# Zoo Bazaar

URS final version

Date: 01/12/2023

Alaa Bolbol – 3948374

Erfan Alizada – 459296

Aya Shikh Suliman – 4668839

## Table of versions:

| Version | Status | Start date | Release date |
|---|---|---|---|
| 0.1 | Released | 10/14/2022 | 10/21/2022 |
| 0.2 | Released | 10/23/2022 | 10/23/2022 |
| 0.3 (Final) | Released | 01/12/2023 | 01/13/2023 |

# Table of Contents:

# Agreement/decisions:

## Waterfall release:

Our team agreed with the client that the highest priority is animal and employee management, this includes being able to add employees to the system, remove them from the system, assign the task to an employee, update employee information, and view it. For animals also adding new animals to the system, updating their details such as location, healthstatus, and personal information.

Managing animals' welfare is more complex compared to employee management.

Adding, removing, changing, and viewing all employees and animal information will be done by the manager including attendance and scheduling. Resource planners have access to theinformation they need to create schedules such as feeding timetable schedules to manage animal's welfare

Caretakers should be able to view data of the animals they are responsible.Employees can however view their own information.

Employees in the future will have a website in which they can log in to view and adjust theirpersonal information and view information about their work shifts.

All these functionalities will be available in the desktop application. As of now, after the firstclient meeting, we will be working on the website.

On our website, we will be having online ticket sales, and customer complaint handling, as anadditional feature we would have statistics on our animals.

In addition to these functionalities, we also agreed to design a logo for Zoo Bazaar,which is going to be seen on this document's title page.

## Iteration 1:

After the second interview with the client, we decided as a team that the highest priority is finishing the whole implementation for animals since the schedule for employees is dependent on that. This includes the general table, the feeding timetable, and the health situation table.

Another thing that we are going to do, is store data in a database. For that, we are going to make a database design and then we are going to write queries to apply the CRUD functionalities and at the end connect the logic layer to the database layer.

In addition to these functionalities, we also agreed to edit some things in our UI to have a perfect design for the client needs.

# Functional & non-functional requirements:

## Waterfall release:

- Functional requirements:
  - Administration:
    - Employees:
      - FR-01: an admin should be able to search through the employees by ID, name, last name, etc.
      - FR-02: an admin should be able to promote an employee.
      - FR-03: an admin can edit the information of an employee.
      - FR-04: an admin should be able to remove an employee.
      - FR-05: an admin should be able to add an employee with a unique id.
    - Animals:
      - FR-06: an admin should be able to search through the animals by ID, name, type, etc.
      - FR-07: an admin should be able to remove an animal.
      - FR-08: an admin should be able to add an animal with a unique id.
  - Login:
    - Administration:
      - FR-09: Administrator should be able to login and be directed to the part of the system that is only meant for the administration.
    - Scheduler:
      - FR-10: schedulers should be able to login and be directed to the part of the system that is only meant for the resource planners.

- Non-functional requirements:
  - The desktop application should be only used by administration and resource planners.
  - The system should be able to handle about 100000 users without performance deterioration.
  - No one can access the application without having an ID and a password.
  - Resource planners cannot access the features of the administration.
  - The administration is not able to access the features of the resource planners.
  - Images are loaded fast.

Iteration 1:
- Functional requirements:
  - Administration:
    - Animals:
      - FR-01: an admin should be able to edit an animal.
      - FR-02: an admin should be able to view the feeding timetable of an animal.
      - FR-03: an admin should be able to edit the feeding timetable.
      - FR-04: an admin should be able to view the animals that have sicknesses or allergies.
      - FR-05: an admin should be able to edit the details of the sickness or the allergy of an animal.
      - FR-06: an admin should be able to add a sickness or an allergy for an animal.
  - Scheduler:
    - Animals:
      - FR-07: a scheduler should be able to view the information of an animal.
      - FR-08: a scheduler should be able to add a feeding time for an animal
      - FR-09: a scheduler should be able to assign an employee to feed an animal.

# Use cases:

## Waterfall release:

- **Use case-01:** Search for employees
  **Actor:** Admin
  **Requirements:** FR-01

  **Main Success Scenario:**
  1. Actor opens the required page in employee section.
  2. System requires information to search with ID.
  3. Actor fills in the Id.
  4. System shows the required employee.

