Logo

Description automatically generated with medium confidence

2022

Done by: Stefan Popescu (4186354), Ilja Pronins (306810), Andrei Tifrea (4150104), Ard Koornstra (4058127)

PRJ-CB01 – Group06 – version 0.4

5/31/2022

Tutor: Andre Postma

User Requirement Specifications

REVISION HISTORY

|  |  |
| --- | --- |
| Version | Changes |
| 0.1 | -Initial version |
| 0.2 | -Updates according to feedback  -GUI added |
| 0.3 | -Information of first iteration added  -Website GUI |
| 0.4 | -Added new use cases  -Added new functional requirements  -Modified revision history and date |
| 0.5 | -Added new use functional requirements (ticket management)  - Added new use cases |

Contents

Contents

[1. Agreements with the client 5](#_Toc400482930)

[2. User Requirements 5](#_Toc1442303950)

[2.1 Functional requirements 6](#_Toc934468744)

[2.1.1 C# application – Employee Administration 7](#_Toc1210422625)

[2.1.2 C# Application – Animal Management 7](#_Toc90833599)

[2.1.3 C# Application – Habitat Management 8](#_Toc608186286)

[2.1.4 C# Application – Ticket Management 9](#_Toc1758112889)

[2.1.5 Website – Employees 9](#_Toc893801790)

[Non-Functional requirements 10](#_Toc1160941993)

[3. Use Cases C# Application 10](#_Toc1912794620)

[UC-01: Adding a new employee to the system. 10](#_Toc363671065)

[UC-02: Removing an employee from the company’s system. 11](#_Toc411379684)

[UC-03: Assigning an employee to a work shift. 12](#_Toc602200026)

[UC-04: Logging into the application 12](#_Toc823319595)

[UC-05: Logging out of the application. 13](#_Toc1438998353)

[UC-06: Indicating that an employee is no longer part of the company. 14](#_Toc1247487890)

[UC-07: Editing employee’s data. 14](#_Toc533575826)

[UC-08: Canceling a work shift. 15](#_Toc959048186)

[UC-09: Adding a new animal to the system. 16](#_Toc854816142)

[UC-10: Removing an animal from the zoo’s system. 16](#_Toc2006099684)

[UC-11: Editing animal’s data. 17](#_Toc694857138)

[UC-12: Log into the website 18](#_Toc190960911)

[UC-13: Log out the website 18](#_Toc769468094)

[UC-14: View employee schedule on the website 19](#_Toc1794661592)

[UC-15: Update employee profile information on the website 19](#_Toc1077617408)

[UC-16: Auto assign employees to shifts. 20](#_Toc513418824)

[UC-17: Adding a new habitat. 20](#_Toc370169783)

[UC-18: Removing a habitat. 20](#_Toc295903373)

[UC-19: Modifying habitat details. 21](#_Toc1858604323)

[UC-20: Searching for habitat by id. 21](#_Toc75762233)

[UC-21: Viewing habitat details. 21](#_Toc1888307424)

[UC-22: Assigning responsible employee to a habitat. 22](#_Toc1119523860)

[UC-23: Assigning an animal to a habitat. 22](#_Toc1105191297)

[UC-24: Unassign an animal from a habitat. 23](#_Toc1829823213)

[UC-25: Send an alert message to all employees 23](#_Toc12620443)

[UC-26: View alert messages. 24](#_Toc1944248801)

[UC-27: Send an email to the new employee 24](#_Toc728897539)

[UC-28: Generate a new ticket 24](#_Toc1341876222)

[UC-29: Update ticket prices/discounts 25](#_Toc1850001083)

[4. GUI for Desktop Application 26](#_Toc1913753644)

[Login screen 26](#_Toc237759151)

[Main Menu 26](#_Toc1705140105)

[Animal Management 27](#_Toc527885218)

[Add Animal 28](#_Toc982853266)

[View Animal Details 29](#_Toc509845776)

[Update Animal Details 29](#_Toc1194754505)

[Remove Animal 30](#_Toc1779620559)

[Assign to Habitat 30](#_Toc109783284)

[Habitat Management 31](#_Toc1129761852)

[Animals in habitats 31](#_Toc1752318921)

[Employee Administration 32](#_Toc652790768)

[Employee Scheduling 33](#_Toc2128582519)

[Add Shift 34](#_Toc1162230627)

[4. GUI for Website 36](#_Toc70153949)

[Home screen (before logging in) 36](#_Toc1349115969)

[Login page 36](#_Toc363888603)

[Messages page 37](#_Toc727544195)

[Personal information 38](#_Toc916383739)

[Schedule overview 39](#_Toc110219645)

1. Agreements with the client

The first six weeks of working on this project, we have agreed to create a C# application, that will allow our client’s administration to manage their employees,

manager being able to view information on employees and animals, manually schedule work shifts, and display them in a calendar.

In the next phase, starting from week 6, the agreements got extended. The C# application should adapt the scheduling algorithm to fit the habitat management as well (feeding of animals/habitat). Furthermore, the client wishes for a web application to be created which will feature announcements as well as the logged employee’s schedule. The employee also has the option of providing 2 dates a month on which he is unable to work.

# User Requirements

## Functional requirements

The application’s functional requirements can be seen below, prioritized following the MoSCoW model. This is a technique, which is used to reach common understanding with stakeholders on the importance placed on each of the functionalities regarding a product. The term MoSCoW itself is an acronym derived from the first letter of each of four prioritization categories: M - Must have S - Should have C - Could have W - Won't have.

