# 



**<NEXUS AQUARIUM>**

**Documentation**

|  |  |
| --- | --- |
| **Authors** | **C2108L-Group 1** |
| 1. **Cao Thái Tuân** 2. **Nguyễn Minh Hoàng** 3. **Đặng Anh Tuấn** 4. **Nguyễn Quốc Việt** |
| **Date** | **27/05/2022** |
| **Instructor** | **Trần Văn Điệp** |

* **Hà Nội, 2022 -**

**Index**

[1. Problem Definition 3](#_Toc104548152)

[1.1. Problem Abstraction 3](#_Toc104548153)

[1.2. The Current System 3](#_Toc104548154)

[1.3. The Proposed System 4](#_Toc104548155)

[1.3.1. Boundaries of the System 5](#_Toc104548156)

[1.3.2. Hardware and Software Requirements 5](#_Toc104548157)

[2. Customer Requirements Specification 6](#_Toc104548158)

[2.1. Users of the System 6](#_Toc104548159)

[2.2. System functions 6](#_Toc104548160)

[ Customers. 6](#_Toc104548161)

[3. System Designs 10](#_Toc104548162)

[3.1. Entity Relationship Diagram 10](#_Toc104548163)

[3.2. Database Design 14](#_Toc104548164)

[3.3. Sitemap 20](#_Toc104548165)

[3.4. System functions design 21](#_Toc104548166)

[4. Task sheet 53](#_Toc104548167)

[5. Validation Checklists 55](#_Toc104548168)

# Problem Definition

## Problem Abstraction

The Earth's surface is covered by 71% of water, this is also the habitat of many species of animals, creatures, and plants. They have created an extremely diverse and rich underwater ecosystem.

Currently, climate and environmental changes on earth are changing and destroying the habitats of many animals. In which marine ecosystems are affected the most, many species are on the verge of extinction. Nexus Aquarium provides an ideal habitat to help protect animals from the destruction of nature by humans to maintain rare and endangered species of animals.

Nexus Aquarium is one of the city's largest and most famous aquariums. The aquarium is very huge and comprises marine life animals and they are fed and breeding. It is the big center for organizing the school events for study and so on. For the children to visit an aquarium is a big thing as they love sea creatures.

This is a project requested by Aptech, with a website design for Nexus Aquarium to provide useful information for children about life under the vast ocean.

The website is required to provide a user-friendly environment and navigation. Important categories should be stated at the top of the website. Also, there should be a good look and a relatively complete color combination.

The website will be developed for the Windows Platform using HTML5, JavaScript, and Geolocation. The site should work well in all leading browsers including Chrome, IE, Firefox, etc…

## The Current System

With the information provided on the website of Nexus Aquarium, viewers can follow the activities going on here through articles on the website.

The interface at the "Home" page will be information a way to off the news about the species living at the aquarium, articles about events held here, and articles about life in conflict. the animals, creatures, and plants of the ocean.

Forward to the "Animals" page which will list the creatures that live in the Nexus aquarium, articles about the animals will give the little ones more knowledge about them.

In addition, the “Events” and “News” pages will feature the latest news on activities to be held at the Nexus Aquarium as well as news-sharing articles around here.

Users can obtain other information about Nexus Aquarium by visiting the "About Us" page and contacting us directly. I adopt information outside of the “Contact Us” page. Users can give us feedback by submitting information on the “Feedback” page after logging in to the user.

Users can view admission ticket details and book tickets online on the website through the booking page on the “Visit” page. In addition, visitors can book tickets to an ongoing event right on the website while reading the content of that event after logging in to the user.

## The Proposed System

**The purpose is to help users access and use to follow news on the website easily. We have created a website with some of the following features:**

* Registration and login function.
* Function to correct information, retrieve passwords, and confirm email on the user's side.
* Function to add, edit, and delete information, and images to articles about events, news, and animals.
* The function of displaying and searching for article content and animal information through the filter has been optimized.
* Function to schedule events on the website and schedule visits to the aquarium.
* Function to search, receive and send feedback from customers.

## Boundaries of the System

Users can only use functions such as registering, logging in, viewing articles, booking tickets, and sending feedback to the system for the current system.

