>
</table>

<table>
<tr><td><b>Use case ID</b></td><td>UR - 13</td></tr>
<tr><td><b>Use Case Name</b></td><td>Admin can approve a refund</td></tr>
<tr><td><b>Description</b></td><td>When customer requests a refund then based on the scenario (No of days before the booking) admin approves the refund</td></tr>
<tr><td><b>Actors</b></td><td>Admin</td></tr>
<tr><td><b>Precondition</b></td><td>User has cancelled the booking and applied for the refund</td></tr>
<tr><td><b>Post Condition</b></td><td>Admin approves the funding and initiates the refund</td></tr>
<tr><td><b>Frequency of Use</b></td><td>Medium</td></tr>
<tr><td><b>Flow of Events</b></td><td>
<table>
<tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1</td><td>Customer cancelled the booking and applies for the refund</td><td>System Redirects the request to the admin</td></tr>
<tr><td>2</td><td>Admin approves or declines the request based on the scenario</td><td>System Reflects the admins decision to the user</td></tr>
</table>
</td></tr>
<tr><td><b>Variations</b></td><td></td></tr>
<tr><td><b>Exceptions</b></td><td></td></tr>
</table>

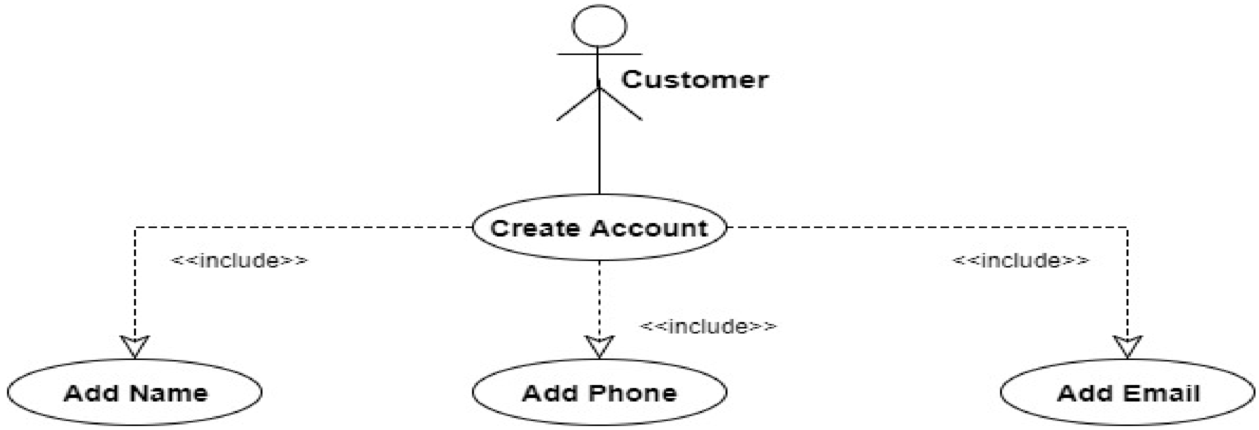
