Lab-2

IT314_Guesthouse_booking_system_16

GROUP NO: 16

Members:

202001156 - KHAMAR AYUSH KINCHITKUMAR
202001146 - MAKADIYA MUDIT BHAVESHBHAI
202001138 - GANDHI SOMIN
202001129 - DEEP JAIMESH SHAH
202001159 - RADADIYA DARPAN VITTHALBHAI
202001161 - GAJERA AYUSH KAMLESHBHAI
202001165 - PATEL HARSH BHARATBHAI
202001175 - ANKIT LOCHAN
202001170 - RAVAL KEVIN KIRITKUMAR
202001126 - PATEL AYUSH MUKESHBHAI

Needs for the project :

1. **Convenience**
    - There are many Users who want to book stays as per their **convenience** from wherever they want.
    - Users want the process of booking to be **simple and straightforward**.
    - Users want **Different modes of payment** for booking (which is a headache in offline booking).

2. **Security**
    - Many hotels are not verified, and it's hard for the users to trust such stays. With our booking system, we display only **verified ones.**.
    - With the **past record** of the hotels, users can easily navigate for their best stay.

3. **Transparency**
    - Customers wish to get the same they saw online in the pictures, but usually hotels don’t provide facilities up to that mark..
    - In online payments, there are many **hidden charges** that customers have to pay. We offer full **transparency** in terms of extra charges (e.g. sgst, cgst, transaction fees etc.).

4. **Renting Property**
    - The system will work either way like also providing the users to add their property in the system and get rentals for the same.

5. **Comparison**
    - In search of the best deal, users want to **compare between multiple hotels**, according to their pricings. With our website this can be easily done.
    - Even hotels can see reviews of customers given by other hotels. (According to their stays.)

6. **Need for business**
    - With our website, hotels will get **more opportunities** and ultimately will increase their business.

Features of the project solution:

1. **Verified Places** :
   Guesthouses, offices and other land available for rent will be verified before listing

2. **Clear portfolio of locations** :
   Pictures will be available to clearly show each and every aspect of the location and all the facilities should be clearly listed so that the customer knows what he is paying for.

3. **Authentication** :
   User account will be secured and the access to the account will only be allowed after password authentication.

4. **Nearby locations** :
   The system will allow the user to find available rooms near to his/her current location.

5. **Easy booking** :
   The system will be user friendly and suitable for all age groups of users to make bookings.

6. **Premium member** :
   Users will have the option to upgrade to premium membership allowing them to avail facilities like longer cancellation period, priority bookings, offers and coupons.

7. **Cancellation** :
   The facility of cancellation will be hassle free and it will be available 24 hours prior for normal members and 48 hours prior for premium members.

8. **Feedback and rating** :
   Each guesthouse will be awarded a rating based on customer satisfaction and customer frequency. This rating will be clearly visible to the customer and reviews will be available for reading as well.

9. **Customer Support** :
   Customer support will be available 24x7. In case of any inconvenience the customer support team will communicate with the owner/provider, the user just has to state his problem to the customer support team. The customer support team will act as an intermediary between the customer and the owner/provider.

10. **Document verification** :
    Customers will have to verify the documents needed to make a booking only once on the system. Then for every booking the verification will be taken care of by the system.

Functional Requirements:

1. **User Management:**
    - Users should be able to **sign up, log in, and edit their profile information**.
    - Admins should be able to **manage user accounts,** including adding, editing, and deleting users.
    - Users should be able to **reset their passwords** if they forget them.
    - Users should be able to apply for premium membership.

2. **Hotel/Guesthouse/Property Management:**
    - Property owners should be able to register their property for booking/renting.
    - Property owners should be able to add their portfolio like hotel campus images, amenities, room information and pricing/renting details.
    - The admin should be able to verify and validate the documents and the licenses of the registered property.
    - Admin should be able to add, delete, and update the property.

3. **View Property:**
    - System should **view the list of property/hotel** based on the **user's filters** which consist of attributes like location, price range, ratings, type of property etc.

4. **Room Management:**
    - The system should be able to **show the availability of rooms in specific hotels** based on the check-in check-out time, category of room, and other amenities.
    - Admins should be able to add, edit, and delete rooms, including room details such as the room number, type, size, and amenities.
    - Admins should be able to vary room availability and pricing for different dates.

5. **Booking Management:**
    - Users should be able to **apply for the waiting list** of the rooms in a specific hotel.
    - System should **approve booking** in the waiting list **based on the priority like higher for premium and then regular members**.
    - Users should be able to make bookings for rooms, specifying the check-in and check-out dates.
    - System should **verify the user documents** while booking.
    - Admins should be able to view, approve, and reject bookings.

- Admins should be able to update the booking status, such as from "pending" to "approved."
- Users should **receive email confirmations** for their bookings.

6. **Cancellation Management:**
   - System should validate the user before carrying out the cancellation procedure.
   - System should **allow cancellation only if the user satisfies the cancellation policy** of the hotel/guesthouse.