### C# application – Employee Administration

|  |  |
| --- | --- |
| FR | Priority (MSCW) |
| FR-01: User must be able to manually assign work shifts for each of the employees. | M |
| FR-02 User must be able to auto assign all of the unassigned employees to a work shift | S |
| FR-03: User must be able to unassign work shifts for an employee. | M |
| FR-04: User must be able to display all of the employees’ work shifts in a calendar. | M |
| FR-05: User must be able to store data about the employees into the system: -First name; Last name; Gender; Birthday; Username; Password; Email address; Contract type; Start date and end date; Hourly wage; department; Status | M |
| FR-06: User should be able to edit data about the employees. | M |
| FR-07: User must be able to remove employees from the company’s system. | M |
| FR-08: User should be able to hire employees ahead of time. | S |
| FR-09: User should be able to indicate that an employee is no longer part of the company. | S |
| FR-10: User must be able to view the list of employees and search for individuals by ID. | M |
| FR-11: User must be able to log in. | M |
| FR-12: User must be able to log out. | M |

### 2.1.2 C# Application – Animal Management

|  |  |
| --- | --- |
| FR | Priority (MSCW) |
| FR-13: User must be able to manually assign an animal to a habitat. | M |
| FR-14 User could be able to assign all animals to specific habitats based on their perks and relationships | C |
| FR-15: User must be able to unassign an animal from their habitat. | M |
| FR-16: User could be able to display all the animals in their habitats. | C |
| FR-17: User must be able to store data about the animal into the system: - ID; Name; Species; Sex; Birth date; Habitat; Status, date of arrival and departure | M |
| FR-18: User should be able to edit data about the animals. | M |
| FR-19: User must be able to remove animals from the zoo’s system. | M |
| FR-20: User should be able to adopt new animals ahead of time. | S |
| FR-21: User should be able to indicate that an animal is no longer part of the zoo. | S |
| FR-22: User must be able to view the list of animals and search for individuals by ID. | M |
| FR-23: Animals must be assigned to a valid type of habitat. | M |

### 2.1.3 C# Application – Habitat Management

|  |  |
| --- | --- |
| FR–23 User must be able to add a new habitat. | M |
| FR-24 User must be able to remove a habitat. | M |
| FR-25 User must be able to modify a habitat. | M |
| FR-26 User must be able to search a habitat by ID. | C |
| FR-27 User must be able to view animals in a selected habitat. | M |
| FR-28 User must be able to add a habitat manager to the Habitat. | C |
| FR-29 User must be able to view habitats | M |

### 2.1.4 C# Application – Ticket Management

|  |  |
| --- | --- |
| FR-42 User must be able to generate a new ticket. | M |
| FR-43 User must be able to update ticket prices and discounts. | M |

### 2.1.5 Website – Employees

|  |  |
| --- | --- |
| FR-30: User must be able to log in. | M |
| FR-31: User must be able to log out. | M |
| FR-32: Employee will be able to change his/her Profile Picture through the website. | C |
| FR-33: User must be able to view his work-schedule. | M |
| FR-34: An employee can indicate a non-preferable working day. | S |
| FR-35: Employee will be able to change his/her Family name through the website. | M |
| FR-36: Employee will be able to change his/her Password through the website. | M |
| FR-37: Employee will be able to change his/her Address through the website. | M |
| FR-38: Employee must be able to see his profile information | M |
| FR-39: Employee manager must be able to submit an alert message | M |
| FR-40: Employee must be able to see alert messages | M |
| FR-41: Automatically send an email to the new hired employee | M |

## Non-Functional requirements

NFR-01: The Program can be extended in the future. The code is maintainable and readable, which allows future developers to easily understand how the application works and allows them to add new features without having to communicate with the previous developers.

NFR-02: The program is reliable, which means the state of the application is going to be stable and will not have frequent crashes. This can be achieved by writing a high-quality code, which can avoid many of the errors a program can usually encounter while running. Proper error-handling will be implemented as well as different object-oriented concepts such as inheritance, interfaces and others.

# 3. Use Cases C# Application

## UC-01: Adding a new employee to the system.

* + FR: [05](#_FR-01:_User_), [08](#_FR-07:_User_should_1)
  + Actor: Administrator
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: New worker has been employed to the company
  + Main success scenario:

1. The administrator presses “Add employee” button.
2. New form “Register employee” pops up.
3. The administrator puts in the personal details of the new employee, first working date, hourly wage and all other data that are mentioned in FR-04.
4. The administrator presses the add button.
5. The software adds the new employee’s details to the system and sends a confirmation message box.
   * Extensions:

4a: There is an option where user can cancel action by pressing cancel button.

5a: There is one or more fields with missing or syntactically incorrect information.

1. The system shows a notification, that you must input information in the required fields.
2. Return to MSS step 3.

## UC-02: Removing an employee from the company’s system.

* + FR: 07
  + Actor: Administrator
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: An employee has been fired or left the company for various reasons.
  + Main success scenario:

1. Administrator inputs ID of the employee and presses the search button.
2. The employee is found and displayed on the screen.
3. Administrator selects it and presses the delete button.
4. The software removes the employee from the system and displays confirmation text.
   * Extensions:

2a: Employee with this name has not been found.

