AnimalDex - Web Application Documentation

Creators and developers

1940729 Baldi
1951836 Catalano
1946039 Dieni
1954414 Di Paolo

Project Introduction

AnimalDex is a distributed web application designed to raise awareness about Italian wildlife. The project aims to encourage people to learn more about the animals in Italy, promoting interaction among users and providing a platform to report and manage encounters with injured animals by contacting veterinary operators directly.

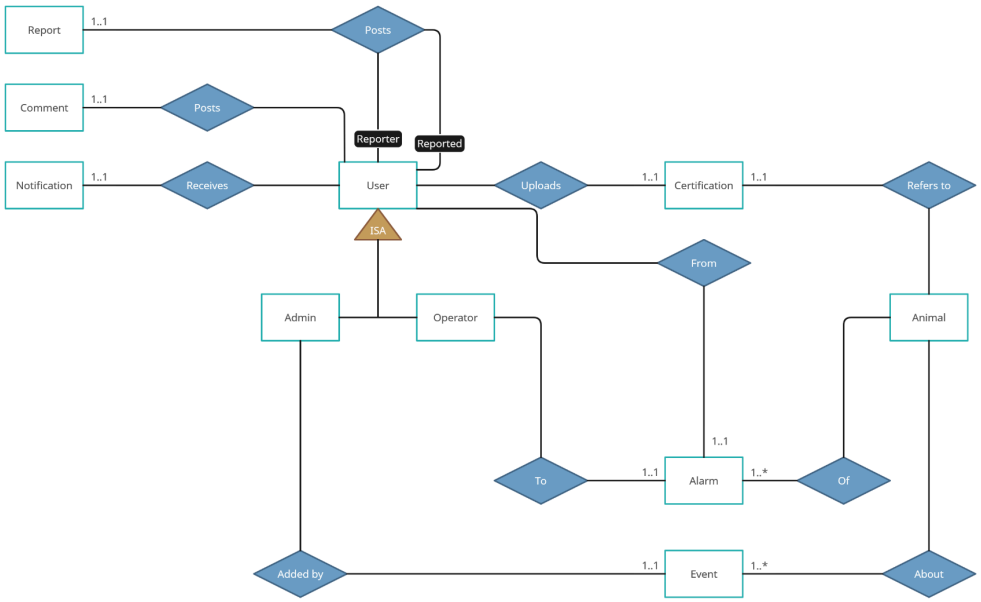
The application is designed to be both educational and entertaining, accessible to anyone interested in Italian fauna.

Project Phases

1. Design and Modeling

During the design phase, we developed several diagrams and documents to describe the application's functionalities and structure. The tools used for modeling include:

- **ER Diagram**: An entity-relationship diagram representing the main entities of the application, such as users, animals, operators, etc., along with their relationships, attributes and constraints.



ER Entities, Relations and Attributes:

User (mail, password, username, name, surname, points, birthday, setting_1, setting_2, setting_3, profileImage)
Admin (usermail)
Operator (usermail, operator_code)
Comment (commentID, date, text)
Notification (NotificationID, text)
Report (ReportID, status, date)
Certificate (Image)
Animal (AnimalID, name, description, regions, standardPoints)
Event (EventID, StartDate, EndDate, BonusPoints)
Alarm (AlarmID, position, time, status, info, UserEvaluation, OperatorEvaluation)

- **Microservices Design**: The application is based on microservices to manage the core functionalities (User Management, Certification, Notifications, Forum, Animal Recognition).
- **Mockups**: Visual prototypes of the user interface created with Balsamiq, representing the final appearance of the application and facilitating the understanding of the user flow.

2. Technologies Used

The project was developed using various technologies, each with specific features and capabilities. These are divided between the **backend** (for users, administrators, and operators), the **backend** of the central server, and the frontend.

Backend for Users/Administrators/Operators:

- **RabbitMQ**: Manages asynchronous communication between microservices.
- **Node.js**: JavaScript framework used for building efficient user server.
- **Flask**: Python framework used for API routing.
- **Spring Boot**: Java framework for building microservice applications.
- **Google OAuth**: Google-based authentication system.
- **PostgreSQL**: Relational database.

Backend for the Central Server:

- **Spring Boot**: Manages the central application logic.
- **PostgreSQL**: Relational database.

Frontend:

- **Thymeleaf**: Java template engine for building HTML interfaces.
- **HTML, CSS, JavaScript**: Technologies used for frontend development.

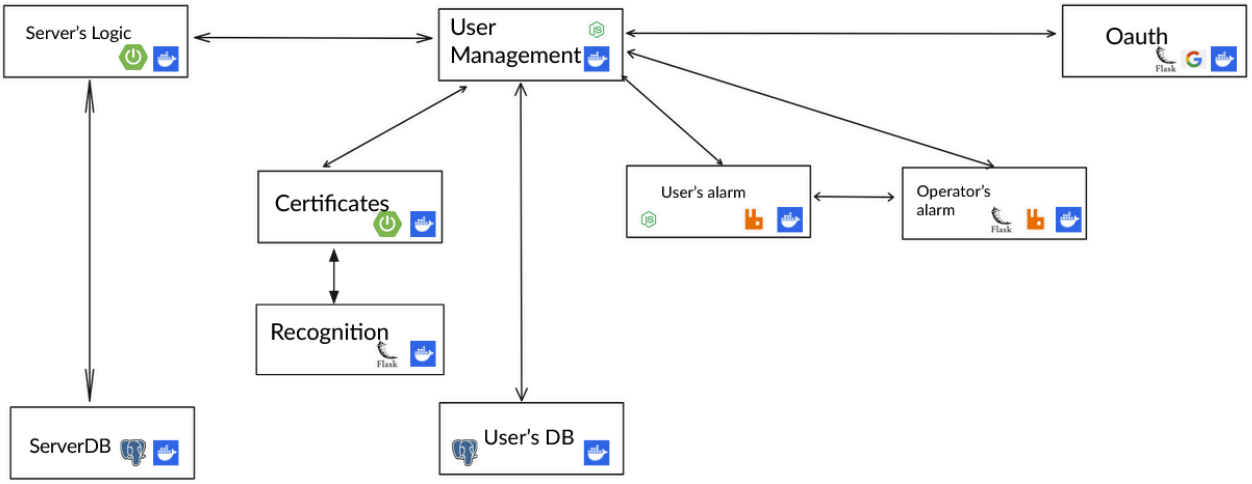
3. Microservices

The application is divided into microservices, each managing specific functionalities:

- **User Management**: Registers, authenticates, and manages user, admin and operator information.
- **Certificates**: Issues certifications to users who contribute to the project.
- **Notifications**: Sends alerts and updates to users.
- **Forum**: A space for users to discuss and share experiences.
- **Animal Recognition**: Helps users identify animal species they encounter.

4. Application Architecture

The architecture follows the microservices paradigm, with different components working together:



5. Function Points

The team used **Function Points** to evaluate the software's functional size, assessing the interactions between users, the database, and external services. This approach allowed the team to quantify the complexity of the project and estimate the resources needed.

Reference Table

RET/DET	1-19 DET	20-50 DET	51+ DET
0-1 RET	Low 6/4	Low 6/4	Med. 9/7
2-5 RET	Low 6/4	Med. 10/7	High 15/10
6+ RET	Med. 10/7	High 15/10	High 15/10

EI

FTR/DET	1-4 DET	5-15 DET	16+ DET
0-1 FTR	Low 1	Low 2	Med. 3
2 FTR	Low 3	Med. 4	High 5
3+ FTR	Med. 5	High 6	High 7

EO/EQ

FTR/DET	1-4 DET	5-15 DET	16+ DET
0-1 FTR	Low 2/1	Low 3/2	Med. 4/3
2 FTR	Low 4/3	Med. 5/4	High 6/5
3+ FTR	Med. 5	High 6	High 7

Computed Function Points

Table Name	EIF	ILF	DET	RET	FP
Notification	0	1	3	1	6
Certificate	1	1	4	1	6
Event	0	1	8	1	6
Comment	0	1	3	1	6
Report	0	1	5	1	6
Alarm	0	1	10	1	6
Animal	0	1	5	1	6
Operator	0	1	6	1	6
User	1	1	14	1	6

Total amount of FP
119

EI	FTR	DET	FP
User/Admin deletion	1	8	2
Registration User/Admin	1	14	2
Registration Operator	1	6	2
Operator Deletion	1	3	1
User/Admin modification	1	10	2
Operator modification	1	5	2
Delete comment	1	2	1
Add comment	1	3	1
Modify comment	1	2	1
Uploading certificate	1	3	1
Modification certificate	1	1	1
Modify/Insert profile image	1	1	1
Login Operator	1	2	1
Login Admin/User	1	2	1
Admin/User add report	1	5	2
Ban user	1	1	1
Add event	1	8	2
Modify user evaluation by operator	1	2	1
Modify operator evaluation by user	1	2	1
Add Alarm by User	1	10	2
Modify Alarm by Operator	1	2	1

EO	FTR	DET	FP
AnimalDex filter per User	2	4	4
AnimalDex filter per region	1	4	2
AnimalDex filter per name	1	4	2
Sort AnimalDex	1	4	2
Forum filter by username	1	3	2
Forum filter by date	1	3	2
View map	2	2	4
View scoreboard	1	2	2
Show winner	1	2	2

EQ	FTR	DET	FP
View Animal	1	4	1
View User	1	11	2
View notification	1	1	1
View certificated Animal	2	6	4
View report	1	5	2
View Alarm	1	8	2
View evalutions	1	6	2

6. Cost Estimation - Cocomo II

The **Cocomo II** model was used to estimate the project's cost based on its complexity and the resources employed. This methodology provided a precise estimate of development costs, broken down into the different phases of the project.

COCOMO II - Constructive Cost Model

Monte Carlo Risk: Off
Auto Calculate: Off

Software Size
Sizing Method: Function Points
Unadjusted Function Points: 119
Language: Java

Software Scale Drivers

Precedentedness	Low	Architecture / Risk Resolution	Low	Process Maturity	Very Low
Development Flexibility	Nominal	Team Cohesion	High		

Software Cost Drivers

Product		Personnel		Platform	
Required Software Reliability	Low	Analyst Capability	Very Low	Time Constraint	Nominal
Data Base Size	Low	Programmer Capability	Low	Storage Constraint	Nominal
Product Complexity	Very Low	Personnel Continuity	Nominal	Platform Volatility	Low
Developed for Reusability	Low	Application Experience	Low	**Project**	
Documentation Match to Lifecycle Needs	Nominal	Platform Experience	Low	Use of Software Tools	Nominal
		Language and Toolset Experience	Low	Multisite Development	Very Low
				Required Development Schedule	Low

Maintenance Off

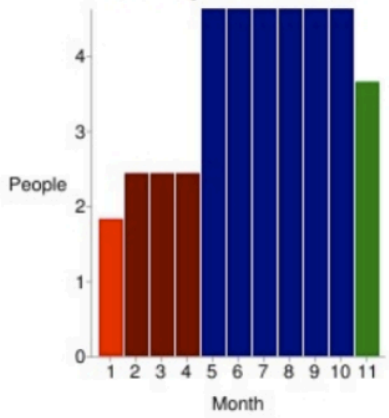
Software Labor Rates
Cost per Person-Month (Dollars):
Calculate

Software Development (Elaboration and Construction)

Effort = 37.1 Person-months
Schedule = 9.8 Months
Cost = $37141

Total Equivalent Size = 6307 SLOC
Effort Adjustment Factor (EAF) = 1.53

Staffing Profile



Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	2.2	1.2	1.8	$2228
Elaboration	8.9	3.7	2.4	$8914
Construction	28.2	6.1	4.6	$28228
Transition	4.5	1.2	3.7	$4457