## Hardware and Software Requirements

**Hardware:**

* Intel Core i3 Processor or higher
* 4GB RAM or higher-
* Color SVGA
* 80 GB Hard Disk space
* CD-ROM or DVD-ROM drive

**Software:**

* Windows 8, 10, or higher
* HTML5 and JavaScript supporting browser
* Bootstrap 3 or higher and jQuery plugins, font awesome 5.
* PHP and framework Laravel 8 or higher.
* VisualCode or other HTML editor.

# Customer Requirements Specification

## Users of the System

Users can only use functions such as registering, logging in, viewing articles, booking tickets, and sending feedback to the system.

1. **Customers.**

* Log in and log out.
* View articles about animals, related articles, and events held at Nexus Aquarium
* Book tickets to events or book tickets to visit Nexus Aquarium.
* Search information content by keyword.
* Send feedback to the system.
* Manage user account information.

1. **Admin.**

* User account management.
* Information management of articles (Animals, News, Event).
* Management of customer feedback.
* Manage booking information of each customer.
* Searching for user and article information

## System functions

## Customers.

1. **View detail.**

|  |  |
| --- | --- |
| **Input** | Select an article to view detailed content |
| **Process** | Access data to get article information |
| **Output** | Display detailed content of the selected article |

1. **Search and filter.**

|  |  |
| --- | --- |
| **Input** | Select the keyword you want to search |
| **Process** | Access data to get article information |
| **Output** | Display search content by keyword |

1. **Booking request**

|  |  |
| --- | --- |
| **Input** | Submit a booking request for an event you want to participate in, including an aquarium visit |
| **Process** | Access data to get information about the event requested to be booked |
| **Output** | Display event information is booked and waiting for confirmation |

1. **Booking confirm**

|  |  |
| --- | --- |
| **Input** | Select pre-requested booking data |
| **Process** | Access the requested booking information |
| **Output** | Display successful booking confirmation and send it to the system. |

1. **Registration User**

|  |  |
| --- | --- |
| **Input** | Enter user information in the form |
| **Process** | Authenticate, submit the form, create a new users account, and complete the transition to login |
| **Output** | Notify back to the user (failed or succeeded) |

1. **Feedback**

|  |  |
| --- | --- |
| **Input** | Enter the content you want to send feedback |
| **Process** | Generate customer feedback data stored in the database |
| **Output** | Notify users who have successfully submitted feedback |

1. **Manage user account**

|  |  |
| --- | --- |
| **Input** | Select user information |
| **Process** | Access data to get user information |
| **Output** | Display user information and allow editing (add, edit, delete information) |

* **Admin.**

1. **View user info**

|  |  |
| --- | --- |
| **Input** | Select the information the user wants to display |
| **Process** | Check the user information stored in the data |
| **Output** | Display user information |

1. **Add new articles**

|  |  |
| --- | --- |
| **Input** | Enter new post information |
| **Process** | Validate, submit, and insert a new record in the database |
| **Output** | Show the new article |

1. **Edit articles**

|  |  |
| --- | --- |
| **Input** | Input new information to be edited and updated |
| **Process** | Validate, submit, and update database |
| **Output** | Show the article after edit |

1. **Delete articles**

|  |  |
| --- | --- |
| **Input** | Select the article to delete |
| **Process** | Delete the record in the database |
| **Output** | Return the home site after deleting the article |

1. **Search order**

|  |  |
| --- | --- |
| **Input** | Enter search name (article, user) |
| **Process** | Search in database |
| **Output** | Display search result |

1. **Manage user feedback**

|  |  |
| --- | --- |
| **Input** | Select user feedback |
| **Process** | Check user feedback stored in a database |
| **Output** | Show response content |

1. **Manage user booking**

|  |  |
| --- | --- |
| **Input** | Select customer information to book tickets |
| **Process** | Check the booking information stored in the database |
| **Output** | Show booking information |

# System Designs

## Entity Relationship Diagram

* **Entity-relationship**
* Entity-relationship of tables in animals

Diagram

Description automatically generated

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

**Image 1: Entity-relationship - Animals**

* Entity-relationship of tables in Posts

Diagram

Description automatically generated

**Image 2: Entity-relationship – Posts**

* Entity-relationship of tables in accounts

**Diagram

Description automatically generated**

**Image 3: Entity-relationship – Account**

* Entity-relationship of tables in event and booking

Diagram

Description automatically generated

**Image 4: Entity-relationship – Event and Booking**

## Database Design

* **Table relationship**

*The relationship between the tables in the database is represented through a diagram:*