7. **Payment Management:**
   - System should support the **pay later option** but should also ask advance payment for confirming booking.
   - The system should support **multiple payment methods**, such as credit card,debit cards and UPI etc.
   - Users should be able to securely enter and store their payment information.
   - Admins should be able to view payment information and track payments.

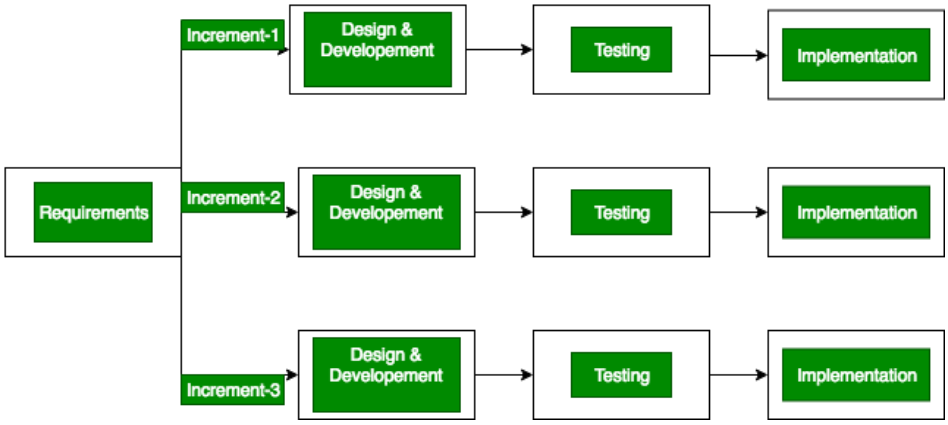
8. **User Support:**
   - Users should be able to reach out for **customer support** via contact or email.
   - Users should be able to **file complaints regarding any hotel/property**.
   - Users should be able to **give feedback regarding the hotel/guesthouse**.
   - Admins should be able to **view and respond to customer support requests**.

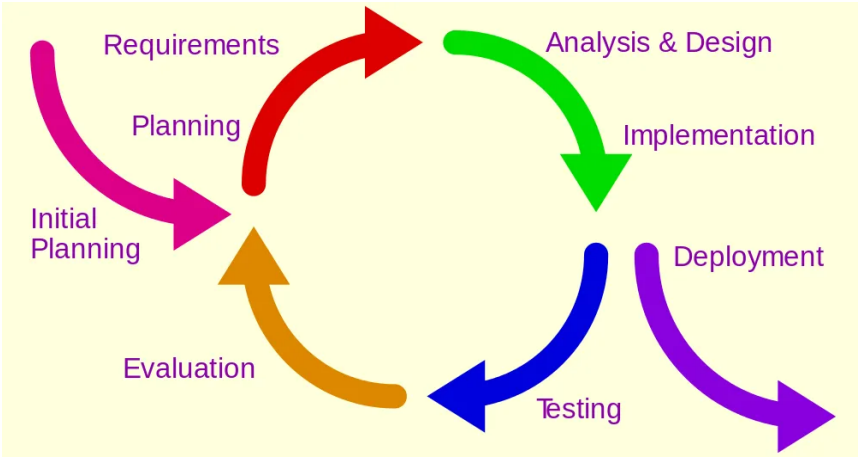
Non functional requirements:

1. **Performance:**
    - The system should be quick and responsive, with minimal lag times for a better user experience.
    - The system should be able to handle high volumes of concurrent users and transactions.
2. **Scalability:**
    - The system should be designed to accommodate growth and be easily scalable to meet increasing demands.
    - The system should be good enough to handle a vast database of the users and the property details.
3. **Availability:**
    - The system should have a high availability rate and be designed to minimize downtime.
    - The system should have robust disaster recovery mechanisms in place.
4. **Security**:
    - The system should have strong security measures to protect sensitive information, such as user passwords and payment information.
    - The system should comply with relevant data protection regulations.
5. **Usability:**
    - The system should be user-friendly and intuitive, with a clear and simple navigation structure.
    - The system should be accessible and usable for users with disabilities.
6. **Interoperability:**
    - The system should be compatible with a range of devices and browsers.
    - The system should be able to integrate with other systems, such as payment gateways, APIs, etc.
7. **Reliability:**
    - The system should have a high level of reliability, with minimal errors and bugs.

8. **Maintainability:**
    - The system should be designed to be easily maintainable, with clear documentation and an organized code structure.
    - The system should have the capability to be updated and improved over time.

