



Bunk'O'Bed

-Make your stay away from home feel like home

Sayf Hussain Z

2018103059

Suryaa V S

2018103610

We define ourselves as “ A social website that connects people who have space to spare with those who are looking for a place to stay”.

Niche in the Market:

Lots of homeowners find it difficult to rent out their property for extended stay. Bunk'O'Bed connects these homeowners to people who are looking for affordable housing for a short period of time.

Scope:

Hotels all over the world offer a very limited set of facilities for a very steep price, Bunk'O'Bed provides users the opportunity to search for accommodations that suit their needs and budgets. Tourist whether it be local or international would prefer to stay in the warmth of a house with amenities that just aren't present in hotels. Guests can now choose to live in an entire apartment by themselves at an affordable price. Tourists now get an opportunity to stay in a local neighbourhood and get to know the place better which couldn't be done with a regular hotel. Bunk'O'Bed doesn't just cater to tourist, but to anyone looking for short term accommodations at an affordable price.

People who own multiple homes in a city get the perfect opportunity to supplement their income by renting out properties for a short term and with very little maintenance of the property. Elderly people with spare rooms who do not want to rent out their rooms on a

permanent basis can now rent out their rooms on short term basis to prospective guests.

Anyone moving to a new city in search of work need not commit to a long term rental plan right away and can instead ease into the city by choosing a short term accommodation suiting their needs.

Implementation:

Creating an account:

Users are required to enter details regarding themselves like their phone number, ssn/aadhar card details and if they are the host some paperwork for the property to prove ownership.

Adding a listing:

Homeowners who are willing to rent out their homes or spare room register onto the site and make a listing with details on the type of property, amenities offered, the location with pictures of the property. Amenities can include loads of features like availability of home cooked food, the electrical appliances that the guests can use, the furnishing available, access to recreational facilities. The host also decides the price of stay for each day.

Categorising properties:

Guests can log in into the system and search for listings based on various factors like the city, the cost, the type of property whether it's a single room, an apartment, or even a bungalow. They can also search based on the amenities offered like access to a swimming pool, the furnishing available.

Making a reservation:

Users can choose a listing and read through its description, pictures posted and even check the profile of the host. Users can also request to talk to the host even before they have made the reservation for the property. Contact details are kept private until a reservation is

made and whenever a reservation is made the host and the guest are notified with each other's contact details.

Rating system:

All users (including the hosts and the guests) have a rating which is visible to everyone. The rating for a host is generated based on reviews of other guests whom the host has previously hosted and for the guests its based on the hosts at whose properties they have stayed.

Grievance support:

24/7 user support is available that takes up queries from the users and gets back to them at the earliest with solutions for their problems.

Contacting freelance photographers:

Homeowners can contact freelance photographers through the site and request them to click pictures of the property.

Blacklisting listings:

The admin goes through the rating system and if he finds listings violating user policy he can blacklist the listing and remove the listing and also temporarily ban the user.

Agency bookings

Travel agencies can connect with the hosts and offer stay at apartments instead of traditional hotels in their travel itinerary.

Actors:

Host: A special type of user with permissions to add listings for multiple properties.

Guest: A user who primarily uses the system to browse through the different listings in various categories and makes a reservation for the desired property.

Admin: Maintains control over the entire system, integrity of the system and acts upon the directions of the grievance system to ban users either temporarily or permanently.

Customer support: Responds to the queries or complaints made by any kind of user to assist them. Customer support also passes on information about each user to the admin.

Travel Agencies: Another special user with the permission to include multiple hosts in the travel plans promoted by them.

Photographers: Photographers can contact different hosts offering their services at a nominal cost. They also play a role of validating users to eliminate fraudsters uploading fake pictures for their listings.

Use cases:

- I. Create an account.
- II. Logging in.
- III. Rating system.
- IV. Grievance system.
- V. Add a listing.
- VI. Contact freelance photographers.
- VII. Make payment to photographer.
- VIII. Request to contact host.
- IX. Categorising properties.
- X. Making a reservation.
- XI. Make payment.
- XII. Disclosure of details.
- XIII. Blacklisting users.
- XIV. Temporary Ban.
- XV. Contact agencies and advertisers.
- XVI. Details verification.
- XVII. Register agency.
- XVIII. Upload travel itinerary.
- XIX. Bulk bookings

Use case: Creating account

Primary actor: Host, Guest.