**Timeline

Description automatically generated**

**Image 5: Table Relationship**

* **Table detail**
* **Table rule\_account**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | BigInt (20) | -primary key  -unsigned | Rule\_account id |
| Name | Varchar (10) |  | Rule\_account name |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

* **Account**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | BigInt (20) | -primary key  -unsigned | account id |
| Fullname | Varchar (50) |  | User name |
| Email | Varchar (100) |  | User email |
| Phone | Varchar (16) |  | Phone number |
| Address | Varchar (200) |  | User address |
| Gender | Varchar (6) |  | User gender |
| DOB | Date |  | Date of birthday |
| Token | Varchar (255) |  | Token |
| Id\_rule | Begin (20) | Foreign key | Rule id |
| Confirmed | tinyint (1) |  | Confirm email |
| Confirmation | Varchar (255) |  | Confirm code |
| Password | Varchar (255) |  | User password |
| Image | Varchar (255) |  | Image avatar |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

* **Categoties\_animal**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | BigInt (20) | -primary key  -unsigned | Category animal id |
| Name | Varchar (100) |  | Category name |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

* **Ranges**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | BigInt (20) | -primary key  -unsigned | Range id |
| Name | Varchar (255) |  | Range name |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

* **Animals**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | BigInt (20) | -primary key  -unsigned | Animal id |
| Normal\_name | Varchar (50) |  | Animal name |
| Scientific | Varchar (255) |  | Scientific |
| Habitat | Long Text |  | Habitat |
| Overview | Long Text |  | Overview |
| Diet | Long Text |  | Diet |
| Size | Long Text |  | Size |
| Population\_status | Long Text |  | Population\_status |
| Long thumbnail | Varchar (255) |  | Long image |
| Short thumbnail | Varchar (255) |  | Short image |
| HrefParam | Varchar (255) |  |  |
| Id\_Category | UnsignedBigInt (20) | Foreign key | Id\_Category |
| Id\_range | UnsignedBigInt (20) | Foreign key | Id\_range |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

* **Authors**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | bigInt (20) | -primary key  -unsigned | Author id |
| Name | Varchar (200) |  | Author name |
| DOB | date |  | Date of birthday |
| Title | Varchar (255) |  | Job title |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

* **Categories\_post**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | BigInt (20) | -primary key  -unsigned | Category posts id |
| Name | Varchar (100) | Not null | Category posts name |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

* **Posts**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | BigInt (20) | -primary key  -unsigned | account id |
| Title | Varchar (255) |  | account name |
| Description | Long text |  | Description |
| Short thumbnail | Varchar (255) |  | Short image |
| Long thumbnail | Varchar (255) |  | Long image |
| Id\_author | UnsignedBigInt (20) | Foreign key | Author id |
| Id\_category\_posts | UnsignedBigInt (20) | Foreign key | category\_posts id |
| Content | Long text |  | Content |
| Href\_Param | Long text |  |  |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

* **Feedback\_categories**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | BigInt (20) | -primary key  -unsigned | Feedback category id |
| Name | Varchar (255) |  | Feedback category name |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

* **Feedbacks**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | BigInt (20) | -primary key  -unsigned | Feedback id |
| Id\_user | UnsignedBigInt (20) | Foreign key | User id |
| Id\_feedback\_category | UnsignedBigInt (20) | Foreign key | Feedback category id |
| Content | Long text |  |  |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

* **Event\_categories**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | BigInt (20) | -primary key  -unsigned | Event category id |
| Name | Varchar (30) |  | Event category name |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

* **Event**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | BigInt (20) | -primary key  -unsigned | Event id |
| Id event\_category | UnsignedBigInt (20) | Foreign key | event\_category id |
| Title | Varchar (255) |  | Title |
| image | Varchar (255) |  | image |
| Start\_date | Date time |  | Start\_date |
| End\_date | Date time |  | End\_date |
| Price | float |  | Price |
| Location | Varchar (255) |  | Location |
| Href\_Param | Varchar (255) |  | Href\_Param |
| Content | Long text |  | Content |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

