

**The Bird Clinic**

**Software Requirement Specification**

|  | |
| --- | --- |
| Project Code | NJS1706G3 |
| Document Code | NJS1706G3 - v1.0 |
| Group Member | Đặng Kim Minh Tiến- SE172051  Phan Thiên Ân- SE172048  Vũ Minh Nhật- SE172011  Nguyễn Huy Bách- SE172021  Phạm Vĩnh Sơn- SE172457 |
| Supervisor | Mr. Lại Đức Hùng |

**TP HCM,17/06/2023**

**TABLE OF CONTENTS**

[**1 Introduction**](#_heading=h.1fob9te) **4**

[1.1 Purpose](#_heading=h.3znysh7) 4

[1.2 Scope](#_heading=h.2et92p0) 4

[1.3 Definitions, Acronyms, and Abbreviations](#_heading=h.tyjcwt) 4

[1.4 References](#_heading=h.3dy6vkm) 5

[1.5 Overview](#_heading=h.1t3h5sf) 6

[**2 Overall Description**](#_heading=h.4d34og8) **7**

[2.1 User Requirement Specification](#_heading=) 7

[2.1.1. Guest Requirement:](#_heading=h.82oang71uua2) 7

[2.1.2. Customer Requirement:](#_heading=h.ascn86gmw5l1) 7

[2.1.3. Doctor Requirement:](#_heading=h.l964fdhref3x) 7

[2.1.4. Staff Requirement:](#_heading=h.omexua3rsq57) 8

[2.1.5. Admin Requirement:](#_heading=h.crhwuunu30og) 8

[2.2 Business Rules](#_heading=h.2s8eyo1) 8

[**3 FUNCTIONAL Requirements**](#_heading=h.17dp8vu) **9**

[3.1 Use Cases Diagram](#_heading=h.3rdcrjn) 9

[3.2 Use Case Detail](#_heading=) 10

[UC-01 Register](#_heading=h.4kz8gd5qxgw8) 10

[UC-02 Login](#_heading=h.bx242ssyc2o9) 12

[UC-03 View Profile](#_heading=h.c33z2u7c13f2) 13

[UC-04 Edit Profile](#_heading=h.i1e3mad5ayty) 15

[UC-05 Change Password](#_heading=h.2s7ysnhhjzfr) 17

[UC-06 View Users](#_heading=h.827a0gtg8z3) 18

[UC-07 Edit Customer Profile](#_heading=h.f51d7q73dpy9) 20

[UC-09 Update Medical Record](#_heading=h.8q6igkrijvkd) 22

[UC-10 View Medical Record](#_heading=h.jrd5xhozjh17) 24

[UC-11 View Services](#_heading=h.yj1z5jkh7guq) 26

[UC-12: Create Feedback](#_heading=h.i6jl5nxmxvos) 27

[UC-13: View FeedBack List](#_heading=h.qwyzy8uilk7n) 29

[UC-14 View Feedback Detail](#_heading=h.uf4mpdfup92r) 31

[UC-15: View Blog](#_heading=h.3iuui6gggntl) 33

[UC-16: Delete Blog](#_heading=h.vc2r3w2ti2pm) 36

[UC-17: Create Blog](#_heading=h.4oaqxzrun1hl) 38

[UC-18: Edit Blog](#_heading=h.kx9vf0kga5po) 40

[UC-19 Add Bird](#_heading=h.uu88p6am3dtk) 43

[UC-20: Delete Bird](#_heading=h.o244fg7rajzc) 45

[UC-21: View Bird Profile](#_heading=h.3hmufj3vy936) 46

[UC-22: Edit Bird Profile](#_heading=h.a9v5hc66hdfe) 48

[UC-23: Create Reservation](#_heading=h.l681qlc0zhxx) 50

[UC-24: Update Reservation](#_heading=h.2ebctd04yry) 54

[UC-25: Delete Reservation](#_heading=h.4olf1plmdqad) 58

[UC-26: Cancel reservation](#_heading=h.tgznebhvrpgy)60

[UC-27: View Reservation](#_heading=h.t30girx8eeax) 61

[UC-28: Create Doctor](#_heading=h.9rs9kuqh6ku) 63

[UC-29: Reply Feedback](#_heading=h.fp95o87vohkp) 66

[UC-30: Create Staff](#_heading=h.lml6da3q3b5z) 67

[UC-31: Choose Doctor’s timeslots](#_heading=h.pgo31f309ohm) 69

[UC-32: Filter Doctor](#_heading=h.xsfevc4hnw1m) 72

[UC-33: Logout](#_heading=h.rzmln1j64q5b) 73

[UC-34: Delete Users’ Account](#_heading=h.tf30oormr534) 74

[UC-35: Suspend/Activate Users’ Account](#_heading=h.qhwstt75w8jr) 76

[UC-36: Update Service](#_heading=h.kt4tre2sz0x) 78

[UC-37: Add service](#_heading=h.hp50utnfhbpf) 80

[UC-38: Delete service](#_heading=h.1i9yn7x1joji) 81

[**4 NON-FUNCTIONAL Requirements**](#_heading=h.26in1rg) **84**

[4.1 Usability](#_heading=h.lnxbz9) 84

[4.2 Reliability](#_heading=h.35nkun2) 84

[4.3 Performance](#_heading=h.1ksv4uv) 84

[**5 Supporting Information**](#_heading=h.44sinio) **85**

# Introduction

## Purpose

This document illustrates the Software Requirements Specification (SRS) for the **BirdLover** project. The main purpose of this document is to give details about the scope (functional and non-functional) software requirements. Also, this document also describes the design constraints for the logic sub-component in . Along with all the information above, the system’s interface design detail is included in this document.

## Scope

**BirdLover** basically is an online website usable for mobile devices, computer and laptop. **BirdLover** is created to satisfy customer’s demand for an online website which supports all of the problems related to their bird, specifically in the health issue. In **BirdLover**, customers and doctors can have an environment to contact each other, therefore providing the best quality for both customers and their bird.

Specifically, Customer can have a look at the available clinic’s services to choose what is suitable for the bird’s needs. After taking a view about the whole services part, Customer can start making an appointment if they want to book the slot to the clinic based on Customer schedule and the Doctor available time slot. Doctor can receive the appointment information that Customer have made and then decide whether to accept or not. After finishing checking the Bird, Doctor can make a Medical Record to note about the bird’s status, the suitable medicine and finally the total money for the whole process. There is also a blog part where the clinic’s Staff will put information for the Guest and Customer to view. There is a use-case model to show out the correlation relationship between the user and the system, functional and non-functional requirements of the project, an ERD diagram and a relation schema for further research and details.

## Definitions, Acronyms, and Abbreviations

| **Acronym** | **Definition** |
| --- | --- |
| **Guest** | A person using the application without signing in. |
| **Customer** | A person who signed in as a customer, the purpose is to make appointments for the bird and manage the bird’s medical record. |
| **Doctor** | A person who signed in as a doctor, the purpose is to manage the bird’s medical record, give out diagnoses and assign medicine for the bird. |
| **Admin** | A person who signed in as an admin, the purpose is to manage staff, customers and create new accounts. |
| **Staff** | A person who signed in as staff to manage blogs, orders, services, appointment and answer feedback from customers |
| **Reservation** | The process of scheduling appointments with doctors for medical consultations or treatments |
| **Medical Record** | A digital record containing information about a bird's health history, diagnosis, and treatment details |
| **Feedback** | Opinions, comments, or suggestions provided by customers or users about their experience with the clinic's services |
| **Blog** | A section on the website where informative articles or posts related to bird health and care are published |

## References

Refer Source:

**+ Website:**

* <https://www.theanimaldoctors.org/>
* <https://www.petvet.org.uk/>
* <https://www.thebirdclinic.com/>
* <https://animalclinic.co.uk>

Refer User Case Detail: **Grab-Work** Capstone Project

* [Grabwork-Final-Report\_Full (1).docx](https://docs.google.com/document/d/1jCKsVnymHWciNKa1Ck4h6_vv8La6f1Cd/edit?usp=sharing&ouid=113453348335674169476&rtpof=true&sd=true)

## Overview

- The second chapter introduces the Overall Description of this document which provides a better view of the functionality of the website, including the informal requirements to establish the specification required for this project.

- The third chapter of this document is about the Functional Requirements for this project. It briefly describes the functionality of the project in the technical aspect, therefore giving this document a better understanding of what this website can do. Use-cases and details are also included in this chapter.

- The fourth chapter of this document is about the Non-Functional Requirements for this project. It

- The fifth chapter of this document is about the Database of this project. The main purpose for this chapter is to describe the details about the database used in the product and related diagrams and information (specifically, ERD and Relation Schema). There will be details about what is included in the database, the relation of each database and how the database will receive the data and the organization in each database.

# Overall Description

## User Requirement Specification

### 2.1.1. Guest Requirement:

**Guest** is a person who doesn’t have access to the system. Guests can only use some functions in the system. To use all functions, guests must login. These are some functions guest can use:

* Login
* Register
* View Service
* Create Feedback
* Create Reservation
* View Feedback

### 2.1.2. Customer Requirement:

**Customer** is an authorized user who has permission to log in to the system. With customer role, user can perform the following functions:

* View Service
* Create Feedback
* Create Reservation
* Cancel Reservation
* View Feedback
* View and Edit profile
* Manage their birds

### **2.1.3. Doctor Requirement**:

**Doctor** is an authorized user who has permission to log in to the system. With customer role, user can perform the following functions:

* Manage Medical Records
* View Reservation
* Manage Blog

### **2.1.4. Staff Requirement**:

**Staff** is an authorized user who has permission to log in to the system. With customer role, user can perform the following functions:

* Manage Reservation
* Manage Blog
* Manage Feedback
* Manage Service

### **2.1.5. Admin Requirement**:

**Admin** is an authorized user who has permission to log in to the system. With customer role, user can perform the following functions:

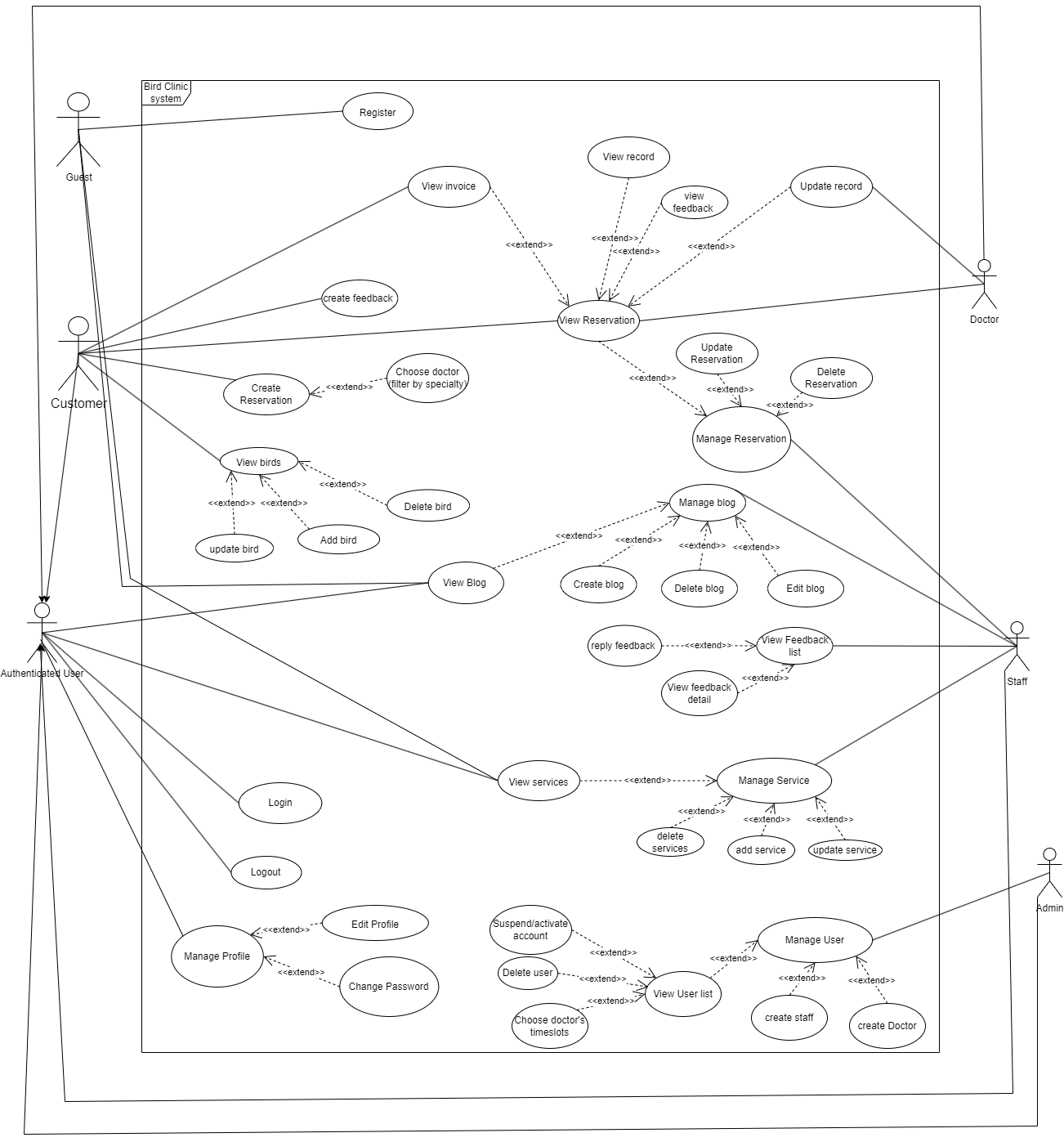
* Manage Customer
* Manage Admin Dashboard
* Manage Reservation

## Business Rules

| **ID** | **Rule Definition** |
| --- | --- |
| BR-01 | Patient must have reservation before visiting the clinic |
| BR-02 | Each Patient must be registered in the system |
| BR-03 | Medical Records should include previous visits, diagnoses, treatment,… |
| BR-04 | The system should generate accurate bills for services provided, including consultations, procedures, medications, and other treatments. |
| BR-05 | The system should provide a user-friendly interface that is accessible and intuitive for users of all levels of technical expertise. |
| BR-06 | The system should provide reports and analytics on various aspects, such as patient statistics, revenue,…and reports should be customizable and accessible by Admin |
| BR-07 | The system should provide users with the ability to rate and provide feedback on their visit to clinic |
| BR-08 | The system should facilitate communication between the clinic and patients/owners, such as appointment reminders, follow-up instructions,… |

# FUNCTIONAL Requirements

## Use Cases Diagram



Link to Diagram: [BirdClinic](https://drive.google.com/file/d/1ZuutLwrnbW-vKp1nW2H3l2OBR9_R2Cl-/view?usp=sharing)

## Use Case Detail

#### UC-01 Register

| **USE CASE-01 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC001 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Register | | | |
| **Author** | Pham Vinh Son | | | |
| **Date** | 20/05/2023 | **Priority** | Normal | |
| **Actor:**   * Guest   **Summary:**   * This use case allows User register to become a customer to use more function   **Goal:**   * The goal is to help a guest become a customer by create an account   **Triggers**   * Guest clicks Register button/link   **Preconditions:**   * A guest that doesn’t have account   **Post Conditions:**  - After creating account successfully, guest gains access to their account on the website  **Main Success Scenario:**   | **No** | **Action** | **System Response** | | --- | --- | --- | | 1 | Guest clicks the register button/links | System showing register form for user with: - Upload Photo: (image, required, maximum 1 picture)  - First Name: String(text input, required, length at least 8 characters.)  - Last Name: String(text input, required, length at least 8 characters.)  - Date of Birth: Date (date input,required)  - Gender: string  - Email: String(free input text, required, regular expression format \b[\w.%+-]+@[\w.-]+\.[a-zA-Z]{2-6}\b)  - Phone: Number: (number input, required, length from 9-11 characters.)  - Address: String(text input, required, length at least 8 characters.)  - District: String(text input, required, length at least 8 characters.)  - City: String(text input, required, length at least 8 characters.)  - Password: String(text input, required, length at least 8 characters.)  - Confirm Password: String(text input, required, length at least 8 characters.) | | 2 | Guest input required information in register for user page | The system will check validate of input (Exceptions: EX1) | | 3 | Guest clicks next button | System will display register form for user’s bird with  - Upload Photo(free input text, optional) - Bird Name: String(text input, required, length at least 8 characters.)  - Breed: String(text input, required, length at least 8 characters.)  - Hatching Date: Date  - Gender: Boolean  - Weight: float  - Band/Microchip : String(text input, required, length at least 8 characters.)  - Feather Color: String(text input, required, length at least 8 characters.) | | 4 | Guest input required information in register for bird page | The system will check validate of input **(Exceptions 1,2)** | | 5 | Guest clicks Done button | Guest and their bird’s information will be added into system |   **Alternative Scenario:**  N/A  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | Guest input valid information | The system will send notification and prompt them to enter correct information | | **2** | Guest input not enough information | The system will prompt them to enter full required information |   **Business Rules:**   * Only Guest can do this function | | | | |

#### UC-02 Login

| **USE CASE-02 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC002 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Login | | | |
| **Author** | Nguyen Huy Bach | | | |
| **Date** | 20/05/2023 | **Priority** | Normal | |
| **Actor:**  Users  **Summary:**  Allow users to login to the system  **Goal:**  users can access their personal accounts on a website securely  **Triggers**  Users want to login to the bird clinic website  **Preconditions**:   * Users must have a registered account with email address and password on website * Users email address and password must be secured in website database * User’s account must be active and not suspended, banned   **Post Conditions:**   * Users login successfully to the website   **Main Success Scenario:**   | Step | Actor action | System response | | --- | --- | --- | | 1 | User access login system | System showing input field include:   * Email:String(free input text, required, regular expression format \b[\w.%+-]+@[\w.-]+\.[a-zA-Z]{2-6}\b)   Password:String(text input, required, length at least 8 characters.) **(Exception 3)** | | 2 | User enter login form include email, password and click “login” | System verifies email and password then redirect user to Home Page **(Exception 1,2)** |   **Alternative Scenario: (N/A)**  **Exceptions:**   | No | Cause | System Response | | --- | --- | --- | | 1 | Unregistered email | system show error message “email or password is not correct” | | 2 | Incorrect password | system show error message “email or password is not correct” | | 3 | Invalid email or password format | system show error message prompt users to input correct format |   **Business Rules:**   * Only guest with registered account can login | | | | |

#### UC-03 View Profile

| **USE CASE-3 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC003 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Profile | | | |
| **Author** | Pham Vinh Son | | | |
| **Date** | 20/05/2023 | **Priority** | Normal | |
| **Actor:**   * Customer, Admin, Doctor   **Summary:**   * This use case allows authenticated user to view their profile   **Goal:**   * The goal is to help authenticated users can view their detailed profile   **Triggers**   * User clicks Profile button   **Preconditions:**   * A user that has authenticated account   **Post Conditions:**  - System show actor’s detailed profile  **Main Success Scenario:**   | **No** | **Action** | **System Response** | | --- | --- | --- | | 1 | User click the Profile button | The system retrieves to profile page and show user’s profile  - First Name String - Last Name: String - Date of Birth: Date - Gender:String - Email: String - Phone: String - Address: String - District: String - City: String |   **Alternative Scenario:**  **N/A**  **Exceptions:**  **N/A**  **Business Rules:**   * Only user have authenticated account can view their profile | | | | |

#### 

#### 

#### 

#### UC-04 Edit Profile

| **USE CASE-4 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC004 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit Profile | | | |
| **Author** | Pham Vinh Son | | | |
| **Date** | 20/05/2023 | **Priority** | Normal | |
| **Actor:**   * Customer, Admin, Doctor   **Summary:**   * This use case allows authenticated user to manage their profile   **Goal:**   * The goal is to help authenticated users manage and update their profile   **Triggers**   * User clicks Profile button   **Preconditions:**   * A user that has authenticated account   **Post Conditions:**  - After updating profile successfully, account’s information is updated accurately in the system  **Main Success Scenario:**   | **No** | **Action** | **System Response** | | --- | --- | --- | | 1 | User click the Profile button | The system retrieves to profile page and show user’s profile  - First Name: String (text input, required, length at least 8 characters. - Last Name: String (text input, required, length at least 8 characters. - Date of Birth: Date - Gender: boolean - Email: String(free input text, required, regular expression format \b[\w.%+-]+@[\w.-]+\.[a-zA-Z]{2-6}\b) - Phone Number: String (Number input, required, length from 9-11 characters). - Address: String (text input, required, length at least 8 characters. - District: String (text input, required, length at least 8 characters. - City: String (text input, required, length at least 8 characters. | | 2 | User change their specify profile information |  | | 3 | User click Done button | The system will update new information accurately  **(Exception 1, 2)** |   **Alternative Scenario:**  **N/A**  **Exceptions:**   | **No** | **Cause** | **System response** | | --- | --- | --- | | 1 | User input invalid fields | System will show a message with corresponding fields | | 2 | System fails on saving changes to database | System will show a message and redirect user to save again |   **Business Rules:**   * Only user have authenticated account can view their profile * Information must be valid before saving to database | | | | |

