SRS Document

(41) – Trivago

Alshaymaa Abdelaty

4 / 4 / 2024

Department	Section	ID	Name
SC	6	20201700734	محمد محمود رجب محمد
SC	6	20191700570	محمد محسن السيد عبدالرحمن محمد
CSYS	2	20201700729	محمد مجدي أحمد محمد علي
SC	4	20201700481	عبد الرحمن محمد احمد عواد
CSYS	3	20201700075	احمد محمد عبد الرؤف محمد
SC	4	20201700525	عمر احمد محمود حسين

1.Introduction:

Trivago makes it easy to find your ideal hotel and compare prices from various sites. Just enter the place you want to visit and the dates you want, and let Trivago compare prices for your choice of hotels for experiences. To refine your results, you can use filters by price, distance, or number of stars. If you want to request a room in a small hotel or a suite in a luxury hotel, configure your trivago process for online booking. You can search through the large group of all members to stay in the Arab world.

2.User Requirements:

1. **User Registration and Login**:
   - Users should be able to create accounts and log in securely to access personalized features.
   - Registration should collect necessary information and provide account recovery options.
2. **Hotel Search Functionality**:
   - Users can search for hotels based on location, preferred dates, and other relevant criteria.
   - The search results should display relevant hotels along with essential details.
3. **Price and Feature Comparison**:
   - Users should be able to compare hotel prices, amenities, and features side by side.
   - The system should present clear information to help users make informed decisions.
4. **Hotel Reservation**:
   - Users can book hotel rooms directly through the website.
   - Users can reserve hotel rooms after selecting their preferred hotel.
   - The system should handle reservations, cancellations, and modifications seamlessly.
5. **Manage Accounts**:
   - Admins can manage user accounts, including creating, updating, and deleting accounts.
6. **Check Balance**:
   - The manager checks the user's bank balance to ensure it is sufficient for processing the payment for the hotel reservation.
   - Users should receive a notification after making a payment for their hotel reservation.

3.Functional Requirements:

1.User Login:

- **Description/Action**: Users should be able to log in securely using their registered accounts.
- **Requirement/Input**: User-provided login credentials (username/email and password).
- **Source**: User input.
- **Pre-condition**: User account must be registered.
- **Post-condition**: User gains access to personalized features.

2.Hotel Search:

- **Description/Action**: Users can search for hotels based on location, preferred dates, and other preferences.
- **Requirement/Input**: User-provided search criteria (location, dates, filters).
- **Source**: User input.
- **Pre-condition**: User is logged in.
- **Post-condition**: Search results display relevant hotels.

3.Hotel Comparison:

- **Description/Action**: Display hotel options with prices, amenities, and user ratings for comparison.
- **Requirement/Input**: List of hotels retrieved from the search results.
- **Source**: Search results.
- **Pre-condition**: User has performed a hotel search.
- **Post-condition**: User can compare hotel features.

4.Hotel Reservation:

- **Description/Action**: Users can reserve a hotel room after selecting their preferred hotel.
- **Requirement/Input**: User-provided reservation details (check-in/check-out dates, room type, guest count).
- **Source**: User input during the booking process.
- **Pre-condition**: User has selected a hotel and provided necessary details.
- **Post-condition**: Reservation is confirmed, and the room is reserved for the specified dates.

5.Manage Accounts:

- **Description/Action**: Admins can manage user accounts, including creating, updating, and deleting accounts.
- **Requirement/Input**: Admin-provided account details (username, email, password, role, etc.).
- **Source**: Admin input during account management.
- **Pre-condition**: Admin is logged in.
- **Post-condition**: User accounts are created, updated, or deleted as needed.

6.Check Bank:

- **Description/Action**: System verifies bank account details before finalizing hotel reservation.
- **Requirement/Input**: User-provided payment information (credit card, PayPal, etc.).
- **Source**: User input during booking.
- **Pre-condition**: User has selected a hotel and provided payment details.
- **Post-condition**: Reservation is confirmed, and payment is processed.

4.Non-Functional Requirements:

1. **Performance:**

- **Description**: The system should respond quickly to user requests (e.g., search results, booking confirmation).
- **Quality Attribute**: Performance
- **Constraints**: Optimize database queries, use caching, and minimize latency.

