

**Centre Name: ACE-THUDUC-1-FPT**

**Address : 62 Street 36, Van Phuc Residential Area, Hiep Binh Phuoc Ward, Thu Duc City**

**Travel Tour**

|  |  |  |
| --- | --- | --- |
| **Suppervisor** | **Mr.Pham Cong Danh** | |
| **Batch** | **T5.2303.E0** | |
| **Group** | **Group 03** | |
| **No** | **Full name** | **Student code** |
| 1 | Tran Nhat Tan | Student1323263 |
| 2 | Nguyen Quang Huy | Student1468178 |
| 3 | Truong Nguyen Nguyen Chuong | Student1468205 |
| 4 | Nguyen Duy Anh | Student1468183 |

*Mont: 01 Year 2024*

This is to certify that

|  |  |
| --- | --- |
| **Mr.** | **NGUYỄN DUY ANH** |
| **Mr.** | **NGUYỄN QUANG HUY** |
| **Mr.** | **TRƯƠNG NGUYỄN NGUYÊN CHƯƠNG** |
| **Mr.** | **TRẦN NHẬT TÂN** |

Have successfully Designed & Developed

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **TRAVELTOUR** | | |

Submitted by:

|  |
| --- |
| **Mr. PHAM CONG DANH** |

Date Of Issue:

|  |  |
| --- | --- |
|  |  |

Authorized Signature:

|  |  |
| --- | --- |
|  |  |

**CONTENT**

[ACKNOWLEDGE 3](#_Toc156132546)

[INTRODUCTION 4](#_Toc156132547)

[PROBLEM DEFINITION 4](#_Toc156132548)

[CUSTOMER REQUIREMENT SPECIFICATIONS 5](#_Toc156132549)

[USER/INTERFACE 5](#_Toc156132550)

[Hardware / Software requirement 7](#_Toc156132552)

[TASK SHEET REVIEW 1 8](#_Toc156132553)

[REVIEW 2 9](#_Toc156132554)

[USECASE 9](#_Toc156132555)

[Data Flow Diagram (DFD) 32](#_Toc156132556)

[SITE MAP 33](#_Toc156132557)

[ENTITY RELATIONSHIP DIAGRAM (ERD) 35](#_Toc156132558)

[TASK SHEET REVIEW 2 36](#_Toc156132559)

# ACKNOWLEDGE

We extend our deepest gratitude and appreciation to all those who have contributed to the successful completion of our Semester 2 project as part of the Advance Diploma of Software Engineering program. As a collaborative effort, this endeavor involved the dedicated efforts of our team of four students who have strived tirelessly to deliver a robust and innovative project.

First and foremost, we would like to express our heartfelt thanks to our esteemed teacher, Mr. Danh, whose guidance and mentorship have been instrumental throughout the development of our project. His wealth of knowledge, unwavering support, and constructive feedback have been invaluable, shaping not only our technical skills but also fostering a deep understanding of the software engineering principles.

The foundation of our project lies in the integration of various technologies, and we want to acknowledge the pivotal role played by PHP, Laravel, MySQL, HTML, CSS, and JavaScript. These tools have been the building blocks of our project, enabling us to create a dynamic and feature-rich application. The hands-on experience gained in utilizing these technologies has not only enriched our technical prowess but also provided a practical insight into real-world software development.

The collaborative nature of this project has allowed us to cultivate effective teamwork and communication skills. Working as a cohesive unit, we learned the importance of coordination, task delegation, and problem-solving. These invaluable skills extend beyond the realm of software engineering and are applicable to various aspects of our academic and professional lives.

Furthermore, we want to express our gratitude to our fellow team members for their commitment and diligence. Each member has brought a unique set of skills and perspectives, contributing to the overall success of the project.

In conclusion, this project has been a journey of growth, learning, and camaraderie. We are sincerely thankful to everyone who has been part of this endeavor, shaping it into a rewarding experience that will undoubtedly influence our future endeavors in the field of software engineering.

# INTRODUCTION

Greetings! We are excited to present our project in the realm of software engineering. As part of our Advanced Diploma studies, our team of four diligent students has undertaken the development of a comprehensive tour and travel website. The primary objective is to provide visitors with a user-friendly platform for exploring, discovering, and booking various tours.

We aim to deliver a website that is visually appealing and technologically robust. The emphasis lies in creating a seamless user experience, allowing travelers to effortlessly navigate through a diverse array of tours and gain insights into different destinations. With intuitive booking features, our goal is to simplify the reservation process, providing users with a convenient and efficient means to plan their desired adventures. We invite you to join us on this technological journey as we unveil a travel website that not only showcases our technical expertise but also enhances the wanderlust of every virtual explorer.

# PROBLEM DEFINITION

Our project deals with the common struggles travelers face when trying to find and book tours. Right now, it's not easy for people to explore different travel options. The current methods are a bit confusing and take a long time to figure out. Also, the current solutions don't always use the latest technologies, making the experience less smooth. Our goal is to fix these issues by creating a simple tour and travel website. We're using technologies like PHP, Laravel, MySQL, HTML, CSS, and JavaScript to make it happen. The plan is to make booking tours much easier, providing a place where users can easily check out, plan, and book their favorite trips. Our hope is that this project will make the whole travel experience better and more straightforward for everyone.

# CUSTOMER REQUIREMENT SPECIFICATIONS

# 1.USER/INTERFACE

# Input:

1. **Forms and Text Fields:** Users can input information through forms and text fields. This could include filling out personal details, selecting preferences, or entering search queries.
2. **Buttons and Links:** Users interact with buttons and links to navigate through different sections of the website, submit forms, or trigger specific actions.
3. **Dropdowns and Selections:** Users can make selections from dropdown menus or choose options from lists to customize their experience.
4. **Checkboxes and Radio Buttons:** Users can provide input by selecting checkboxes or radio buttons to indicate preferences or choices.
5. **Uploads:** If applicable, users may upload files, images, or documents using file input fields.

**Process:**

1. **Form Submission:** When users fill out forms and submit them, the website processes the input, which may involve data validation and verification.
2. **Dynamic Updates:** Based on user input or interactions, certain sections of the webpage may dynamically update without requiring a full page reload.
3. **Backend Processing:** If the webpage involves server-side processing (using technologies like PHP and Laravel), the server processes the input and returns relevant data or updates.
4. **Client-Side Scripting:** JavaScript can be used for client-side scripting to enhance user interaction, validate input, and dynamically modify the content without reloading the entire page.

**Output:**

1. **Displaying Results:** The webpage outputs information based on user input, such as displaying search results, updated content, or personalized recommendations.
2. **Error Messages:** If there are issues with the input, the webpage may output error messages to guide users in correcting their input.
3. **Confirmation Messages:** Users may receive confirmation messages for successful actions, such as submitting a form or completing a transaction.
4. **Visual Changes:** The webpage may undergo visual changes to reflect the processed information, such as updating images, text, or layout dynamically.

**2. ADMIN**

**Input**

1. **Login Credentials:** The system requires input in the form of username and password for uthentication during the login process.
2. **Configuration Settings:** Administrators may input or modify configuration settings, such as system preferences, security parameters, or feature toggles.
3. **Data Upload:** Administrators might upload data, such as user information, product details, or other relevant content, to the system.

**Process :**

1. **Authentication:** The system processes login credentials to authenticate administrators and grant access to the admin panel.
2. **Authorization:** Once authenticated, the system checks for appropriate authorization levels to determine the scope of actions an administrator can perform.
3. **Configuration Processing:** Changes made to configuration settings are processed by the system to implement the updated preferences or rules.
4. **Data Management:** The system processes uploaded data, ensuring proper validation, storage, and integration into the database.

**Output:**

1. **Admin Dashboard:** After successful login, the system outputs an admin dashboard, providing an overview of system status, analytics, or key metrics.
2. **Error Messages:** In case of incorrect login credentials or invalid input during configuration, the system outputs error messages to guide administrators.
3. **Activity Logs:** The system generates logs to record administrator activities, helping to track changes and monitor system interactions.
4. **System Status Updates:** If configuration settings are modified, the system outputs updates to reflect changes in system behavior or functionality.
5. **Confirmation Messages:** After successful actions, such as data uploads or configuration updates, the system outputs confirmation messages to inform administrators.

**Additional Aspects:**

1. **Role Management:** The system allows administrators to manage user roles and permissions, defining who can access certain features or perform specific actions.
2. **Security Measures:** Input validation, secure authentication protocols, and encryption are crucial components to ensure the security of admin interactions.
3. **Notification System:** The system may include a notification system to alert administrators about critical events, updates, or potential issues.
4. **Backup and Recovery:** Administrators may have the capability to initiate backup processes and recovery procedures in case of data loss or system failures.

**Hardware / Software requirement**

**Hardware**:

**Web Server**

* Processor Intel Core I3 or higher.
* Memory 6 GB RAM or greater.
* Modem/ADSL Internet access is required.

**Client**

* Processor Intel Core I3 or higher.
* Memory 6 GB RAM or greater.
* Monitor Super VGA (1024x768) or higher resolution.
* Modem/ADSL Internet access is required.

**Software**:

**Server:**

* Web server: XAMPP
* Operation System: Window 7 or later.
* Databases: MySQL.
* PHP - Laravel framework
* Browser: Google Chrome version 35.