#### 

#### 

#### 

#### 

#### 

#### 

#### UC-05 Change Password

| **USE CASE-5 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC005 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Change Password | | | |
| **Author** | Pham Vinh Son | | | |
| **Date** | 20/05/2023 | **Priority** | Normal | |
| **Actor:**   * Customer, Admin, Doctor   **Summary:**   * This use case allows authenticated user to change their password   **Goal:**   * The goal is to help authenticated users change their old password into new password   **Triggers**   * User want to change Password   **Preconditions:**   * A user that has authenticated account   **Post Conditions:**  - The password is changed successfully  **Main Success Scenario:**   | **No** | **Action** | **System Response** | | --- | --- | --- | | 1 | User click the Profile button | The system retrieves to change password page and display: - Old Password: String (text input, required, length at least 8 characters. - New Password: String (text input, required, length at least 8 characters. - Input new password again: String (text input, required, length at least 8 characters. | | 2 | User input new password | The system will check validate input **(Exception 1, 2)** | | 3 | After finish changing information, user click Change Password button | The system will update user new password |   **Alternative Scenario:**  **N/A**  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | 1 | User input wrong old password | system send notification and prompt user to input again | | 2 | user input invalid new password | system send notification and prompt user to input again |   **Business Rules:**   * Only user have authenticated account can change their password | | | | |

#### UC-06 View Users

| **USE CASE-6 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC006 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Users | | | |
| **Author** | Pham Vinh Son | | | |
| **Date** | 30/05/2023 | **Priority** | Normal | |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to view user’s information in the bird clinic system   **Goal:**   * The goal of this use case is to display all the user list and show user’s details   **Triggers**   * The view user function is triggered admin need view customer list   **Preconditions:**   * The admin must have access to the clinic's customer management system. * User information is already stored in the system.   **Post Conditions:**  - User information is accurately shown in the clinic's system.  **Main Success Scenario:**   | **No** | **Action** | **System Response** | | --- | --- | --- | | 1 | In the admin’s homepage, the actor clicks the “User list” button | The system displays 3 lists of users with these info:  List 1 (Customer):   * id: string * fullname: string * age: int * gender: string * Phone: String * address: string * status: boolean   List 2 (Staff):   * id: string * fullname: string * Phone: String * DOB: date * account status: boolean   List 3 (Doctor):   * id: string * fullname: string * Phone: String * DOB: date * Specialty: string * account status: boolean   **(Exception 1)** | | 2 | Admin select user’s profile to view | For Customer:   * id: string * customer’s profile img * fullname: string * age: int * gender: string * Phone: String * address: string * account status: boolean * list of birds: array + bird’s img * list of previous appointments.   For Doctor:   * id: string * fullname: string * Phone: String * DOB: date * Specialty: string * degree: string * YearOfExp: tinyint * account status: boolean * show No.SuccessfulAppointment   For Staff:   * id: string * fullname: string * Phone: String * DOB: date * account status: boolean |   **Alternative Scenario:**  **N/A**  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | 1 | User not found | The system display “No user was found" |   **Business Rules:**   * Only Admin role can manage user profile | | | | |

#### UC-07 Edit Customer Profile

| **USE CASE-7 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC007 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit Customer Profile | | | |
| **Author** | Pham Vinh Son | | | |
| **Date** | 30/05/2023 | **Priority** | Normal | |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to manage customer’s information in the bird clinic system   **Goal:**   * The goal of this use case is to effectively handle and update customer information and interactions in the clinic's system.   **Triggers**   * admin wants to update customer information, record new interactions, or retrieve customer details.   **Preconditions:**   * The admin must have access to the clinic's customer management system. * Customer information is already stored in the system.   **Post Conditions:**  - Customer information is accurately updated in the clinic's system.  **Main Success Scenario:**   | **No** | **Action** | **System Response** | | --- | --- | --- | | 1 | Admin open the customer management system | The system presents a list of existing customers:   * Customer’s id: String(text input, required, length at least characters.) * Customer’s name: String(text input, required, length at least 8 characters.) * Customer’s bird: String(text input, required, length at least 8 characters.) | | 2 | Admin search the customer information by name, contact information, unique identifier | The system shows information of desired customer   * First name: String(text input, required, length at least 8 characters.) * Last Name: String(text input, required, length at least 8 characters.) * Email: String(free input text, required, regular expression format \b[\w.%+-]+@[\w.-]+\.[a-zA-Z]{2-6}\b) * Address: String(text input, required, length at least 8 characters.) * City: String(text input, required, length at least 8 characters.) * District: String(text input, required, length at least 8 characters.) * Phone: String(number input, required, length from 9 to 11 characters.) * Appointment date: Date * Bird: String(text input, required, length at least 8 characters.) * Gender: String | | 3 | Admin select customer’s profile to view or update their information | | 4 | Admin verifies the updated information and save changes in the system | The system will update new information accurately |   **Alternative Scenario:**  **N/A**  **Exceptions:**   | **No** | **Cause** | **System response** | | --- | --- | --- | | 1 | System fails on saving changes | System notifies the user that an error has occurred | | System redirect user to update the information again |   **Business Rules:**   * Only Admin role can manage customer profile | | | | |

#### UC-09 Update Medical Record

| **USE CASE-9 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC009 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Update Medical Record | | | |
| **Author** | Phan Thien An | | | |
| **Date** | 31/05/2023 | **Priority** | Normal | |
| **Actor:**  Doctor.  **Summary:**  This sub-use case allows doctors to update a bird's medical record with the latest information.  **Goal:**  Update the bird's medical record with accurate and current information.  **Triggers**  The doctor selects the "Update Record" option for a specific bird's medical record.  **Preconditions:**  - The doctor has access rights to update medical records.  - There are existing birds and their corresponding medical records in the system.  **Post Conditions:**  The bird's medical record is updated with the new information.  **Main Success Scenario:**   | **No** | **Actions** | **System response** | | --- | --- | --- | | **1** | Doctor selects the “Update Record” button **(Exception 2)** | System displays the bird’s current medical record details in form-type with editable fields:   * Bird’s Image: image * Bird’s Name: String(text input, required, length at least 8 characters.) * Gender: boolean * Breed: String(text input, required, length at least 8 characters.) * Band: String(text input, required, length at least 8 characters.) * Age: Float(number input, required, * Microchip: String(text input, required, length at least 8 characters.) * Weight: Number * Hatching Date: Date * Color: String(text input, required, length at least 8 characters.) * Medicine: String(text input, required, length at least 8 characters.) * Days for taking medicine: Number * Diagnosis: String(text input, required, length at least 8 characters.) * Appointment Date: Date * Booking Date: Date | | **2** | Doctor modifies the necessary fields **(Exception 3)** |  | | Doctor finish updating the bird’s medical record | System confirms changes and save the updated medical record to the list **(Exception 1)** | |  | System shows a notification that changes have been saved successfully. | | **3** |  | System redirect user to the medical record list |   **Alternative Scenario:**  **N/A**  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | System fails on saving changes to the list | System notifies user that an error has occurred | | System redirect user to confirm and save the medical record again | | **2** | Unauthorized access | System display an error message | | System redirect users login or registration page | | **3** | The doctor cancels the update process | System discards any changes made | | System redirect user to the medical record details |   **Business Rules:**  - Only authorized doctors can update medical records.  - Each bird's medical record can be updated with accurate and current information. | | | | |

#### UC-10 View Medical Record

| **USE CASE-10 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC010 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Medical Record | | | |
| **Author** | Phan Thien An | | | |
| **Date** | 31/05/2023 | **Priority** | Normal | |
| **Actor:**  Doctor.  Customer  **Summary:**  This sub-use case allows doctors and customers to view a bird's medical record.  **Goal:**  Access and review the bird's medical record.  **Triggers**  The doctor or customer selects the "View Record" option for a specific bird's medical record.  **Preconditions:**  - The user has access rights to view medical records.  - There are existing birds and their corresponding medical records in the system.  **Post Conditions:**  The doctor or customer has reviewed the bird's medical record.  **Main Success Scenario:**   | **No** | **Actions** | **System response** | | --- | --- | --- | | **1** | Doctor or customer selects the “View Record” option. **(Exception 1)** | System displays the bird’s current medical record details:   * Bird’s Image: image * Bird’s Name: String * Gender: String * Breed: String * Band: String * Age: Float * Microchip: String * Weight: Number * Hatching Date: Date * Color: String * Medicine: String * Medical Description: String * Days for taking medicine: Number   And customer details:   * Customer’s Name: String * Customer’s Image: Image | | **2** | Doctor or customer reviews the details in the medical record |  | | **3** | Doctor or customer finish reviewing the details in the medical record |  | | Doctor or customer clicks the “Back” button | System redirects to the Medical Report page |   **Alternative Scenario:**  **N/A**  **Exceptions:**   | No | Cause | System Response | | --- | --- | --- | | 1 | Unauthorized access | System display an error message that user are lack of authorization to access | | Redirect users to login or registration page | | 2 | No medical record is found | System display a notice that no medical record was found |   **Business Rules:**  - Only authorized doctors and customers can view the medical records.  - Each bird's medical record must be shown with accurate and newly updated information.  - Customers can only view the exact medical record for the specific bird. | | | | |

#### UC-11 View Services

| **USE CASE-11 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC011 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Services | | | |
| **Author** | Nguyen Huy Bach | | | |
| **Date** | 01/06/2023 | **Priority** | Low | |
| **Actor:**  Auth User  Guest  **Summary:**  This use case describe the process of how the users and guests can view services in the bird clinic website  **Goal:**  The goal is to allow authorize users and guest to view the service  **Triggers**  Users/Guests clicks into each service in the website  **Preconditions:**   * Service available on the website   **Post Conditions:**   * User get direct to service page   **Main Success Scenario:**   | No | Actor | System | | --- | --- | --- | | 1 | User access service system | Present a list of available services:   * Name of service: String * Image: image * Short describe about service: String | | 2 | User select desired service | System show information of selected service   * Service’s name: String * Image: image * Description about service: String * Price: Float |   **Alternative Scenario:**  N/A  **Exceptions:**  N/A  **Business Rules:**   * The chosen service must be showed with information detail to the authenticated user and guest * The service must be available and updated with new information * Customer and User must be able to choose which service to use | | | | |