* **Booking**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | BigInt (20) | -primary key  -unsigned | booking id |
| Id\_user | UnsignedBigInt (20) | Foreign key | User id |
| Id\_event | UnsignedBigInt (20) | Foreign key | Event id |
| Detail | Varchar (255) |  | Detail |
| Quantity | Int (11) |  | Quantity |
| Price | float |  | Price |
| Arival\_date | Date time |  | Arival\_date |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

* **Rand\_backgrounds**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| id | BigInt (20) | -primary key  -unsigned | Rand id |
| Name | Varchar (255) |  | Rand name |
| Created\_at | Timestamp |  | Create time |
| Updated\_at | Timestamp |  | Update time |

## Sitemap

Diagram

Description automatically generated

**Image 6: Site Map**

## System functions design

* **User’s Interface**
* **Header**

**Graphical user interface, website

Description automatically generated**

**Image 7: Header website**

* **Body**

**A picture containing text, reptile

Description automatically generated**

**Image 8: Body website**

* **Footer & contact**

**Graphical user interface, text

Description automatically generated**

**Image 9: Footer & contact website**

* **Login**

**Graphical user interface, text, application, email

Description automatically generated**

**Image 10: Login page**

* **Register**

**Graphical user interface

Description automatically generated with medium confidence**

**Image 11: Register page**

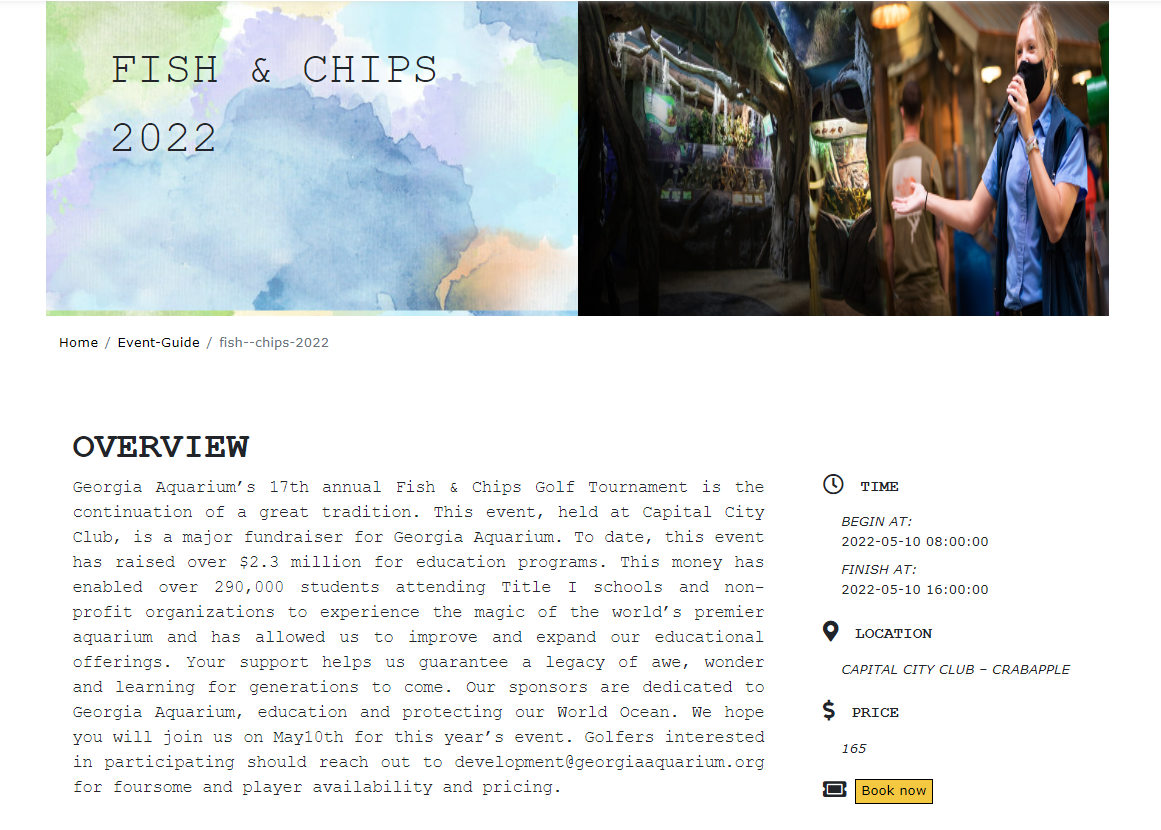
* **Feedback**

**Graphical user interface, application

Description automatically generated**

**Image 12: Feedback page**

* **View event detail**

****

**Image 13: View the event page**

* **Booking**

**Graphical user interface

Description automatically generated**

**Image 14: Booking page**

* **Booking form**

**Graphical user interface, application

Description automatically generated**

**Image 15: Booking form page**

* **Admin’s Interface**
* **Admin home page**

**Graphical user interface, text, application, email

Description automatically generated**

**Image 16: Admin home page**

* **Admin information page**

**Graphical user interface, text, application

Description automatically generated**

**Image 16: Admin home page**

* **Admin add a new page**

**Graphical user interface, text, application, email

Description automatically generated**

**Image 17: Admin adds a new page**

* **Admin edit page**

**Graphical user interface, text, application, email

Description automatically generated**

**Image 17: Admin edit page**

* **Admin views the user detail**

**Table, timeline

Description automatically generated**

**Image 18: Admin views the user detail page**

* **Admin view feedback detail**

**Graphical user interface, text, application, table

Description automatically generated**

**Image 19: Admin views feedback detail page**

* **Algorithm**