Goal of the use case: Allow people to register onto the system by entering their details. The entered details are then validated by the admin after which a verification email is sent to the user.

Use cases involved at a lower level of abstraction:

- I. Enter details.
- II. Validate details.
- III. Send verification mail.
- IV. Validate mail.
- V. Pay registration fee.

Use case: Rating System

Primary actor: Host, Guest.

Goal of the use case: Allow any user to provide ratings to other users they have interacted with (i.e rented their property to or had rented their property). The users also have the option of adding a detailed review of the host, guest and the property. The new rating for the host, guest and the property is also calculated here.

Use cases involved at a lower level of abstraction:

- I. Add rating for host
- II. Add rating for guest
- III. Add rating for property
- IV. Add a review
- V. Update rating

Use case: Logging in

Primary actors: Host, Guest

Goal of the use case: Registered users are required to enter their credentials before accessing their account. The entered credentials are cross checked with existing records before the user is allowed access to any features reserved for registered users.

Use case: Adding a listing

Primary actors: Host.

Goal of the use case: Users can list their property on the website. Users are required to enter information about the property such as the location, size of the property and the features available.

Use case: Contact freelance photographers

Primary actors: Host.

Goal of the use case: The host can contact photographers to take pictures of the property to increase the trust factor of the property as the pictures of the property come from a third party.

Use case: Make payment for photographer

Primary actors: Host

Goal of the use case: The photographers is paid for his services.

Use case: Categorising properties

Primary actors: Guest.

Goal of the use case: Users can categorise properties while searching for them based on various factors like location, type of property, amenities available.

Use case: Making a reservation

Primary actors: Guest

Goal of the use case: Users can Finalize details with the host and make a reservation for the property for the specified dates.

Use case: Disclosure of details

Primary actor: Guest

Goal of the use case: Contact details are exchanged between the host and the user which were kept private before this stage.

Use case: Make payment

Primary actor: Guest.

Goal of the use case: The user can now make payment for the rented property. This payment is first held in the system, the admin then confirms with the host if the guest has checked in and then transfers money to the host.

Use cases involved at a lower level of abstraction:

- I. Make payment
- II. Withhold payment
- III. Confirm check-in with the host.
- IV. Transfer money to the host.

Use case: Blacklisting user

Primary actor: Admin.

Goal of the use case: The admin blacklists users with lower ratings and users who are reported repeatedly. If a user gets blacklisted more than 5 times the user is banned.

Use case: Temporary ban

Primary actor: Admin.

Goal of the use case: User who are blacklisted more than 5 times in a period of 6 months are banned from accessing the system.

Use case: Contact agencies and advertisers

Primary actor: Admin.

Goal of the use case: The admin can invite various travel agencies to use the platform and can also contact advertisers to market the platform efficiently.

Use case: Details verification

Primary actor: Admin

Goal of the use case: The Admin has to verify the information entered by the user while listing properties like check ownership of the property.

Use case: Register Agency

Primary actor: Travel Agency

Goal of the use case: Travel agencies can register with the system to integrate the features offered by the platform into their travel plans.

Use case: Upload Travel Itinerary

Primary actor: Travel Agency.

Goal of the use case: Travel agencies can create travel plan for tourists with the housing provided by various hosts on the system and can upload the itinerary on the platform.

Use case: Bulk bookings

Primary actor: Travel Agency.

Goal of the use case: The travel agency can rent out multiple properties offered by multiple hosts at the same time.

Use case: Help desk

Primary actors: Host, Guest, Photographers, Customer support.

Goal of the use case: Users can send in their complains and queries which are then replied to by the customer support.