  **Extensions:**
  3. There is no employee with such an Id.End
  scenario.

- **Use case-02:** Promote an employee
  **Actor:** Admin
  **Requirements:** FR-02

  **Main Success Scenario:**
  1. Actor fills in the needed info to find the employee by ID.
  2. System shows the required employee.
  3. Actor promotes the shown employee.
  4. Employee is promoted in the
     system.

  **Extensions:**
  4. The employee is an Admin, can't be promoted to higher.
  End scenario.

- **Use case-03:** Edit the information of an employee
  **Actor:** Admin
  **Requirements:** FR-03

  **Main Success Scenario:**
1. Actor fills in the id of an employee.
2. System shows the required employee and all his information.
3. Actor changes the information that needs to be changed or updated.
4. System changes / updates the information about
   that employee.

  **Extensions:**
2. employee id doesn't exist.
End scenario.

- **Use case-04:** Remove employee
  **Actor:** Admin
  **Requirements:** FR-04

  **Main Success Scenario:**
1. Actor opens the required page in employee section.
2. System requires information about the employee.
3. Actor fills in the employee Id.
4. System shows the required employee and all his information.
5. Actor removes the shown employee.
6. System shows employee is
   removed.

  **Extensions:**
3. Employee id doesn't exist.
End scenario

- **Use case-05:** Add an employee
  **Actor:** Admin
  **Requirements:** FR-05

  **Main Success Scenario:**
1. Actor opens the required page in employee section.
2. Actor fills in the employee information with an unique ID in order to add a new employee.
3. Actor adds the employee.
4. System shows that employee is added.

  **Extensions:**
2. ID already exist.
End scenario.

- **Use case-06:** Search for animals
  **Actor:** Admin
  **Requirements:** FR-06

  **Main Success Scenario:**
1. Actor opens the search page in animal section.
2. System requires information to search with ID.
3. Actor fills in the id.
4. System shows the required animal/animals.

  **Extensions:**
3. There is no animal with such an Id.
End scenario.

- **Use case-07:** Remove animals
  **Actor:** Admin
  **Requirements:** FR-07

  **Main Success Scenario:**
1. Actor opens the tab related to removing animals.
2. System requires information to identify with ID.
3. Actor fills in the ID.
4. System removes the required animal.

  **Extensions:**
3. There is no animal with such an Id.
   End scenario.

- **Use case-08:** Add animal
  **Actor:** Admin
  **Requirements:** FR-08

  **Main Success Scenario:**
1. Actor opens the required tab in animal section.
2. System requires data for the animal.
3. Actor fills in the animal id with the animal information such as type and name.
4. System adds the animal.

  **Extensions:**
3. ID already exist.
End scenario.

- **Use case-09:** Login
  **Actor:** Admin
  **Requirements:** FR-09

  **Main Success Scenario:**
1. Actor runs the application.
2. Actor inserts the username and password that belongs to the administration.
3. Actor enters the application successfully.
4. System shows the related home page to Admin.

  **Extensions:**
3. a) Password or username incorrect: return to step 2.
   b) Password or username incorrect.

End scenario.

- **Use case-10:** Login
  **Actor:** Scheduler
  **Requirements:** FR-10

  **Main Success Scenario:**
1. Actor runs the application.
2. Actor inserts the username and password that belongs to the Resource planners.
3. Actor enters the application successfully.
4. System shows the related home page
   to Resource planner.
   End success scenario.

  **Extensions:**

3. a) Password or username incorrect: return to step 2.
   b) Password or username incorrect.

End scenario

# Iteration 1:

- **Use case-01:** Edit the information of an animal
  **Actor:** Admin
  **Requirements:** FR-01

  **Main Success Scenario:**

  1. Actor fills in the id of an animal.
  2. System shows the required animal and all his/her information.
  3. Actor changes the information that needs to be changed or updated. The actor can change/update any information except for the id.
  4. System changes/updates the information about that employee.

  **Extensions:**

  2. Animal id doesn't exist.
  End scenario.

- **Use case-02:** View the feeding timetable
  **Actor:** Admin
  **Requirements:** FR-02
  **Pre-condition:** feeding time for the animals

  **Main Success Scenario:**

  1. Actor opens the feeding timetable.
  2. System shows the feeding timetable.
  3. Actor checks the table.
  End scenario.

- **Use case-03:** Edit the information of the feeding timetable.
  **Actor:** Admin
  **Requirements:** FR-03

  **Main Success Scenario:**

  1. Actor opens the feeding timetable.
  2. System shows the feeding timetable.
  3. Actor changes the type of food or the amount of food if there is a need.
  4. System changes / updates the information in the feeding timetable.

  **Extensions:**

  5. Feeding timetable doesn't exist.
  End scenario.

- **Use case-04:** View the animals that have sicknesses or allergies.
  **Actor:** Admin
  **Requirements:** FR-04
  **Pre-condition:** health situation table for the animals

  **Main Success Scenario:**
1. Actor opens the health situation page.
2. System shows the animals with allergies and sicknesses.
End scenario.
- FR-06: an admin should be able to add a sickness or an allergy for an animal.

  - **Use case-05:** Edit the details of the sickness/allergy of an animal
    **Actor:** Admin
    **Requirements:** FR-05
    **Pre-condition**: there are animals with allergies and sicknesses

    **Main Success Scenario:**
1. Actor opens the health situation page.
2. System shows the health situation page.
3. Actor changes/updates the details of an allergy/sickness of an animal.
4. System changes/updates the information.
End scenario.

  - **Use case-06:** Add an animal in the sicknesses/allergies.
    **Actor:** Admin
    **Requirements:** FR-07
    **Pre-condition:** an animal has an allergy or a sickness.

    **Main Success Scenario:**
1. Actor opens the health situation page.
2. System shows the general schedule.
3. Actor checks the schedule.

    **Extensions:**
2. General schedule doesn't exist.
End scenario.

- **Use case-07:** View the information of an animal
  **Actor:** Scheduler
  **Requirements:** FR-07


  **Main Success Scenario:**
4. Actor opens the general schedule.
5. System shows the general schedule.
6. Actor checks the schedule.

  **Extensions:**
2. General schedule doesn't exist.
End scenario.


- **Use case-08:** Add a feeding time for an animal
  **Actor:** Scheduler
  **Requirements:** FR-08


  **Main Success Scenario:**
1. Actor selects time, animal, food type, and amount of food.
2. Actor adds the selected information.
3. System adds the information into the table.

  **Extensions:**
1. Actor doesn't select the right
information.
End scenario.

- **Use case-09:** Assign an employee to feed an animal
  **Actor:** Scheduler
  **Requirements:** FR-09


  **Main Success Scenario:**
4. Actor selects an employee and an animal.
5. Actor assigns the selected information.
6. System adds the information into the feeding timetable.

  **Extensions:**
1. Actor doesn't select the right
information.
End scenario.

Iteration 2:

- **Functional requirements:**
- Animals:

  - Administration:
    - FR-01: an admin should be able to add an animal.

  - Scheduler:
    - FR-02: a scheduler should be able to add a feeding time for an animal
    - FR-03: a scheduler should be able to view the special feeding table
    - FR-04: a scheduler should be able to search and see an animals health situation.

- Employee:

  - Administration:

    - FR-01: an admin should be able to add job type for an employee (part-time, fulltime).
    - FR-02: an admin should be able to add shift type for the schedule of an employee.
    - FR-03: an admin should be able to view the attendance for employee.

  - Employee:

    - FR-04: an employee should be able to login via the website.
    - FR-05: an employee should be able to give his/her information about the shifts (preference, working days, request for vacation)
    - FR-06: an employee should be able to view his/her schedule.

# Iteration 2:

## Animals:

- **Use case-01:** Add an Animal
   **Actor:** Admin
   **Requirements:** FR-01

   **Main Success Scenario:**
1. System shows the required information such as type of animal, location in the zoo, amount of food needed for this animal and animal sickness or allergy if needed.
2. Actor can add a name for an animal but it's not necessary.
3. Actor confirms the information.
4. System adds the animal with a unique id.

- **Use case-02:** Schedular adds a feeding time
   **Actor:** scheduler
   **Requirements:** FR-02

   **Main Success Scenario:**
   1. Actor opens the feeding time page.
   2. Actor selects an animal type.
   3. System shows the amount of animal, food and needed time to feed all these animals.
   4. System shows a full description about the animals location.
   5. The scheduler adds a feeding time for these animals.
   6. The scheduler confirms the information.
   7. The system creates the feeding timetable with the feeding times being assigned to each available employees.

- **Use case-03:** Schedular Views the special feeding table
    **Actor:** scheduler
    **Requirements:** FR-03

    **Main Success Scenario:**
1.  Actor opens the special feeding time page.
2.  System shows the special feeding table.

- **Use case-04:** Schedular adds a feeding time
    **Actor:** scheduler
    **Requirements:** FR-04

    **Main Success Scenario:**
1.  Actor opens the Health page
2.  System shows all the animals and their health situation.
3.  Actor searches for a specific animal by id.
4.  System shows the animal's health situation.

 **Extensions:**

 3a. ID doesn't exist.

 End scenario.

Use Cases Employees:

- Use case-01: add an employee
  **Actor:** Admin
  **Requirements:** FR-01

  **Main Success Scenario:**
  1. Actor opens the required page in employee section.
  2. Actor fills in the employee information with a unique ID in order to add a new employee.
  3. Actor selects type of job (part-time or full-time).
  4. Actor adds the employee.
  5. System shows that employee is added.

  **Extensions:**
     3. ID already exist. End scenario.

- Use case-03: view the attendance for an employee
  **Actor:** Admin
  **Requirements:** FR-03

  **Main Success Scenario:**
  1. Actor opens the required page in employee section.
  2. System requires information to search with ID.
  3. Actor fills in the employee Id.
  4. System shows the required employee and all his information.
  5. System shows the attendance of selected employee.

  **Extensions:**
- employee id doesn't exist.
  End scenario.

- **Use case-04: login via the website**
    **Actor:** Employee
    **Requirements:** FR-04


    **Main Success Scenario:**
1. Actor opens the required page on the website.
2. Actor inserts the username and password that belongs to the employee
3. Actor enters the application successfully.
4. System shows the required employee and all his information.

    **Extensions:**
    1. a) Password or username incorrect:
    return to step 2.
    b) Password or username incorrect. End
    scenario.

- **Use case-05: view schedule page for an employee**
    **Actor:** Employee
    **Requirements:** FR-05


    **Main Success Scenario:**
1. Actor opens the required page in employee section.
2. System shows the schedule.
3. Actor can view his/her schedule.

    **Extensions:**
1. A message there is no schedule made yet.

Iteration 3 (Final):
Functional requirements:
- Client:
  - Client:
    - FR-01: a client should be able to buy a ticket.
    - FR-02: a client should be able to leave comment.
    - FR-03: a client should be able to contact the administration.
- Employee:
  - Employees:
    - FR-04: an employee should be able to login via the website.
    - FR-05: an Employee should be able to add a report.
    - FR-06: an Employee should be able to ask for vacation.
- Scheduler:
  - Scheduler:
    - FR-07: scheduler should be able to manage the shifts.
    - FR-08: scheduler should be able to manage the reports.
- Administration:
  - Administration:
    - FR-09: Administrator should be able to reply to a client's message.

## Iteration 3:

## Client:

- **Use case-01:** Buy a ticket
  **Actor:** Client
  **Requirements:** FR-01

  **Main Success Scenario:**
1. Actor opens the required page.
2. Actor selects the amount of visitors and a date.
3. Actor buys the ticket.
4. System shows a message "Payment successful".
5. System sends the ticket/tickets to actor's provided email address.

  **Extensions:**
  3a. Payment wasn't successful. Show a proper message.
  End scenario.

- **Use case-02:** Leaving a comment
  **Actor:** Client
  **Requirements:** FR-02

  **Main Success Scenario:**
1. Actor opens the required page.
2. Actor leaves a comment.

- **Use case-03:** Contact the administration
  **Actor:** Client
  **Requirements:** FR-03

  **Main Success Scenario:**
1. Actor opens the required page.
2. Actor provides the following information: Name, phone Number, Email and the message.
3. Actor submits the contact form.
4. System shows a message "Sent successfully, we contact you as soon as possible".

## Employees:

- **Use case-04:** Login
  **Actor:** Employee
  **Requirements:** FR-04

  **Main Success Scenario:**
1. Actor opens the required page.
2. Actor gives the password and the email address.
3. Actor logs in successfully.

  **Extensions:**
  2a. Incorrect password or email address.
  Return to step 2.
  2b. Contact the administrator.
  End scenario.

- **Use case-05:** Add a report
  **Actor:** Employee
  **Requirements:** FR-05

**Main Success Scenario:**
1. Actor opens the required page.
2. Actor scans the related barcode to an animal.
3. System loads the animal's information.
4. Actor adds the report.

**Extensions:**
2a. Unknown barcode error. Show a message "Contact administration ". End scenario.

- **Use case-06:** Vacation request
  **Actor:** Employee
  **Requirements:** FR-06

**Main Success Scenario:**
1. Actor opens the required page.
2. Actor selects the date and submit.
3. System shows a message "Request sent successfully".

## Scheduler:

- **Use case-07:** Add a shifts
   **Actor:** Scheduler
   **Requirements:** FR-07

   **Main Success Scenario:**
1. Actor opens the required page.
2. Actor selects the employee and a date.
3. Actor submits the information.
4. System shows the information being added.

- **Use case-08:** Delete a shifts
   **Actor:** Scheduler
   **Requirements:** FR-07

   **Main Success Scenario:**
1. Actor opens the required page.
2. Actor selects a shift and submit.
3. System removes the shift.
4. System shows a message "Shift removed successfully".

- **Use case-09:** Edit a shifts
   **Actor:** Scheduler
   **Requirements:** FR-07

   **Main Success Scenario:**
1. Actor opens the required page.
2. Actor selects a shift and adds the new information such as a date or employee.
3. System Edits the shift.
4. System shows a message "Shift Edited successfully".

- **Use case-10:** View the reports
  **Actor:** Scheduler
  **Requirements:** FR-08

  **Main Success Scenario:**
1. Actor opens the required page.
2. System loads the available reports.

- **Use case-11:** Delete a report
  **Actor:** Scheduler
  **Requirements:** FR-08

  **Main Success Scenario:**
5. Actor opens the required page.
6. Actor selects a report and submit.
7. System removes the report.
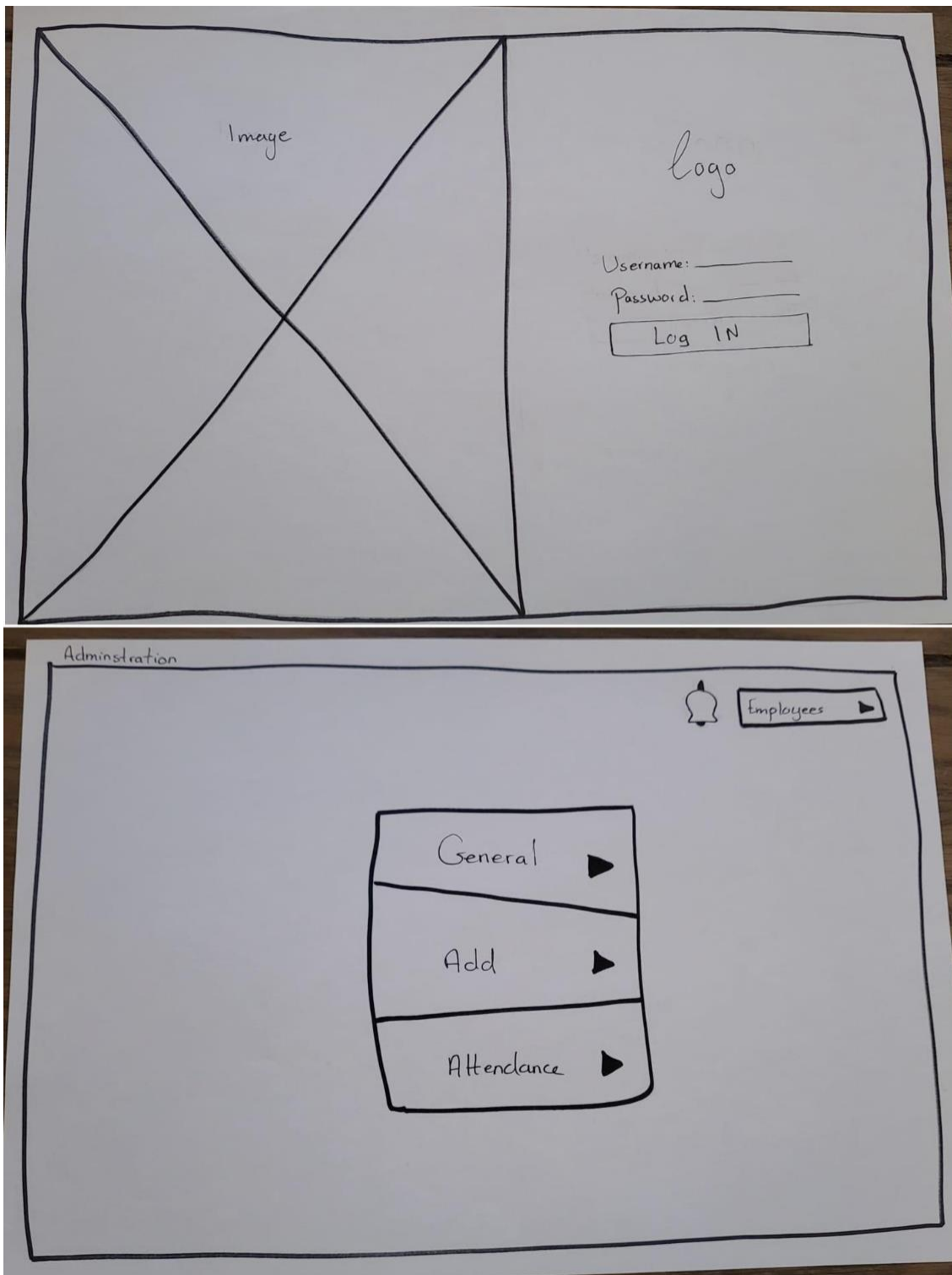8. System shows a message "report removed successfully".

## Administrator:

- **Use case-12:** Reply to Client's message
  **Actor:** Administrator
  **Requirements:** FR-09

  **Main Success Scenario:**
5. Actor opens the required page.
6. System shows the messages.
7. Actor selects a message and reply.
8. System shows a message "message successfully sent".

Paper prototypes:

Adminstration : General : Employees

Name: [_____] Last name: [_____] ID: [_____]

| ID | Name | Last name |
|----|------|-----------|
| 1 | aya | shikh suliman |

[ Edit ]    [ Promote ]    [ Remove ]

goes to a
dashboard
where information
is found about the
employee

---

Adminstration: Add: Employees

Employees:

ID: auto generated
Password: auto generated
first name: _____
last name: _____
Email : _____
Phone number: _____
Position: _____
Age: [ 1 ▼ ]
Address: _____
CV: file.pdf
Profile Picture: .png

[ Add ]

Adminstration

🔔 | Animals ▶ |

| General | ▶ |
| Nutrition | ▶ |
| Health | ▶ |

---

Adminstration : Attendance : Employees

◀ Today's date ▶

| Employee name | Availability |
|---|---|
| Erfan Alizada | Sick |

28
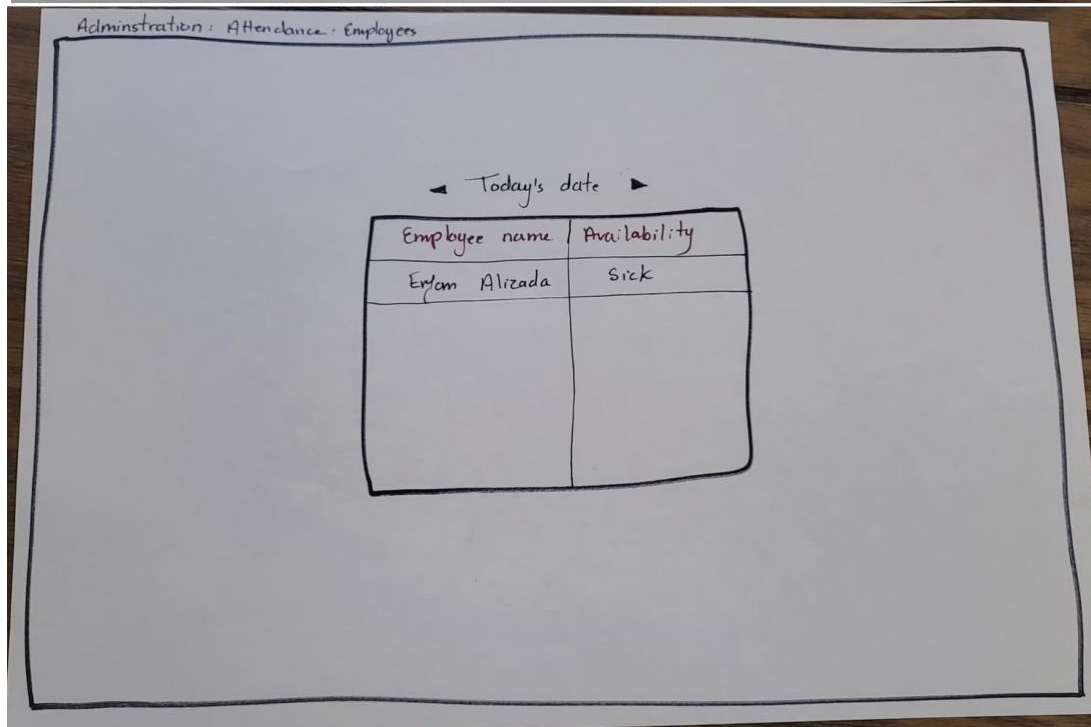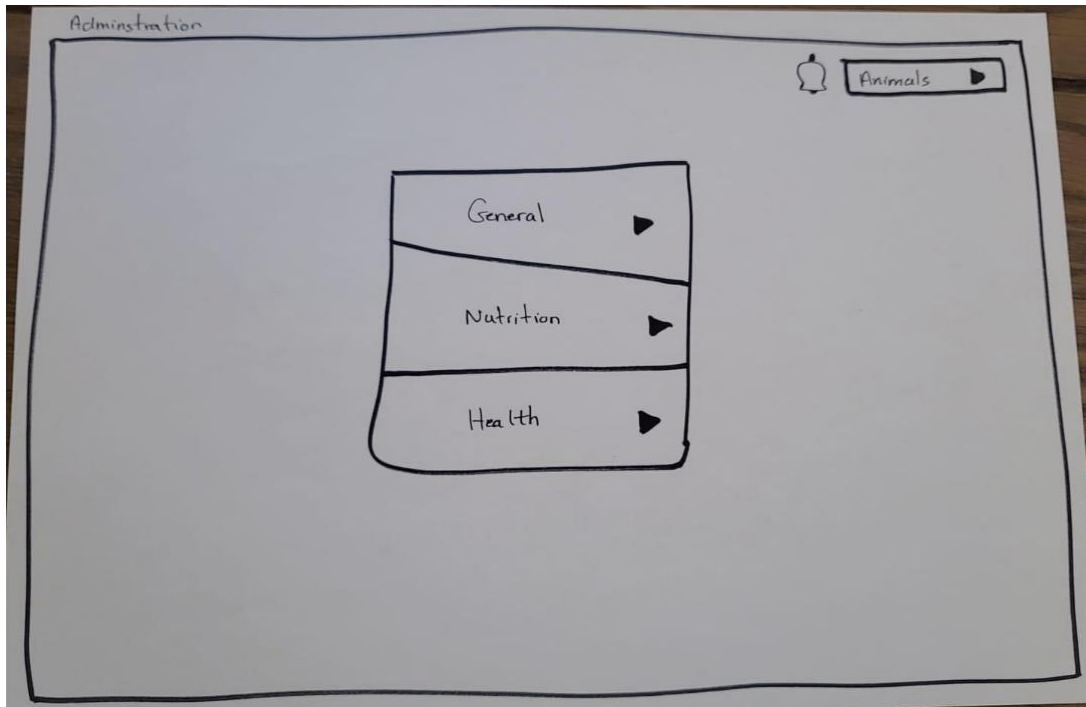
Administration: General: Animals

ID: [____]   Name: [____]   Type: [____]

| ID | Name | Type |
|----|------|------|
| 1  | Nars | Cat  |

[ Remove ]  [ Edit ]  [ Find ]  [ Add ]

         Dashboard   location

---

Administration: Nutrition: Animals

◀  Today's  Date  ▶

| Time  | ID | Animal | Type of feed | Amount of Feed |
|-------|-----|--------|--------------|----------------|
| 09.00 | 2   | Sam    | Plants       | 20 gram        |

| ID | Name | Allergies | Sicknesses | Current situation |
|----|------|-----------|------------|-------------------|
|    |      |           |            |                   |
|    |      |           |            |                   |
|    |      |           |            |                   |

→ Scroll

Resource planner

🔔 [ Employees ▶ ]

| General | ▶ |
|---------|---|
| Reports | ▶ |

Resource planner: General: Employees

ID: [____] Name: [_____] Last name: [____]

Assign     Employee        → to another form

Schedule   Employee        → to another form

Resource planner: Reports: Animals reports uploaded by employees

Date: [____]   Number: [____]   Name: [____]

| Number | Date | Name |
|--------|------|------|
|        |      |      |

Download

Resource planner:



General ▶

Create ▶

🔔 Animals ▶

---

Resource planner: General: Animals



ID: ▭  Name: ▭  Type: ▭

| ID | Name | Type |
|----|------|------|
|    |      |      |

s