User Requirements Document



~ Ziyadatan Imad Aqil Mardi

Date: 16-02-2023

Last edited: 08-06-2023

Version: 3.5

# User Requirements

1.1. Functional Requirements

The desktop application will be mainly used to insert new information, look for this information and edit this information.

|  |  |
| --- | --- |
| Desktop Application | MoSCoW |
| FR-01: Admin can log in to desktop application | Must |
| FR-02: Admin can Log out from desktop application | Must |
| FR-03: User can add new animals | Must |
| FR-04: User can remove existing animals | Must |
| FR-05: User can view animals | Must |
| FR-06: User can edit animal information | Could |
| FR-07: User can add new shelters | Must |
| FR-08: User can remove existing shelters | Must |
| FR-09: User can view shelters | Must |
| FR-10: User can edit shelter information | Could |
| FR-11: User can view Customer adoption orders | Must |
| FR-12: User can Decline Customer’s order | Must |
| FR-13: User can Accept Customer’s order | Must |

|  |  |
| --- | --- |
| Web Application | MoSCoW |
| FR-14: Customer can Register an account to the website | Must |
| FR-15: Customer can log in to the website | Must |
| FR-16: Customer can search for animal by name | Must |
| FR-17: Customer can search for animal by shelters | Must |
| FR-18: Customer can search animals by species | Should |
| FR-19: Customer can view animal details | Must |
| FR-20: Customer can view Available Shelters | Must |
| FR-21: Customer can search specific shelter | Should |
| FR-22: Customer can do an Advanced Filter Search | Could |
| FR-23: Customer can request to adopt animal | Must |

1.2. Non-Functional Requirements

1. NFR-01: Maintainable and extendable. Proper OO principles must be applied to ensure good maintainability and extensibility of the code base.
2. NFR-02: Bug free system. Appropriate testing techniques must be used when implementing the system to ensure proper functioning.
3. NFR-03: Secure software. Only authorized people may make use of the system and can only access data for which they are authorized. Passwords and user input must also be managed appropriately.

# Use Case / Wireframe

2.1. Uses Cases

* Desktop Use Cases
* UC-01: Admin logs in to desktop Application

**Requirements**: FR-01

**Actor**: Admin

**Pre-Condition**: Must have the admin login credentials

**Trigger**: Admin requests to add animals

**Main Success Scenario**:

* + 1. System shows log in page
    2. Actor fills in required parameter
    3. Actor is logged in
    4. End Use Case

**2a. Extensions**: credentials invalid

* + 1. System display error message
    2. Return to step 2
* UC-02: Admin logs out of the desktop Application

**Requirements**: FR-02

**Actor**: Admin

**Pre-Condition**:-

**Trigger**: Admin requests to add animals

**Main Success Scenario**:

1. Actor clicks on log out button
2. Confirm choice
3. Actor is logged out
4. End Use Case

**2a. Extensions**: Actor cancels logging out

1. End Use Case

* UC-03: User add new animals

**Requirements**: FR-03

**Actor**: Admin

**Pre-Condition**:-

**Trigger**: Admin requests to add animals

**Main Success Scenario**:

1. System requests animals information
2. Actor fills in required parameter
3. Animal added
4. End Use Case

**2a. Extensions**: Information does not fit the required parameter

1. System display error message
2. Return to step 2

* UC-04: User remove existing animals

**Requirements**: FR-04

**Actor**: Admin

**Pre-Condition**: FR-03, must have at least 1 animal in the database **Trigger**: Admin requests to remove animals

**Main Success Scenario**:

* + 1. Actor selects the desired animal to be deleted
    2. System shows confirmation message
    3. Animal removed
    4. End Use Case
* UC-05: User view animal information

**Requirements**: FR-05

**Actor**: Admin

**Pre-Condition**: FR-03, must have at least 1 animal in the database

**Trigger**: Admin requests to view animal

**Main Success Scenario**:

* + 1. Actor opens view animals tab
    2. System shows animals
    3. End Use case
* UC-06: User edit existing animals

**Requirements**: FR-06

**Actor**: Admin

**Pre-Condition**: FR-03, must have at least 1 animal in the database **Trigger**: Admin requests to edit animal information

**Main Success Scenario**:

* + 1. Actor selects desired animal to be edited
    2. Actor fills in required parameter
    3. Animal edited
    4. End Use Case

**2a. Extensions**: Information does not fit the required parameter

* + 1. System display error message
    2. Return to step 2
* UC-07: User add new shelters

**Requirements**: FR-07

**Actor**: Admin

**Pre-Condition**:-

**Trigger**: Admin requests to add shelters

**Main Success Scenario**:

* + 1. System requests shelters information
    2. Actor fills in required parameter
    3. Shelter added
    4. End Use Case

**2a. Extensions**: Information does not fit the required parameter

* + 1. System display error message
    2. Return to step 2

* UC-08: User remove existing shelters

**Requirements**: FR-08

**Actor**: Admin

**Pre-Condition**: FR-07, must have at least 1 shelter in the database **Trigger**: Admin requests to remove animals

**Main Success Scenario**:

* + 1. Actor selects desired shelter to be removed
    2. System shows confirmation message
    3. Shelter removed
    4. End Use Case
* UC-09: User view shelter information

**Requirements**: FR-09

**Actor**: Admin

**Pre-Condition**: FR-03, must be at least 1 shelter in the database **Trigger**: Admin requests to view shelter

**Main Success Scenario**:

1. Admin requests to view shelters
2. System shows shelters
3. End Use Case

* UC-10: User edit existing shelters

**Requirements**: FR-10

**Actor**: Admin

**Pre-Condition**: FR-06, must have at least 1 shelter in the database **Trigger**: Admin requests to edit animal information