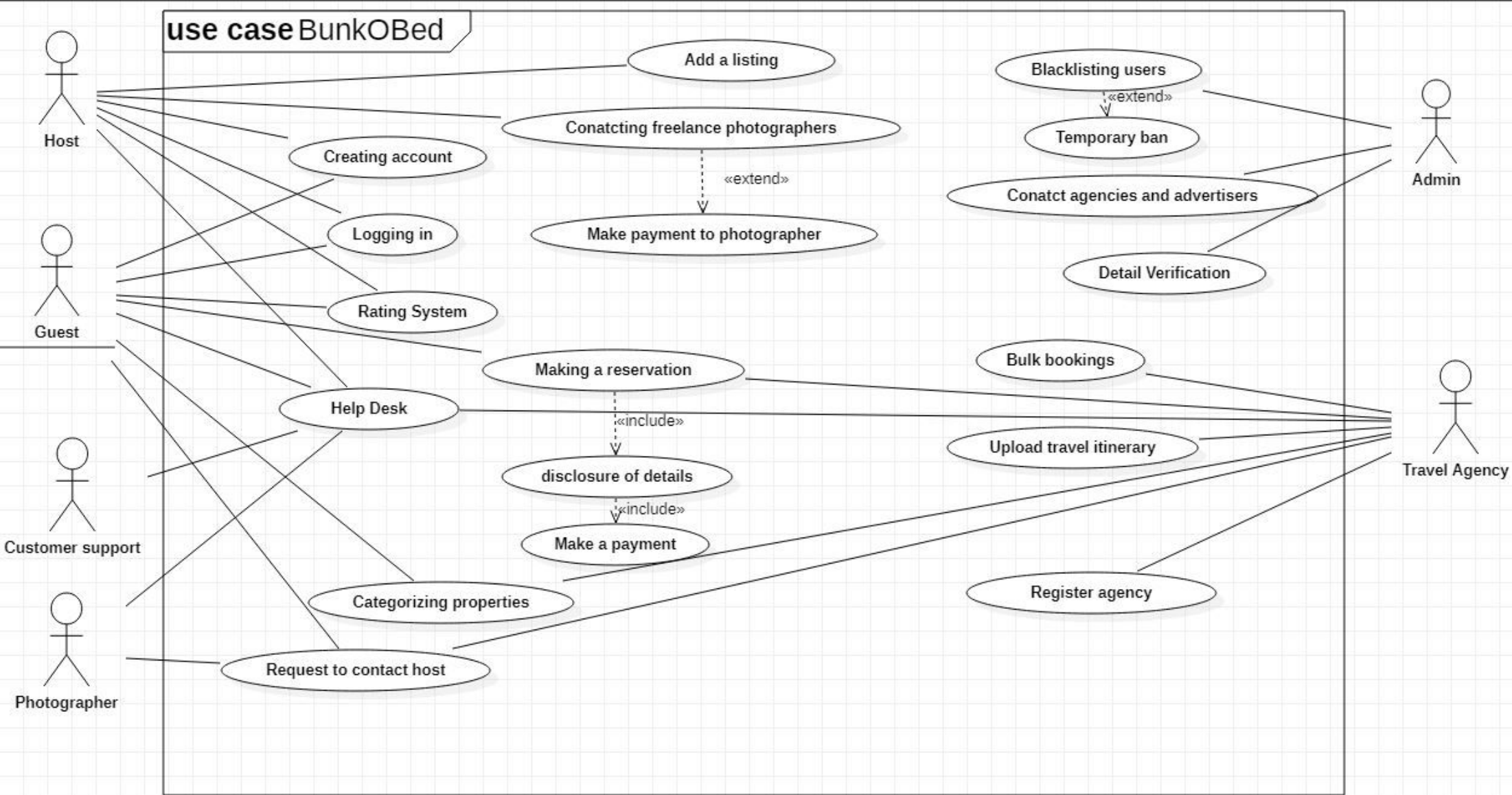
Use cases involved at a lower level of abstraction:

- I. Report abuse
- II. Avail refund
- III. Schedule meeting with admin
- IV. Contact company