Developer Notes	

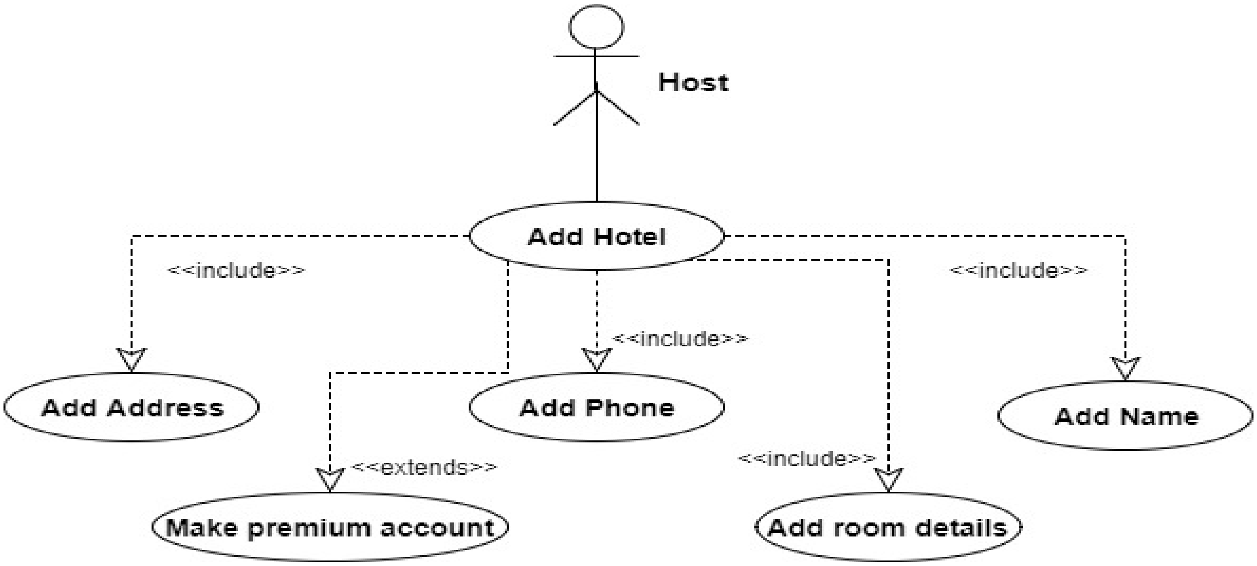
Use Cases

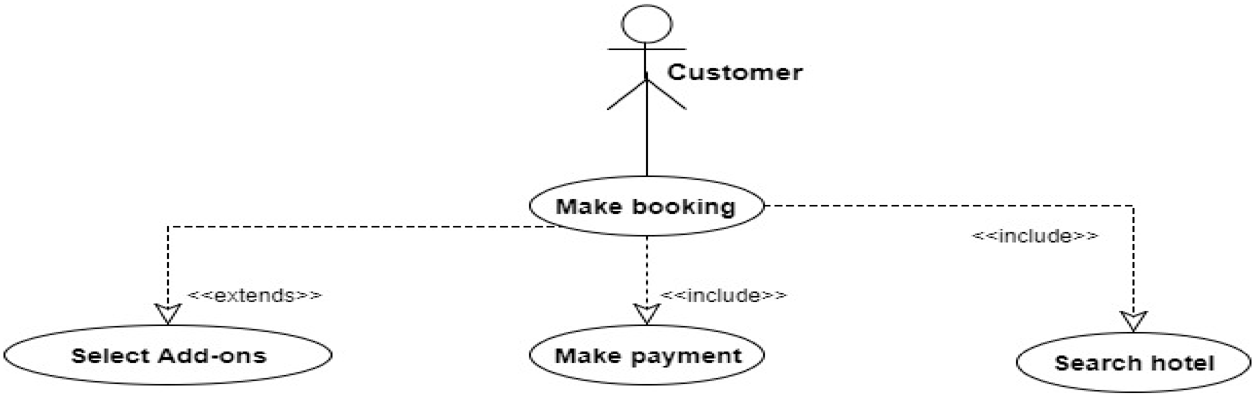
- **Use Case Overview**



- **Sub Diagrams**

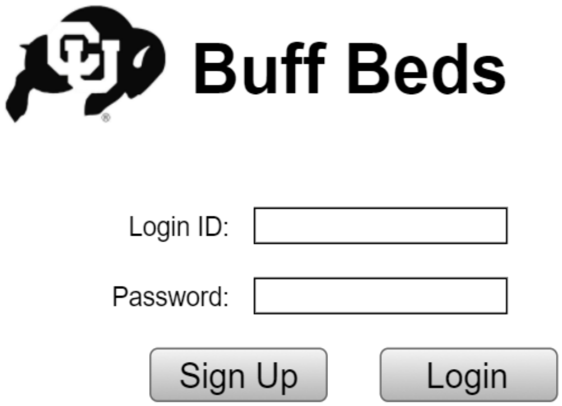




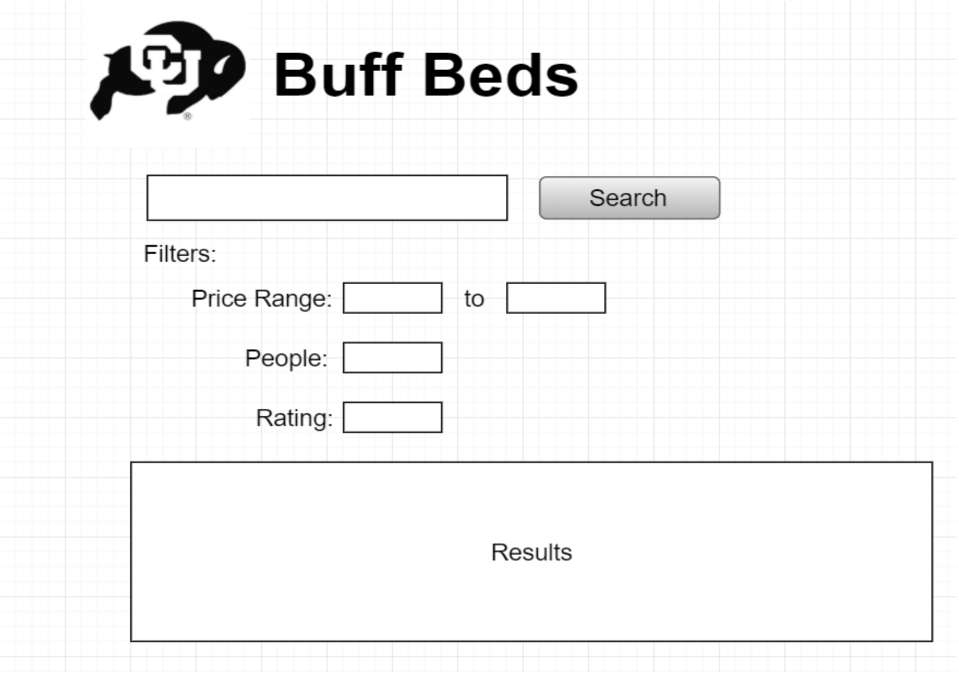


UI Mockups

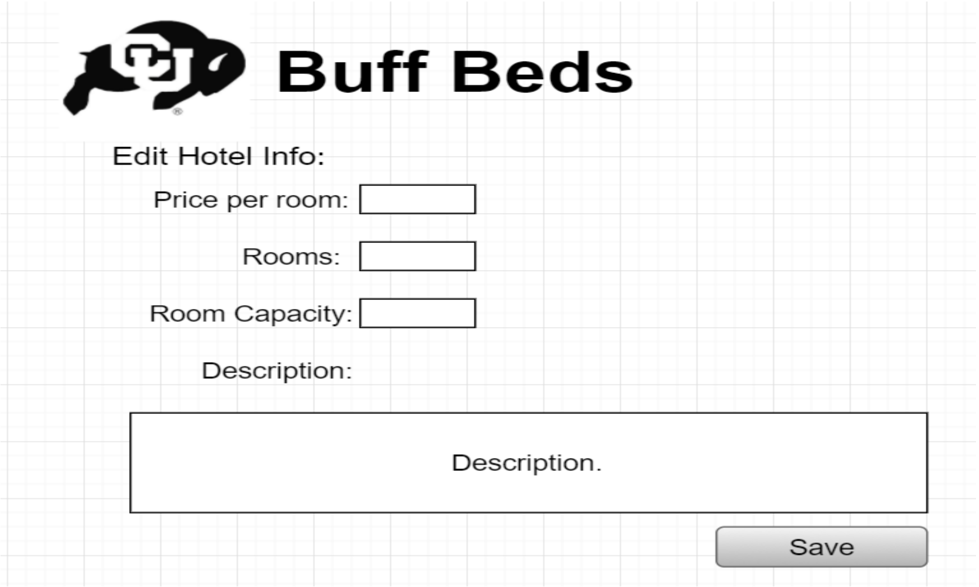
Mockup 1: Login



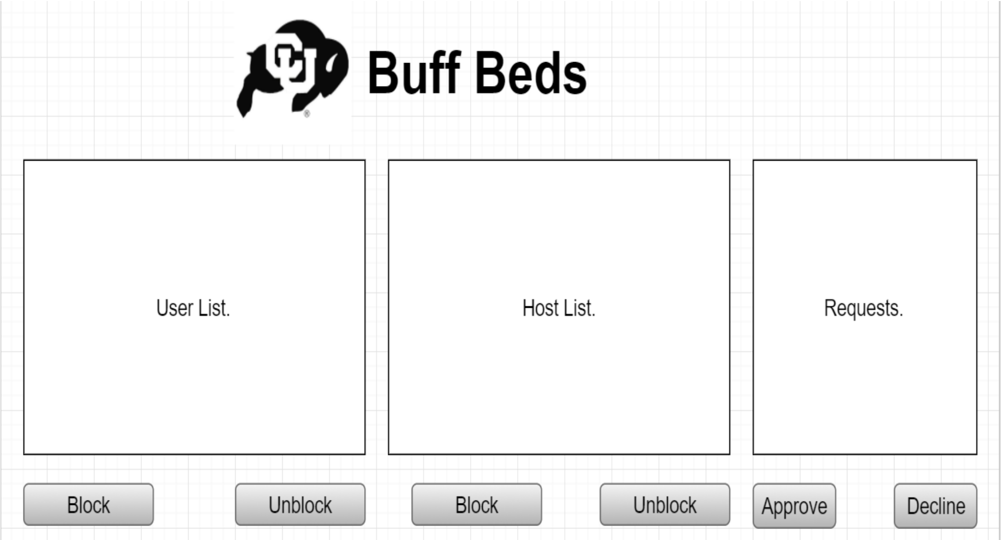
Mockup 2: Customer Search Page



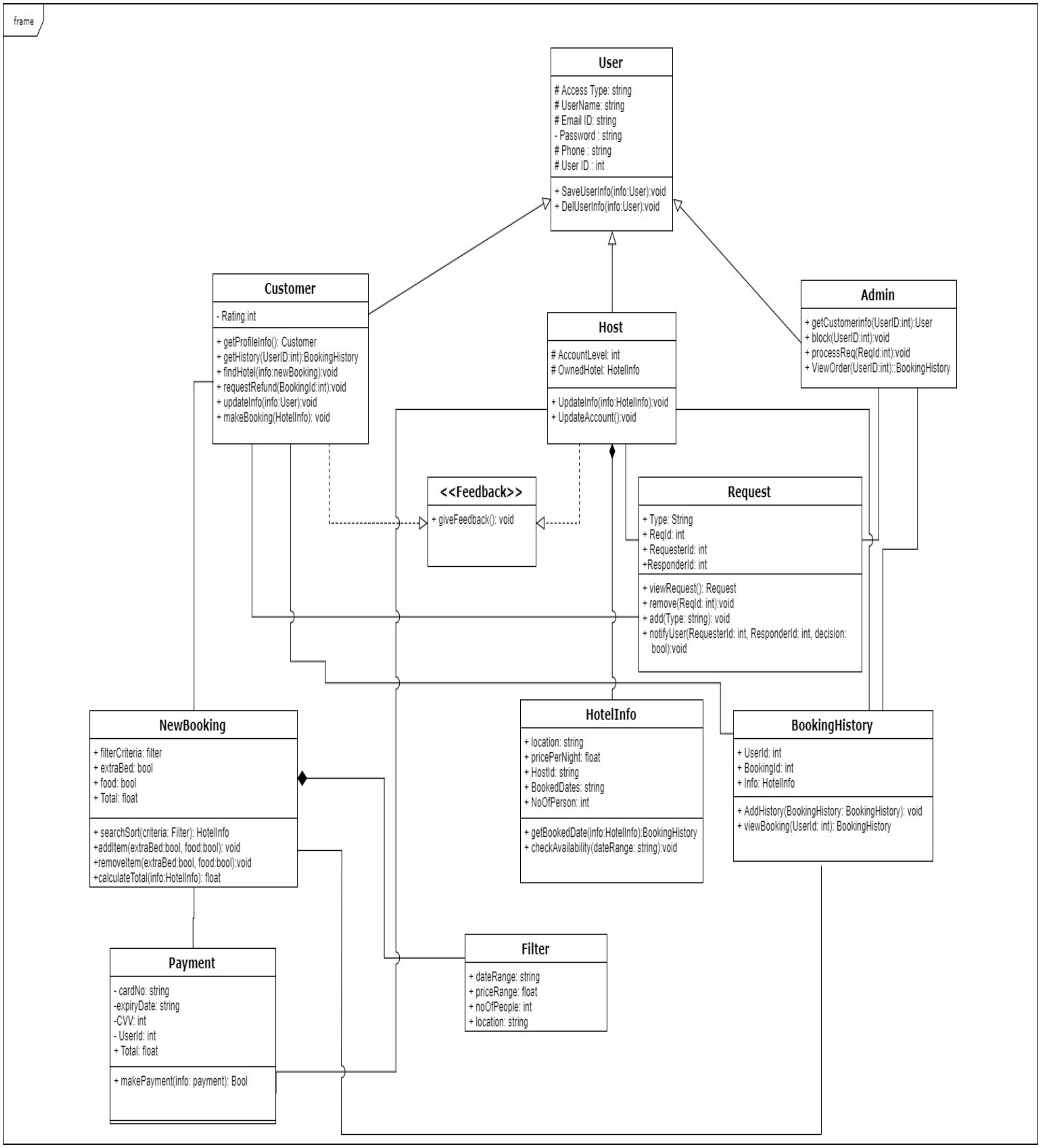
Mockup 3: Host Update Hotel info



Mockup 4: Admin View



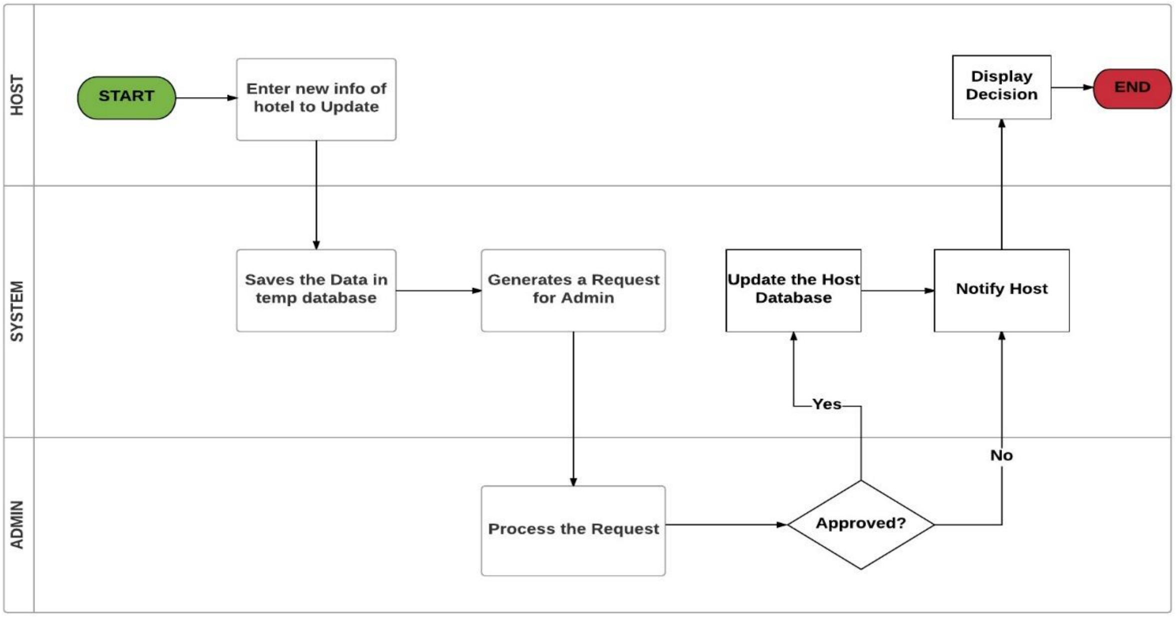
Class Diagram


frame
