

Functional Requirements:

1. Users can create an account and log in.
2. Users can plan a Moon journey with customizable activities, schedule and hotels.
3. Users can view available hotels and activities.
4. Users can track their journey progress and receive notifications.
5. Hotel Admins can add hotels, menus for that hotel and activities.
6. System Admins can add or remove hotels, activities and rockets.
7. Users can pay for the chosen plan.

Non Functional Requirements:

1. Response time for page loading must be under 3 seconds.
2. Data must be securely stored and modified using transactions.
3. Passwords must be stored using encryption.
4. The interface must be user-friendly and easy to navigate.
5. During registration verification link must be sent by the system.
6. Scheduled task must reset the availability status for rooms and rocket seats.

Use Cases:

Use case	Actor	Description
Sign up	User	Allows users to register
Login/Logout	User/Hotel Admin/System Admin	Allows users to log in to the system or log out
Plan Journey	User	Users select Hotel, activities and launch time
Add new hotel and activities	Hotel Admin/System Admin	Hotel admin can add hotel and activities for their email, System admin can do the same with the hotel admin email
Send verification link	System	Sends verification email
Add new rocket	System Admin	System Admin can add new rockets
Send a ticket	System	Sends a ticket after successful payment via email
Payment	User	Allows users to pay for their journey
Reset availability status for rooms and rocket seats	System	Scheduled task resets availability status

Objects, Classes And Relationships:

Classes:

1. User – Attributes: id, firstName, lastName, email, password, role, verificationToken, enabled
2. Role – Enum type – PASSENGER, HOTEL_ADMIN, SYSTEM_ADMIN
3. Rocket – Attributes: id, launchDay, launchTime, seatsTotal, seatsTaken
4. Hotel – Attributes: id, name, email, activities, rooms
5. Room – Attributes: id, name, capacity, price, hotel
6. RoomBooking – Attributes: id, room, user, checkInDate, checkOutDate
7. Activity – Attributes: id, name, duration, hotel, price
8. ActivityBooking – Attributes: id, activity, user, date
9. Order – Attributes: id, user, hotel, activities, rocket, totalPrice, orderDate

Relationships:

1. Hotel can have many activities and rooms
2. User has three roles
3. Room must have one hotel
4. RoomBooking must have one room, and one user
5. Activity must have one hotel
6. ActivityBooking must have one activity and one user
7. Order can have one user, many RoomBooking, one hotel, one rocket and many ActivityBooking