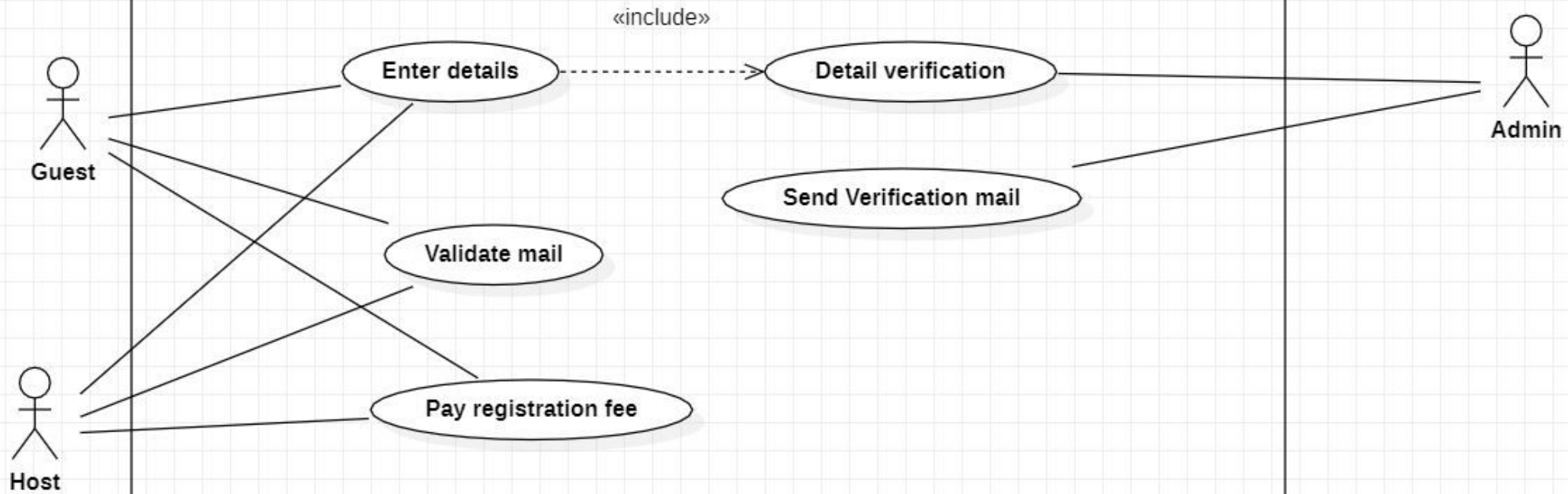
Use case: Request to contact host

Primary actor: Guest, Photographers, Travel Agency.

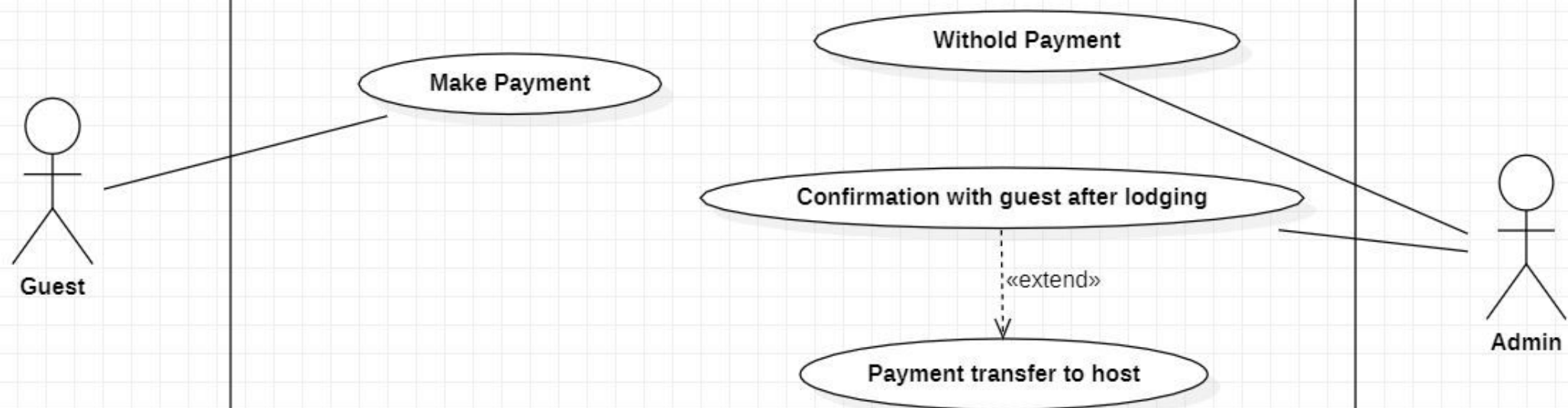
Goal of the use case: Guest, Photographers and Travel agencies can request to contact host to either discuss details or sign them up for their travel plans.