* **RULES FOR DRAWING DIAGRAM HANDLING OF BUSINESS PROCEDURES**

|  |  |
| --- | --- |
|  | Steps to take |
|  | Check/Approve/Decide |
|  | Products/Documents |
|  | Begin |
|  | End |
|  | Referencing process |
|  | Automated process |
|  | Processes can be performed or skipped |
|  | Steps to perform outside the system |

1. **User/Customer**

* **Registration**

This process is activated when the customer visits the website and chooses to register.



* **Describe the steps in the business process**

|  |  |  |  |
| --- | --- | --- | --- |
| Step code | Step name | Group of people | Describe |
| 1 | The user chooses to register for a new account | User | When the customer opens the link to the website and selects Register |
| 2 | Fill in your account information | User | The account information includes:   * Fullname * Phone * Email * DOB * Gender * Address * Password * Confirm-Password |
| 3 | Error message | System | Display an error message to the user when the required information is missing |
| 4 | Add new users | System | The system checks the validity of the information and adds a new account to the system. |
| 5 | Error message | System | Display an error message to the user when the process of adding a new account fails. |
| 6 | Go to the homepage | System |  |

* **Login**

When the user accesses the website and logs into the system



* **Describe the steps in the business process**

|  |  |  |  |
| --- | --- | --- | --- |
| Step code | Step name | Group of people | Describe |
| 1 | The user chooses to log in | User | When the customer opens the link to the website and selects Login |
| 2 | Fill in your account information | User | The account information includes:   * Email * password |
| 3 | Error message | System | Display an error message to the user when one of the two pieces of information is missing |
| 4 | Check login information | System | The system checks the validity of the information. |
| 5 | Error message | System | Display an error message to the user when the login credentials check fails |
| 6 | Follow the next procedure | System |  |

* **Searching**

After the user has successfully logged into the system



* **Describe the steps in the business process**

|  |  |  |  |
| --- | --- | --- | --- |
| Step code | Step name | Group of people | Describe |
| 1 | Select search function | User | User selects search function after successful login |
| 2 | Enter the keyword you want to find | User | Enter any characters to search |
| 3 | Provide  suggested keywords | System | Based on some criteria to display suggested keywords with the word you are typing:  **- User's search history**  **- Keywords related to the word being entered**  **- Categories that users are interested in**  **- Hot and trending keywords** |
| 4 | Press search | User |  |
| 5 | Show list of search results | System | The list of search results is displayed in groups:  - Creature name  - Species  - Posts  - Event  - All |