User
# Access Type: string
# UserName: string
# Email ID: string
- Password : string
# Phone : string
# User ID : int
+ SaveUserInfo(info:User):void
+ DelUserInfo(info:User):void
Customer
- Rating:int
+ getProfileInfo(): Customer
+ getHistory(UserID:int):BookingHistory
+ findHotel(info:newBooking):void
+ requestRefund(BookingId:int):void
+ updateInfo(info:User):void
+ makeBooking(HotelInfo): void
Host
# AccountLevel: int
# OwnedHotel: HotelInfo
+ UpdateInfo(info:HotelInfo):void
+ UpdateAccount():void
Admin
+ getCustomerinfo(UserID:int):User
+ block(UserID:int):void
+ processReq(ReqId:int):void
+ ViewOrder(UserID:int)::BookingHistory
<<Feedback>>
+ giveFeedback(): void
Request
+ Type: String
+ ReqId: int
+ RequesterId: int
+ResponderId: int
+ viewRequest(): Request
+ remove(ReqId: int):void
+ add(Type: string): void
+ notifyUser(RequesterId: int, ResponderId: int, decision: bool):void
NewBooking
+ filterCriteria: filter
+ extraBed: bool
+ food: bool
+ Total: float
+ searchSort(criteria: Filter): HotelInfo
+addItem(extraBed:bool, food:bool): void
+removeItem(extraBed:bool, food:bool):void
+calculateTotal(info:HotelInfo): float
HotelInfo
+ location: string
+ pricePerNight: float
+ HostId: string
+ BookedDates: string
+ NoOfPerson: int
+ getBookedDate(info:HotelInfo):BookingHistory
+ checkAvailability(dateRange: string):void
BookingHistory
+ UserId: int
+ BookingId: int
+ Info: HotelInfo
+ AddHistory(BookingHistory: BookingHistory): void
+ viewBooking(UserId: int): BookingHistory
Payment
- cardNo: string
-expiryDate: string
-CVV: int
- UserId: int
+ Total: float
+ makePayment(info: payment): Bool
Filter
+ dateRange: string
+ priceRange: float
+ noOfPeople: int
+ location: string