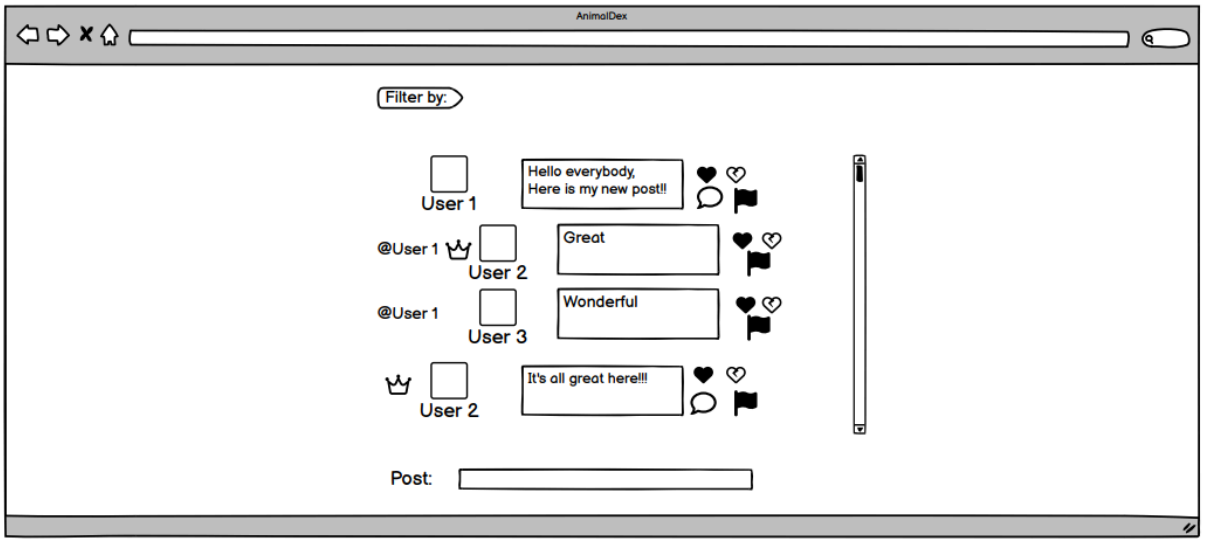
Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.3	1.1	2.8	0.6
Environment/ CM	0.2	0.7	1.4	0.2
Requirements	0.8	1.6	2.3	0.2
Design	0.4	3.2	4.5	0.2
Implementation	0.2	1.2	9.6	0.8
Assessment	0.2	0.9	6.8	1.1
Deployment	0.1	0.3	0.8	1.3

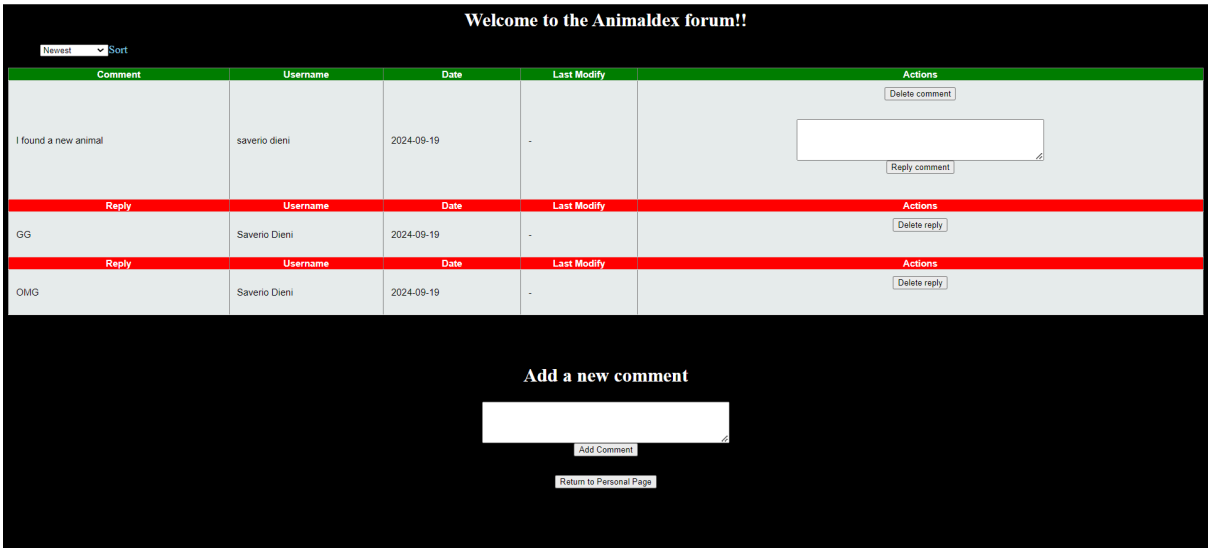
7. User Interface Mockups

Mockups created with Balsamiq provide a visual representation of the application's main screens. These prototypes were used during the design phase to preview the user interface and improve the final user experience.