**Client:**

* Operation System Window 7 or higher.
* Browser Google Chrome , MS-Edge, Firefox

# TASK SHEET REVIEW 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project: Travel tour** | | | **Date of preparation of activity plan:** | | | |
| **#** | **Task** | **Prepared by** | **Start date** | **days** | **Team member name** | **status** |
| 1 | Acknowledgement | Chương | 13/1/2024 | 1 | Chương | completed |
| 2 | Introduction | Chương | 13/1/2024 | 1 | Chương | completed |
| 3 | Problem definition | Chương | 13/1/2024 | 1 | Chương | completed |
| 4 | Customer requirement specification | Chương | 13/1/2024 | 1 | Chương | completed |
| 5 | Scope of work | Chương | 13/1/2024 | 1 | Chương | completed |
| 6 | Hardware/software requirement | Chương | 13/1/2024 | 1 | Chương | completed |
| 7 | Task Sheet | Huy | 13/1/2024 | 1 | Huy | Completed |
| Review | | | Signature of instructor | | | |
|  | | |  | | | |
|  | | | **Mr.Pham Cong Danh** | | | |

# REVIEW 2

# 1 .USECASE

**1.1 Usecase for visitor**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Login | |
| **Actors** | Customer | |
| **Description** | Customer who has registered an account can login | |
| **Requirements** | Customer provides username and password | |
| **Pre-conditions** | Visitor must sign up an account | |
| **Post-conditions** | Success: Visitor is logged in to website | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Visitor click ‘Login’ button  on Home Page          3. Actor inputs Email and  Password, then click the  ‘Login’ button | 2. System redirects to Login Page with the following controls:  -“Email” text field  -“Password” text field  -“Login” button  4. System checks the information  5. System redirects to Home page |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs invalid email or password. | System redirects to Login page with the following controls:   * “Email” text field * “Password” text field * “Login” button   System shows message:  “invalid email and password” |

**Login**

**Register**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Register | |
| **Actors** | User | |
| **Description** | Customer who has signup an account can login | |
| **Requirements** | Customer signups an account on the website and becomes a member | |
| **Pre-conditions** | Customer provides their registration information | |
| **Post-conditions** | Success: Visitor creates a User account in this website | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Visitor click ‘signup’ button on  Home Page                      3. Visitor inputs signup’s information and click “signup” button | 2. System redirects to signup Page and display Registration form with the following controls:  - “Full name” text field   * “Email” text field * “Address” text field * “Phone” text field * “Password” text field * “Register” button * “Login” button if the user has already registered  1. System validates the information      1. System inserts the account into database      1. System show success message |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1 | - Actor left “Username” text field blank. | System displays message to customer: - “Username cannot be blank”. |
| 2 | - Username has already existed. | - “This Username has already existed”. |
| 3 | - Actor left “Password” text field blank. | - “Password cannot be blank”. |
| 4 | - Password has not matched regular expression condition (at least 6 letters) | - “Password must have at least 6 letters |
| 5 | - Actor left “Email” text field blank. | - “Email cannot be blank”. |
| 6 | - Email has already existed. | - “This Email has already existed”. |
| 7 | - Actor input invalid email format | - “Email has invalid format” |
| 8 | - Actor left “Phone” text field blank. | - “Phone cannot be blank”. |
| 9 | - Input phone is not numeric format or phone length is not between 8 and 14 characters. | - “The phone number should be numeric only and must be between 8 and 14 digits”. |
| 10 | - Actor left “Address” text field blank. | - “Address cannot be blank”. |

**Search tour**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Search tour | |
| **Actors** | Registered Customer/ New Customer | |
| **Description** | Customer can search tour | |
| **Requirements** | Customer provides search phrase | |
| **Pre-conditions** | N/A | |
| **Post-conditions** | Success: Show tour that matches search phrase | |
| Fail: No tour is shown | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs search phrase    2.Actor clicks tour’s name in list or view every tour that matches search phrase | System shows list of tours that match the search phrase |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| |  |  | | --- | --- | | Actor enter is not wrong | | | tour |  | | The system does not show the search tour |

###### **VIEW TOUR DETAIL:**

|  |  |  |
| --- | --- | --- |
| **Use case name** | View tour Detail | |
| **Actors** | Registered Customer/ New Customer | |
| **Description** | Customer can view tour detail | |
| **Requirements** | Customer select tour to show tour’s detail | |
| **Pre-conditions** | N/A | |
| **Post-conditions** | Success: Display tour detail matching the selected tour | |
| Fail: No tour is shown | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| Actor select tour | System shows tour’s detail matching selected tour |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| Actor no select tour | The system does not show the  select tour |

###### **CREATE BOOKING (ADD TO BOOKING):**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Create Booking (Add To Booking) | |
| **Actors** | Registered Customer | |
| **Description** | Customer can add tour to their booking | |
| **Requirements** | Customer must choose at least one tour | |
| **Pre-conditions** | Customer must log in before adding tour | |
| **Post-conditions** | Success: Actor add tour to booking successfully | |
| Fail: Redirects to Login Pages | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| Actor chooses a tour and click “Add to booking” | System adds selected tour to booking by id of this tour |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| |  |  | | --- | --- | | Actor inputs more tours | | | than there are in stock |  | | System shows messages error |

###### **VIEW BOOKING**

|  |  |  |
| --- | --- | --- |
| **Use case name** | View Booking | |
| **Actors** | Registered can view their booking | |
| **Description** | Customer can view their booking | |
| **Requirements** | N/A | |
| **Pre-conditions** | Customer must log in before view booking | |
| **Post-conditions** | Success: Customer can view their booking | |
| Fail: Redirects to Login Pages | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| Actor click on “Booking” icon. | System shows booking and added tours |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| |  |  | | --- | --- | | 1. Actor clicks “x”– Close | | | button- on table |  |      |  |  | | --- | --- | | 2. Actor change quantity of a | | | tour |  |     3. Actor clicks “x” – close button – in the right side of each tour. | 1. System redirects to current page. 2. System update quantity of this tour into booking. 3. System removes this   tour from booking |

**Checkout**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Check Out | |
| **Actors** | Check out and Place booking | |
| **Description** | Customer can booking tours | |
| **Requirements** | Customer must has at least 1 tour in booking | |
| **Pre-conditions** |  | |
| **Post-conditions** | Success: Booked successfully | |
| Fail: Redirects to Login Pages | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor click “Booking” icon      1. Actor inputs booking   information then click “Book” button     1. Actor clicks “Check out with” button.      1. Actor inputs Card owner’s name, card number and security code if actor choose   payment by credit card     1. Actor views booking’s status in Booking page | 1. System shows booking and added tours      1. System shows tour in booking with calculated total price and show following controls:  * “Customer name” text field. * “Phone” text field. * “Address” text field. * “Email” text field. - * “Payment” table.      1. Systems shows Payment tab type according to type of   payment actors choose     1. System validates information     5.. System redirects to Booking page and show booking’s status |
| **Alternative flow** | **Actor’s actions:** | **System’s responses:** |
| |  |  | | --- | --- | | 1. Actor clicks “x”– Close | | | button- on table |  |      |  |  | | --- | --- | | 2. Actor clicks “X” button on | | | each tour |  |      |  |  | | --- | --- | | 3. Actors clicks “Home” | | | button. |  | | 1. System redirects to booking page.      1. System removes tour from booking.      1. System redirect to Home pages. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs more tours than there are in system. 2. Actor left “Billing name” text field blank. 3. Actor left “Payment” dropdown blank. 4. Actor inputs invalid card information | The system will report the corresponding error:  1. “Tour out of order”.   1. “Booking name must not be blank”. 2. “Must choose a payment method”   4 .“Invalid card”. |

###### **VIEW BOOKING HISTORY**

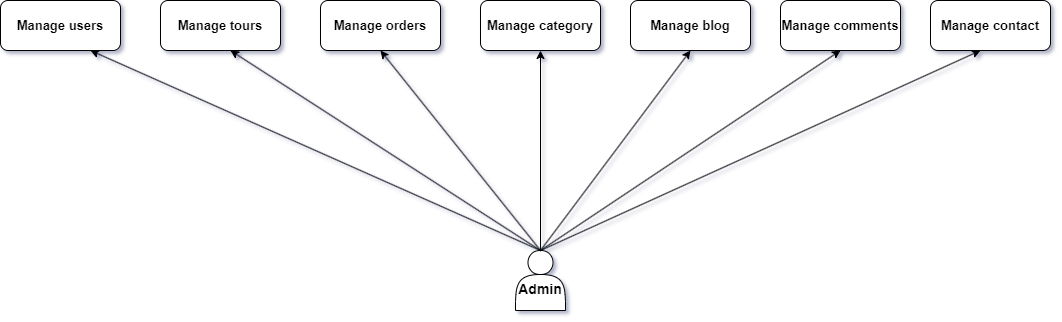
|  |  |  |
| --- | --- | --- |
| **Use case name** | View Booking History | |
| **Actors** | Registered Customer | |
| **Description** | Customer can view their booking history | |
| **Requirements** | N/A | |
| **Pre-conditions** | Customer must log in to website | |
| **Post-conditions** | Success: Customer can view their booking. | |
| Fail: Customer cannot see their booking. | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses “Login” icon on Menu      1. Actor chooses “Booking history” on sides bar      1. Actor selects the booking that they want to view in   “Booking history” and clicks “View” Button. | 1. System redirects to Login page      1. System redirects to   “Booking history” page and displays a list of existing  bookings and their status     1. System validates inputs by id of Booking and displays all   booking’s detail and its status |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor no select tour     |  |  | | --- | --- | | 2. Actors clicks “Home” | | | button |  | | 1. The system does not show the select tour 2. System redirect to Home pages |

###### **CANCEL BOOKING**

###### 

|  |  |  |
| --- | --- | --- |
| **Use case name** | Cancel Booking | |
| **Actors** | Registered Customer | |
| **Description** | Customer can cancel their booking | |
| **Requirements** | Customer provides booking’s id to cancel | |
| **Pre-conditions** | Customer must log in to website and only can cancel when this booking has not been confirmed by admin | |
| **Post-conditions** | Success: Cancel successfully. | |
| Fail: Booking cannot cancel. | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses “Login” icon on Menu      1. Actor chooses “Booking history” on sides bar     3.. Actor selects the booking that they want to cancel in  “Booking history”    4. Clicks “Cancel” Button. | 1. System redirects to Login page      1. System redirects to   “Booking history” page and displays a list of existing bookings and their status     1. System validates inputs by id of booking      1. System change booking status in database and system displays message |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| |  |  | | --- | --- | | 1. Actors clicks “Home” | | | button |  | | 1. System redirect to Home pages |