Activity Diagram

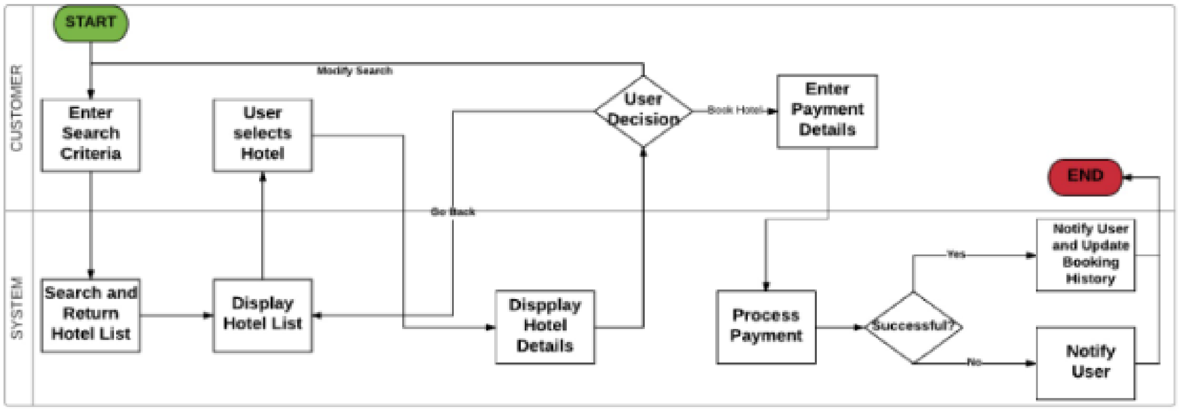
UR - 10 Host can update the Hotel info and UR - 12 Admin can approve request for request for update information

BUFF BEDS | October 25, 2017

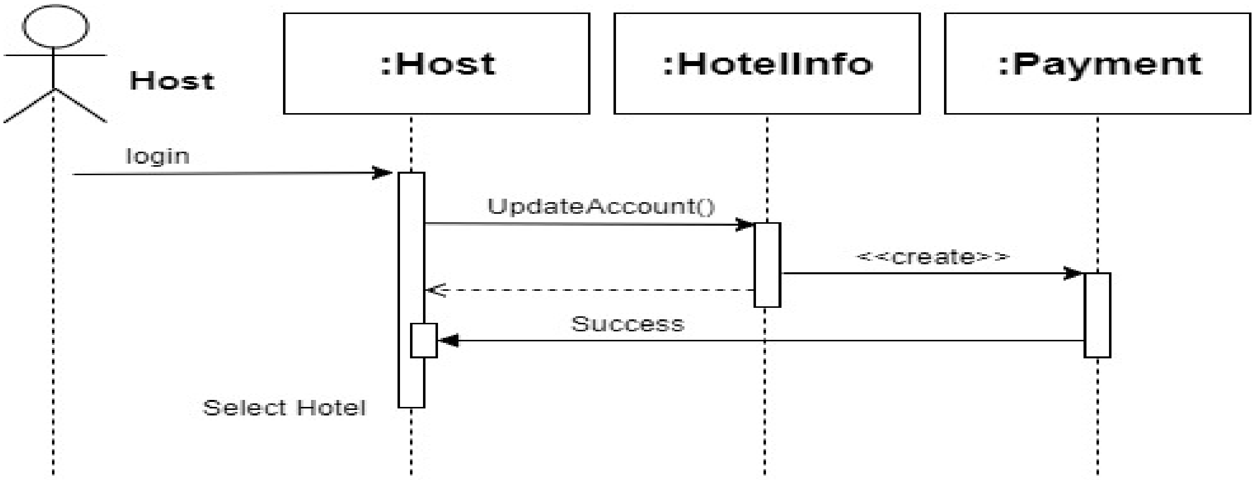


UR - 05 Customer can search and sort hotels or rooms and UR - 06 Customer can checkout the hotel/room