Forum

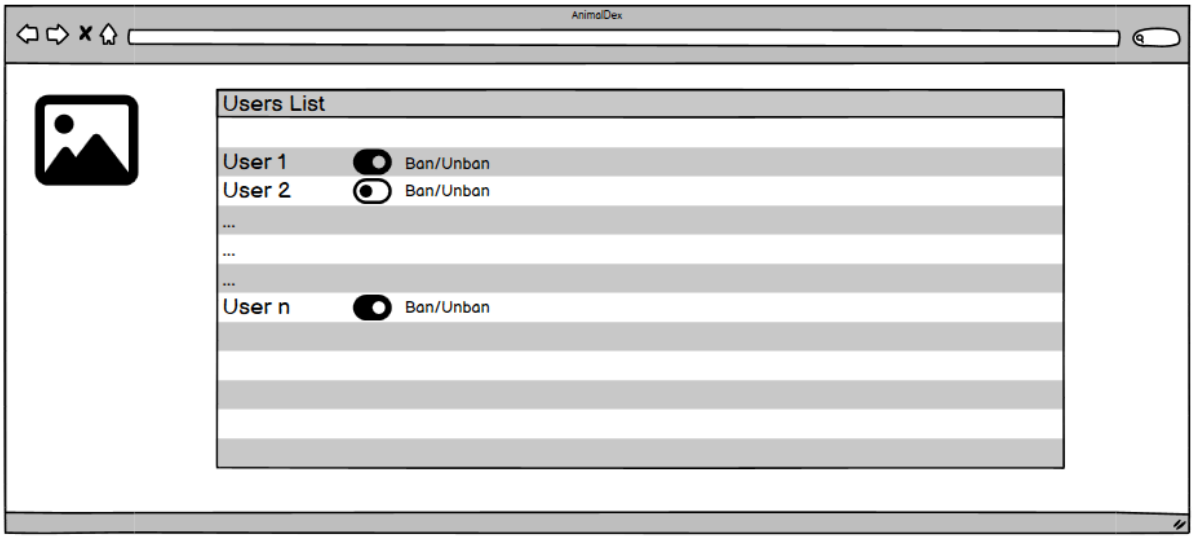


Forum mockup



Final version

Users list



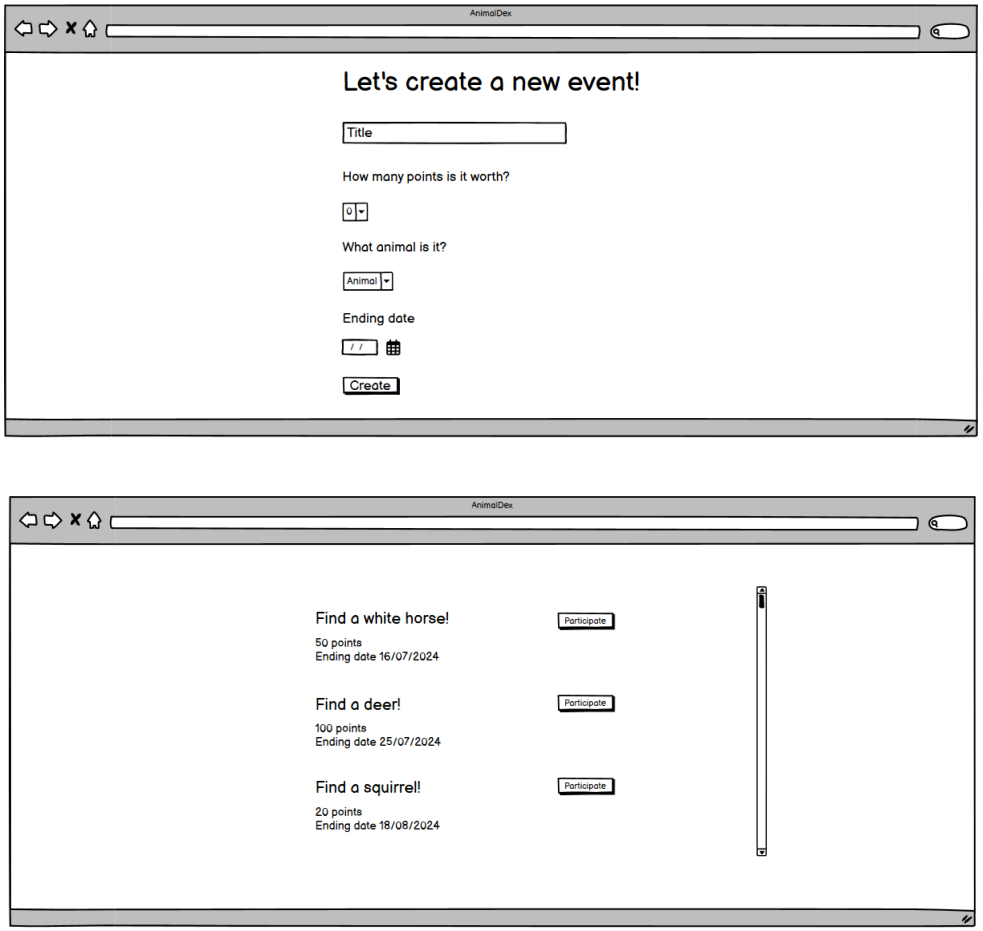
Users list mockup

Users List

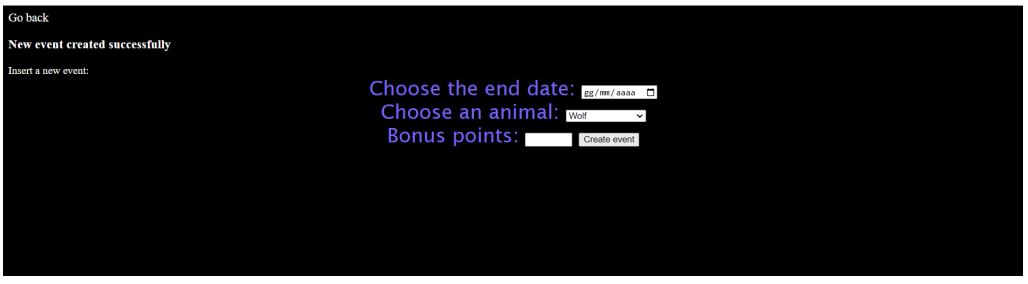
User ID	Email	Username	First Name	Surname	Points	Birthday	Fav Animal	Forum Notify	Emergency Notify	Administrator	
999	doe@mail.com	user999	John	Doe	0	1990-01-01	1	false	false	false	Permanent Ban
1000	berardi@mail.com	user1000	Mimmo	Berardi	0	1994-08-01	1	false	false	false	Permanent Ban