**1.2 Usecase For Admin**

****

###### **LOGIN**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Login | |
| **Actors** | Admin | |
| **Description** | Admin who has registered an account can login | |
| **Requirements** | Admin provides username and password | |
| **Pre-conditions** | Admin must signup an account | |
| **Post-conditions** | Success: Admin is logged in to website | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor click ‘Login’ button  on Home Page          3. Actor inputs Email and  Password, then click the  ‘Login’ button | 2. System redirects to Login Page with the following controls:   * “Email” text field * “Password” text field * “Login” button 4. System checks the information     5. System redirects to Home page |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs invalid email and password. | System redirects to Login page with the following controls:   * “Email” text field * “Password” text field * “Login” button   System shows message:  “invalid email and password” |

###### **ADD TOUR**

|  |  |  |
| --- | --- | --- |
| **Use case name** | add tour | |
| **Actors** | Admin | |
| **Description** | add new tour information | |
| **Requirements** | tour information | |
| **Pre-conditions** | tour information must pass all input validations | |
| **Post-conditions** | Success: new tour is added to the website | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor click ‘add tour’  button on management page    3. Actor inputs all tours information, then click the  ‘add’ button | 2. System redirects to add tour section with the input information  4. System checks the information    5. System redirects to management page |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs invalid  tour information | System redirects to management page with the error message:  “invalid input information” |

###### **EDIT TOUR**

|  |  |  |
| --- | --- | --- |
| **Use case name** | edit tour | |
| **Actors** | Admin | |
| **Description** | edit tour information | |
| **Requirements** | tour information | |
| **Pre-conditions** | tour information must pass all input validations | |
| **Post-conditions** | Success: new tour is added to the website | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor click edit tour’  button on management page          3. Actor inputs all tours information, then click the  ‘save’ button | 2. System redirects to edit tour section with the input information 4. System checks the information    5. System redirects to management page |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs invalid  tour information | System redirects to management page with the error message:  “invalid input information” |

###### HIDE/DELETE TOUR

|  |  |  |
| --- | --- | --- |
| **Use case name** | Hide/Delete tour | |
| **Actors** | Admin | |
| **Description** | hide tour from website | |
| **Requirements** | none | |
| **Pre-conditions** | none | |
| **Post-conditions** | Success: tour is hide from the website | |
| Fail: Recheck | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor click hide or delete tour’  button on management page | 2. System redirects to management page |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| none | -1. Hide successfully -2. Delete successfully |

###### **VIEW TOUR**

|  |  |  |
| --- | --- | --- |
| **Use case name** | view tour | |
| **Actors** | Admin | |
| **Description** | view tour information | |
| **Requirements** | tour information | |
| **Pre-conditions** | none | |
| **Post-conditions** | Success: tour full information is show up | |
| Fail: Recheck | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor click view tour’  button on management page | 2. System redirects to tour details section |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| none | none |

###### **DISABLE CUSTOMER**

|  |  |  |
| --- | --- | --- |
| **Use case name** | disable customer | |
| **Actors** | Admin | |
| **Description** | disable customer | |
| **Requirements** | none | |
| **Pre-conditions** | none | |
| **Post-conditions** | Success: disabled customer is disable to login to the system | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor click disable customer button on management page | 2. System redirects  management page |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| none | none |

###### EDIT ADMIN ACCOUNT INFORMATION

|  |  |  |
| --- | --- | --- |
| **Use case name** | edit admin account information | |
| **Actors** | Admin | |
| **Description** | edit admin account information | |
| **Requirements** | new admin account information | |
| **Pre-conditions** | new admin account information must pass all input validations | |
| **Post-conditions** | Success: new admin account information are updated to the website | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor click edit account information button on  management page          3. Actor inputs all account information, then click the  ‘save’ button | 2. System redirects to edit account information section with the input information 4. System checks the information    5. System redirects to management page |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs invalid information | System redirects to management page with the error message:  “invalid input information” |

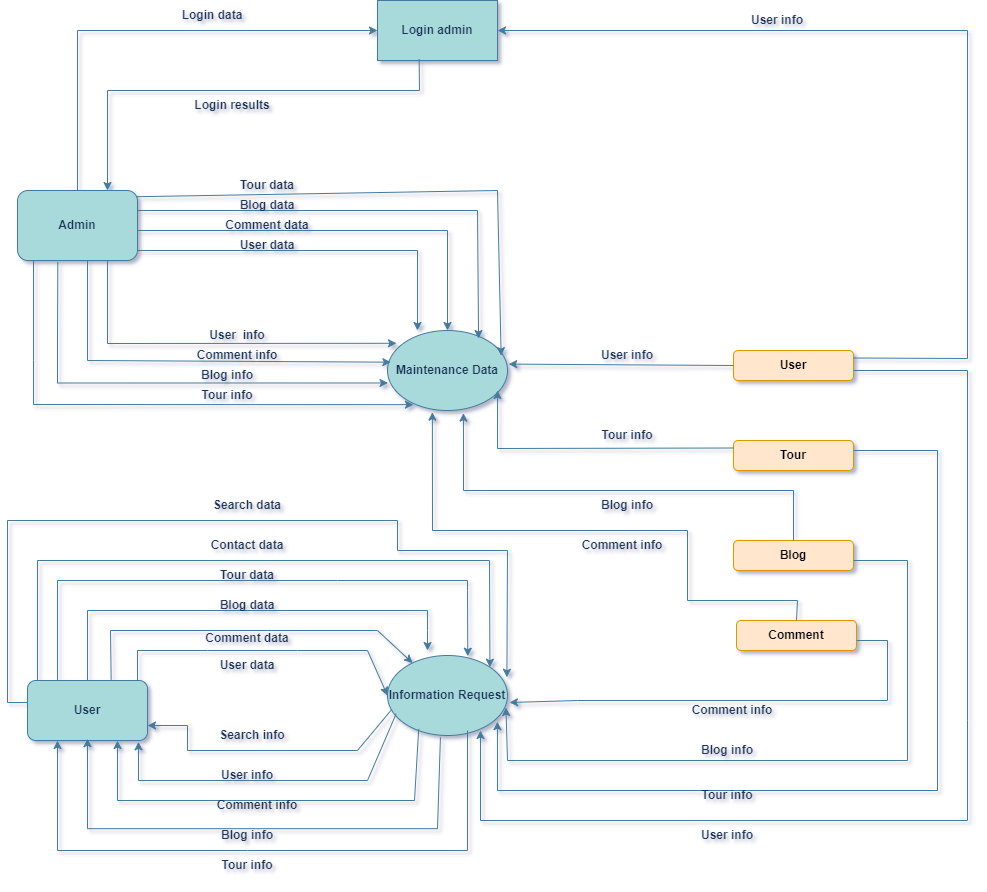
###### **SEARCH TOUR**

|  |  |  |
| --- | --- | --- |
| **Use case name** | search tour | |
| **Actors** | Admin | |
| **Description** | search tour | |
| **Requirements** | name of tour to be searched | |
| **Pre-conditions** | none | |
| **Post-conditions** | Success: the result is show up | |
| Fail: not found | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor click search tour  button on management page          3. Actor inputs name of tour to search | 2. System redirects to search tour section   1. System checks the information      1. System return the result |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| none | none |

###### **SEARCH USER**

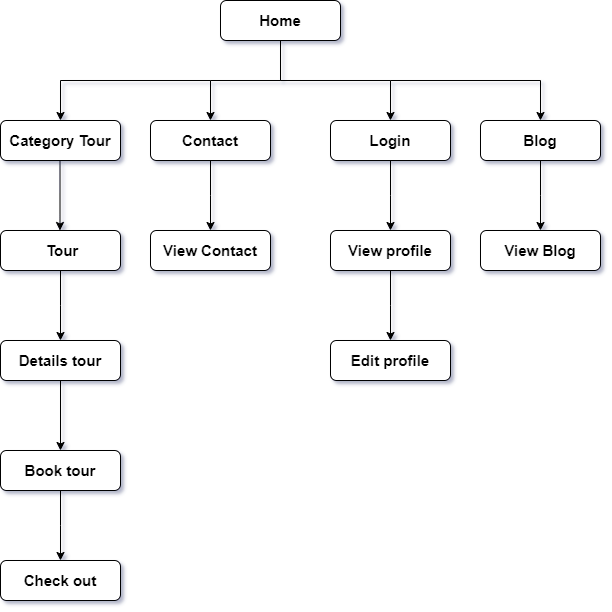
|  |  |  |
| --- | --- | --- |
| **Use case name** | search user | |
| **Actors** | Admin | |
| **Description** | search user | |
| **Requirements** | name of user to be searched | |
| **Pre-conditions** | none | |
| **Post-conditions** | Success: the result is show up | |
| Fail: not found | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor click search user  button on management page  3. Actor inputs name of user to search | 2. System redirects to search user section   1. System checks the information      1. System return the result |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |

**2.Data Flow Diagram (DFD)**

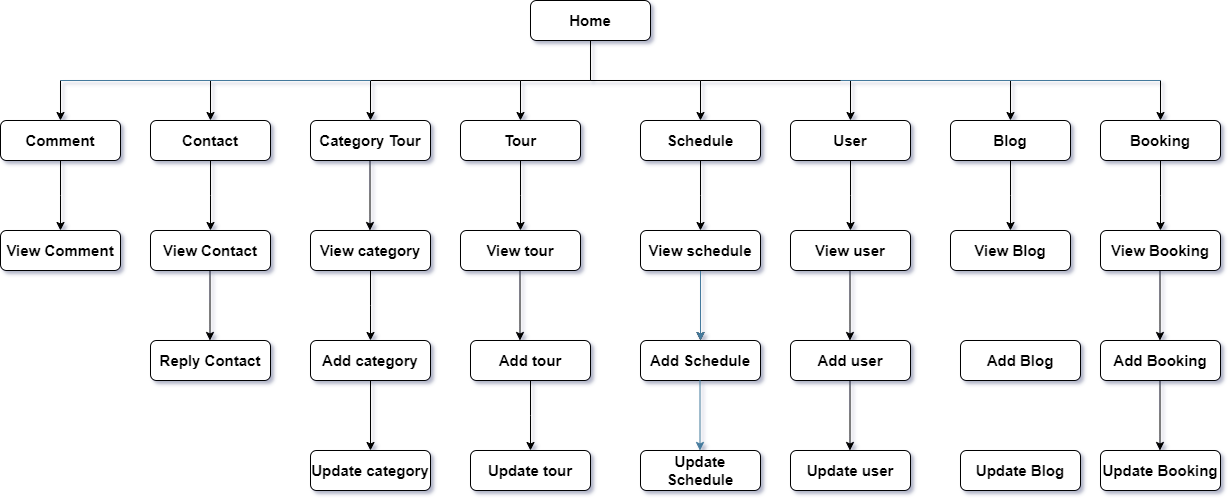
****

# SITE MAP

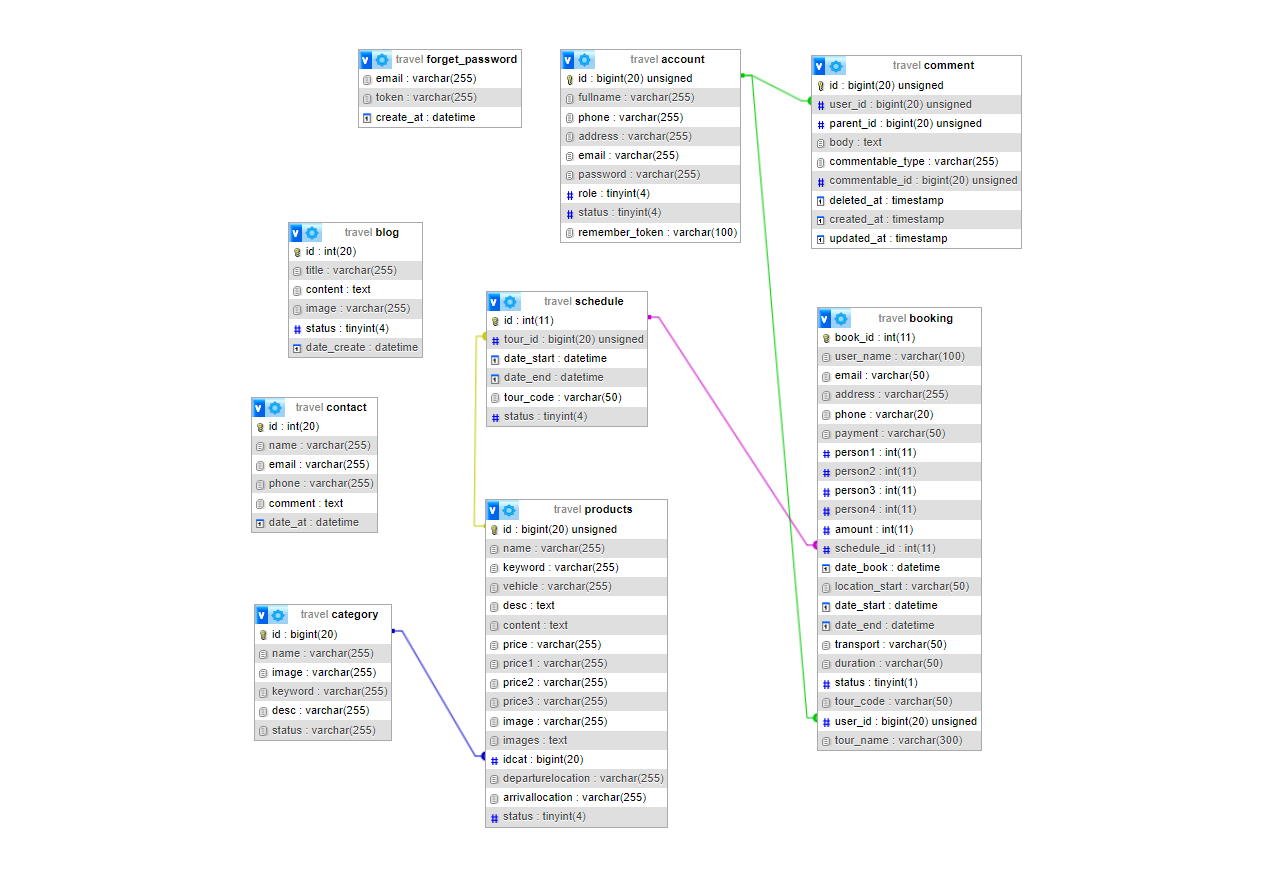
1. **USER**



1. **ADMIN**



**ENTITY RELATIONSHIP DIAGRAM (ERD)**

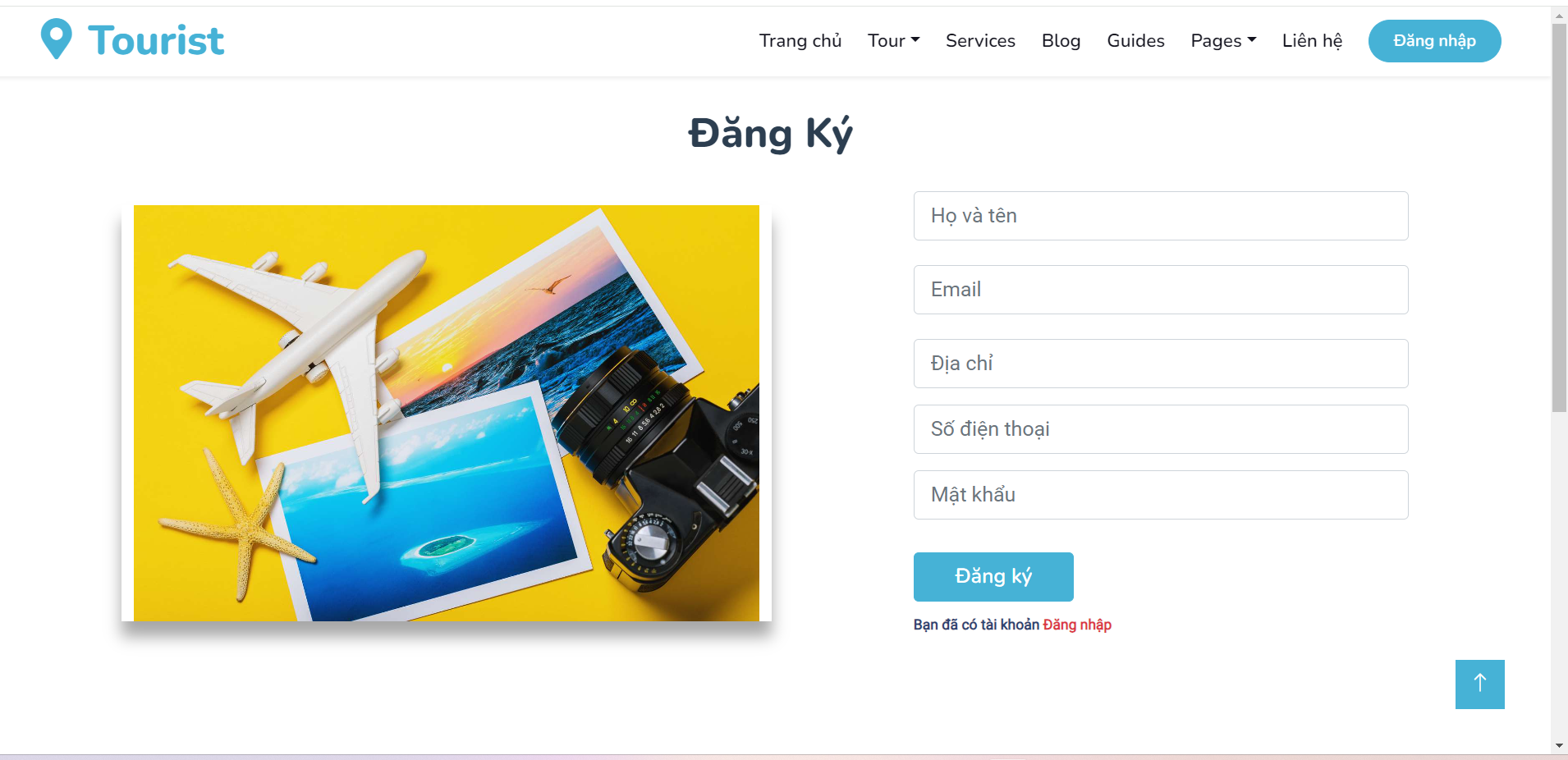


# TASK SHEET REVIEW 2

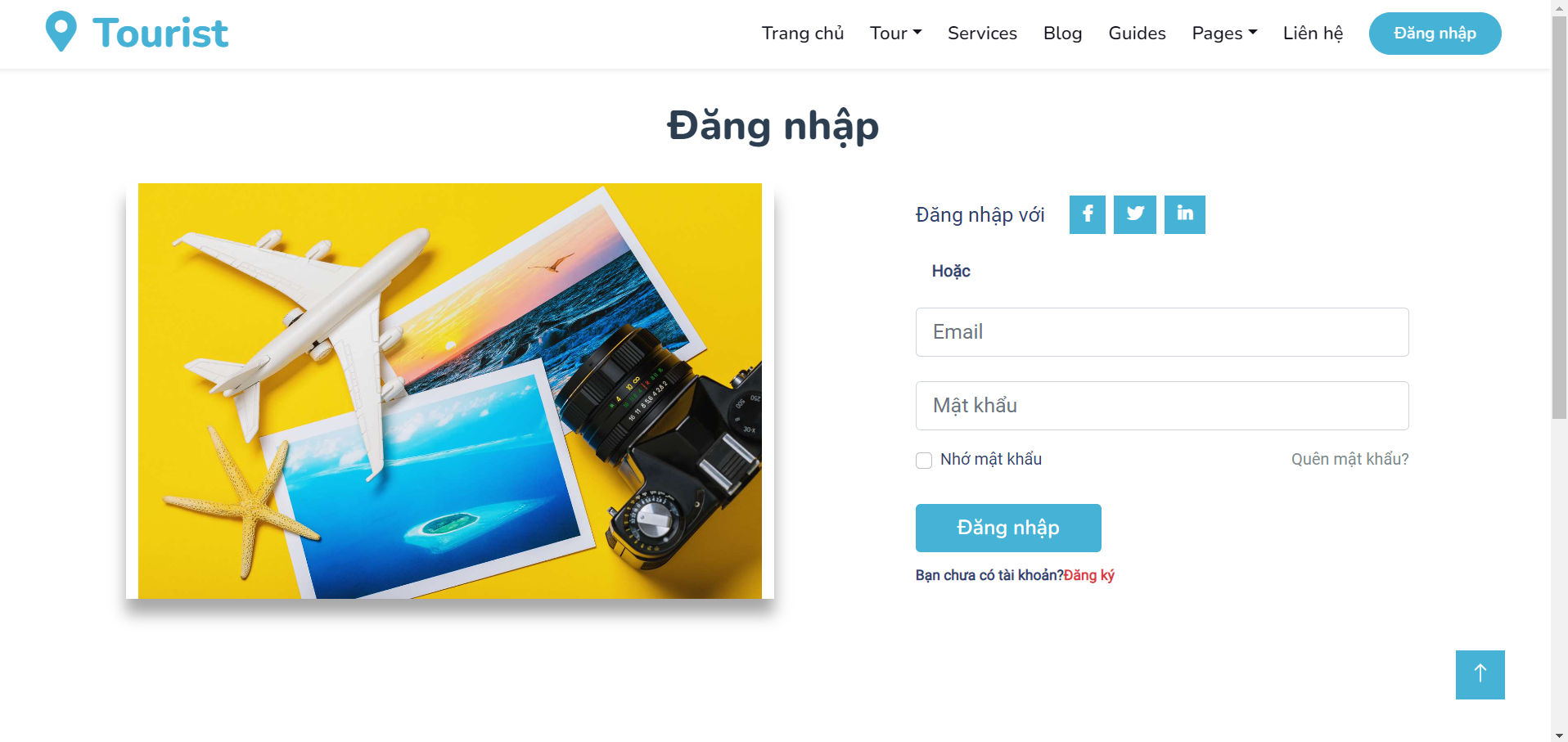
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project: Travel tour** | | | **Date of preparation of activity plan:** | | | |
| **#** | **Task** | **Prepared by** | **Start date** | **days** | **Team member name** | **status** |
| 1 | Use case | Chương | 13/1/2024 | 1 | Chương | completed |
| 2 | Data Flow Diagram(DFD) | Huy | 13/1/2024 | 1 | Huy | completed |
| 3 | Site Map | Huy | 13/1/2024 | 1 | Huy | completed |
| 4 | Entity Relationship Diagram (ERD) | Huy | 13/1/2024 | 1 | Huy | completed |