4a: If the employee was scheduled to work shift in future a message will pop up to inform the user.

1. System displays a message that, there is no employee with this ID.
2. End of case.

## UC-03: Assigning an employee to a work shift.

* + FR: [01](#_FR-01:_User_must), [03](#_FR-03:_User_must), 04
  + Actor: Administrator, System
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: The administrator wants to assign an employee to a certain work shift
  + Main success scenario:
    1. Actor opens the “Work Shifts Schedule” tab.
    2. Actor selects the date; he wants to assign a shift for.
    3. Actor selects the type of shift; he wants to assign.
    4. System displays a list with available employees for the selected date and shift type.
    5. Actor selects an employee from the list.
    6. Actor selects a habitat he wants to assign the employee to work on.
    7. Actor presses the assign button.
    8. System sends a confirmation message that the action has been successfully completed.
    9. System displays the shift in the calendar.

## UC-04: Logging into the application

* + FR: [11](#_FR-10:_User_must)
  + Actor: User
  + Pre-condition: Application must be already running on the device.
  + Trigger: User wants to log in his account
  + Main success scenario:
    1. User is shown the login screen.
    2. User enters his credential (username and password)
    3. User presses the “Login” button.
    4. User gets confirmation and access to the application features of either manager or administrator. (Depends on the account he has)
  + Extensions:

3a: User can cancel the action by pressing exit button then the system will be closed.

4a: User with these credentials has not been found.

* + - 1. System displays a message that, there is no user with these login details.
      2. Return to MSS step 2.

## UC-05: Logging out of the application.

* + FR: [12](#_FR-11:_User_must)
  + Actor: Administrator
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: User wants to log out of his account
  + Main success scenario:
    1. Administrator presses the “Log out” button in the top-right corner of the administration’s form.
    2. Administrator gets message: “Are you sure you want to log out?”
    3. Administrator selects: “Yes”.
    4. Administrator is sent back to login form.
  + Extensions:

3a: Administrator selects: “No”.

* + - 1. Administrator remains in the admin form.
      2. End of case.

## UC-06: Indicating that an employee is no longer part of the company.

* + FR: [09](#_FR-08:_User_should), [10](#_FR-09:_User_must), [11](#_FR-10:_User_must)
  + Actor: Administrator
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: Administrator wants to cancel an employee’s contract
  + Main success scenario:
    1. Administrator enters the ID of the employee in the text field on the right of the “Search by ID” button.
    2. Administrators presses the “Search by ID” button.
    3. Employee with this ID is found and displayed in the list below the search button.
    4. Administrator selects the employee from the list and presses the “Cancel contract” button.
    5. New form is displayed.
    6. Administrator selects last working date and reason for removing the employee from the company.
    7. Administrator presses the “Remove” button.
    8. Administrator gets confirmation and the form is closed.
  + Extensions:

7a: User can cancel the action by pressing cancel button.

3a: Employee with this ID has not been found.

* + - 1. Return to MSS step 1.

## UC-07: Editing employee’s data.

* + FR: [06](#_FR-05:_User_should_1)
  + Actor: Administrator
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: Administrator wants to edit an employee’s data
  + Main success scenario:
    1. Administrator enters the ID of the employee in the text field on the right of the “Search by ID” button.
    2. Administrators presses the “Search by ID” button.
    3. Employee with this ID is found and displayed in the list below the search button.
    4. Administrator selects the employee from the list and presses the “Edit personal details” button.
    5. New form “Edit personal info” is displayed.
    6. Administrator fills in the fields related to the information he wants to edit.
    7. Administrator presses the “Edit” button.
    8. Administrator gets confirmation and the form is closed.
  + Extensions:

3a: Employee with this name has not been found.

* + - 1. Return to MSS step 1.

6a: Administrator presses the “Cancel” button.

* + - 1. No Data is changed
      2. Form is closed.
      3. End of case

7a: User can cancel the action by pressing cancel button.

## UC-08: Canceling a work shift.

* + FR: [03](#_FR-02:_User_must)
  + Actor: Administrator, System
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: Administrator wants to cancel an employee’s work shift
  + Main success scenario:
    1. Actor opens the “Work Shift Schedule” tab.
    2. Actor selects a date from the calendar.
    3. System displays all the scheduled shifts for this day.
    4. Actor selects the shift he wants to cancel.
    5. Actor presses the “Remove shift button”.
    6. Actor receives confirmation, that the shift has been removed successfully.
  + Extensions:
  + 3a: No scheduled shifts for the selected day have been found.
    - 1. End of case.

## UC-09: Adding a new animal to the system.

* + FR: 17, 20
  + Actor: Administrator, System
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: New animal has been added to the zoo
  + Main success scenario:

1. Actor presses “Add animal” button.
2. New form “Register animal” pops up.
3. Actor puts in the personal details of the new animal that are mentioned in FR-04.
4. Actor presses the add button.
5. The software adds the new animal’s details to the system and sends a confirmation message box.
   * Extensions:

4a: There is an option where user can cancel action by pressing cancel button.

5a: There is one or more fields with missing or syntactically incorrect information.

