Demo Document

for

Homestay Booking Website Project

Version 1.0

Prepared by

Group 6

Vu Nguyet Hang	22028079	22028079@vnu.edu.vn
Duong Quoc Khanh	22028090	22028090@vnu.edu.vn
Nguyen Thi Thu Trang	22028254	22028254@vnu.edu.vn
Nguyen Van Son	22028020	22028020@vnu.edu.vn
Nguyen Truong Giang	19021259	19021259@vnu.edu.vn

Instructor:	Assoc. Prof. Dr. Dang Duc Hanh
Course:	INT2208E_23
Teaching Assistant:	Kieu Van Tuyen
Date:	04/05/2024

CONTENTS

REVISIONS

Version	Primary Author(s)	Description of Version	Date Completed
#1	Group 6	First Demo	04/05/24

1. Introduction

Golobe, an innovative platform tailored for discerning travelers who seek immersive cultural experiences beyond the conventional confines of hotels. Rooted in the belief that travel should be a transformative journey of discovery and connection, our platform serves as a bridge between travelers and local communities, offering a curated selection of homestays that provide unparalleled insights into the customs, traditions, and daily lives of residents. Through *Golobe*, travelers have the chance to forge genuine connections with hosts who graciously open their homes and share their stories, creating moments of mutual understanding and appreciation. Whether it's a bustling metropolis, a quaint village, or a remote countryside retreat, our platform offers a diverse array of accommodations to suit every traveler’s preference and interest.

2. Demo

Follow this link to find a video of our interactive demo, where the functionality and user experience of our product are presented:
https://youtu.be/eiHkCvNGytw

3. Functions’ Description

1: Login

- Basic Flow:
  1. User navigates to the login page.
  2. User fills their login form with email and password.
  3. User submits the login form.
  4. System verifies the provided credentials.
  5. System grants access to the user's account.
  6. Use case ends.
- Alternative Flow:
  1. Step 4a. If the provided credentials are incorrect, system displays an error message prompting the user to enter valid credentials.
- Exception Flow:
  1. Step 3a. If there are errors in the login form, system prompts the guest to correct the errors and resubmit the form.

2: Change password

- Basic Flow:
  1. User navigates to the "Change Password" section in their account settings.
  2. System prompts the user to enter their current password and the new password.
  3. User enters their current password and the new password, then confirms the new password.

4. System verifies the entered passwords meet the password requirements (e.g., minimum length, special characters).
5. System updates the user's password with the new one.
6. System displays a message confirming that the password has been successfully changed.
7. Use case ends.

- Alternative Flow:
    1. Step 4a. If the entered passwords do not meet the requirements, system displays an error message indicating the password requirements are not met, user re-enters the new password following the requirements, flow returns to step 3 of the basic flow.
- Exception Flow: None
    1. Step 2a. If the user cancels the password change operation, system cancels the password change operation, use case ends without changing the password.

3: Register

- Basic Flow:
    1. User navigates to the registration page.
    2. User fills out the registration form with required details such as name, email, and password.
    3. User submits the registration form.
    4. System verifies the provided information.
    5. System creates a new user account.
    6. System sends a confirmation email to the registered email address.
    7. Use case ends.
- Alternative Flow:
    1. Step 4a. If the provided email is already registered, system prompts the user to log in or reset their password.
- Exception Flow:
    1. Step 5a. If there are errors in the registration form, system prompts the user to correct the errors and resubmit the form.

4: Edit profile

- Basic Flow:
    1. User navigates to the profile editing section.
    2. User selects the option to edit their profile.
    3. User updates the desired profile information such as name, email, or password.

    4. User confirms the changes.
    5. System verifies the updated information and applies the changes.
    6. User receives a confirmation message for the profile update.
    7. Use case ends.
- Alternative Flow:
    1. Step 3a. If the user chooses to change their password, system prompts for the current password before allowing the change.
- Exception Flow:
    1. Step 5a. If there are errors in the updated information, system prompts the user to correct the errors and resubmit the form.