#### UC-12: Create Feedback

| **USE CASE-12 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC012 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create Feedback | | | |
| **Author** | Nguyen Huy Bach | | | |
| **Date** | 01/06/2023 | **Priority** | Normal | |
| **Actor:**  Customer  **Summary:**  Allow customer to create a feedback  **Goal:**  Allow customer to share their feedback and experience to the bird clinic  **Triggers**  Customer clicks on create feedback link on navigation bar  **Preconditions:**   * Customer must login in to the web * Customer have accessed the bird clinic’s services   **Post Conditions:**   * Customer successfully create and submit their feedback to the bird clinic   **Main Success Scenario:**   | step | Actor | System | | --- | --- | --- | | 1 | Customer clicks into the “Appointment” option in the dashboard | System shows out the appointment list with following information:   * Doctor’s full Name: String * Appointment’s Date and Time: Date * Booking Date: Date * Customer’s Name(Doctor’s Page): String * Customer’s bird name(Doctor’s Page): String * Amount: Float * Status: String   And a “View Appointment” button | | 2 | Customer clicks into the “View Appointment” button | System shows out the appointment’s detail page with following information:   * Appointment’s Date and Time: Date * Customer’s bird Name (Doctor’s Page): String * Doctor’s fullName: String * Doctor’s Time Slot: Date * Service’s Name: String * Status: String * Booking Date: Date * Customer’s Name: String * Notes: String   And a “Create Feedback” button | | 3 | Customer open “Create Feedback” | System provide input form with the fields include:  - Full name: string(text input, required, length at least 8 characters.)  - Email: string(free input text, required, regular expression format \b[\w.%+-]+@[\w.-]+\.[a-zA-Z]{2-6}\b)  - Phone: String(number input, required, length from 9 to 11 characters.)  - Yours birds:string(text input, required, length at least 8 characters.)  - Which service: string(text input, required, length at least 8 characters.)  - appointment id: String (text input, required, length at least 8 characters.)  - What you think:string (text input, required, length at least 8 characters.) | | 2 | Customer fill in user name and contact information, write feedback message in provided form | System validate feedback message | | 3 | Customer submit feedback | System stores the feedback in database |   **Alternative Scenario:**  **Exceptions:**   | **No** | **Cause** | **Response** | | --- | --- | --- | | 1 | Invalid or incomplete form submission | The system will send notification and tell Users to finish or enter correct information |   **Business Rules:**   * Only customers can create feedback * The new feedback must be added in the feedback list * Customer can only create feedback after using the service * The created feedback must be matched with the chosen appointment | | | | |

#### UC-13: View FeedBack List

| **USE CASE-13 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC013 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Feedback List | | | |
| **Author** | Nguyen Huy Bach | | | |
| **Date** | 01/06/2023 | **Priority** | Low | |
| **Actor:**  Staffs  **Summary:**  Detail process to view feedback  **Goal:**  Staffs get to view the feedback from customer  **Triggers**  Staffs want to view feedback from customer  **Preconditions:**   * Staffs signed in to the web * At least one feedback   **Post Conditions:**   * Staff get to view feedback on the list     **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | Staffs open “Manage feedback” system | The system present the feedback data in a list   * Date the feedback was created: Date * Name of customer: String * Title of the feedback: String   And “Detail” button to view the feedback content |   **Alternative Scenario: (N/A)**  **Exceptions:**   | **No** | **Cause** | **Response** | | --- | --- | --- | | 1 | No feedback in the list | System display a notice that no feedback are available to view | | 2 | Unauthorized access | System display a notice that user are lack of authorization | | System redirect user to login again |   **Business Rules:**   * Latest feedback will show up on top of the list * There must be at least one feedback available for viewing in the list * Only Staff are able to view the feedback list | | | | |

#### 

#### UC-14 View Feedback Detail

| **USE CASE-14 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC014 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Feedback detail | | | |
| **Author** | Nguyen Huy Bach | | | |
| **Date** | 01/06/2023 | **Priority** | Low | |
| **Actor:**  Staffs  **Summary:**  Detail process to view feedback detail  **Goal:**  Staffs get to view the detail feedback from customer  **Triggers:**  Staffs clicks into the “View Detail” in the feedback list  **Preconditions:**   * Staff signed in to the web   **Post Conditions:**   * Staff get to view feedback detail on the feedback detail page   **Main Success Scenario:**   | **Step** | **Actor action** | **System action** | | --- | --- | --- | | 1 | Staffs click on a desired feedback | The system show the detail information of selected feedback:   * Name of customer: String * Theirs birds: String * Date of medication: Date * title of the feedback: String * Content of the feedback: String | | 2 | Read the feedback detail |  |   **Alternative Scenario:**   | **No** | **Action** | **Response** | | --- | --- | --- | | 1 | Staff click on manage reservation and get to reservation list, filter reservation that status is done or canceled | Present the reservation list from customer include:   * Name of customer: string * ID of customer: integer * Date and time of appointment: integer * Which doctor: string * Which service: string * “feedback” button | | 2 | Staff click on “feedback” button | Show the detail information of selected feedback:   * Name of customer: String * ID of customer: integer * Bird’s Name:string * Gender: string * Age: Float * species: String * Date and time of appointment: Date * Content from customer: String | | 3 | Staff read the feedback |  |   **Exceptions:**  N/A  **Business Rules:**   * Feedback for Appointments which status are in processing can not be viewed * Only Staff are able to view the feedback detail * Each Feedback information must be matched with the specific appointment made by the customer. | | | | |

#### 

#### 

#### 

#### UC-15: View Blog

| **USE CASE-15 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC016 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Blog | | | |
| **Author** | Phan Thien An | | | |
| **Date** | 31/05/2023 | **Priority** | Normal | |
| **Actor:**  Authenticated User.  **Summary:**  This use case allows authenticated users to view a specific blog post on the bird clinic website.  **Goal:**  Allow staff members to edit and update existing blog posts to update new information and ensure that the blogs are correctly informed.  **Triggers**  The staff want to edit content and information in blog posts.  **Preconditions:**  - Authenticated users are logged into the system.  - At least one blog exists.  **Post Conditions:**  Blog posts are updated with new content and information in the blog database.  **Main Success Scenario:**   | **No** | **Actions** | **System response** | | --- | --- | --- | | **1** | Authenticated user clicks on the blog section **(Exception 2)** | System navigates user to the blog section | | **2** | Authenticated user selects a specific blog post from the list | System retrieves and displays the details of the chosen blog:   * Title: String * Author: String * Post Date: Date * Content: String   **(Exception 1)** |   **Alternative Scenario:**  Search Bar for Blog Posts:   | **No** | **Actions** | **System response** | | --- | --- | --- | | **1** | Authenticated user clicks on the blog section **(Exception 2)** | System navigates user to the blog section | | **2** | Authenticated user clicks on a search bar on the page |  | | Authenticated user enters relevant keyword into the search bar | | Authenticated user clicks on the search button | System performs searches and present matching result **(Exception 1)** | | **3** | Authenticated user clicks on the desired blog post | System retrieves and display information of the blog post:   * Title: String * Tag: String * Author: String * Context: String |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | 1 | There isn’t any blog in the blog list | System display an error that there are no blog in the blog list to view | | 2 | Unauthorized access | System display error message that user need to login to another account | | System redirect users to login or registration page |   **Business Rules:**  - Only authenticated staff members can edit blog posts.  - The staff members can only edit the blog posts they have created.  - The staff member can modify the title, content, and category of the blog post.  - The edited blog post should be updated in the database with the modified information, including the update timestamp. | | | | |

#### 

#### 

#### 

#### 

#### UC-16: Delete Blog

| **USE CASE-16 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC016 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Delete Blog | | | |
| **Author** | Phan Thien An | | | |
| **Date** | 31/05/2023 | **Priority** | Normal | |
| **Actor:**  Staff member  **Summary:**  This use case allows staff members to delete an existing blog from the blog management system.  **Goal:**  Enable staff to remove unwanted or outdated blog posts from the website.  **Triggers**  The staff clicks into the “delete” button at incorrect or outdated blog posts in the blog list.  **Preconditions:**  - Staff members are logged into the system.  - At least one blog exists.  **Post Conditions:**  Blog posts are deleted from the blog database.  **Main Success Scenario:**   | **No** | **Actions** | **System response** | | --- | --- | --- | | **1** | Staff member selects a specific blog post in the blog list. **(Exception 3)** | System shows out the blog details with information:   * Title: String * Tag: String * Author: String * Context: String   And a “Edit”, “Delete” button | | **2** | Staff member confirms the deletion by clicking into the “Delete” button **(Exception 1)** | System retrieves and removes the blog post from the database **(Exception 2)** | | **3** |  | System confirms successfully delete | | System redirects staff to the blog management section |   **Alternative Scenario:**  **N/A**  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | 1 | Users decide not to make any changes | System redirect user to the blog list page | | 2 | System fails on saving changes to the list | System notifies users that an error has occurred while saving | | System redirect user to confirm the deletion again | | 3 | Unauthorized access | System display an error message about lack of authorization | | System redirect users to login or registration page | | 4 | No existing blog | System shows out an error message that there isn’t any blog in the list |   **Business Rules:**  - Only authenticated staff members can delete blog posts.  - The staff members can only delete the blog posts they have created.  - When a blog post is deleted, it should be permanently removed from the database and no longer accessible to users. | | | | |

#### UC-17: Create Blog Post

| **USE CASE-17 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC017 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create Blog Post | | | |
| **Author** | Phan Thien An | | | |
| **Date** | 31/05/2023 | **Priority** | Normal | |
| **Actor:**  Staff Member  **Summary:**  This use case is used for creating a new blog post in the bird clinic website's blog management system.  **Goal:**  Allow the staff members to create and publish new blog posts on the website.  **Triggers**  The staff want to create a new blog for sharing information.  **Preconditions:**  Staff members are logged into the system.  **Post Conditions:**  New blog posts are created and added to the blog database.  **Main Success Scenario:**   | **No** | **Actions** | **System response** | | --- | --- | --- | | **1** | Staff selects the “Create blog” option **(Exception 3)** | System presents a form to enter details for the blog post:   * Author: String(text input, required, length at least 4 characters.) * Title: String(text input, required, length at least 4 characters.) * Content: String(text input, required, length at least 8 characters.)   **(Exception 4)** | | **2** | Staff member enter the required information for the blog post **(Exception 1)** |  | | The staff add Tag: String(text input, required, length at least 4 characters) for further searching | The system validates the blogs information and format | |  | The system saves the tag and the new information of the blog post **(Exception 2)** | | **3** |  | System redirects the staff to the blog management section |   **Alternative Scenario: N/A**  **Exceptions:**   | **No** | **Cause** | **System response** | | --- | --- | --- | | **1** | Users decide not to make any changes | System redirect user to the blog list management section. | | **2** | System fails on saving changes to the list | System notifies users that an error has occurred | | System redirect user to confirm creating the blog again. | | **3** | Unauthorized access | System display an error message | | Redirect users to login or registration page | | **4** | Incorrect format | System display a error in the specific field that user has entered wrong format |   **Business Rules:**  - Only authenticated staff members can create blog posts.  - The staff member must provide the title, content, and select a category for the blog post.  - The blog post should be saved in the database with the relevant information. | | | | |
|  |  |  |  |  |