1. The system shows a notification, that you must input information in the required fields.
2. Return to MSS step 3.

## UC-10: Removing an animal from the zoo’s system.

* + FR: 19,22
  + Actor: Administrator
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: An animal has been bought or is being transferred.
  + Main success scenario:

1. Administrator inputs ID of the animal and presses the search button.
2. The animal is found and displayed on the screen.
3. Administrator selects it and presses the delete button.
4. The software removes the animal from the system and displays confirmation text.
   * Extensions:

2a: Animal with this name has not been found.

3a: If the animal was scheduled to work shift in future a message will pop up to inform the user.

1. System displays a message that, there is no animal with this ID.

End of case.

## UC-11: Editing animal’s data.

* + FR: 18,
  + Actor: Administrator
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: Administrator wants to edit an animal’s data
  + Main success scenario:
    1. Administrator enters the ID of the animal in the text field on the right of the “Search by ID” button.
    2. Administrators presses the “Search by ID” button.
    3. Animal with this ID is found and displayed in the list below the search button.
    4. Administrator selects the animal from the list and presses the “Edit animal” button.
    5. New form “Edit animal” is displayed.
    6. Administrator fills in the fields related to the information he wants to edit.
    7. Administrator presses the “Edit” button.
    8. Administrator gets confirmation and the form is closed.
  + Extensions:

3a: Animal with this name has not been found.

* + - 1. Return to MSS step 1.

6a: Administrator presses the “Cancel” button.

* + - 1. No Data is changed
      2. Form is closed.
      3. End of case

7a: User can cancel the action by pressing cancel button.

## UC-12: Log into the website

* + FR: 30
  + Actor: Employee
  + Pre-condition: User must be an employee of Zoo Bazaar.
  + Trigger: User opens the web application.
  + Main success scenario:

1. User enters username and password.

2. User presses the “Log in” button.

3. User’s credentials are approved by the system.

4. System redirects user to “Home” page.

* + Extensions:

3a: User’s credentials are not recognized by the system.

− System displays an error message. − Return to step 1.

## UC-13: Log out the website

* + FR: 31
  + Actor: Employee
  + Pre-condition: User must be logged in on the Employees website.
  + Trigger: User opens the web application.
  + Main success scenario:

1. User clicks the “Log out” button on the navigation bar.

2. System logs the user out.

3. User can no longer access the website (until they login).

## UC-14: View employee schedule on the website

* + FR: 33
  + Actor: Employee
  + Pre-condition: User must be logged in on the Employees website.
  + Trigger: User opens the web application.
  + Main success scenario:
    1. Actor navigates to the schedule page.
    2. System displays all the shifts for this month.

## UC-15: Update employee profile information on the website

* + FR: 32, 35, 36, 37, 38
  + Actor: Employee
  + Pre-condition: User must be logged in on the Employees website.
  + Trigger: User opens the web application.
  + Main success scenario:
    1. User navigates to the profile page.
    2. System displays user’s profile information.
    3. User modifies the field (family name/password/address)
    4. User presses the “Confirm” button
    5. User gets a success message
  + Extensions:

5a: User gets message “Fields are not in the correct format”.

Go back to step 2

## UC-16: Auto assign employees to shifts.

* + FR: 02
  + Actor: Administrator, System
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: The administrator wants to auto assign employees for the shifts of the present month
  + Main success scenario:
    1. Actor opens the “Work Shifts Schedule” tab.
    2. Actor presses the “Auto schedule” button.
    3. System displays a confirmation message “”.

## UC-17: Adding a new habitat.

* + FR: 23
  + Actor: Administrator
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: The administrator wants add a new habitat to the system.
  + Main success scenario:
    1. Actor opens the “Habitat Management” form.
    2. Actor selects the habitat from the list of habitats.
    3. Actor presses the “Add habitat” button.
    4. Actor enters valid Habitat title, type, capacity, amount of required employees and feeding type.
    5. Actor presses “Save” button.
    6. Created habitat is added to the list of habitats.

## UC-18: Removing a habitat.

* + FR: 24
  + Actor: Administrator
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: The administrator wants to remove a habitat from the system.
  + Main success scenario:
    1. Actor opens the “Habitat Management” form.
    2. Actor selects the habitat from the list of habitats.
    3. Actor presses the “Remove habitat” button.
    4. Actor presses “Ok” button of the dialog box, to confirm removal of selected habitat.
    5. Created habitat is removed from the list of habitats.

## UC-19: Modifying habitat details.

* + FR: 25
  + Actor: Administrator
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: The administrator wants to remove a habitat from the system.
  + Main success scenario:
    1. Actor opens the “Habitat Management” form.
    2. Actor selects the habitat from the list of habitats.
    3. Actor presses the “Update Habitat Details” button.
    4. Actor updates the details of habitat and presses “Save” button.

## UC-20: Searching for habitat by id.

* + FR: 26
  + Actor: Administrator
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: The administrator wants to find a habitat by specified id.
  + Main success scenario:
    1. Actor opens the “Habitat Management” form.
    2. Actor enters the habitat id in “Habitat ID:” box.
    3. Actor presses the “Search” button.
    4. Habitat with specified habitat id is displayed in the list of habitats.

## UC-21: Viewing habitat details.

* + FR: 27, 29
  + Actor: Administrator
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: The administrator wants to get overview of habitat details.
  + Main success scenario:
    1. Actor opens the “Habitat Management” form.
    2. Actor selects the habitat from the list of habitats.
    3. Actor presses the “View Habitat Details” button.
    4. Actor presses “Back” button to return to the main form of habitat management.

## UC-22: Assigning responsible employee to a habitat.

* + FR: 28
  + Actor: Administrator
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: The administrator wants to find a habitat by specified id.
  + Main success scenario:
    1. Actor opens the “Habitat Management” form.
    2. Actor selects the habitat from the list of habitats.
    3. Actor presses the “Assign Manager to Habitat” button.
    4. Actor selects employee from “New responsible employee” drop down list.
    5. Actor presses “OK” button to confirm the changes and return to the main form of habitat management.

## UC-23: Assigning an animal to a habitat.

* + FR: 13,14,16
  + Actor: Administrator, System
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: The administrator wants to place an animal in a habitat
  + Main success scenario:
    1. Actor navigates to the “Habitat Management” form.
    2. System displays a list of habitats as well as different commands.
    3. Actor selects a habitat and navigates “Manage animals”.
    4. System displays a new form with 2 commands, one to assign, the other to unassign.
    5. Actor selects the animal he wants to assign to the habitat.
    6. Actor navigates the “<-”.
    7. System sends a success message.

Extensions:

7a. System sends an error message specifying the action cannot be completed due to animal-habitat incompatibility.

Go back to MSS step 5.

## UC-24: Unassign an animal from a habitat.

* + FR: 15,16
  + Actor: Administrator, System
  + Pre-condition: Must be logged in as an administrator.
  + Trigger: The administrator wants to remove an animal from a habitat
  + Main success scenario:
    1. Actor navigates to the “Habitat Management” form.
    2. System displays a list of habitats as well as different commands.
    3. Actor selects a habitat and navigates “Manage animals”.
    4. System displays a new form with 2 commands, one to assign, the other to unassign.
    5. Actor selects the animal he wants to unassign from the habitat.
    6. Actor navigates the “->”.
    7. System sends a success message.

## UC-25: Send an alert message to all employees

* + FR: 39
  + Actor: Administrator, System
  + Pre-condition: Must be logged on the website as an administrator.
  + Trigger: The administrator wants to send an alert message to all employees
  + Main success scenario:
    1. Actor navigates to the “Message”.
    2. System displays a page designated to sending alert messages.
    3. Actor fills in the message field.
    4. Actor presses the “Send” button.
    5. System displays the message that has been sent.

## UC-26: View alert messages.

* + FR: 40
  + Actor: Employee, System
  + Pre-condition: Must be logged in on the website.
  + Trigger: The employee wants to see all alert messages
  + Main success scenario:
    1. Actor navigates to the Messages page.
    2. System displays all the messages submitted by the employee manager.

## UC-27: Send an email to the new employee

* + FR: 41
  + Actor: Ticket manager
  + Pre-condition: Must be logged on the website as an administrator.
  + Trigger: The administrator wants to add a new employee
  + Main success scenario:
    1. UC-01
    2. The employee receives an email with the login credentials for the website.

## UC-28: Generate a new ticket

* + FR: 42
  + Actor: Administrator
  + Pre-condition: Must be logged on the website as a ticket manager.
  + Trigger: The administrator wants to generate a new ticket
  + Main success scenario:
    1. The actor selects ticket management functionality from the general menu, by pressing “Tickets” button.
    2. The actor enters the following data: date, number of adults, number of children, ticket type and confirms entered data by pressing the confirmation button.

## UC-29: Update ticket prices/discounts

* + FR: 43
  + Actor: Ticket manager
  + Pre-condition: Must be logged on the website as a ticket manager.
  + Trigger: The actor wants to update current ticket prices/discounts.
  + Main success scenario:
    1. The actor presses “Update prices” in the ticket management system of the Desktop app.
    2. The actor enters new values for the ticket prices/discount to be changed.
    3. The actor confirms that entered data is correct and should be save to the database by pressing the confirmation button.

# 4. GUI for Desktop Application

## Login screen

Graphical user interface, application

Description automatically generated

After opening the application, the user will see the Login screen. Here they can enter their credentials to access the rest of the system.