5: Search room

- Basic Flow:
    1. Guest navigates to the search page.
    2. Guest selects destination, check-in and check-out dates, number of guests, and any additional filters.
    3. Guest submits the search criteria.
    4. System retrieves and displays available rooms matching the search criteria.
    5. Guest views detailed information about available rooms.
    6. Use case ends.
- Alternative Flow:
    1. Step 4a. If no rooms match the search criteria, system displays a message indicating no results found.
- Exception Flow:
    1. Step 3a. If there are errors in the search criteria, system prompts the guest to correct the errors and resubmit the form.

6: Book room

- Basic Flow:
    1. Guest selects a room to book from the search results.
    2. Guest provides necessary booking information such as personal details, payment information, and any special requests.
    3. Guest confirms the booking.
    4. System verifies the booking details and processes the payment.
    5. System confirms the booking and sends a confirmation email to the user.

6. Use case ends.

- Alternative Flow:
    1. Step 4a. If payment processing fails, system prompts the guest to try again or use a different payment method.
- Exception Flow:
    1. Step 2a. If there are errors in the booking form, system prompts the guest to correct the errors and resubmit the form.

7: Manage room

- Basic Flow:
    1. Host navigates to the room management section.
    2. Host selects an option to add, modify, or remove rooms, provides necessary room information such as room type, availability, pricing, and amenities.
    3. Host confirms the changes.
    4. System updates the room information accordingly.
    5. Use case ends.
- Alternative Flow:
    1. Step 3a. If the host chooses to remove a room, system prompts for confirmation before proceeding.
- Exception Flow:
    1. Step 2a. If there are errors in the room information form, system prompts the host to correct the errors and resubmit the form.

8: View guest's booking

- Basic Flow:
    1. The host selects the "View Guest Booking" option from the room management view.
    2. System retrieves the booking details associated with the provided booking ID and room ID.
    3. System displays the booking details to the host.
    4. Use case ends.
- Alternative Flow:
    1. Step 3a. If the booking ID or room ID is invalid or does not exist, system displays an error message indicating that the booking details could not be found, flow returns to step 1 of the basic flow.
- Exception Flow:

1. Step 2a. If the system encounters an error retrieving the booking details, system displays an error message indicating the issue encountered, flow returns to the appropriate step based on the host's choice.

9: View my booking

- Basic Flow:
  1. Guest selects the "View My Booking" option from the menu.
  2. System retrieves the booking details associated with the guest’s account.
  3. System displays the booking details to the guest.
  4. Guest reviews the booking details.
  5. Use case ends.
- Alternative Flow:
  1. Step 3a. If the guest has no bookings, system displays a message indicating that the user has no bookings and use case ends.
- Exception Flow:
  1. Step 2a. If the system encounters an error retrieving the booking details, system displays an error message indicating the issue encountered, flow returns to the appropriate step based on the guest's choice.

4. Challenges/Proposed Solutions

- As data volumes grow, traditional relational databases may struggle to scale horizontally across multiple servers, leading to performance degradation and limited capacity for handling concurrent requests.

→ Leveraging MongoDB’s scalability, flexibility, performance, and rich data capabilities.

- Storing and managin a large number of images can strain local storage resources, especially for the current system that allows user-generated uploads.

→ Using Cloudinary – a cloud-based storage that enable system to deliver fast, engaging, and visually appealing experiences without the need for manual capacity planning.

- The problem of ensuring authenticity: verifying the authenticity of homestay experiences and hosts’ cultural knowledge can be challenging, especially in regions with diverse cultural backgrounds.

→ Utilizing user reviews and ratings to maintain quality standards (ongoing functionality).

5. Data structures and Algorithms

There are some notable data structures and alogirthms used in the project:

- Bcrypt hasing function: a cryptographic algorithm that hashes passwords in a way that is secure agianst brute-force attacks.
- sessionStorage: a web storage API that’s used to store data on the client side for the duration of the session. In this project, sessionStorage is used to persist the state of destination, checkIn, checkOut variables across page reloads.