#### 

#### 

#### 

#### 

#### 

#### UC-18: Edit Blog

| **USE CASE-18 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC018 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit Blog | | | |
| **Author** | Phan Thien An | | | |
| **Date** | 31/05/2023 | **Priority** | Normal | |
| **Actor:**  Staff member  **Summary:**  This use case allows staff members to edit an existing blog from the blog management system.  **Goal:**  Allow staff members to edit and update existing blog posts to update new information and ensure that the blogs are correctly informed.  **Triggers**  The staff want to edit content and information in blog posts.  **Preconditions:**  - Staff members are logged into the system.  - At least one blog exists.  **Post Conditions:**  Blog posts are updated with new content and information in the blog database.  **Main Success Scenario:**   | **No** | **Actions** | **System response** | | --- | --- | --- | | **1** | Staff selects a specific blog post in the blog list **(Exception 3)** | System retrieves details of the blog post:   * Author: String * Title: String * Post Date: Date * Content: String * Tag: String   And a “Edit”, “Delete” button | | **2** | Staff clicks into the “Edit” button in a specific blog | System shows out a form with previous information:   * Author: String(text input, required, length at least 4 characters.) * Title: String(text input, required, length at least 8 characters.) * Content: String(text input, required, length at least 8 characters.) * Tag: String(text input, required, length at least 4 characters) | | **2** | Staff edits in the form **(Exception 1)** | System validated the information’s form | | Staff submits the form | | **3** |  | System confirms successful update of the blog post **(Exception 2)** | | System redirects the staff to the blog management section |   **Alternative Scenario: (Not Available)**  **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | **1** | User decide not to make any changes | System redirect user to the blog list page | | **2** | System fails on saving changes to the list | System display notifies users that an error has occurred | | System redirect user to confirm and saves changes again | | **3** | Unauthorized access | System display an error message | | redirect users to login or registration page | | **4** | Invalid or missing form data | System display an error and highlights the invalid fields for the user to edit again |   **Business Rules:**  - Only authenticated staff members can edit blog posts.  - The staff members can only edit the blog posts they have created.  - The staff member can modify the title, content, and category of the blog post.  - The edited blog post should be updated in the database with the modified information, including the update timestamp. | | | | |

#### 

#### 

#### 

#### 

#### 

#### 

#### UC-19 Add Bird

| **USE CASE-19 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC019 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Add Bird | | | |
| **Author** | Vu Minh Nhat | | | |
| **Date** | 01/06/2023 | **Priority** | Normal | |
| **Actor:**  Customer  **Summary:**  System allows customer to add their bird to the bird clinic  **Goal:**  Customer add their bird to the bird clinic accurately  **Triggers:**  Customer want to add their bird to the bird clinic  **Preconditions:**   * Customer must login in to the web * Customer must have a bird requiring medical care   **Post Conditions:**   * Customer successfully create and adding their bird to the bird clinic   **Main Success Scenario:**   | **Step** | **Actor** | **System** | | --- | --- | --- | | 1 | Customers open “Add bird” | System showing input form include:   * Bird’s Image: image * Bird’s Name: String(text input, required, length at least 8 characters.) * Gender: String(text input, required, length at least 8 characters.) * Breed: String(text input, required, length at least 8 characters.) * Band: String(text input, required, length at least 8 characters.) * Age: Float(number input, required) * Microchip: String(text input, required, length at least 8 characters.) * Weight: Float(number input, required) * Hatching Date: Date * Color: String(text input, required, length at least 8 characters.) | | 2 | Customers fill in information in provided form then click “Submit” | System validate bird’s information inputted and store it in database |   **Alternative Scenario:**  N/A  **Exceptions:**   | No | Cause | System Response | | --- | --- | --- | | 1 | Customers inputs invalid or incomplete form submission | The system show error message |   **Business Rules:**   * Only logged user can add their bird * Staffs and Admin can view new bird * Bird’s weight must be equal or less than 5 | | | | |

#### UC-20: Delete Bird

| **USE CASE-8 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC008 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Delete Bird | | | |
| **Author** | Vu Minh Nhat | | | |
| **Date** | 31/05/2023 | **Priority** | Normal | |
| **Actor:**  Customer.  **Summary:**  This use case allows authenticated users to delete a bird in their " Bird List ".  **Goal:**  - Allow a customer to request the deletion of their bird record from the bird clinic system, ensuring that the record is permanently removed.  **Triggers**  - The customer decides to remove their bird record from the system.  - The customer wants to remove a bird record that is no longer needed or is no longer relevant.  **Preconditions:**  - The customer is authenticated and authorized to perform delete tasks.  - There is an existing bird record in the system that needs to be deleted.  **Post Conditions:**  The bird record is successfully deleted from the system  **Main Success Scenario:**   | **No** | **Actions** | **System response** | | --- | --- | --- | | **1** | The customer selects the " Bird List " option in Customer Profile. | The system presents a list of bird records available for deletion with following information:   * Bird’s Image: Image * Bird’s Name: String * Gender: String * Breed: String * Age: Float * Hatching Date: Date * Color: String | | **2** | The customer selects the specific bird record to be deleted. | The system prompts for confirmation of the deletion. | | **3** | The customer confirms the deletion. | The system removes the bird record from the system and updates the database accordingly. | |  | The system displays a success message confirming the deletion of the bird record. |   **Alternative Scenario: (N/A)**  **Exceptions:**   | **No** | **Actions** | **System response** | | --- | --- | --- | | **1** | Confirmation Canceled | System redirect the Customer to the bird list | | **2** | No bird in the list | The system displays a message indicating that no bird records are available for deletion. | | The system redirects the Customer back to the bird list | | **3** | System fails on saving changes: | System notifies user that an error has occurred while saving the list to the database | | System return user to confirm and delete the selected bird again |   **Business Rules:**  - Only a customer can initiate the request for the deletion of their bird record.  - Deletion of a bird record is permanent and cannot be undone.  - Once a bird record is deleted, any associated information or data linked to that record is also removed from the system. | | | | |

#### UC-21: View Bird Profile

| **USE CASE-21 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC021 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Bird Profile | | | |
| **Author** | Pham Vinh Son | | | |
| **Date** | 20/05/2023 | **Priority** | Normal | |
| **Actor:**  - Customer  **Summary:**  - This use case allows actor to view their birds’s profile  **Goal:**  - The goal is to help actor to view their bird detailed profile  **Triggers**  - Actors click Bird Profile button  **Preconditions:**  - Actors’s bird profile is already registered  **Post Conditions:**  - System displays actor bird detailed profile  **Main Success Scenario:**   | **No** | **Action** | **System Response** | | --- | --- | --- | | 1 | Actor click the Bird Profile button | The system retrieves to bird profile page and show bird’s profile   * Bird’s Image: Image * Bird’s Name: String * Gender: string * Breed: String * Band: String * Age: Float * Microchip: String * Weight: Float * Hatching Date: Date * Color: String |   **Alternative Scenario:**  **N/A**  **Exceptions:**  **N/A**  **Business Rules:**  - Only Customer can view their bird profile | | | | |

#### 

#### UC-22: Edit Bird Profile

| **USE CASE-22 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC022 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit Bird Profile | | | |
| **Author** | Pham Vinh Son | | | |
| **Date** | 20/05/2023 | **Priority** | Normal | |
| **Actor:**  - Customer  **Summary:**  - This use case allows actor to edit their bird’s profile  **Goal:**  - The goal is to help actor to change their bird’s information  **Triggers**  - Actors clicks Bird Profile button  **Preconditions:**  - Actor’s bird profile is already registered  **Post Conditions:**  - After updating profile successfully, actor bird’s profile is updated accurately in the system  **Main Success Scenario:**   | **No** | **Action** | **System Response** | | --- | --- | --- | | 1 | Actor click the Bird Profile button | The system retrieves to bird profile page and show bird’s profile  - Bird Name: String(text input, required, length at least 8 characters.)  - Breed: String(text input, required, length at least 8 characters.)  - Hatching Date: Date  - Gender: String(text input, required, length at least 8 characters.)  - Weight: Float(number input, required, enter value less than 5)   - Band/Microchip: String(text input, required, length at least 8 characters.)  - Feather Color: String(text input, required, length at least 8 characters.) | | 2 | Actor change their bird specify profile information and click Done button | The system will update new information accurately | |   **Alternative Scenario:**  **N/A**  **Exceptions:**  N/A  **Business Rules:**  - Only Customer can edit their bird  - Information must be valid before saving to database  - Bird weight can not larger than 5kg | | | | |