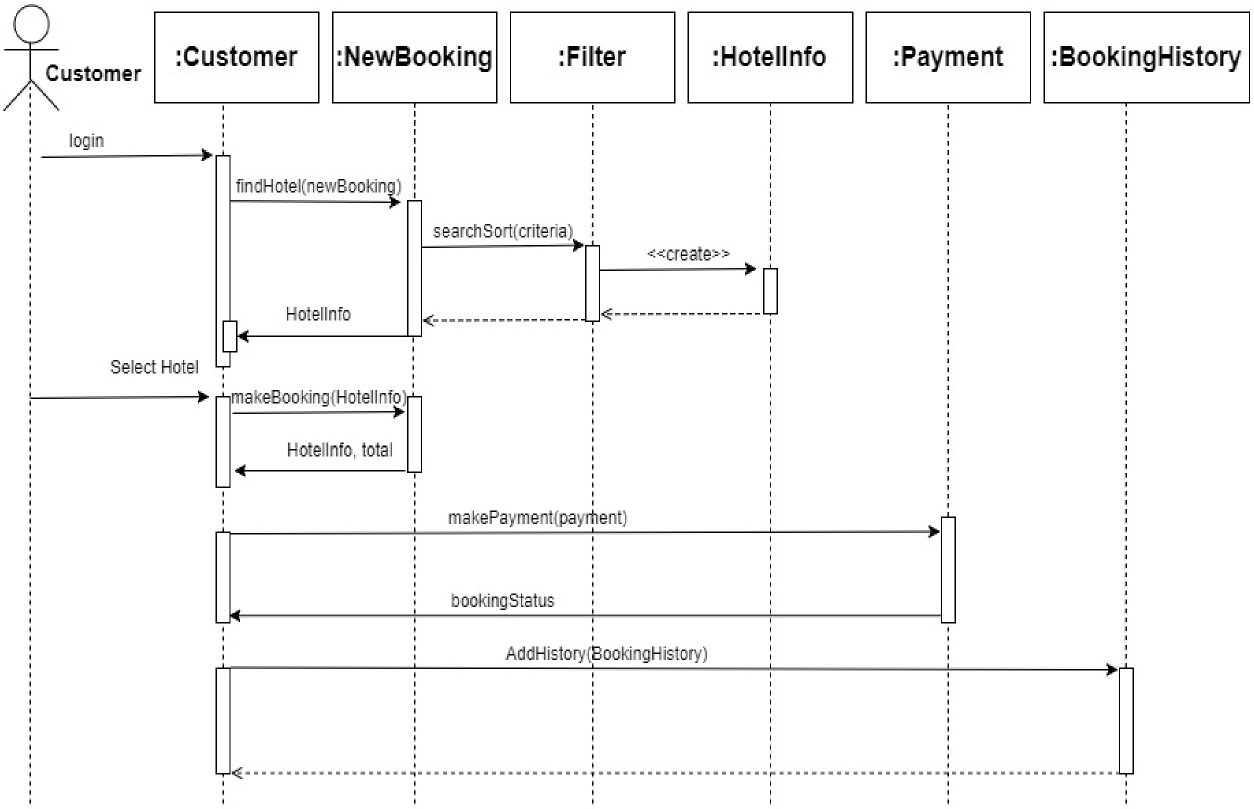
BUFF BEDS | October 25, 2017



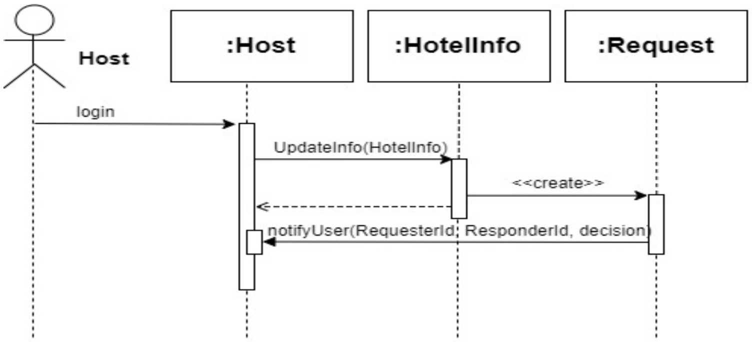
Sequence Diagram



UR09: Hosts can upgrade their Account



UR03, UR05 and UR06: Customer can login, Customer can search and sort hotels or rooms, Customer can checkout the hotel/room



UR10: Host can update the info