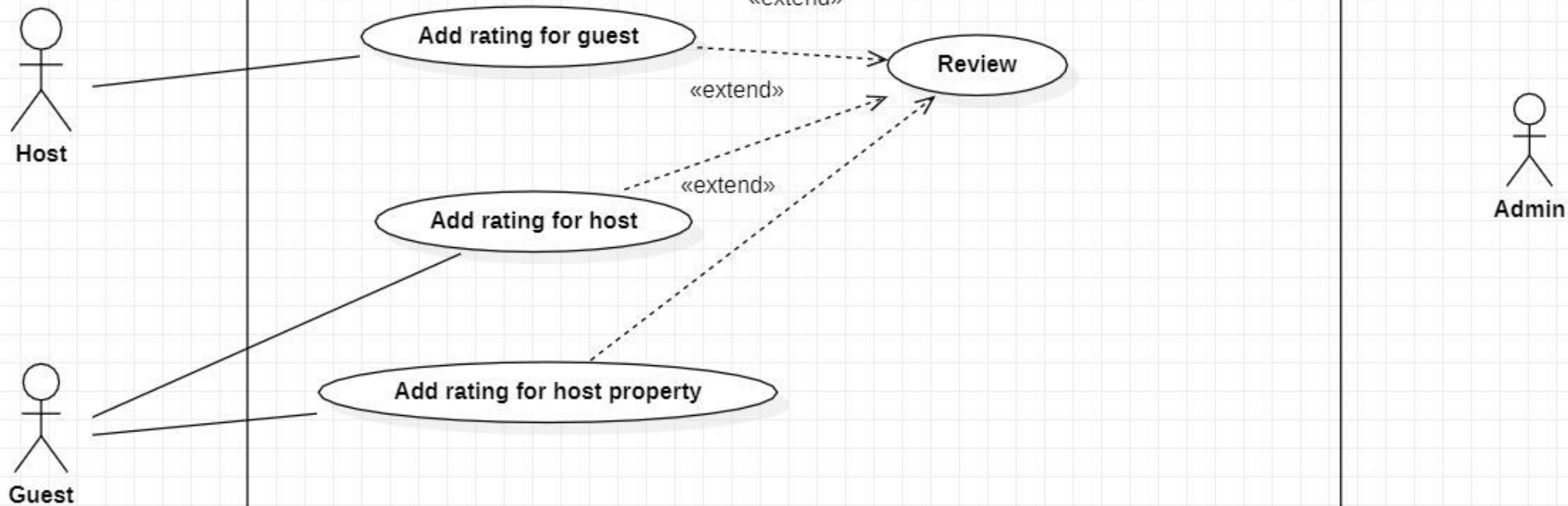
use case Creating account



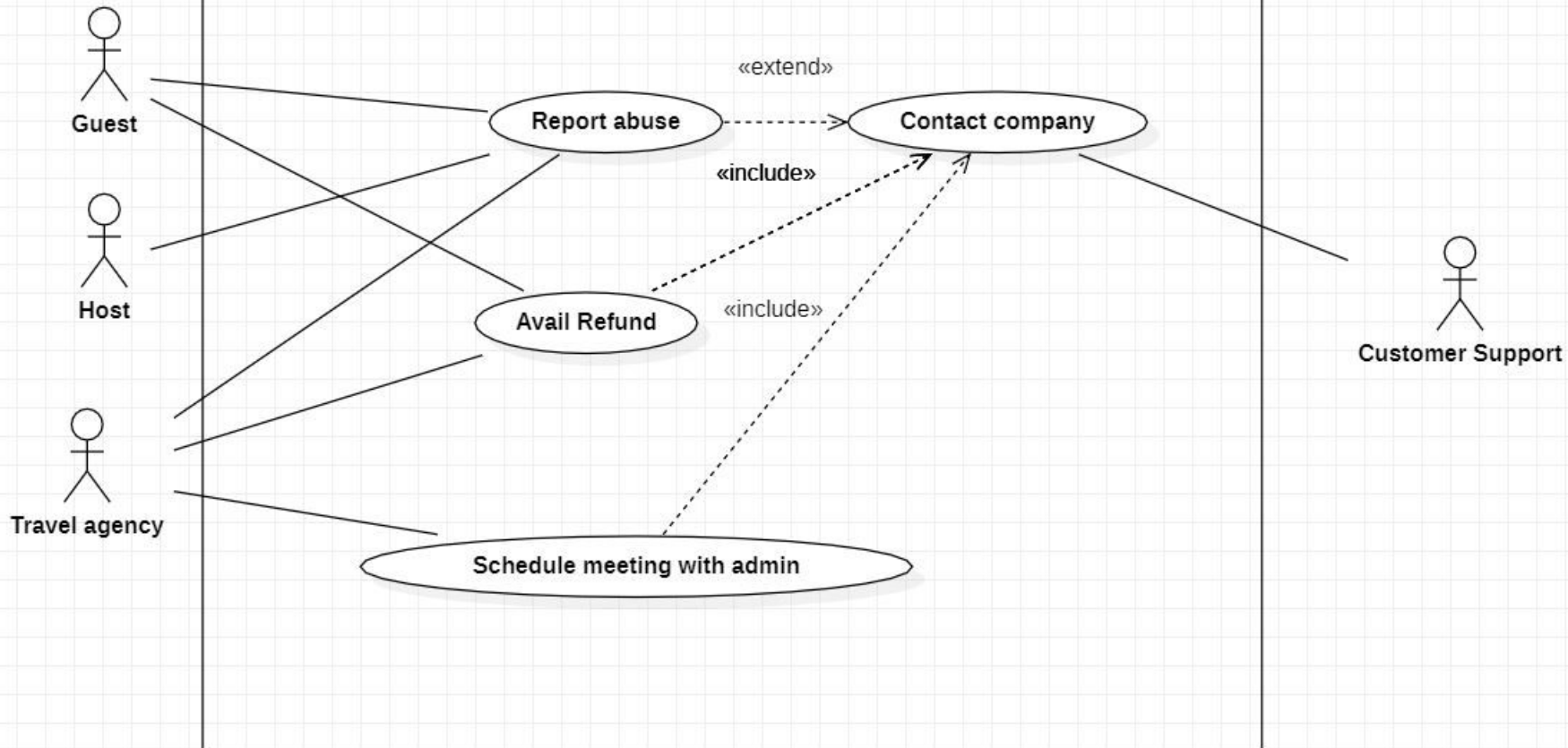
use case Make a payment



use case Rating System



use case Help desk



Domain classes:

- I. User
- II. Host
- III. Guest
- IV. Property listings
- V. Rating
- VI. Transaction
- VII. Admin
- VIII. Photographers
- IX. Queries
- X. Costumer support
- XI. Travel agencies
- XII. Meeting
- XIII. Refund
- XIV. Rented properties
- XV. Bulk bookings

Host ,Guest and Travel agencies are subclasses of User

Associations and Multiplicity:

Admin **Manages** User (1 to 1..*)

User **Has** User details (1 to 1)

Host **Adds a** property listings (1 to 1..*)

Guest **Books a** Property listing (1 to *)

Property listing **Added to** Rented Properties (1 to *)

Guest **Initiates** transaction (1 to *)

Travel agency **Initiates** Transaction (1 to *)

Transaction **Forwards to** Host (1 to 1)

User **Makes a** Queries (1 to *)

Queries **Resolved by** Customer support (1 to 1..*)

Customer support **initiates** Refund (1 to *)

Refund **Requires** Transaction (1 to 1)

User **Has a** Rating (1 to 1)

Property **Has a** Rating (1 to 1)

Host **Hires a** photographer (1 to *)

Meeting **Scheduled with** admin (1 to 1)

Travel agency **calls for a** meeting (1 to 1)

Travel agency **requests** bulk bookings (1 to *)

Attributes:

1. User

- User ID
- Password
- Name
- Number of blacklists

2. User details

- Bank account details
- Aadhar number
- PAN card ID
- Email id
- Phone number

3. Guest

- Number of bookings made

4. Host

- Number of properties registered

5. Travel Agency

- Government Registration ID
- Number of bookings made

6. Rating

- Number of stars
- Comments

7. Customer support

- Staff ID
- Location
- Rating
- Languages spoken

8. Refund

- Issue
- Guest ID

9. Queries

- Type of Query
- Grievance
- User contact details
- Status

10. Photographers

- Photographer ID
- Name
- Location

11. Admin

- Admin ID
- Position of responsibility
- Password

12. Meeting

- Meeting ID
- Travel agency ID
- Date
- Time

13. Bulk bookings

- Travel agency
- Number of people
- Duration of travel
- Accommodation type

14. Property listing

- Property ID
- City
- Address
- Type of property
- Host ID

15. Rented Property

- Property ID
- Guest ID
- Booking dates
- Cost of booking

16. Transaction

- Transaction ID
- Amount
- Mode

Functions:

1. User:
 - Get_userID()
 - Cancel_booking()
 - Make_payment()
2. Guest:
 - Calculate_cost()
3. Transaction:
 - Get_bank_details()
 - With_hold_payment()
 - Make_payment(mode)
4. Queries:
 - Categorise()
 - Check_status()
5. Travel agency:
 - Calculate_cost()
6. Host:
 - Notify_host()
 - Hire_photographers()
7. Customer support:
 - Change_status()
8. Refund:
 - Set_refund(amount)
9. User details
 - getBankdetails()
10. Property listings
 - Get_cost_per_day()
11. Ratings
 - Addtosuslist()
12. Rented Property
 - Get_booking_dates()
13. Meeting

- Block_date_time(userid,adminid)

14.Admin

- Modify_blacklist(user)
- Ban(user)

CRC:

Class	Attributes	Functions	Collaborations
User	<ul style="list-style-type: none">• User ID• Password• Name• Number of blacklists	<ul style="list-style-type: none">•Get_userID()•Raise_query()	<ul style="list-style-type: none">•User details•Admin•Queries•Ratings
User details	<ul style="list-style-type: none">•Bank account details•Aadhar number• PAN card ID• Email id• Phone number	<ul style="list-style-type: none">•getBankdetails()	<ul style="list-style-type: none">•User
Guest	<ul style="list-style-type: none">• Number of bookings made	<ul style="list-style-type: none">•Cancel_booking()•Make_payment()	<ul style="list-style-type: none">•Property listings•Transaction
Host	<ul style="list-style-type: none">• Number of properties registered	<ul style="list-style-type: none">• Notify_host()•Hire_photographers()	<ul style="list-style-type: none">•Property listings•Photographers•Transaction
Travel agency	<ul style="list-style-type: none">• Government Registration ID• Number of bookings made	<ul style="list-style-type: none">• Calculate_cost()	<ul style="list-style-type: none">•Transaction•Meetings•Bulk bookings
Rating	<ul style="list-style-type: none">• Number of stars• Comments	<ul style="list-style-type: none">• Addtosuslist()	<ul style="list-style-type: none">•User•Property listings
Customer support	<ul style="list-style-type: none">• Staff ID• Location• Rating• Languages spoken	<ul style="list-style-type: none">• Change_status()	<ul style="list-style-type: none">•Refund•Queries
Refund	<ul style="list-style-type: none">• Issue• Guest ID	<ul style="list-style-type: none">• Set_refund(amount)	<ul style="list-style-type: none">•Customer support•Transaction
Queries	<ul style="list-style-type: none">• Type of Query• Grievance• User contact details• Status	<ul style="list-style-type: none">• Categorise()• Check_status()	<ul style="list-style-type: none">•Customer support•User
Photographers	<ul style="list-style-type: none">• Photographer ID• Name• Location		<ul style="list-style-type: none">•Host
Admin	<ul style="list-style-type: none">• Admin ID• Position of responsibility• Password	<ul style="list-style-type: none">•Modify_blacklist(user)• Ban(user)	<ul style="list-style-type: none">•User•Meeting

Meeting	<ul style="list-style-type: none"> • Meeting ID • Travel agency ID • Date • Time 	<ul style="list-style-type: none"> • Block_date_time(userid,adminid) 	<ul style="list-style-type: none"> •Travel agency •Admin
Bulk bookings	<ul style="list-style-type: none"> • Travel agency • Number of people • Duration of travel • Accommodation type 		<ul style="list-style-type: none"> •Travel agency
Property listing	<ul style="list-style-type: none"> • Property ID • City • Address • Type of property • Host ID 	<ul style="list-style-type: none"> • Get_cost_per_day() 	<ul style="list-style-type: none"> •Host •Rented property
Rented Property	<ul style="list-style-type: none"> • Property ID • Guest ID • Booking dates • Cost of booking 	<ul style="list-style-type: none"> • Get_booking_dates() 	<ul style="list-style-type: none"> •Property listings
Transaction	<ul style="list-style-type: none"> •Transaction ID •Amount •Mode 	<ul style="list-style-type: none"> •with_hold_payment() •make_payment(mode) 	<ul style="list-style-type: none"> •Travel Agency •Guest •Host •Refund