#### 

#### 

#### UC-23: Create Reservation

| **USE CASE-23 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC023 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create Reservation | | | |
| **Author** | Dang Kim Minh Tien | | | |
| **Date** | 20/5/2023 | **Priority** | High | |
| **Actor:**   * Customers   **Summary:**   * Allow Customers to create a new appointment online via the clinic’s website for their bird.   **Goal:**   * Enable Customers to book appointments efficiently and accurately.   **Triggers:**   * Customers click book an appointment.   **Preconditions:**   * Customers need to login to the website * Customers must know the service they want to book with. * Customers must know the date and time they want to book the appointment. * Customers need to provide enough information about them and their bird.   **Post Conditions:**   * The appointment is booked successfully in the system. * The Staff is notified of the new appointment.   **Main Success Scenario:**   | **Step** | **Actor** | **System** | | --- | --- | --- | | 1 | Customers click into “book now” button | System redirect to booking page that showing a list of doctor:   * name: string * age: float * speciality: string * degree: string | | 2 | Customers click into “Book all” button | System shows out all available time slots from Monday to Friday | | 3 | Customer select time slot |  | | 4 | Customer confirm the time slot by click into “Next” button | System redirect customer to an appointment form with saved information:   * Full Name: String * Email: String * Phone: String | | 5 | Customer click on “Choose your bird” button to choose the wanted bird in the list and fill in description form (optional) | System shows out the roll-out list of Customer’s bird and Doctor’s Speciality | | 6 | Customer finish filling form and click on “Confirm and Submit” | System saves the appointment’s information to the database. | |  | System notices that appointment has been booked successfully with additional informations:   * Doctor’s name: String * Time slot: date & time |   **Alternative Scenario:**  AC1: Book Appointment based on Doctor’s Speciality:   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | Customer choose the specific Speciality from the list | System shows the list of Doctors:   * Full Name: string * Speciality: string * Address: string   match with the same chosen Speciality field | | 2 | Customer choose the specific Doctor by click into “Book Now” button in that Doctor’s card | System shows out the chosen Doctor’s available time slots | | 3 | Customer choose a time slot |  | | 4 | Customer confirm the select time slot by click into “Next” button | System redirect customer to an appointment form with saved information:   * Full Name: string * Email: string * Phone: integer * Name: string * Speciality | | 5 | Customer click on “Choose your bird” button to choose the wanted bird in the list and fill in description form (optional) | System shows out the list of Customer’s bird list including:   * Bird’s Name:string | | 6 | Customer confirm all the information in the appointment form by click into “Confirm and Submit” button | System saves the appointment’s information to the database. | |  | System notices that appointment has been booked successfully with additional informations:   * Doctor’s name:string * Time slot: date & time |   AC2: Book appointment on the Doctor’s profile page   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | Customer click on the “View Profile” button in each Doctor’s card on the list | System redirect Customer to the Doctor’s Profile Page with following informations:   * Doctor’s Full Name: String * Speciality: String * YearOfExperience: tinyint * Degree: String * Age: float * Academic Title: String | | 2 | Customer confirm booking the specific Doctor by click into “Book Appointment” button in the Doctor’s Profile Page | System shows out the chosen Doctor’s available time slots | | 3 | Customer choose the suitable time slot |  | | 4 | Customer confirm the select time slot by click into “Next” button | System redirect customer to an appointment form with saved information:   * Full Name: String * Email: String * Phone: integer | | 5 | Customer click on “Choose your bird” button to choose the wanted bird in the list and fill in description form (optional) | System shows out the roll-out list of Customer’s bird list including:   * Bird’s Name: String | | 6 | Customer confirm all the information in the appointment form by click into “Confirm and Submit” button | System saves the appointment’s information to the database. | |  | System notices that appointment has been booked successfully with additional informations:   * Doctor’s name:String * Time slot: date & time |   **Exceptions:**   | **No** | **Cause** | **System Response** | | --- | --- | --- | | 1 | Guests/Customer fails to show up for the appointment without cancellation | System put them in blacklist | | 2 | Guests/Customers enter incorrect information, such as invalid email address or phone number | System prompt them to enter correct information |   **Business Rules:**   * Appointments may be subject to cancellation fees if the user/client fails to show up without cancellation. * The booking system may have blackout dates or times when appointments cannot be booked. * Each Appointment only consists of 1 time slot. | | | | |

#### 

#### 

#### 

#### 

#### 

#### UC-24: Update Reservation

| **USE CASE-24 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC024 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Update Reservation | | | |
| **Author** | Dang Kim Minh Tien | | | |
| **Date** | 23/06/2023 | **Priority** | High | |
| **Actor:**  Staff  **Summary:**  This use case describes the process of how the Staff and Customer update a specific reservation made by a customer in the bird clinic system.  **Goal:**  System allows Staff members to update customer’s reservations in the bird clinic, such as assigning a doctor to appointments which don't have a Doctor yet, setting status for each appointment, while Customers can cancel the appointment.  **Triggers**   * Staff members access the reservation management section of the system and select a reservation that requires updating.   **Preconditions:**   * Staff members are authenticated and have the authority to confirm reservations. * There is a pending reservation in the system that needs to be confirmed.   **Post Conditions:**   * The reservation is successfully updated with new information. * Customers, Staff and Doctors are notified about the changes by the system.   **Main Success Scenario:**   | **Step** | **Action** | **System Response** | | --- | --- | --- | | 1 | Staff members click into the “Appointment” button in the dashboard | System presents the appointment list with following informations:   * Customer’s fullName: String * Appointment’s Date and Time: Date * Bird’s fullName: String * Bird’s Breed: String * Doctor’s fullName: String * Status: String | | 2 | Staff members click into “Edit” button in the appointment management section | System redirect to the Appointment detail page with saved informations and fields for each information:   * Customer’s fullName: String(text input, required, length at least 8 characters.) * Appointment’s Date and Time: Date * Bird’s fullName: String(text input, required, length at least 8 characters.) * Bird’s Breed: String(text input, required, length at least 8 characters.) * Doctor’s fullName: String(text input, required, length at least 8 characters.) * Status(Processing/Confirmed/CheckIn/Cancel/Complete): String | | 3 | Staff members fill in the new required information to the specific field |  | | 4 | Staff members confirm all the information by clicking into “Confirm and Submit” button | System save the new appointment into the database | |  | System redirect Staff members to the appointment management section | | 5 | Staff set the appointment’s status by clicking one of three options “Complete”, “Cancel”, “Check in” | System update the Status of the appointment based on Staff members’ choice |   **Alternative Scenario:**  AC1: Update after creating a new appointment   * **Customer:**  | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | Customer finish creating a reservation in the website | System notices that appointment has been booked successfully with additional informations:   * Doctor’s name: String * Time slot: Date   And a “View Appointment” button | | 2 | Customer clicks into “View Appointment” button | System redirect Customer to the Customer’s Profile page  including Customer’s appointment list with information:   * Customer’s fullName:string * Appointment Time:datetime * Bird’s fullName:string * Bird’s Breed:string * Doctor’s fullName:string * Status:string | | 3 | Customer clicks into “Edit” button next to the specific Customer’s appointment in the list | System redirect to the Appointment detail page with saved informations and fields for each information:   * Customer’s fullName: String(text input, required, length at least 8 characters.) * Appointment’s Date and Time: Date * Bird’s fullName: String(text input, required, length at least 8 characters.) * Bird’s Breed: String(text input, required, length at least 4 characters.) * Doctor’s fullName: String(text input, required, length at least 8 characters.) | | 4 | Customer fills in the new required information to the specific field |  | | 5 | Customer confirms all the information by clicking into “Confirm and Submit” button | System save the new appointment into the database | |  | System redirect Customer to Customer’s appointment list |   **Exceptions:**   | **No** | **Cause** | **Response** | | --- | --- | --- | | 1 | No existing appointment | System notices Staff members that “No appointment right now” | | 2 | Unauthorized access | System denies access to the Appointment Management section | | System display an error message to the unauthorized Staff members | | 3 | System fails on saving changes to the list | System fails on saving changes to the database | | System display an error that system failed on saving the information | | System display an error message to the unauthorized Staff members | | 4 | User decide not to make any change | System redirect user to the reservation list page |   **Business Rules:**   * Only the Customer who made the reservation or the authorized Staff are allowed to update reservations. * The reservation being updated must exist in the system and be associated with the Customer's account. * The Staff member must have the necessary permissions and privileges to update the reservation. * Any changes made to the reservation should adhere to the clinic's policies and availability of resources. * The system may send a notification or confirmation email to the Customer, informing them of the updated reservation details. * Customers can only change the Status from “Processing” to “Cancel”. | | | | |

#### [UC-25: Delete Reservation](#_heading=h.4olf1plmdqad)

| **USE CASE-25 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC025 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Delete Reservation | | | |
| **Author** | Dang Kim Minh Tien | | | |
| **Date** | 22/06/2023 | **Priority** | High | |
| **Actor:**  Staff  **Summary:**  This use case describes the process of deleting an existing reservation from the bird clinic system.  **Goal:**  The goal of this use case is to allow Staff members to delete their reservations from the bird clinic system when they are no longer needed or requested.  **Triggers**   * The Staff members logs into their account on the clinic’s website and accesses to the reservation management section   **Preconditions:**   * The Staff member is authenticated and has a valid account on the clinic’s website * There is at least one existing reservation associated with the customer’s account that needs to be deleted   **Post Conditions:**   * The reservation is successfully deleted from the system   **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | Staff logins and clicks into “Appointment” button on the dashboard **(Exception 2)** | System presents a list of reservation including:   * Doctor’s full Name: String * Appointment’s Date and Time: Date * Booking Date: Date * Amount: Float **(Exception 1)** | | 2 | Staff confirms deleting appointment by clicks into “Delete” button in the specific appointment in the appointment list | System confirm and delete the chosen appointment **(Exception 3)** | |  | System notices to Customers that the appointment is successfully deleted |   **Alternative Scenario:(N/A)**  **Exceptions:**   | **No** | **Cause** | **Response** | | --- | --- | --- | | 1 | No existing Reservation | System displays a message indicating that no existing reservations are found. | | 2 | Unauthorized Access | System denies access to the Appointment Management section. | | System displays an error message indicating the lack of authorization | | System redirect the Staff to the Login Page | | 3 | System fails on saving changes to list | System display an error that system failed on saving the information | | System redirect Staff to confirms and deletes the reservation again | | 4 | User decide not to make any changes | System redirect user to the reservation list page |     **Business Rules:**   * Only staff members are allowed to delete reservations in the system. * The reservation being deleted must exist in the system. * The staff member must have the necessary permissions and privileges to delete the reservation. * Once a reservation is deleted, the corresponding appointment slot and resources associated with the reservation may become available for other reservations. * The customer may be notified of the reservation deletion via a notification or confirmation email. | | | | |

#### 

#### [UC-26: Cancel Reservation](#_heading=h.tgznebhvrpgy)

#### 

| **USE CASE-25 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC026 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Cancel Reservation | | | |
| **Author** | Dang Kim Minh Tien | | | |
| **Date** | 22/06/2023 | **Priority** | High | |
| **Actor:**  Customer  **Summary:**  This use case describes the process of canceling an existing reservation from the bird clinic system.  **Goal:**  The goal of this use case is to allow Customers to cancel their reservations from the bird clinic system if they have a reason to not come to the appointment.  **Triggers**   * The Customer logs into their account on the clinic’s website and accesses to the reservation list page   **Preconditions:**   * The Customer is authenticated and has a valid account on the clinic’s website * There is at least one existing reservation associated with the customer’s account that the customer choose to cancel   **Post Conditions:**   * The reservation’s status is successfully set to cancel in the system   **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | Customer clicks into “Appointment” button on the dashboard **(Exception 2)** | System presents a list of reservation including:   * Doctor’s full Name: String * Appointment’s Date and Time: Date * Booking Date: Date * Amount: Float **(Exception 1)** | | 2 | Customer clicks into the specific appointment in the appointment list | System presents the reservation’s detail information of the selected appointment:   * Appointment’s Date and Time: Date * Customer’s bird Name (Doctor’s Page): String * Doctor’s fullName: String * Doctor’s Time Slot: Date * Service’s Name: String * Status: String * Booking Date: Date * Customer’s Name: String * Notes: String   And a “Cancel” button | | 3 | Customer clicks into the “Cancel” button in the appointment detail page | System present a form for user to write out the reason for canceling the appointment:   * Reason: String(text input, length equal or less than 100 characters) | | 4 | Customer can choose to write the reason or directly cancel the appointment by clicking into “Submit” button in the form **(Exception 4)** | System confirms and save changes to the database **(Exception 3)** | | System automatically set the appointment’s status into “Cancel” | | 5 |  | System redirect user to the reservation list page in the website |   **Alternative Scenario:(N/A)**  **Exceptions:**   | **No** | **Cause** | **Response** | | --- | --- | --- | | 1 | No existing Reservation | System displays a message indicating that no existing reservations are found. | | 2 | Unauthorized Access | System denies access to the Appointment Management section. | | System displays an error message indicating the lack of authorization | | System redirect the Staff to the Login Page | | 3 | System fails on saving changes to list | System display an error that system failed on saving the information | | System redirect user to confirms and deletes the reservation again | | 4 | User decide not to make any changes | System redirect user to the reservation list page |     **Business Rules:**   * Only customers are allowed to cancel reservations in the system. * The reservation being canceled must exist in the system. * The customer must have the necessary permissions and privileges to cancel the reservation. * Once a reservation is canceled, the corresponding appointment slot and resources associated with the reservation may become available for other reservations. * The customer may be notified of the reservation cancellation via a notification or confirmation email. | | | | |