* **View detail and book a ticket**

When users log into the system



* **Describe the steps in the business process**

|  |  |  |  |
| --- | --- | --- | --- |
| Step code | Step name | Group of people | Describe |
| 1 | Select a result from the returned list | User |  |
| 2 | Get details provided or available categories | System | **The details include:**  - Event name  - Organization address  - Start and end time  - Pictures (if any)  - Event entry price  - General information, ongoing services  - Description (if any) |
| 3 | Show detailed information | System |  |
| 3.1 | Error message | System | If the process of obtaining details fails |
| 4 | Choosing to use the service | User | If the user finds the information provided is suitable for his request, the user can click the Submit request button to choose to use the service. |
| 5 | Send require | System | Send booking information including:  - User's name  - Owner's phone number  - Emai  - Notes (If any)  - Detail |
| 5.1 | Error message | System | If the request fails |
| 6 | Successful message | System | If the system sends the request to the user successfully. |
| 7 | Update booking status | System | Orders marked as processing |
| 7.1 | Error message | System | If you can't update your booking status |
| 8 | Complete your order | System | Orders marked as Processed |
| 8.1 | Error message | System | If you can't update your booking status |

* **Process of Viewing User's Profile**

When users log into the system



* **Describe the steps in the business process**

|  |  |  |  |
| --- | --- | --- | --- |
| Step code | Step name | Group of people | Describe |
| 1 | Option to view personal profile | User |  |
| 2 | Get personal information of users | System |  |
| 3 | Return user's personal information | System | **Information includes:**  - Full name  - Email  - Phone number  - Address  - DOB  - Other information |
| 3.1 | Error message | System | If you can't get user information |

* **Feedback**



* **Describe the steps in the business process**

|  |  |  |  |
| --- | --- | --- | --- |
| Step code | Step name | Group of people | Describe |
| 1 | The user enters feedback into the system | User |  |
| 2 | The system receives feedback from the user | System |  |
| 3 | Successful notification and return to the homepage | System |  |
| 3.1 | Error message | System | Error message if no feedback can be sent |

* **Log out**



* **Describe the steps in the business process**

|  |  |  |  |
| --- | --- | --- | --- |
| Step code | Step name | Group of people | Describe |
| 1 | Option to log out of the system | User |  |
| 2 | Proceed to log out users from the system | System |  |
| 3 | Successfully logged out of the system | System |  |
| 3.1 | Error message | System | Error message if logout failed |

1. **Admin**

*In addition to the basic functions of the user, the admin side has several other functions related to managing the content of article information and managing information related to users.*

* **Process of Posting information about marine life, events, articles**



* **Describe the steps in the business process**

|  |  |  |  |
| --- | --- | --- | --- |
| Step code | Step name | Group of people | Describe |
| 1 | Option to  post information | Admin |  |
| 2 | Fill in the information | Admin | Information includes:  **Marine:**  - Names of sea creatures  - Sea creature code  - Picture  - Habitat of sea creatures  - Gender of sea creatures  - Classification of organisms  - Species  **Event:**  - Event name  - Event code  - Picture  - Cost for the event  - Event start time  - Event end time  - Location of the event  - Describe  **Post:**  - Postcode  - Post name  - Description/content of the article  - Picture  - Creation date and modification date  - Status  - Author |
| 3 | Send information | System |  |
| 3.1 | Error message | System | If the upload process fails |
| 4 | Successful message | System | If the new product is successfully posted |

* **Ticket Order Lookup Process**



* **Describe the steps in the business process**

|  |  |  |  |
| --- | --- | --- | --- |
| Step code | Step name | Group of people | Describe |
| 1 | Select the function Lookup ticket orders | Admin |  |
| 2 | Select some search criteria | Admin | Criteria include:  - Ticket booking code  - Time of the ticket order |
| 3 | Search the list of ticket orders | System |  |
| 3.1 | Error message | System | If there is an error while searching |
| 3.2 | Empty message | System | If there are no orders |
| 4 | Show order list | System | If there are orders that satisfy the conditions.  **Information includes:**  - Ticket booking code  - Time  - Customer name  - Customer phone number  - Customer email |

* **Procedure View ticket details**

After looking up the order, the returned results are displayed



* **Describe the steps in the business process**

|  |  |  |  |
| --- | --- | --- | --- |
| Step code | Step name | Group of people | Describe |
| 1 | Choose to view details of an order | Admin |  |
| 2 | Get order details | System |  |
| 2.1 | Error message | System | If there is an error when retrieving information |
| 3 | Show order details | System | **Details include:**  **-** Ticket booking code  - Customer name  - Customer phone number  - Customer email  - Booking date  - Status  - Describe |