| Review | | | Signature of instructor | | | |
|  | | |  | | | |
|  | | | **Mr.Pham Cong Danh** | | | |

REVIEW 3

GUI DESIGN

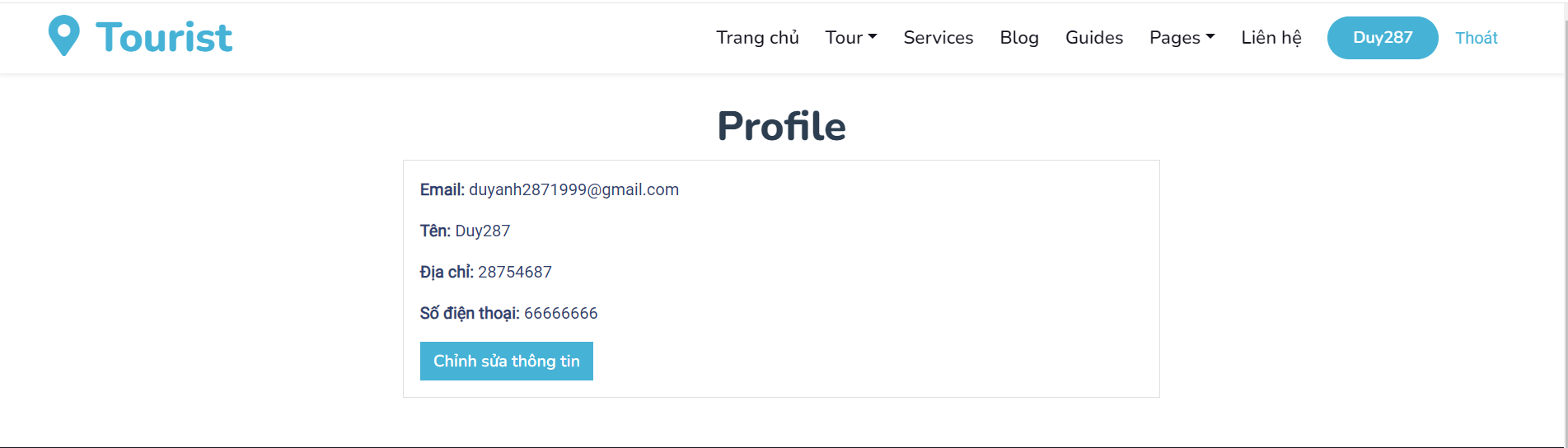
1. User side
   1. Homepage
   2. Register page

User needs to fill all the information to register user account

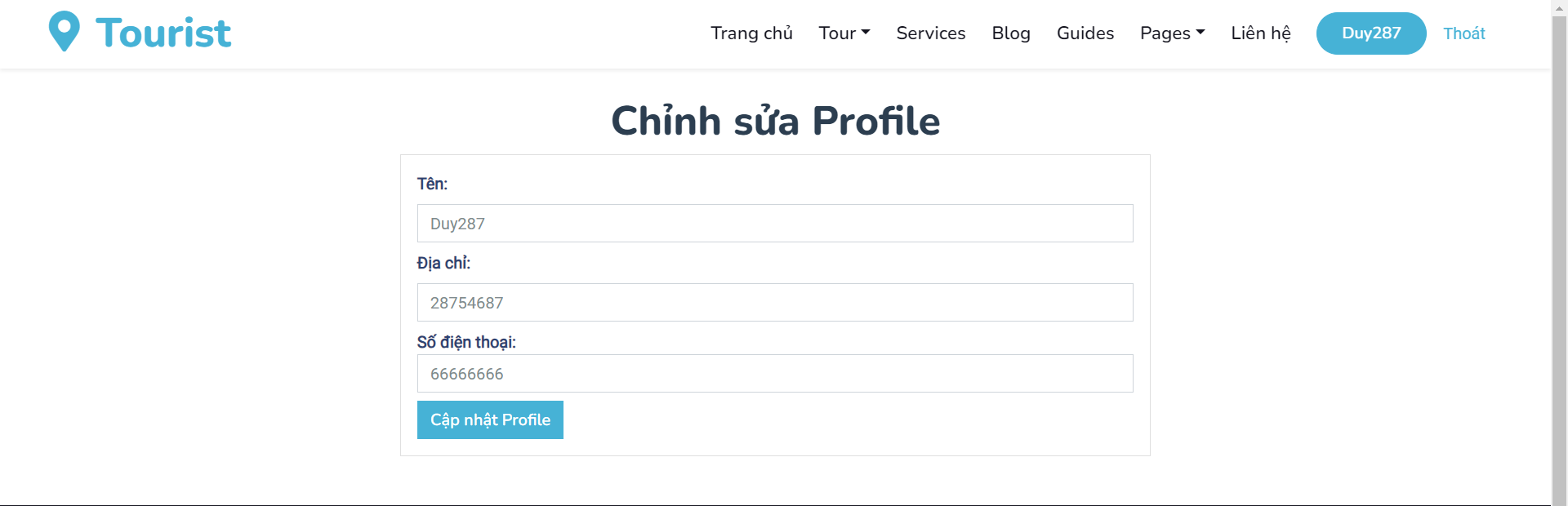
* Name, Email, address, phone number and password
  1. Login form 

***Sử dụng:***

Fill all the required information

* Email, Password(need to match the register info)
  1. Profile page 

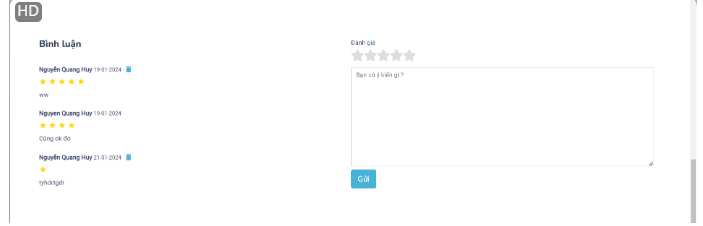
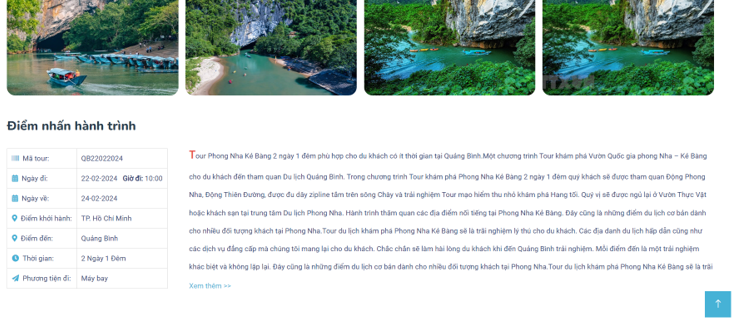
User information and update information o

* + 1. Update information 
  1. Tour page



User can choose tour according to price

User can choose date/month/year to search for tour

* 1. Details page(chưa có hình và thông tin page) 