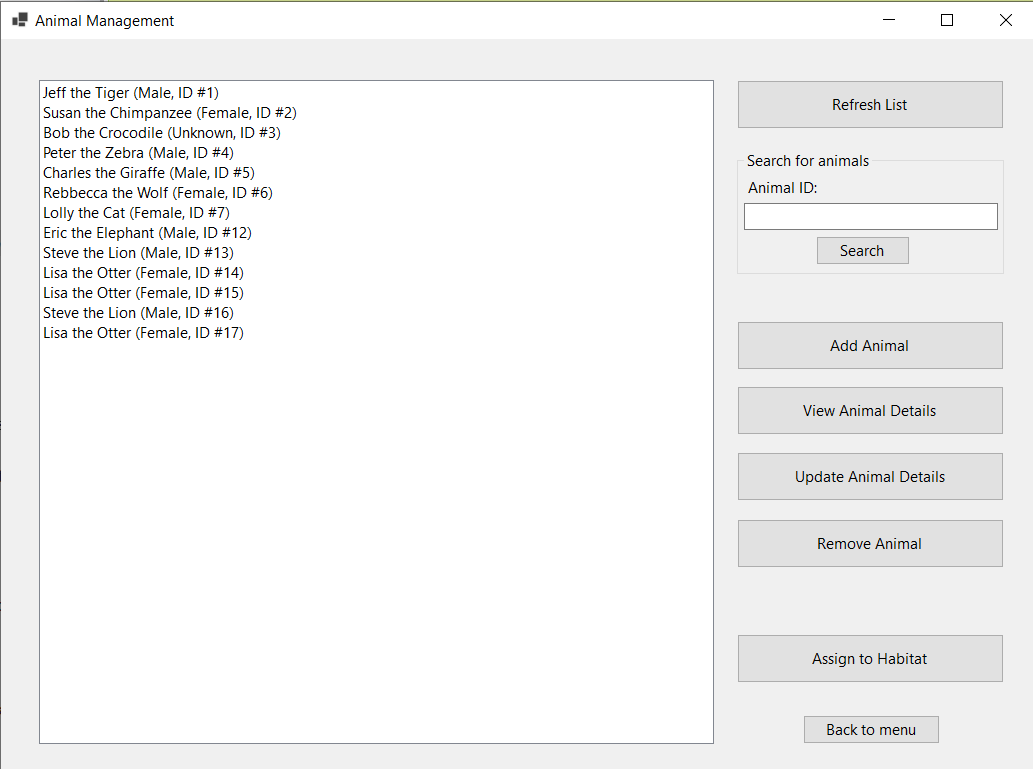
## Main Menu

Graphical user interface, text, application, email

Description automatically generated

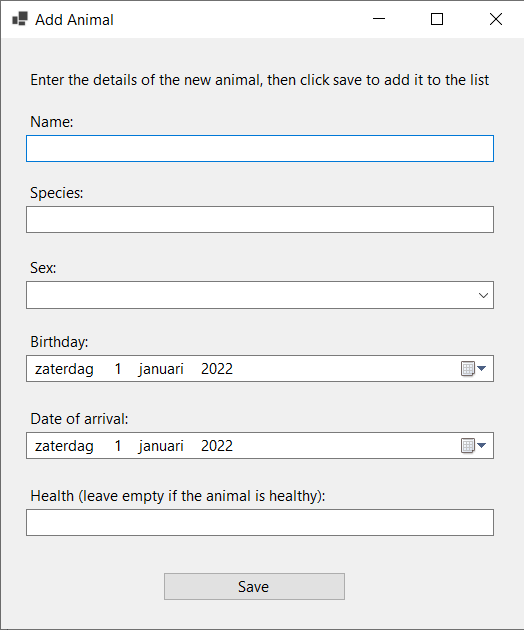
While it does not have any specific functionality of its own, this page allows the user to choose which kind of task they want to do. Each of the buttons leads to a different part of the application. The user is also welcomed with their name.

## Animal Management



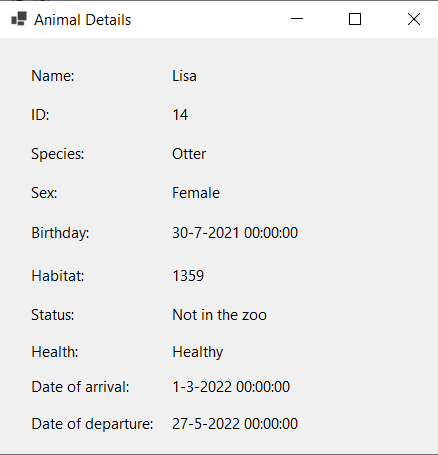
This is where all information about animals is managed. The user can see a list of all animals in the large box, or search for any specific animal in the top right. The other buttons on the side open new windows, to manage the details of specific animals, and to assign them to a habitat.

## Add Animal



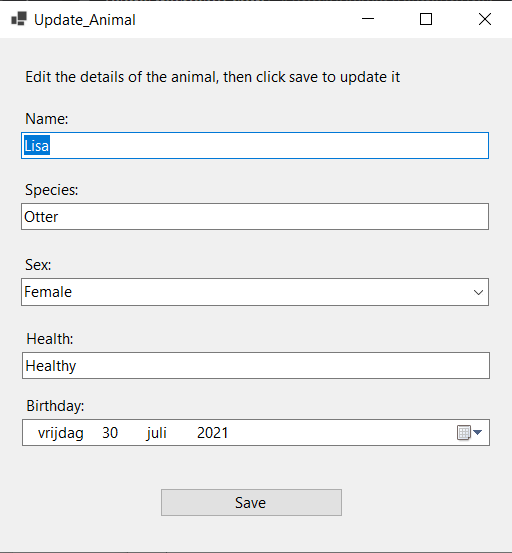
The user can fill in the details of a new animal that will arrive at the zoo, or when an animal is born. This animal is then added to the list.

## View Animal Details



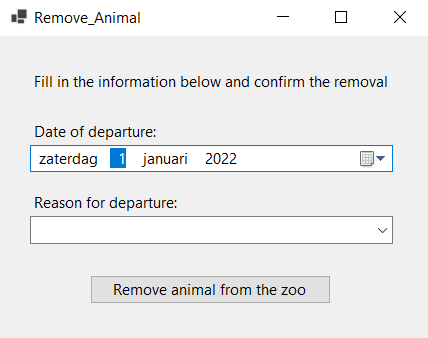
This screen will show all the details of a selected animal.

## Update Animal Details



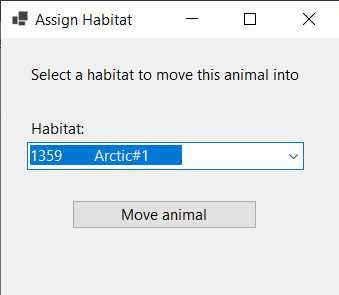
Similar to adding an animal, here the user can update certain information about it. The fields will automatically be filled with the existing information of the selected animal.

## Remove Animal



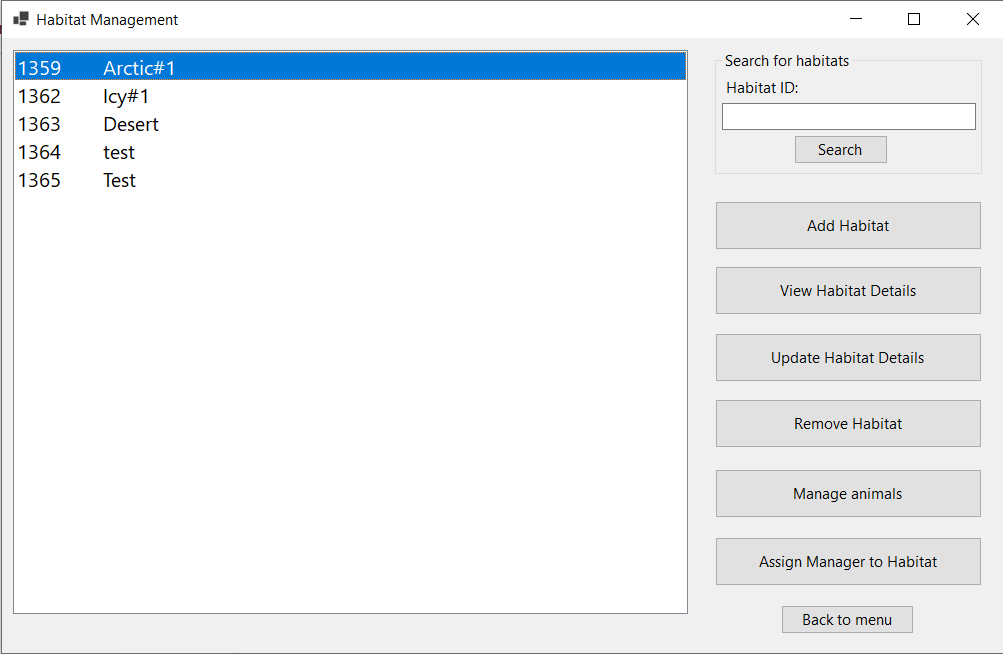
The user can select the date of departure of the animal, and the reason (such as death, or moving it to another zoo).

## Assign to Habitat



This is where an individual animal can be assigned to an existing habitat. More features, such as automatically assigning, will follow later.

## Habitat Management



Similar to the Animal Management screen, here the user can manage habitats. A list of all habitats is shown, with a search function on the side. Creating or modifying a habitat is also possible, as well as assigning an employee to be the manager of this habitat.

## Animals in habitats

Graphical user interface, text

Description automatically generated

## Employee Administration

### 

In this window you will find all the tools needed for the administration of employees. A list will be shown on the left, with a search bar and additional buttons on the right. Just like with animals before, there are options for adding, viewing information, updating this information, and removing employees. There is also a way to see the upcoming shifts for a selected employee.