* **View feedback details**



* **Describe the steps in the business process**

|  |  |  |  |
| --- | --- | --- | --- |
| Step code | Step name | Group of people | Describe |
| 1 | Select to view user feedback details | Admin |  |
| 2 | Get feedback details | System |  |
| 2.1 | Error message | System | If there is an error when retrieving information |
| 3 | Show feedback details | System | **Details include:**  - Customer name  - Customer's phone number  - Customer's email  - Feedback content |

* **Manage and view user information**



**- Describe the steps in the business process**

|  |  |  |  |
| --- | --- | --- | --- |
| Step code | Step name | Group of people | Describe |
| 1 | Option to view personal profile | Admin |  |
| 2 | Get personal information of users | System |  |
| 2.1 | Error message | System | If you can't get user information |
| 3 | Return user's  personal information | System | **Information includes:**  **-** Full name  - Email  - Phone number  - Address  - DOB |

* **Process of managing marine life information, events, articles**



* **Describe the steps in the business process**

|  |  |  |  |
| --- | --- | --- | --- |
| Step code | Step name | Group of people | Describe |
| 1 | Select to view the products and services under management | Admin |  |
| 2 | Get a list of marine  life information, events, and articles provided | System | **The list includes:**  - Code  - Name  - Content/description  - Category |
| 2.1 | Error message if the list  of marine life  information, events, and articles cannot be obtained. | System |  |
| 3 | Display a list of marine life information, events, articles | System |  |
| 4 | Choose to view any content | Admin |  |
| 4.1 | Editing content | Admin |  |
| 4.2 | Submit new information | System |  |
| 4.3 | Error message | System | If you can't send new information |
| 4.4 | Product displayed with newly corrected information | System |  |
| 5.1 | Delete content | System |  |
| 5.2 | Send information about  the content to be deleted | System |  |
| 5.3 | Error message | System | If the product cannot be deleted |
| 5.4 | Content removed from the list | System | If the product is deleted successfully |
| 6.1 | Add new content | System | Refer to the Process of posting information about marine life, events, articles |