User can see all the details about the tour like image, time vv

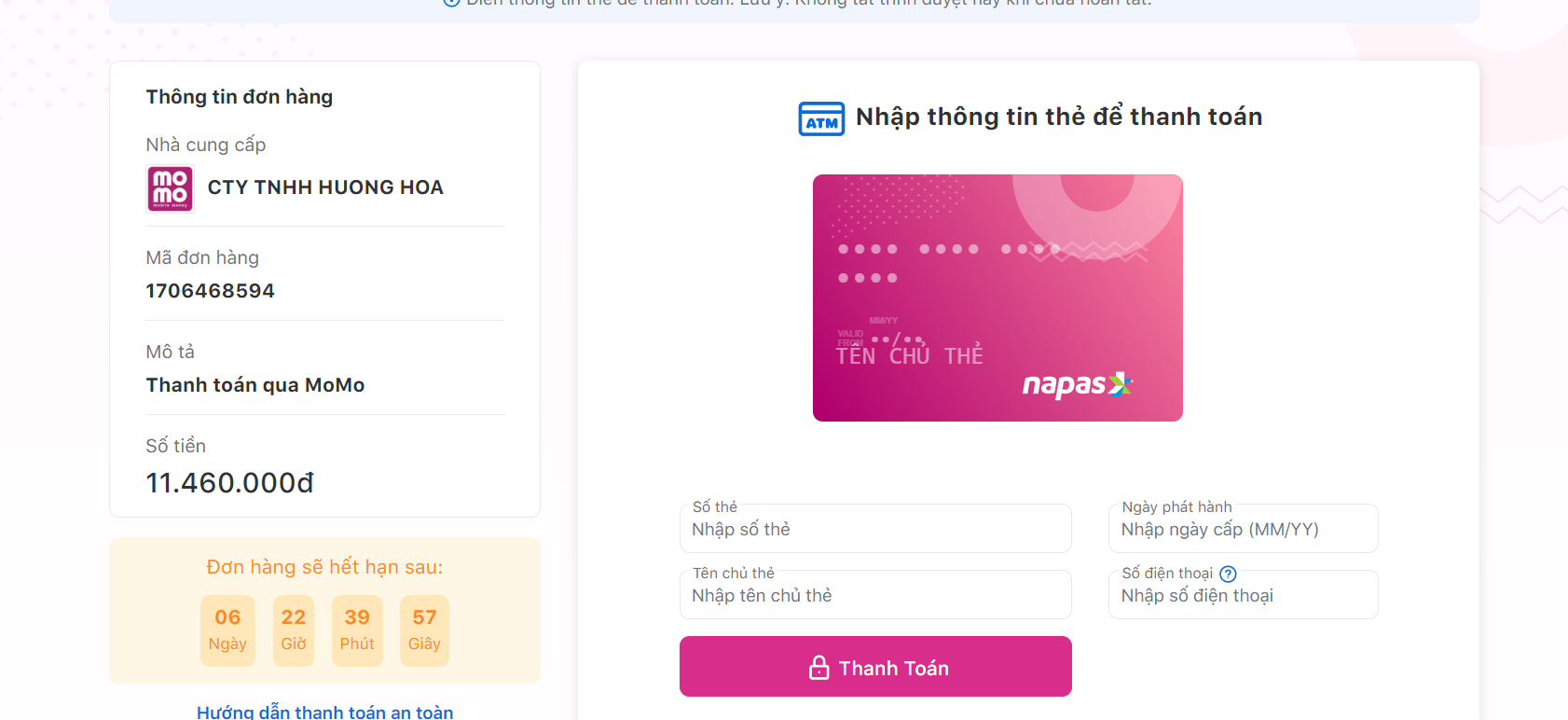
User can comment and rating the tour (need account)

* + 1. Đặt tour(chưa có hình và thông tin page) 

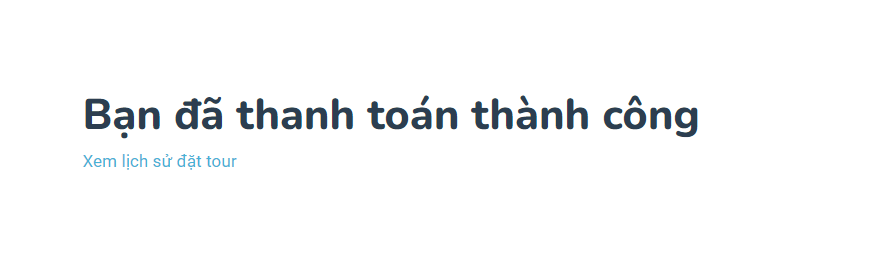
User need to fill in all the information required and then they will see the prices for the tour

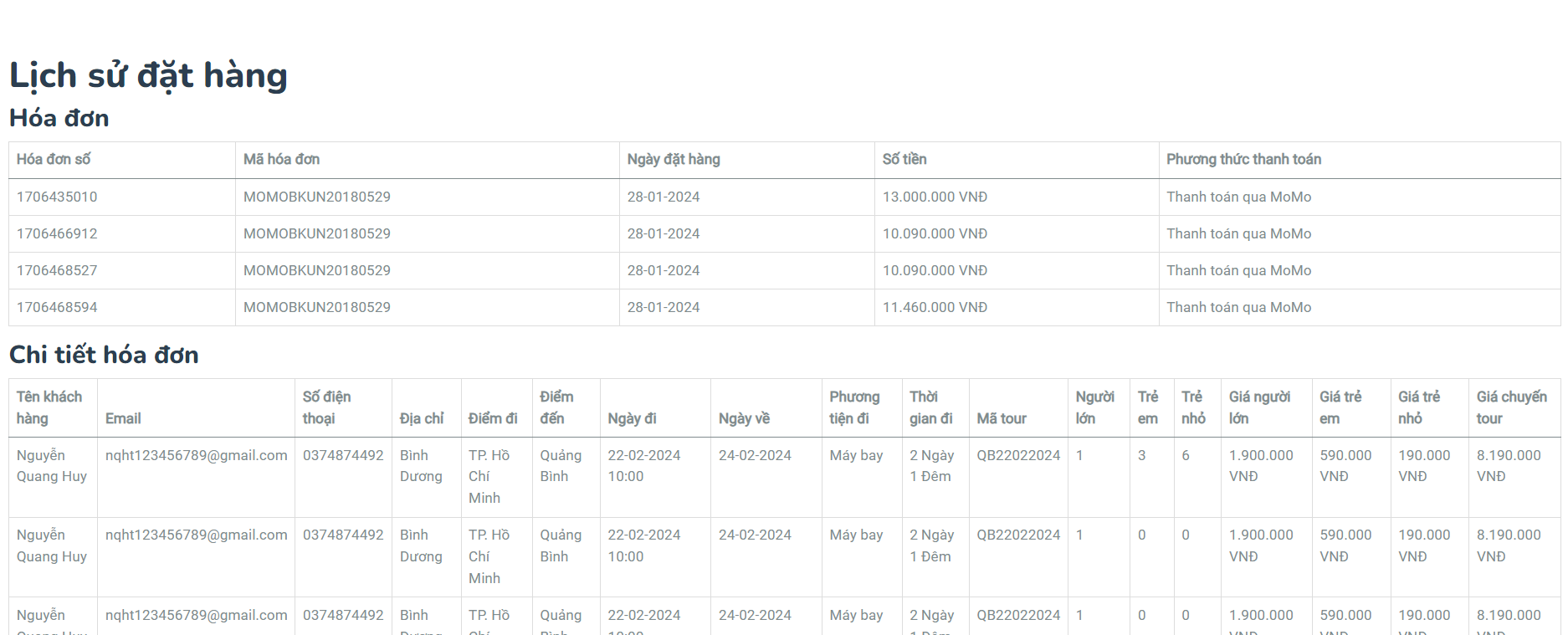
User can also noted thing that they need or want for better tour

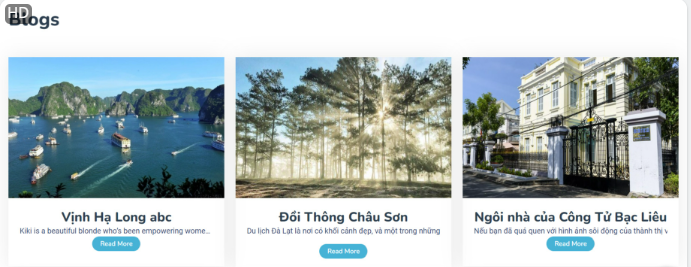
* + 1. Payment****

After user choosing their payment method like momo unother page will open(like below) ****

User need to fill the card info and hit payment button

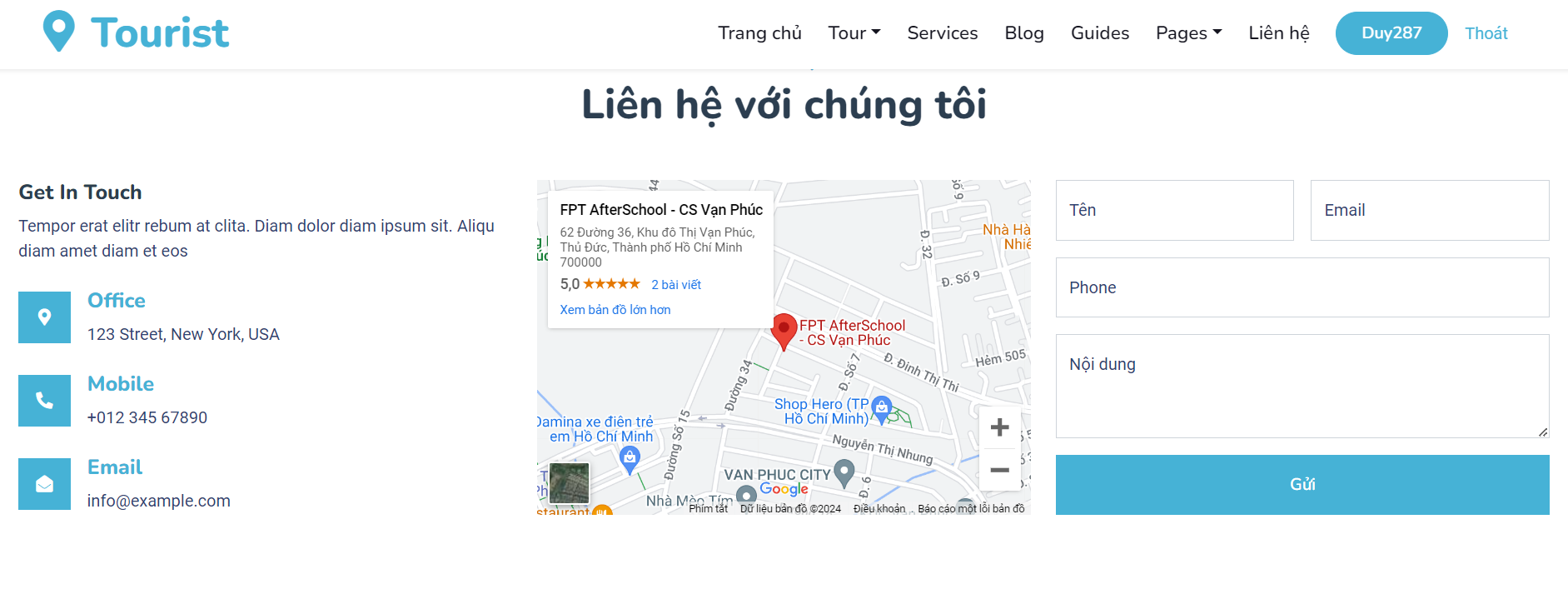
User will recreice notification like this: ****