## Employee Scheduling

Table

Description automatically generated

## Add Shift

A picture containing graphical user interface

Description automatically generated

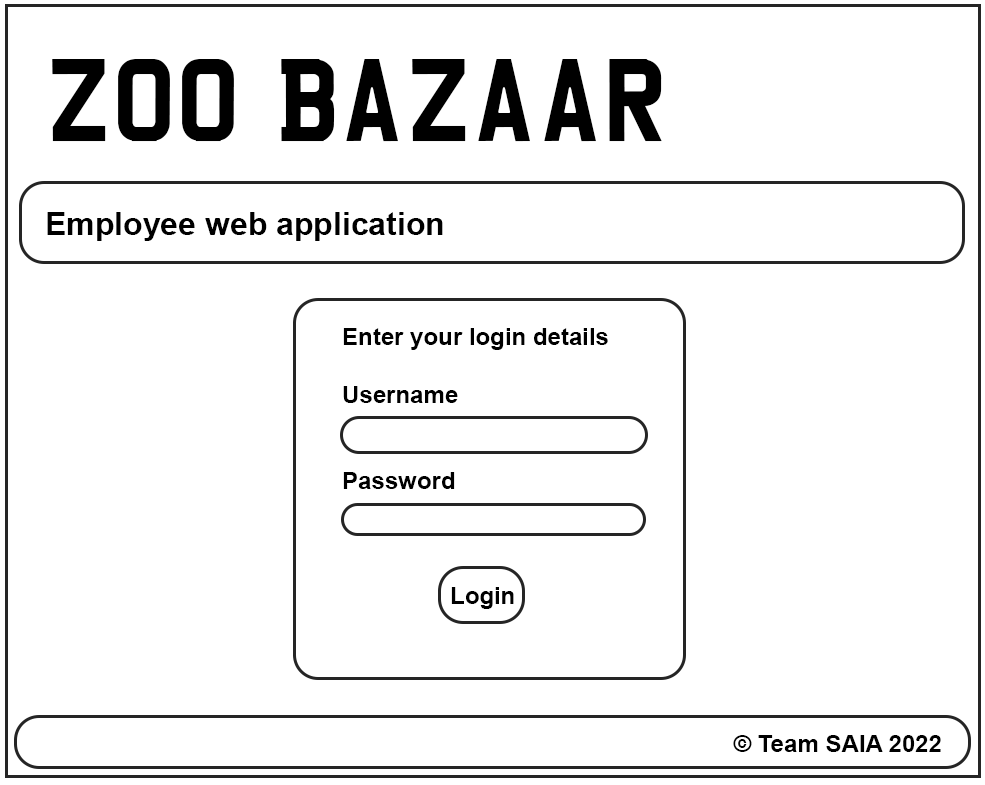
# 4. GUI for Website

## Home screen (before logging in)



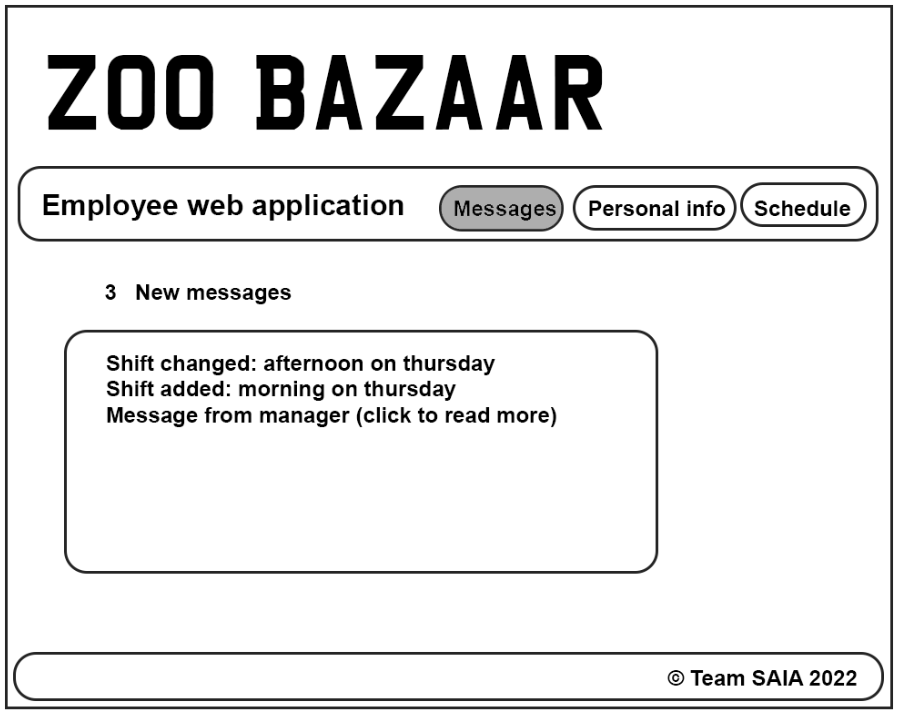
The home screen shows a welcome message, and directs visitors to the login page or to another website.

## Login page



Here the user can fill in their credentials.

## Messages page



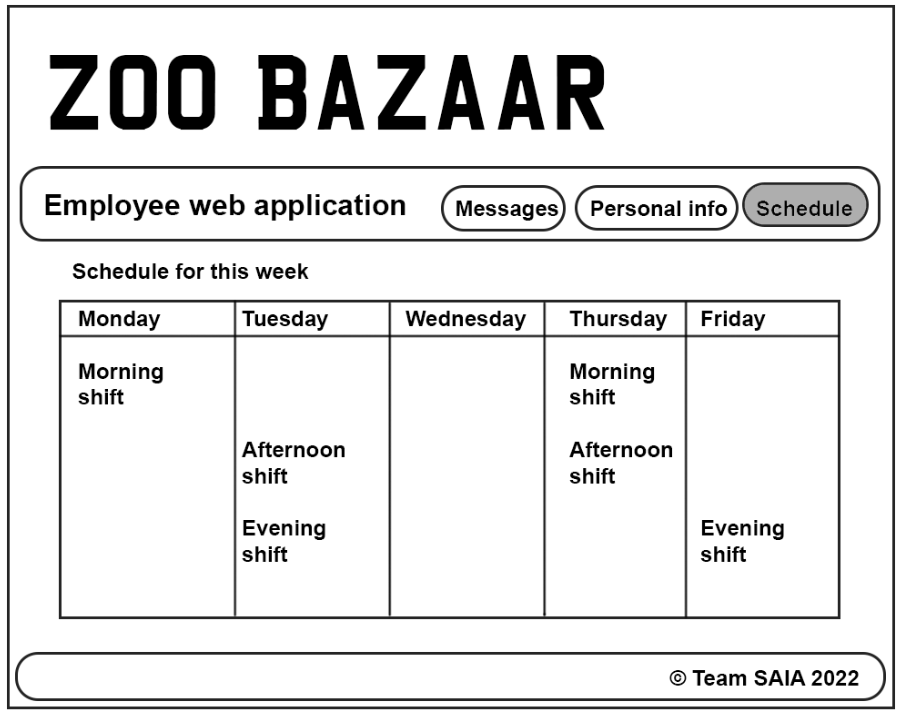
After logging in, you will see the messages or announcements from the manager and the scheduling system.

## Personal information



An overview of the details of the employee who is currently logged in.

## Schedule overview



Here the employee can see their own schedule for the coming week.