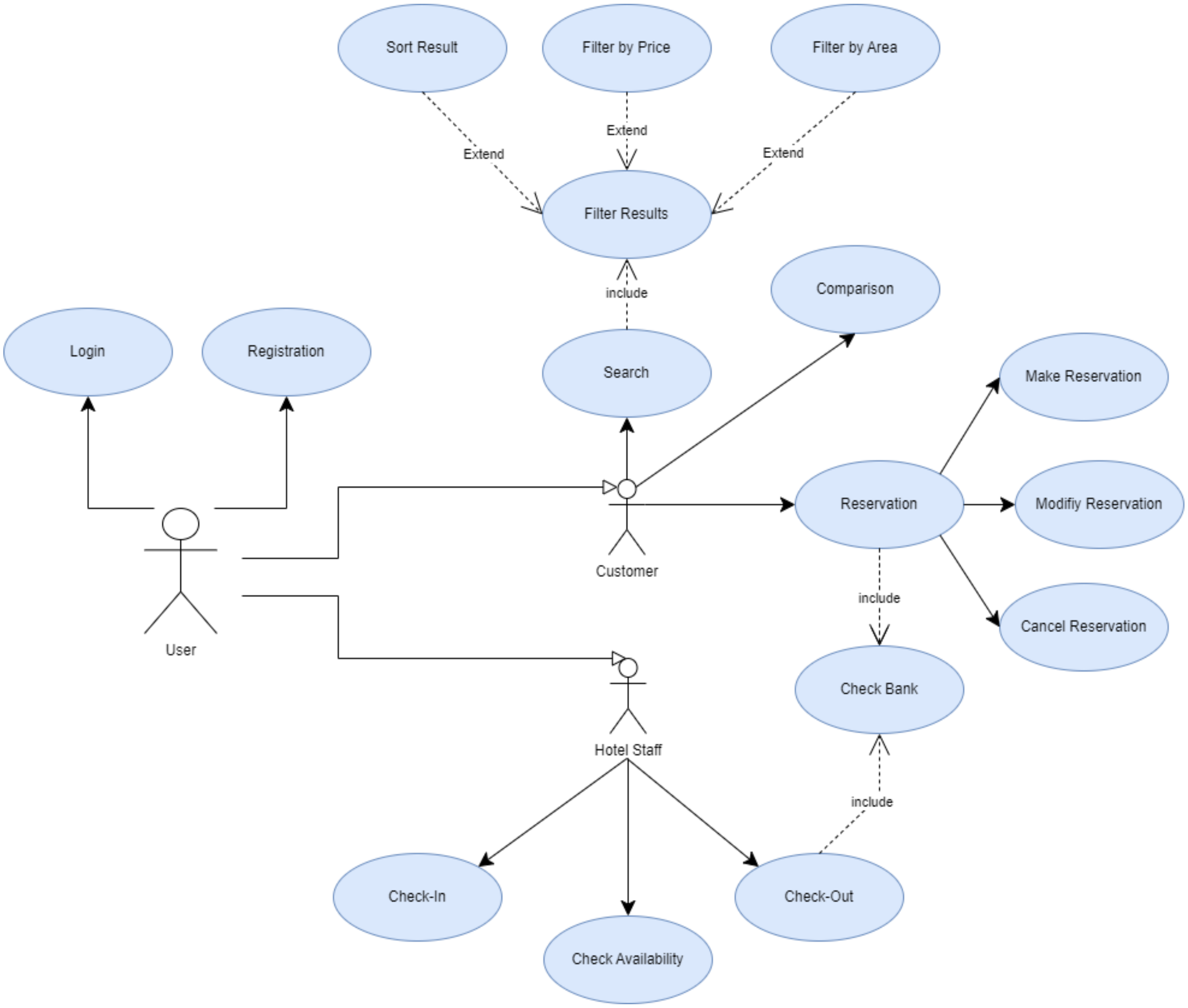
2. **Usability:**

- **Description**: The user interface should be intuitive and easy to navigate.
- **Quality Attribute**: Usability
- **Constraints**: Conduct usability testing and provide clear instructions.

3. **Availability:**

- **Description**: The system should be available 24/7 for users.
- **Quality Attribute**: Availability
- **Constraints**: Implement redundancy and failover mechanisms.

5.Use Case Diagram:



6.Sequence Diagram: