URS document

User Requirements Specification

**Group members:**

Radolina Petrova (4443527)

Stoil Yonchev (4435567)

Martin Todorov (4491114)

Danilo Gutesa (4334558)

Georgi Zhizgov (4243749)

Table of Contents

[Agreements and decision with the client 2](#_Toc178119921)

[Functional & Non-functional requirements 4](#_Toc240125789)

[Functional requirements: 4](#_Toc548708136)

[Non-functional requirements: 4](#_Toc24999729)

[Use cases 5](#_Toc2052161153)

[UC-01: Actor should be able to create an account for a new employee 6](#_Toc417947395)

[UC-02: Actor should be able to update the personal information of an existing employee 6](#_Toc1290687711)

[UC-03: Actor should be able to update their password 7](#_Toc1190150461)

[UC-04: Actor should be able to filter caretakers by specialisation 8](#_Toc697658457)

[UC-05: Actor can view the information about all the other employees 9](#_Toc279415347)

[UC-06: Actor should be able to disable the contract of an employee 10](#_Toc1834902476)

[UC-07: Managers should be able to add a new contract to an employee 11](#_Toc702856777)

[UC-08: Managers should be able to add a new animal 12](#_Toc804383895)

[UC-09: Manager should be able to update animal information 13](#_Toc2030725891)

[UC-10: Manager should be able to edit the information regarding the departure of an animal from the zoo 14](#_Toc1182173850)

[UC-11: Actor should be able to assign caretakers to look after the animals in a cage 15](#_Toc279300564)

[UC-12: Actor should be able to edit the assigned caretaker 16](#_Toc1371904971)

[UC-13: Actor can view different statistics 17](#_Toc251091081)

[UC-14: User should be able to search for an animal by its species 18](#_Toc311227496)

[UC-15: Caretakers should be able to view their schedule in the web application 19](#_Toc1782718803)

[UC-16: Buy ticket without being registered 20](#_Toc2117484584)

[UC-17: Buy ticket without being registered 20](#_Toc720009198)

[UC-18: Employees should be able to request a vacation 21](#_Toc629652445)

[UC-19: Manager should be able to accept/deny a vacation request 23](#_Toc1278101589)

[UC-20: Manager should be able to view complaints from customers 23](#_Toc143081450)

[UC-21: Customer should be able to file a complaint 24](#_Toc723761831)

[Required information 24](#_Toc1659352689)

[Account information 24](#_Toc175813204)

[Employee information 24](#_Toc437244882)

[Contract information 24](#_Toc1444743787)

[Invalid information 25](#_Toc806618340)

[Account information 26](#_Toc927079016)

[Employee information 26](#_Toc1242828443)

[Contract information 26](#_Toc1393592064)

# Agreements and decision with the client

1. Creation of different types of accounts – manager, caretaker, resource planner

* **Manager:** Adding new employees and creating their accounts and contract, adding new animals, manage all the information about the animals, the employees, and their contracts
* **Resource planner:** Creation of schedules, viewing information about the animals and the employees
* **Caretaker:** Viewing the schedules, viewing information about the animals

1. Contents of the feeding timetable

* The type of animals that must be fed
* The time slot the animals must be fed
* The assigned caretakers
* The date

1. Save account personal information regarding:

* First and last name, address, phone and email, emergency contact information, BSN, contracts
* Contract information: start date of employment, duration of the contract, FTE

1. Search functionality

* Search for employees by id or specialisation
* Search for animals by species
* Look up feeding timetable of the animals by date

1. Store information about animals

* Unique id, animal code, personal name, gender, animal type, species, cage number, birth date, reason for arrival, year of arrival, year of departure, reason for departure, diet, weekly feeding iteration, specialist, notes

|  |  |  |  |
| --- | --- | --- | --- |
| *FR* | *Description* | *End user (Authorization)* | *Application* |
| FR-01: | End user must be able to log in | Manager; Employee; Recourse planner; | Desktop; Web; |
| FR-02: | End user must be able to add an employee | Manager; | Desktop; |
| FR-03: | End user must be able to see all the employees and their information | Manager; Resource planner; | Desktop; |
| FR-04: | End user must be able to terminate the contract of an employee | Manager; | Desktop; |
| FR-05: | End user must be able to add an animal | Manager; | Desktop; |
| FR-06: | End user must be able to see all the animals and the information about them | Manager; Recourse planner; Caretaker; | Desktop; |
| FR-07: | End user must be able to set information about the departure of an animal | Manager; | Desktop; |
| FR-08: | End user must be able to update or change the information about employees and animals | Manager; | Desktop; |
| FR-09: | End user must be able to request a change of their personal information | Employee; | Web; |
| FR-10: | End user must be able to add multiple notes to an animal regarding different information | Resource planner; | Desktop; |
| FR-11: | End user must be able to search for an employee by specialization or id | Manager; Recourse planner; | Desktop; |
| FR-12: | End user must be able to change their password | Managers; Employees; Recourse planner; | Web; |
| FR-13: | End user must be able to assign caretakers to look after certain type of animals in a particular time slot | Resource planner; | Desktop; |
| FR-14: | End user must be able to view their schedule | Caretakers; | Web; |
| FR-15: | End user must be able to search for an animal by its species | Manager; Recourse planner; Employees; | Desktop; |

Functional & Non-functional requirements

Functional requirements:

## Non-functional requirements:

* Performance
* Scalability
* Authentication and authorization
* Personal data security
* Maintainability and extendibility

# Use cases

## UC-01: Actor should be able to create an account for a new employee

**Pre-condition:** Actor needs to be logged in manager account on the desktop application

**Actor:** Manager

**Main success scenario:**

1. Actor goes to the employee’s management tab and selects the option to add a new employee
2. System requires information about the account that is being created
3. Actor fills up the required information (pg. 30) and confirms
4. System requires information about the employee, the account is being created for
5. Actor fills up the required information (pg. 30) and confirms
6. System requires information about the contract of the employee
7. Actor fills up the required information (pg. 30) and confirms
8. System sends a confirmation massage that the new account for the employee has been created

**Extensions:**

3a/5a/7a: Empty fields

.1: System displays an empty field message

.2a: Return to MSS step 3/5/7

.2b: End of use case

3b/5b/7b: Invalid information (pg. 31)

.1: System displays invalid information massage

.2a: Return to MSS step 5

.2b: End of use case

## UC-02: Actor should be able to update the personal information of an existing employee

**Pre-condition:** Actor needs to be logged in manager account on the desktop application

**Actor:** Manager

**Main success scenario:**

1. Actor goes to the employee management tab and selects the employee, whose information they want to update
2. System displays the current information of the employee
3. Actor edits the information and saves the changes
4. Systems displays confirmation message

**Extensions:**

3a: Empty fields

.1: System displays an empty field message

.2a: Return to MSS step 3

.2b: End of use case

3b: Invalid information (pg. 31)

.1: System displays invalid information massage

.2a: Return to MSS step 3

.2b: End of use case

## UC-03: Actor should be able to update their password

**Pre-condition:** Actor needs to be logged in their account in the desktop application

**Actor:** Manager, Resource-planner, Caretaker

**Main success scenario:**

1. Actor goes to the settings of the account
2. System displays the information of the currently logged in actor
3. Actor updates the password
4. System sends a confirmation massage that the information is updated

**Extensions:**

3a: Empty fields

.1: System displays an empty field message

.2a: Return to MSS step 3

.2b: End of use case

## UC-04: Actor should be able to filter caretakers by specialisation

**Pre-condition:** Actor needs to be logged in

**Actor:** Manager, Resource planner

**Main success scenario:**

1. Actor goes to the tab with information about the employees
2. System displays all the current employees of the zoo franchise
3. Actor searches for employee by specialisation
4. System displays the filtered information

**Extensions:**

4a: There are no results

.1: System displays no matching information message

.2: End of use case

## UC-05: Actor can view the information about all the other employees

**Pre-condition:** Actor needs to be logged in manager/recourse planner account on the desktop application

**Actor:** Manager/Recourse planner

**Main success scenario:**

1. Actor goes to the employee tab and selects the employee, whose information they want to view
2. System displays the current information of the employee

## UC-06: Actor should be able to disable the contract of an employee

**Pre-condition:** Actor needs to be logged in manager account on the desktop application

**Actor:** Manager

**Main success scenario:**

1. Actor goes to the contracts tab and selects the employee’s contract they want to disable and disables it
2. Systems displays confirmation message

## UC-07: Managers should be able to add a new contract to an employee

**Pre-condition:** User needs to be logged in manager account on the desktop application

**Actor:** Manager

**Main success scenario:**

1. Actor selects the employee, to whom they want to add a new contract
2. System displays the
3. Actor edits the desired information and saves the changes
4. Systems displays confirmation message

## UC-08: Managers should be able to add a new animal

**Pre-condition:** User needs to be logged in a manager account

**Actor:** Manager

**Main success scenario:**

1. Actor selects the animal management tab and selects the option to add a new animal
2. System requires information about the animal that is being added to the system
3. Actor fills up the information and confirms
4. System sends a confirmation massage

## UC-09: Manager should be able to update animal information

**Pre-condition:** User needs to be logged in a manager account

**Actor:** Manager

**Main success scenario:**

1. Actor selects the animal, which information they want to change
2. System displays the information about the animal
3. Actor edits the information they want to update and confirms
4. System sends a confirmation massage

## UC-10: Manager should be able to edit the information regarding the departure of an animal from the zoo

**Pre-condition:** User needs to be logged in a manager account

**Actor:** Manager

**Main success scenario:**

1. Actor selects the animal, which departure data they want to update
2. System requires information about the departure of the animal
3. Actor fills up the information and confirms
4. System sends a confirmation massage

## UC-11: Actor should be able to assign caretakers to look after the animals in a cage

**Pre-condition:** Actor needs to be logged in

**Actor:** Resource planner

**Main success scenario:**

1. Actor goes to the schedule tab and chooses a date from the weekly schedule
2. System displays a daily schedule form
3. Actor chooses the type of the animals, they want to assign caretakers to, and the specific time slot
4. System displays all the cages with animals of that type, that should be fed in that part of the day, and loads all the specialized, available for this week, caretakers
5. Actor chooses two main caretakers and one helper caretaker (optional) for the carnivores and confirms
6. System displays message for successful operation and saves the information into the caretaker’s schedule

**Extensions:**

1a: Actor chooses a past date from the schedule

.2: System displays a daily schedule form

.3: Actor chooses the type of animal they want to assign a caretaker to and the specific time slot

.4: System displays all the cages with animals, that should have been fed in that part of the day, and displays the assigned caretakers for this shift

.5: End of use case

5a: Actor chooses the same caretaker multiple times for the same shift

.2: System displays warning message

.3: Actor edits the chosen caretakers and confirms

.4: System displays confirmation message and saves the information into the caretaker’s schedule

.5: End of use case

## UC-12: Actor should be able to edit the assigned caretaker

**Pre-condition:** Actor needs to be logged in

**Actor:** Resource planner

**Main success scenario:**

1. Actor chooses a date from the weekly schedule
2. System displays a daily schedule form
3. Actor chooses the type of the animals, which assigned
4. System displays all the cages with animals that should be fed in that part of the day
5. Actor chooses a cage they want to edit the assigned caretaker
6. System displays the information about the animals in the cage and the assigned specialized caretakers
7. Actor chooses another caretaker and assigns them to the certain cage
8. System updates the information into the caretaker’s schedule

**Extensions:**

1a: Actor chooses a past date from the schedule

.2: System displays a daily schedule form

.3: Actor choose a time slot

.4: System displays all the cages with animals that were supposed to be fed in that part of the day

.5: Actor chooses a cage

.6: System displays the information about the animals in the cage and the caretaker, that was assigned to them

.2: End of use case

UC-13: Actor can view different statistics

**Pre-condition:** User must be logged in the desktop application

**Actor:** Manager

**Main success scenario:**

1. Actor goes to the statistics tab
2. Actor selects the date and type of statistics they want to view
3. System displays the statistics

UC-14: User should be able to search for an animal by its species

**Pre-condition:** User must be logged in the desktop application

**Actor:** Manager, Resource-planner, Caretaker

**Main success scenario:**

1. Actor goes to the tab with information about the animals
2. System displays all the current animals in the zoo
3. Actor searches for a certain species
4. System displays the filtered information