User can check history of the payment: 

* 1. Blog page

Read all the infor about places and reviews from other tourists

Guide page

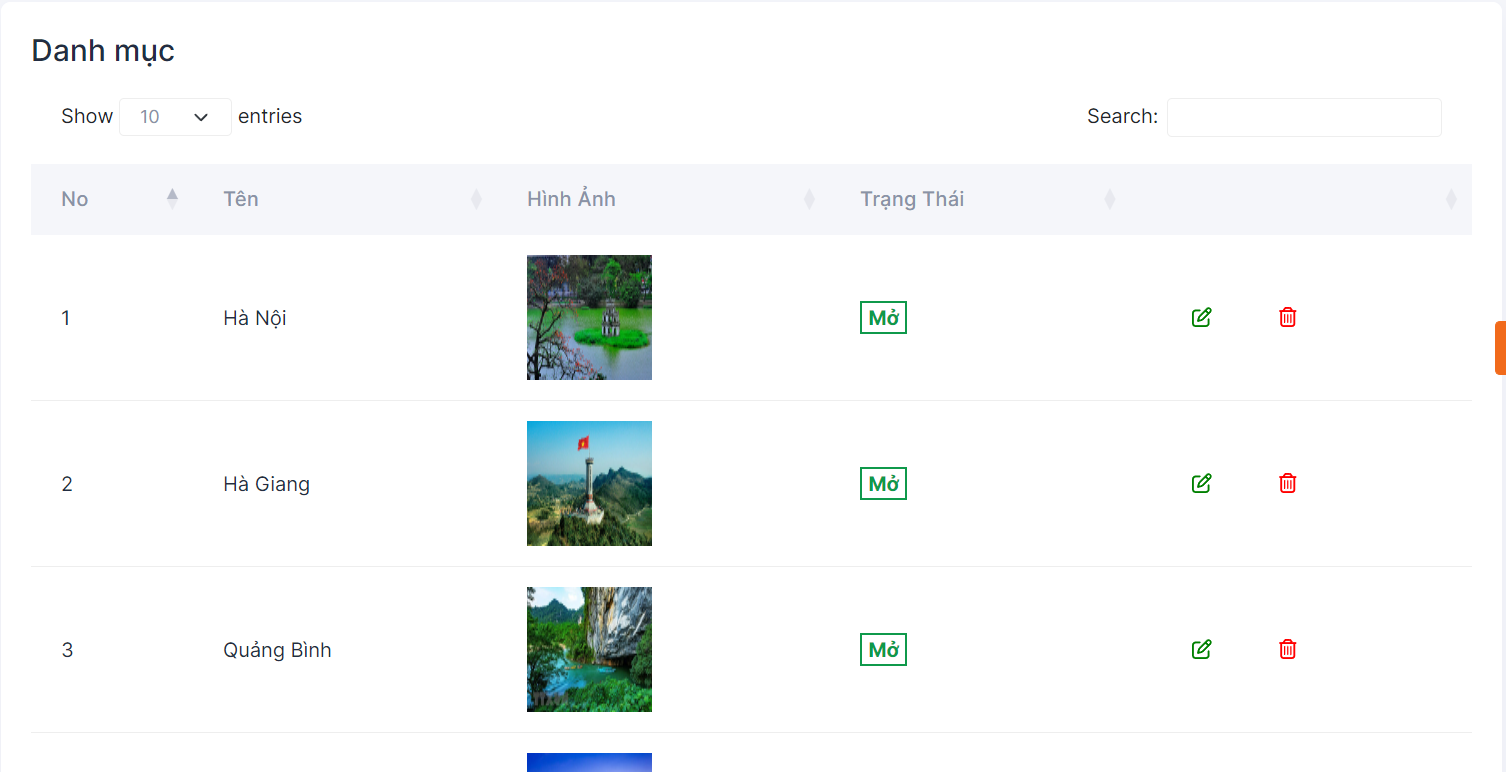
* 1. Contact page 

Suggest and comment about your thought with us to help us improve more(Your information will be secure)