# Task sheet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Start time | Finish time | Implemented by | Evaluation(%) |
| Feasibility study phase | | | | |
| Create memo | 18/04/2022 | 20/04/2022 | Cao Thai Tuan | 30 |
| Contribute opinions | 18/04/2022 | 20/04/2022 | Dang Anh Tuan | 30 |
| Contribute opinions | 18/04/2022 | 20/04/2022 | Nguyen Minh Hoang | 30 |
| Contribute opinions | 18/04/2022 | 20/04/2022 | Nguyen Quoc Viet | 10 |
| Requirement Analysis phase | | | | |
| Create memo | 20/04/2022 | 24/04/2022 | Cao Thai Tuan | 30 |
| Contribute opinions | 20/04/2022 | 24/04/2022 | Dang Anh Tuan | 30 |
| Contribute opinions | 20/04/2022 | 24/04/2022 | Nguyen Minh Hoang | 30 |
| Contribute opinions | 20/04/2022 | 24/04/2022 | Nguyen Quoc Viet | 10 |
| Development phase | | | | |
| BA-Create Database | 25/04/2022 | 26/04/2022 | Cao Thai Tuan | 4% |
| FE-Design Home Page | 25/04/2022 | 26/04/2022 | Nguyen Quoc Viet | 4% |
| FE-Design Animal  Pages (list/detail) | 27/04/2022 | 28/04/2022 | Cao Thai Tuan | 4% |
| FE-Design News Pages (list/detail) | 29/04/2022 | 30/04/2022 | Cao Thai Tuan | 4% |
| FE-Design Event Pages (list/detail) | 02/05/2022 | 03/05/2022 | Cao Thai Tuan | 4% |
| FE-Design Booking Pages (visit & event) | 04/05/2022 | 10/05/2022 | Cao Thai Tuan | 5% |
| FE-Design User Management Pages (list/add/edit/delete) | 27/04/2022 | 30/04/2022 | Nguyen Quoc Viet | 5% |
| FE-Design Feedback Management pages (list/add/edit/delete) | 11/05/2022 | 14/05/2022 | Cao Thai Tuan | 5% |
| FE-Design Login & Register Pages | 15/05/2022 | 17/05/2022 | Cao Thai Tuan | 5% |
| FE-Design About us page | 02/05/2022 | 03/05/2022 | Nguyen Quoc Viet | 4% |
| FE-Design Contact Us page | 04/05/2022 | 05/05/2022 | Nguyen Quoc Viet | 4% |
| FE-Design Ticket Price page | 06/05/2022 | 07/05/2022 | Nguyen Quoc Viet | 4% |
| FE-Design Feedback page | 08/05/2022 | 09/05/2022 | Nguyen Quoc Viet | 4% |
| BE-Design Animal Management logic (list/edit/add/delete) | 25/04/2022 | 28/04/2022 | Dang Anh Tuan | 4% |
| BE-Design User Management logic (list/edit/add/delete) | 26/04/2022 | 29/04/2022 | Nguyen Minh Hoang | 4% |
| BE-Design Posts management and booking logic (list/add/delete/edit) | 29/04/2022 | 03/05/2022 | Dang Anh Tuan | 4% |
| BE-Design Event management logic (list/add/edit/delete) | 03/05/2022 | 06/05/2022 | Nguyen Minh Hoang | 4% |
| BE-Design Random Background, author, range management logic(list/add/edit/delete) | 04/05/2022 | 07/05/2022 | Dang Anh Tuan | 4% |
| Task | Start time | Finish time | Implemented by | Evaluation(%) |
| BE-Design User  Booking logic(add/edit/delete/list) | 08/05/2022 | 11/05/2022 | Dang Anh Tuan | 4% |
| BE-Design Accounts/role-account management  logic (list/add/edit/delete) | 07/05/2022 | 10/05/2022 | Nguyen Minh Hoang | 4% |
| BE-Design  User Login/Logout/Signup/  Forget-Password logic | 11/05/2022 | 14/05/2022 | Nguyen Minh Hoang | 4% |
| BE-Design  User Events pages (list/filter/detail) | 15/05/2022 | 18/05/2022 | Nguyen Minh Hoang | 4% |
| BE-Design  User Animal/News  pages (list/filter/detail) | 12/05/2022 | 15/05/2022 | Dang Anh Tuan | 4% |
| BE-combine code BE and FE | 18/05/2022 | 26/05/2022 | Dang Anh Tuan | 4% |
| Testing phase | | | | |
| FE-Fix Responsive web | 10/05/2022 | 17/05/2022 | Nguyen Quoc Viet | 30% |
| BE-Fix Booking logic | 16/05/2022 | 16/05/2022 | Dang Anh Tuan | 10% |
| BE-Fix Login logic | 19/05/2022 | 19/05/2022 | Nguyen Minh Hoang | 10% |
| BE-Fix Forget-password logic | 20/05/2022 | 20/05/2022 | Nguyen Minh Hoang | 10% |
| BE-Fix Animal  list page’s filter logic | 17/05/2022 | 17/05/2022 | Dang Anh Tuan | 10% |
| FE-Complete web layouts | 18/05/2022 | 25/05/2022 | Nguyen Quoc Viet | 30% |
| Deployment phase | | | | |
| BE-Submit user guide | 21/05/2022 | 25/05/2022 | Nguyen Minh Hoang | 20% |
| BA-Submit project doc | 18/05/2022 | 25/05/2022 | Cao Thai Tuan | 30% |
| BE-Submit developer guide | 23/05/2022 | 26/05/2022 | Dang Anh Tuan | 15% |
| BE-Submit task guide | 25/05/2022 | 26/05/2022 | Dang Anh Tuan | 15% |
| BE-Submit project log | 25/05/2022 | 26/05/2022 | Dang Anh Tuan | 10% |
| Submit a project to Heroku |  |  | Cao Thai Tuan | 10% |

# Validation Checklists

|  |  |
| --- | --- |
| **Option** | **Validated** |
| **Do all text links lead to the appropriate Web site?** | **Yes** |
| **Do all the image links lead to the appropriate Web site?** | **Yes** |
| **Are all the images and links visible on the page?** | **Yes** |
| **Is the Web page taking too long to load?** | **No** |
| **Is the navigation sequence correct through all the Web pages on the site?** | **Yes** |
| **Is the Web page opening properly in all the tested Web browsers?** | **Yes** |
| **Is the Javascript code working as expected in all click events** | **Yes** |