#### 

#### 

#### 

#### 

#### UC-27 View Reservation

#### 

| **USE CASE-26 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC026 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Reservation | | | |
| **Author** | Dang Kim Minh Tien | | | |
| **Date** | 22/06/2023 | **Priority** | Low | |
| **Actor:**  Customer  Doctor  Staff  **Summary:**  This use case describes the process of viewing reservations from the bird clinic system. **Goal:**  The goal of this use case is to allow Customers/Doctors/Staffs to view specific reservations from the bird clinic system.  **Triggers**   * The Customers/Doctors/Staffs accesses to the reservation management section   **Preconditions:**   * Customers/Doctors/Staffs are authenticated and has a valid account on the clinic’s website * There is at least one existing reservation on the reservation list * Doctor’s reservations need to be confirmed   **Post Conditions:**   * The Customers/Doctors successfully view the reservation list.   **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | Customers/Doctors/Staff logins and clicks into “Appointment” button on the dashboard | System presents a list of reservation including:   * Doctor’s full Name: String * Appointment’s Date and Time: Date * Booking Date: Date * Customer’s Name(Doctor’s Page): String * Customer’s bird name(Doctor’s Page): String * Amount: Float * Status: String | | 2 | Customers/Doctors/Staff clicks into specific appointment in the appointment list | System presents the appointment’s detail information of the selected appointment:   * Appointment’s Date and Time: Date * Customer’s bird Name (Doctor’s Page): String * Doctor’s fullName: String * Doctor’s Time Slot: Date * Service’s Name: String * Status: String * Booking Date: Date * Customer’s Name: String * Notes: String |   **Alternative Scenario:(Not Available)**  **Exceptions:**   | **No** | **Cause** | **Response** | | --- | --- | --- | | 1 | No Existing Reservation | System displays a message indicating that no existing reservations are found. | | 2 | Unauthorized Access | System denies access to the Appointment Management section. | | System displays an error message indicating the lack of authorization | | System redirect the Staff to the Login Page |     **Business Rules:**   * Only staff members are allowed to delete reservations in the system. * The reservation being deleted must exist in the system. * The staff member must have the necessary permissions and privileges to delete the reservation. * Once a reservation is deleted, the corresponding appointment slot and resources associated with the reservation may become available for other reservations. * The customer may be notified of the reservation deletion via a notification or confirmation email. | | | | |

#### 

#### 

#### 

#### UC-28 Create Doctor

| **USE CASE-27 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC027 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create Doctor | | | |
| **Author** | Vu Minh Nhat | | | |
| **Date** | 23/06/2023 | **Priority** | Normal | |
| **Actor:**  Admin  **Summary:**  System allow the admin to set the user account’s role into Doctor and add details information for each Doctor  **Goal:**  Allow the admin to create a new doctor account and assign the role of "Doctor" to a specific account.  **Triggers**   * Admin views the user account list and clicks on the Doctor tab, then clicks on the “add Doctor” button.   **Preconditions:**   * The admin is authenticated and has the necessary privileges to create a new doctor account. * The system is operational and accessible.   **Post Conditions:**   * A new doctor account is successfully created with the assigned role of "Doctor" in the system. * The doctor can access functions and features associated with the "Doctor" role.   **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | Admin clicks into “Add Doctor” button | System shows out an Create Account form with following fields:   * UserName: string ((text input, required, length at least 8 characters.)) * Password:string ((text input, required, length at least 8 characters.)) * Role: **Doctor** * Email:string (required)(free input text, required, regular expression format \b[\w.%+-]+@[\w.-]+\.[a-zA-Z]{2-6}\b) * Phone: String(number input, required, length from 9 to 11 characters.) * Fullname:string(text input, required, length at least 8 characters.) * gender: String * DoB: date * DocAge: float * Degree: string(text input, required, length at least 8 characters.) * academicTitle: string(text input, required, length at least 8 characters.) | | 2 | Admin fills in the required information into specific fields and click “Confirm” button | System checks whether any information in the account is invalid **(Exception 1)** | |  | System confirms the new information and creates a new Doctor profile in the bird clinic system **(Exception 2)** |   **Alternative Scenario: N/A**  **Exceptions:**   | **No** | **Cause** | **Response** | | --- | --- | --- | | 1 | Duplicate account | System finds out the duplication(**email/username)** in the information | | System notice to the admin that there has been a duplicated information in the account | | System redirect admin to the Create Account form | | 2 | System fails on creating a new doctor account | System notice to the Admin that there is an error in creating a new doctor account | |  |  | System redirect the Admin to the Create Account form with existing information | | 3 | User decide not to make any change | System redirect user to the doctor’s account list page |   **Business Rules:**   * Only the admin is allowed to create a new doctor account and assign the role of "Doctor." * The doctor's account should be unique and not already exist in the system. * The admin may need to provide specific information for creating the doctor's account, such as their name, contact details, specialization, and credentials. * The doctor account should have appropriate access permissions and privileges to perform functions and features associated with the "Doctor" role. | | | | |

#### UC-29: Reply Feedback

| **USE CASE-28 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC028 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Reply Feedback | | | |
| **Author** | Nguyen Huy Bach | | | |
| **Date** | 20/05/2023 | **Priority** | Low | |
| **Actor:**  Staffs  **Summary:**  Allow staffs to reply back customer’s feedback  **Goal:**  Staffs can reply to customer’s feedback  **Triggers**  Staffs want to reply to specific feedback  **Preconditions**:   * Staff must login into the system * At least one feedback   **Post Conditions:**   * Staff reply feedback successfully   **Main Success Scenario:**   | **Step** | **Actor action** | **System response** | | --- | --- | --- | | 1 | Staffs click “reply” button | System showing customerID and a form with this fields:   * Title: String(text input, required, length at least 8 characters.) * reply content: String(text input, required, length at least 8 characters.) | | 2 | Staff inputs title, reply content and clicks “submit” | System validate imputed information and send it back to customer |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Business Rules:**   * Anonymous feedback will only show feedback title and content | | | | |

#### 

#### 

#### 

#### 

#### 

#### 

#### UC-30: Create Staff

| **USE CASE-29 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC029 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create Staff | | | |
| **Author** | Vu Minh Nhat | | | |
| **Date** | 23/06/2023 | **Priority** | Normal | |
| **Actor:**  Admin  **Summary:**  System allow the admin to create staff account (account’s role is assigned as ‘STAFF’)  **Goal:**  The goal is to allow the admin to create a new account and assign the role of "Staff” to it.  **Triggers**   * Admin views the user list and clicks on the Staff tab, then clicks on the “add Staff” button.   **Preconditions:**   * The admin is authenticated and has the necessary privileges to create a new doctor account. * The system is operational and accessible.   **Post Conditions:**   * A new Staff account is successfully created with the assigned role of "Staff" in the system. * The staff can access functions and features associated with the "Staff" role.   **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | Admin clicks into “Add Staff” button | System shows out an Create Account form with following fields:   * UserName:string(text input, required, length at least 8 characters.) (required) * Password:string(text input, required, length at least 8 characters.) (required) * Role: **Staff** * Email:string(free input text, required, regular expression format \b[\w.%+-]+@[\w.-]+\.[a-zA-Z]{2-6}\b) * Phone: String(number input, required, length from 9 to 11 characters.) * Fullname:string ((text input, required, length at least 8 characters.)) * gender: string * DoB: date | | 2 | Admin fills in the required information into specific fields and click “Confirm” button | System checks whether any information in the account is invalid | |  | System confirms the new information and creates a new Doctor profile in the bird clinic system |   **Alternative Scenario: N/A**  **Exceptions:**   | **No** | **Action** | **Response** | | --- | --- | --- | | 1 | Duplicate account | System finds out the duplication(**email/username)** in the information | | System notice to the admin that there has been a duplicated information in the account | | System redirect admin to the Create Account form |   **Business Rules:**   * Only the admin is allowed to create a new doctor account and assign the role of "Staff." * The staff’s account should be unique and not already exist in the system. * The staff’s account should have appropriate access permissions and privileges to perform functions and features associated with the "Staff" role. | | | | |

#### UC-31: Choose Doctor’s timeslots

| **USE CASE-30 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC030 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Choose Doctor’s Time Slots | | | |
| **Author** | Vu Minh Nhat | | | |
| **Date** | 23/06/2023 | **Priority** | Normal | |
| **Actor:**  Admin  **Summary:**  System allow the admin to choose each Doctor’s account available time slots in the clinic website  **Goal:**  Allow the admin to choose the available time slots for each day of the week for a specific doctor's account.  **Triggers**   * The admin navigates to the user list and clicks into “Doctors” tab * The admin selects the specific doctor’s account for which they want to choose the available time slots.   **Preconditions:**   * The admin is authenticated and has the necessary authorization to manage doctor accounts. * The system is operational and accessible. * The doctor’s account exists in the system.   **Post Conditions:**   * The available time slots for each day of the week are successfully set for the specified doctor’s account. * The system accurately reflects the doctor’s availability for appointments.   **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | Admin clicks into “User List” button in the dashboard | System display user account list with three options:   * Doctor * Staff * Customer | | Admin choose “Doctor” option in the tab | System display the Doctors account list with following information:   * Doctor’s fullName: String * Doctor’s Speciality: String * Doctor’s Age: Int * Account’s Status:string   And a “Add Time Slot” button for each account | | 2 | Admin clicks into “Add Time Slot” for the specific Doctor’s account | System shows out the timetable from Monday to Friday | | 3 | Admin choose the specific day from the timetable | System presents the available time slots for the selected day | | 4 | Admin choose time slots for the selected day based on Doctor’s availability |  | | Admin confirm the chosen time slots by clicking “Save” button | System validates the chosen time slots and checks for any conflicts or overlaps | |  | System updates the Doctor’s schedule with the new time slots for the selected day | | 5 | Admin finish editing all the available time slots for each day in the time table by clicking in the “Finish” button | System updates the doctor’s availability in the database based on the chosen time slots | |  | System notice to the Admin that changes have been made successfully, Doctors are also notified about the new schedule | |  |  | System redirect Admin to the Doctor’s account list |   **Alternative Scenario:(N/A)**  **Exceptions:**   | **No** | **Cause** | **Response** | | --- | --- | --- | | 1 | Overlapping Time Slots | System displays an error message that there are conflicts or overlap in the schedule | | System redirects admin to choose time slots for the selected day again. | | 2 | System fails on saving changes to the schedule | System fails on updating the Doctor’s schedule with the new time slots for the selected day | | System notice to admin that there has been an error while saving the new schedule to the database | | System redirect admin to confirm the chosen time slots again |   **Business Rules:**   * Only the admin is allowed to choose the available time slots for a doctor's account. * The admin can select different time slots for each day of the week. * The system may have limitations on the maximum number of appointments that can be scheduled for each time slot. * The doctor's availability should be accurately reflected in the system to avoid scheduling conflicts. | | | | |