**Extensions:**

4a: There are no results

.1: System displays no matching information message

.2: End of use case

## UC-15: Caretakers should be able to view their schedule in the web application

**Pre-condition:** User needs to be logged in a caretaker account in the web application

**Actor:** Manager

**Main success scenario:**

1. Actor selects the schedule tab
2. System displays the schedule for the current week

## UC-16: Buy ticket without being registered

**Pre-condition:** Actor needs to be logged into his account on the web application

**Actor:** Customer

**Main success scenario:**

1. Actor selects the details for the ticket-type of ticket and date for visit(optional) and enters his first name, last name and an email address to receive the ticket.
2. System processes the information
3. System redirects to Index page and sends the ticket number to the customer
4. End of use case

**Extensions:**

2: The email is invalid

1: System displays a message that the email is invalid

2: End of use case

## UC-17: Buy ticket without being registered

**Pre-condition:** Actor needs to be logged into his account on the web application

**Actor:** Customer

**Main success scenario:**

1. Actor selects the details for the ticket-type of ticket and date for visit(optional)
2. System processes the information
3. System redirects to Index page and sends the ticket number to the customer
4. End of use case

## UC-18: Employees should be able to request a vacation

**Pre-condition:** Actor needs to be logged in their account in the web application

**Actor:** Employee

**Main success scenario:**

1. Actor selects request vacation page
2. System requests start and end date of the vacation
3. Actor chooses start and end date and clicks request
4. System sends the request to managers for approval

**Extensions:**

3: The dates are invalid

1: System displays a message that the dates are invalid

2: End of use case

## UC-19: Manager should be able to accept/deny a vacation request

**Pre-condition:** Actor needs to be logged in a manager account in the desktop application

**Actor:** Manager

**Main success scenario:**

1. Actor selects vacation requests page
2. System displays vacations that are awaiting response
3. Actor chooses to accept or deny the vacation request
4. System indicates that the vacation was approved/denied

## UC-20: Manager should be able to view complaints from customers

**Pre-condition:** Actor needs to be logged in a manager account in the desktop application

**Actor:** Manager

**Main success scenario:**

1. Actor selects complaints page
2. System displays the complaints
3. Actor selects a complaint
4. System displays the information about the complaint

## UC-21: Customer should be able to file a complaint

**Pre-condition:** Actor needs to be logged into his account on the web application

**Actor:** Customer

**Main success scenario:**

1. Actor selects file a complaint page
2. System requires user to input his complaint
3. Actor inputs his complaint and clicks file a complaint
4. System sends the complaints to the managers

## UC-22: Manager should be able to approve details changing of an employee

**Pre-condition:** Actor needs to be logged in a manager account in the desktop application

**Actor:** Manager

**Main success scenario:**

1. Actor selects the request from the employee, who wants to change his details
2. Actor confirms the request
3. System processes the information
4. System prompts that the request was successfully approved
5. End of use case

## UC-23: Manager should be able to decline details changing of an employee

**Pre-condition:** Actor needs to be logged in a manager account in the desktop application

**Actor:** Manager

**Main success scenario:**

1. Actor selects the request from the employee, who wants to change his details
2. Actor declines the request
3. System processes the information
4. System prompts that the request was successfully declined
5. End of use case

## UC-24: Customer should be able to change details

**Pre-condition:** Actor needs to be logged into his account on the web application

**Actor:** Any Employee

**Main success scenario:**

1. Actor inputs the details that he wants to change
2. System processes the information
3. System sends the new information to the manager
4. End of use case

# Required information

## Account information

* Username
* Password

## Employee information

* First name
* Last name
* Address
* Birth date
* Phone
* Email
* Emergency contact
* BSN
* Work position

## Contract information

* Employee ID (Account ID)
* Start date
* End date
* FTE
* Reason
* Valid (true/false)

# Invalid information

## Account information

* Username: already used

## Employee information

* Phone number: already used, more or less than 10 digits (12 if it includes the country code)
* Email: already used, invalid email
* BSN: already used, more or less than 10 symbols

## Contract information

* End date: prior to the start date
* FTE: number smaller than 0 or bigger than 1