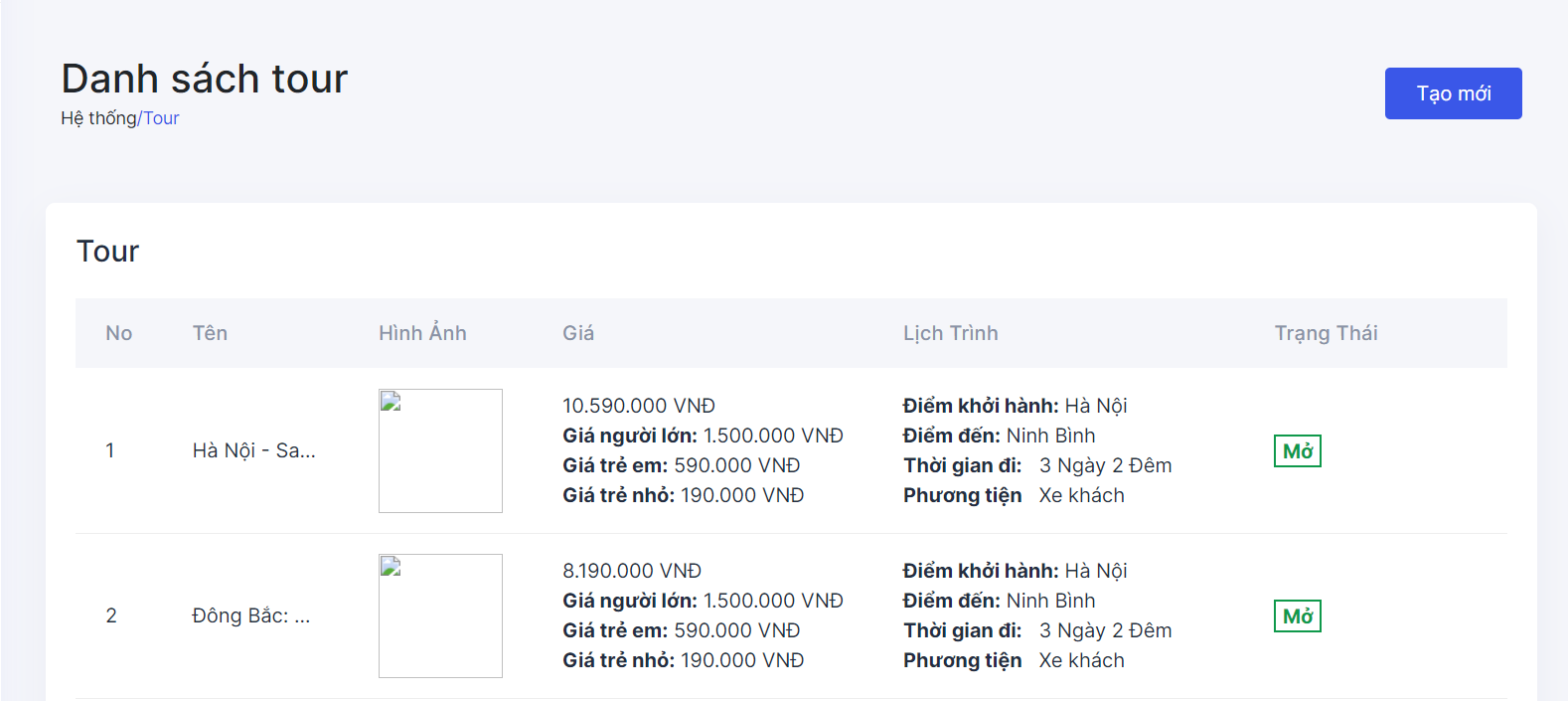
1. Admin side
   1. Admin account list 

Place stored all the user account information

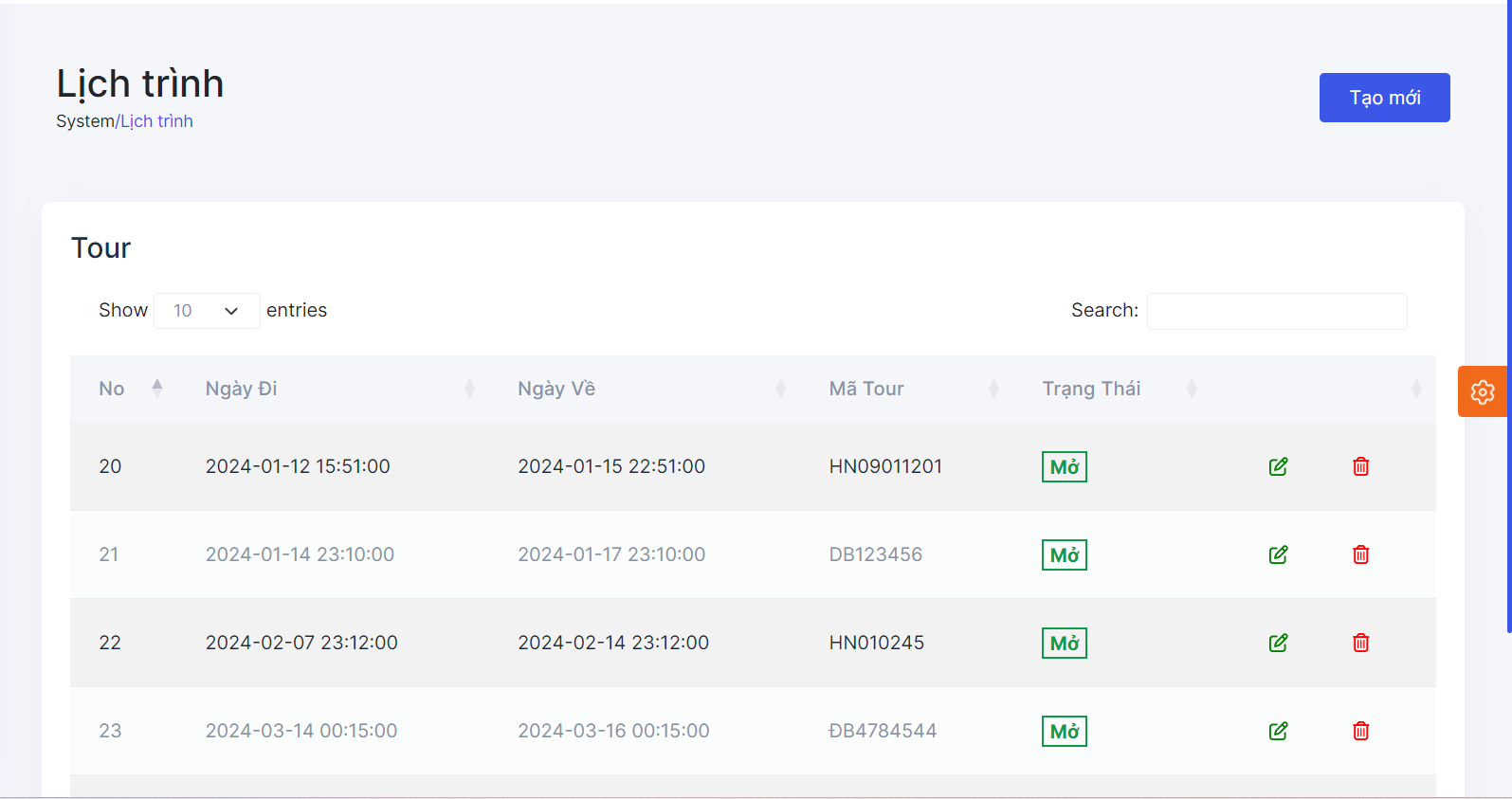
Admin can change role and delete account in this page

* 1. Admin tour list 

List page about all the tour that can be change by admin like status, information and delete tour

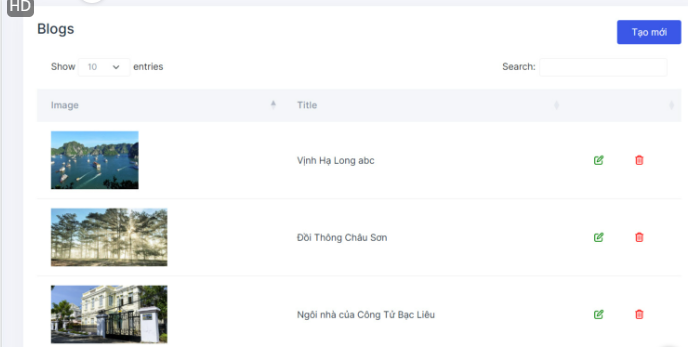
* 1. Admin tour page 

Admin tour list information

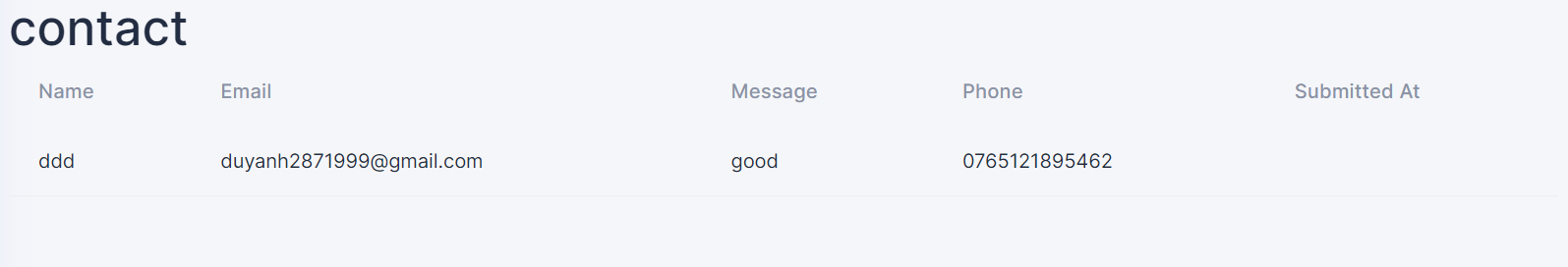
* 1. Admin schedule page 

Time, code and status about the tour manage by admin

Admin can also change status, information and delete tour here

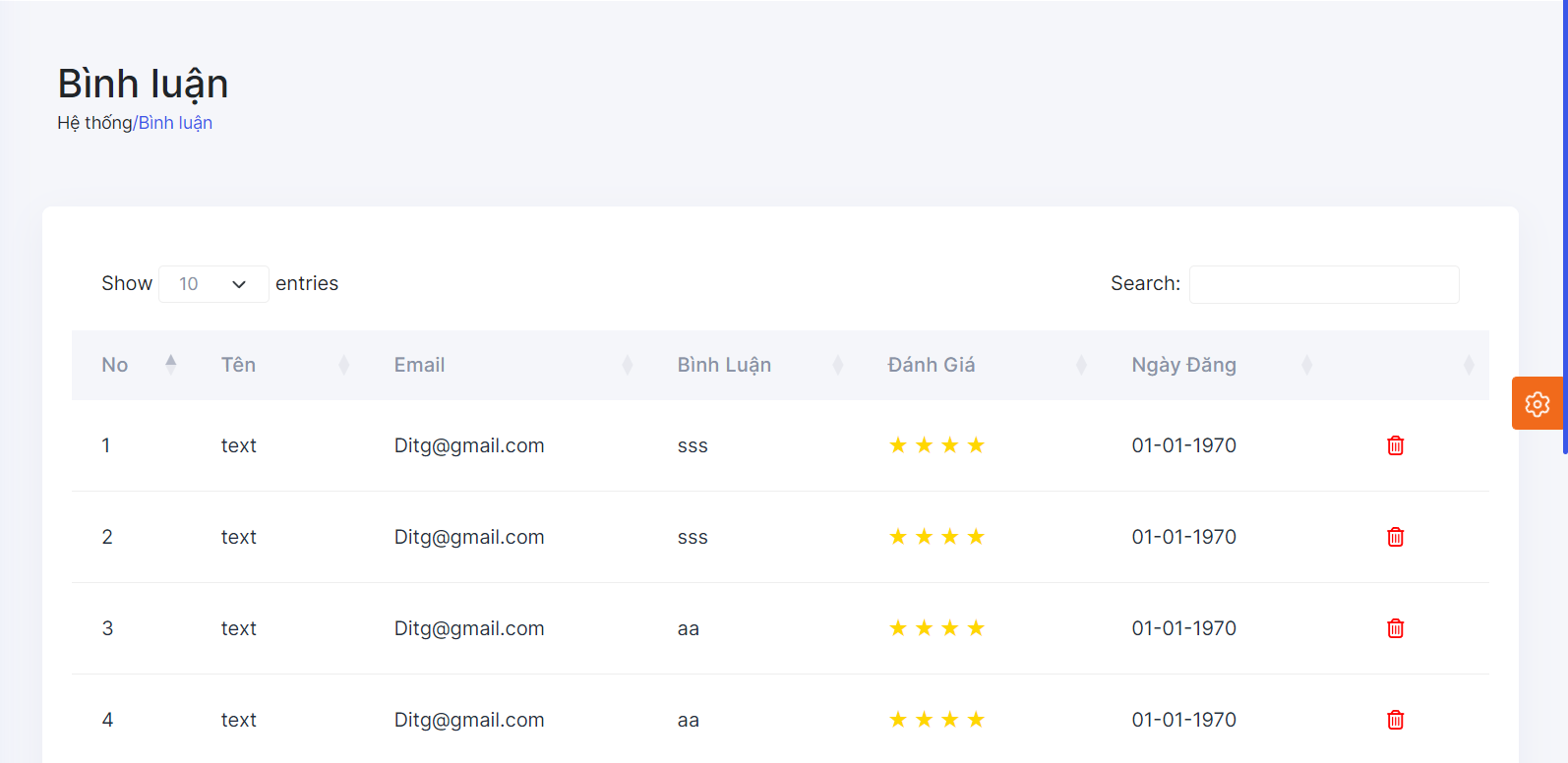
* 1. Admin blog page

Place that contains all the blogs written by tourists  
Admin can change and delete blog in this page

* 1. Admin contact page 

Used:

Place to stored information about customer and message

* 1. Admin comment page 

Used:

Store all the information of user, rating, comment and time

Admin can delete user comment in this page

* 1. Admin Payment page

Used:

Page list of all info about payment of user

Admin can see all the info in this page and can delete it

# TASK SHEET REVIEW 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project: Travel tour** | | | **Date of preparation of activity plan:** | | | |
| **#** | **Task** | **Prepared by** | **Start date** | **days** | **Team member name** | **status** |
| 1 | GUI DESIGN | Duy Anh  Tân Trần | 27/01/2024 | 1 | Duy Anh  Tân Trần | completed |
| Review | | | Signature of instructor | | | |
|  | | |  | | | |
|  | | | **Mr.Pham Cong Danh** | | | |