**Main Success Scenario**:

* + 1. Actor selects desired shelter to be edited
    2. Actor fills in required parameter
    3. Shelter edited
    4. End Use Case

**Extensions**: Information does not fit the required parameter

* + 1. System display error message
    2. Return to step 2
* UC-11: User view Adoption Requests

**Requirements**: FR-11

**Actor**: Admin

**Pre-Condition**: FR-23, Needs to have at least 1 adoption request

**Trigger**: Admin requests to view adoption requests

**Main Success Scenario**:

1. Admin requests to view customer’s order
2. System shows customer’s order
3. End Use Case

**1a. Extensions**: customers has no orders

1. System shows blank page
2. End use case

* UC-12: User Declines Adoption Requests

**Requirements**: FR-12

**Actor**: Admin

**Pre-Condition**: FR-23, Needs to have at least 1 adoption request

**Trigger**: Admin declines adoption requests

**Main Success Scenario**:

1. Admin view customer’s order
2. Admin declines the order
3. System deletes order
4. End Use Case

**Extensions**: -

* UC-13: User Accepts Adoption Requests

**Requirements**: FR-13

**Actor**: Admin

**Pre-Condition**: FR-23, Needs to have at least 1 adoption request

**Trigger**: Admin accepts adoption requests

**Main Success Scenario**:

1. Admin view customer’s order
2. Admin accepts order
3. System binds animal’s data to customer’s id
4. End Use Case

**Extensions**: -

* Website Use Cases
* UC-14: Customer registers in to website

**Requirements**: FR-14

**Actor**: Customer

**Pre-Condition**: customer have email, username, password, first name, last name, address, birthdate (aged 12 or above)

**Trigger**: Customer requests to Register to website

**Main Success Scenario**:

* + 1. System requests user information
    2. Actor fills in required parameter
    3. User is logged in
    4. End Use Case

**2a. Extensions**: Invalid parameters

* + 1. System display error message
    2. Return to step 2
* UC-15: Customer logs in to website

**Requirements**: FR-15

**Actor**: Customer

**Pre-Condition**: Customer must be registered, customer have username and password **Trigger**: Customer requests to log in to website

**Main Success Scenario**:

1. System requests user information
2. Actor fills in required parameter
3. User is logged in
4. End Use Case

**Extensions**: Credentials invalid

1. System display error message
2. Return to step 2

* UC-16: Customer searches for animals by name

**Requirements**: FR-16

**Actor**: Customer

**Pre-Condition:** FR-15

**Trigger**: Customer searches for animals by name

**Main Success Scenario**:

* + 1. Customer uses “Search by name” filter
    2. Customer inputs name to search on search bar
    3. Website shows animals that fits the specified name
    4. End Use Case

**Extensions**: Specified animal cannot be found

* + 1. Website shows nothing
    2. End Use Case

* UC-17: Customer searches animal by shelter

**Requirements**: FR-17

**Actor**: Customer

**Pre-Condition:** FR-15

**Trigger**: Customer searches for animals by shelter

**Main Success Scenario**:

1. Customer uses “Search by shelter” filter
2. Customer inputs name to search on search bar
3. Website shows animals that fits the specified shelter
4. End Use Case

**Extensions**: Specified animal cannot be found

1. Website shows nothing
2. End Use Case

* UC-18: Customer searches animal by species

**Requirements**: FR-18

**Actor**: Customer

**Pre-Condition:** FR-15

**Trigger**: Customer searches for animals by name

**Main Success Scenario**:

1. Customer uses “Species filter” options
2. Customer clicks on search
3. Website shows animals that fits the specified species
4. End Use Case

**Extensions**: Specified animal cannot be found

1. Website shows nothing
2. End Use Case

* UC-19: Customer view animal details

**Requirements**: FR-19

**Actor**: Customer

**Pre-Condition:** FR-15

**Trigger**: Customer requests to view animal’s details

**Main Success Scenario**:

* + 1. Customer clicks on view animal’s details button
    2. Website shows animal details page
    3. End Use Case

**Extensions**: -

* UC-20: Customer views available shelters

**Requirements**: FR-20

**Actor**: Customer

**Pre-Condition: FR-15**

**Trigger**: Customer requests to view available shelters

**Main Success Scenario**:

1. Customer opens Shelter tab
2. System shows available shelters
3. End Use Case

**Extensions**: -

* UC-21: Customer searches for specific shelters

**Requirements**: FR-21

**Actor**: Customer

**Pre-Condition:** FR-15

**Trigger**: Customer searches for specific shelters

**Main Success Scenario**:

1. Customer inputs name to search on search bar
2. Website shows animals that fits the specified name
3. End Use Case

**Extensions**: Specified shelter cannot be found

1. Website shows nothing
2. End Use Case

* UC-22: Customer uses advanced filter search

**Requirements**: FR-22

**Actor**: Customer

**Pre-Condition:** FR-15

**Trigger**: Customer requests to do an advanced filter search

**Main Success Scenario**:

* + 1. Customer goes to advanced Filter page
    2. Actor fills in the desired parameter
    3. Website shows animals that meet the desired criteria
    4. End Use Case

**2a. Extensions**: Parameter doesn’t match

* + 1. Website shows nothing
    2. End Use Case

* UC-23: Customer Requests to adopt animal

**Requirements**: FR-23

**Actor**: Customer

**Pre-Condition: FR-15**

**Trigger**: Customer requests to adopt animal

**Main Success Scenario**:

1. Actor requests to adopt animal
2. Request sent to the admin desktop application
3. End Use Case