Process Model:incremental and iterative

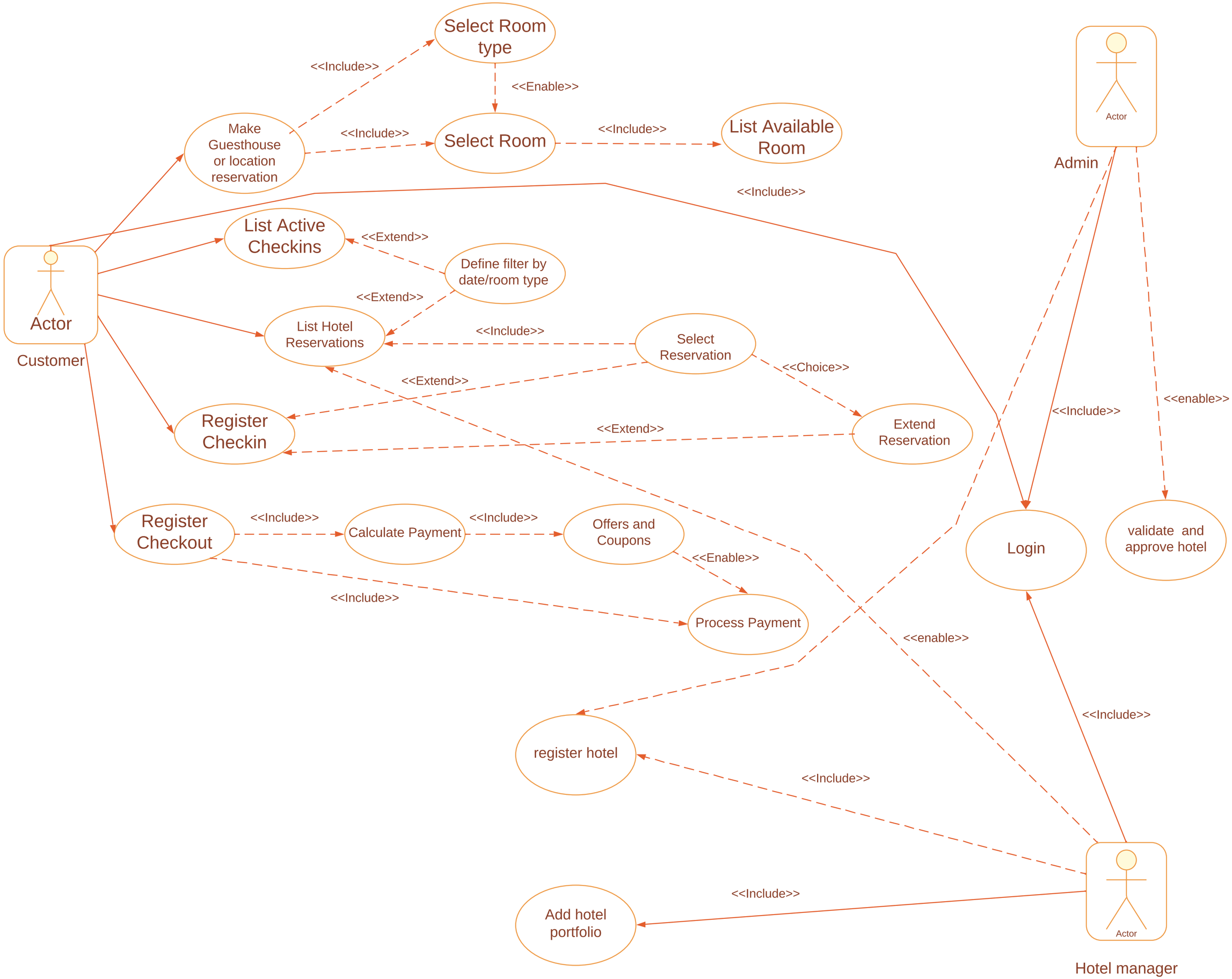




- We will use incremental and iterative models both because we will feature step by step taking the feedback of all the actors present in the use case model.
- It will also help us to detect/debug implementation errors.
- It is also helpful in risk management as our system also includes transaction procedures.
- It is more flexible and less costly to change scope and requirements.

- It is easy to manage each iteration as we develop here module by module.
- It also requires feedback from actors like hotel manager, customers, so the iterative model has an edge over it.
- It is good to use because our projects are lengthy, having a development period of approximately 2 months.

Use case diagram:


Select Room type
<<Include>>
<<Enable>>
Make Guesthouse or location reservation
<<Include>>
Select Room
<<Include>>
List Available Room
<<Include>>
List Active Checkins
<<Extend>>
Define filter by date/room type
<<Extend>>
Actor
Customer
List Hotel Reservations
<<Include>>
Select Reservation
<<Choice>>
<<Extend>>
Register Checkin
<<Extend>>
Extend Reservation
Register Checkout
<<Include>>
Calculate Payment
<<Include>>
Offers and Coupons
<<Enable>>
<<Include>>
Process Payment
Actor
Admin
<<Include>>
<<enable>>
Login
validate and approve hotel
<<enable>>
<<Include>>
register hotel
<<Include>>
<<Include>>
Add hotel portfolio
Actor
Hotel manager