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### UC-32: Filter Doctor

| **USE CASE-31 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC031 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Filter Doctor | | | |
| **Author** | Nguyen Huy Bach | | | |
| **Date** | 20/05/2023 | **Priority** | Normal | |
| **Actor:**  Customer  **Summary:**  System allows customers to filter doctors based on specialty  **Goal:**  Customers can choose the desired doctor.  **Triggers**  Customers click on the filter selection bar.  **Preconditions**:   * Customer must login into the system   **Post Conditions:**   * Customer search for doctor successfully     **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | Customer get to booking system | System showing a doctor list with a filter bar up ahead that include:   * General * Surgery * Nutrition * Genetics * Imaging | | 2 | Customer click on desired option | System filter the list and send back to customer the appropriate doctor |   **Alternative Scenario:**  **N/A**  **Exceptions:**  **N/A**  **Business Rules:**   * Each doctor have a speciality | | | | |

#### [UC-33: Logout](#_heading=h.rzmln1j64q5b)

| USE CASE-32 SPECIFICATION | | | | |
| --- | --- | --- | --- | --- |
| Use-case No. | UC032 | Use-case Version | | 1.0 |
| Use-case Name | Logout | | | |
| Author | Pham Vinh Son | | | |
| Date | 20/05/2023 | Priority | Low | |
| **Actor:**  Customer. Staff, Admin, Doctor  **Summary:**  The system allows actors to logout of the system  **Goal:**  Actors can logout from the system  **Triggers**  Actors click on Logout button  **Preconditions**:   * Actors are logged into the system   **Post Conditions:**   * Actors logout successfully and system will return to home page   **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | Actors click “logout” button | The system display a popup asking actors if they want to logout | | 2 | Actor confirm logout | The system show message that user has logged out successfully |   **Alternative Scenario:**  **N/A**  **Exceptions:**  **N/A**  **Business Rules:**   * Return to home screen after logged out * After logged out, user access the system as role “Guest” | | | | |

#### [UC-34: Delete Users’ Account](#_heading=h.tf30oormr534)

| **USE CASE-33 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC033 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Delete Users’ account | | | |
| **Author** | Đặng Kim Minh Tiến | | | |
| **Date** | 23/06/2023 | **Priority** | Normal | |
| **Actor:**  Admin  **Summary:**  This feature allows actor to delete selected user’s accounts  **Goal:**  The admin delete user’s accounts (Include Customer, Staff and Doctor) successfully  **Triggers**   * The actor wants to delete a user   **Preconditions:**   * The admin is authenticated and has the necessary privileges to manage accounts. * The system is operational and accessible.   **Post Conditions:**   * User's account is successfully deleted, and then the system notifies “Delete successfully”.   **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | In the admin’s homepage, the actor clicks the “User list” button | The system displays 3 lists of users with these info:  List 1 (Customer):   * id: string * fullname: string * age: float * gender: string * Phone: String * address: string * account status: boolean   List 2 (Staff):   * id: string * fullname: string * Phone: String * DOB: date * account status: boolean   List 3 (Doctor):   * id: string * fullname: string * Phone: String * DOB: date * Specialty: string * degree: string * account status: boolean | | 2 | The actor clicks the “Delete” button in the selected row of user | The system displays the message “Are you sure you want to delete this account?” and two boxes “Yes” and “No” | | 3 | The actor clicks the button “Yes” | The system clears all the information of the user and remove from the database |   **Alternative Scenario:**  **N/A**  **Business Rules:**   * Only the admin is allowed to manage users. | | | | |

#### [UC-35: Suspend/Activate Users’ Account](#_heading=h.qhwstt75w8jr)

| **USE CASE-34 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC034 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Suspend/Activate Users’ account | | | |
| **Author** | Đặng Kim Minh Tiến | | | |
| **Date** | 23/06/2023 | **Priority** | Normal | |
| **Actor:**  Admin  **Summary:**  This feature allows actor to suspend/Activate selected user’s accounts  **Goal:**  The admin suspend/Activate user’s accounts (Include Customer, Staff and Doctor) successfully  **Triggers**   * The actor wants to suspend/Activate a user   **Preconditions:**   * The admin is authenticated and has the necessary privileges to manage accounts. * The system is operational and accessible.   **Post Conditions:**   * User's account is successfully suspended/activated, and then the system notifies “Suspend/Active successfully”.   **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | In the admin’s homepage, the actor clicks the “User list” button | The system displays 3 lists of users with these info:  List 1 (Customer):   * id: string * fullname: string * age: float * gender: string * phone: string * address: string * account status: boolean   List 2 (Staff):   * id: string * fullname: string * phone: string * DOB: date * account status: boolean   List 3 (Doctor):   * id: string * fullname: string * phone: string * DOB: date * Specialty: string * degree: string * account status: boolean | | 2 | **For the “activated” accounts :**   * The actor clicks the “**Suspend**” button in the selected row of user   **For the “suspended” accounts:**   * The actor clicks the “**Activate**” button in the selected row of user | The system displays the message “Are you sure you want to suspend/activate this account?” and two boxes “Yes” and “No” | | 3 | The actor clicks the button “Yes” | The system change the account’s status of the selected user |   **Alternative Scenario:**  **N/A**  **Business Rules:**   * Only the admin is allowed to manage users. | | | | |

#### [UC-36: Update Service](#_heading=h.kt4tre2sz0x)

| **USE CASE-36 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC036 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Update Service | | | |
| **Author** | Đặng Kim Minh Tiến | | | |
| **Date** | 23/06/2023 | **Priority** | low | |
| **Actor:**  Staffs  **Summary:**  This feature allows actor to Update service’s price.  **Goal:**  The service’s price is updated accurately.  **Triggers**   * The actor wants to update the service's price.   **Preconditions:**   * The staff is authenticated and has the necessary privileges to manage services. * The system is operational and accessible.   **Post Conditions:**   * The service’s price is updated successfully and then the system notifies “Update successfully”.   **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | In the admin’s homepage, the actor clicks the “Services” button | The system displays a list of users with these info:   * Service’s id * Service name * Reference specialty * Service price | | 2 | Actor clicks into the “Update price” button of the selected service. | System shows a form to input new price (float) | | 3 | The actor input the new price and submit | System update new price to the selected service |   **Alternative Scenario:**  **N/A**  **Business Rules:**   * Only the staff is allowed to manage services. | | | | |

#### 

#### 

#### 

#### 

#### 

#### [UC-37: Add service](#_heading=h.hp50utnfhbpf)

| **USE CASE-37 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC037 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Add service | | | |
| **Author** | Đặng Kim Minh Tiến | | | |
| **Date** | 23/06/2023 | **Priority** | low | |
| **Actor:**  Staffs  **Summary:**  This feature allows actors to create a new service.  **Goal:**  The new service is added to system  **Triggers**   * The actor wants to create a new service   **Preconditions:**   * The staff is authenticated and has the necessary privileges to manage service. * The system is operational and accessible.   **Post Conditions:**   * The new service is added to the system successfully, and then the system notifies “Add successfully”.   **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | In the admin’s homepage, the actor clicks the “Services” button | The system displays a list of users with these info:   * Service’s id * Service name * Reference specialty * Service price | | 2 | The actor clicks the add button | The system shows “Add service” form:   * Service name: String * Reference specialty: choose from specialty db * Service price: float | | 3 | The actor fills the form | System validate the new information | | 4 | The actor clicks the button “submit” | System creates a new service in the bird clinic system |   **Alternative Scenario:**  **N/A**  **Business Rules:**   * Only the staff is allowed to manage services. * The actor can add only one service at once | | | | |

#### 

#### 

#### 

#### 

#### 

#### [UC-38: Delete service](#_heading=h.1i9yn7x1joji)

| **USE CASE-38 SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC038 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Delete service | | | |
| **Author** | Đặng Kim Minh Tiến | | | |
| **Date** | 23/06/2023 | **Priority** | low | |
| **Actor:**staff  **Summary:** This feature allows actor to delete the selected service  **Goal:** The actor delete the selected service successfully  **Triggers**   * The actor wants to delete a service   **Preconditions:**   * The staff is authenticated and has the necessary privileges to manage service. * The system is operational and accessible.   **Post Conditions:**   * the selected service is deleted successfully, and then the system notifies “delete success”.   **Main Success Scenario:**   | **Step** | **Action** | **Response** | | --- | --- | --- | | 1 | In the admin’s homepage, the actor clicks the “Services” button | The system displays a list of users with these info:   * Service’s id * Service name * Reference specialty * Service price | | 2 | The actor clicks the “Delete” button in the selected row of user | The system displays the message “Are you sure you want to delete this **service**?” and two boxes “Yes” and “No” | | 3 | The actor clicks the button “Yes” | The system clears all the information of the **service** and remove from the database |   **Alternative Scenario:**  **N/A**  **Business Rules:**   * Only the staff is allowed to manage services. * The actor can add only one service at once | | | | |

# NON-FUNCTIONAL Requirements

## Usability

* Staffs and doctors only need 1 hours of self-experiencing to understand all the functions
* All text written in English
* Appointment booking system allows visitors to schedule appointments with ease

## Reliability

* Mean Time Between Failures is more than 112 hours per failure
* Percentage of time available is over 80%
* Hour of use: 336 hours
* Maximum available down time around 4.38 hours
* Precision in the system’s output is close to 100%

## Performance

* Response time for a transaction: average 5 seconds, maximum 8 second
* System can handle 5 transaction per second
* Capacity for the number of users the system can accommodate is at least 200 users

# Supporting Information

ERD