2	saveriodieni@gmail.com	Saverio Dieni	null	null	0	null	null	false	false	false	Permanent Ban
1	dieni.1946039@studenti.uniroma1.it	saverio dieni	null	null	175	null	null	false	false	false	Permanent Ban
3	projectanimaldex@gmail.com	Saverio Dieni	null	null	0	null	null	false	false	false	Permanent Ban
4	pino@gmail.com	darkpino	Pino	Gino	0	2024-08-26	null	false	false	true	Permanent Ban

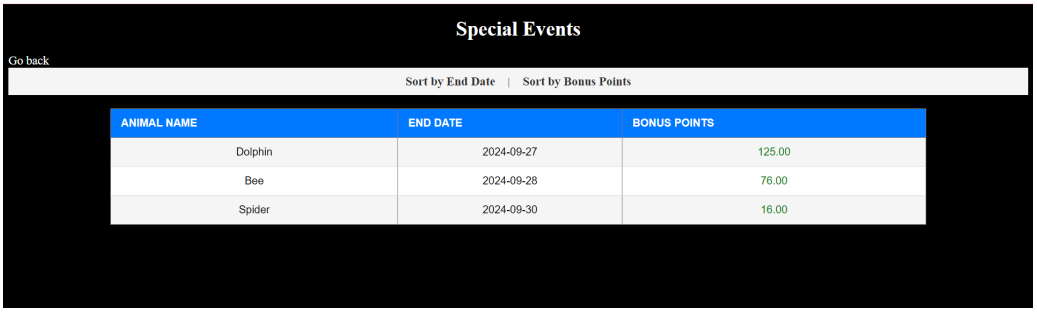
Final version

Special events



Special events mockup

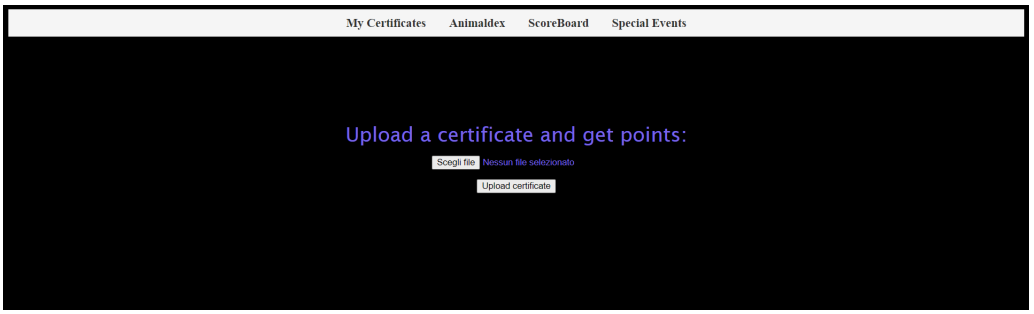




Final version

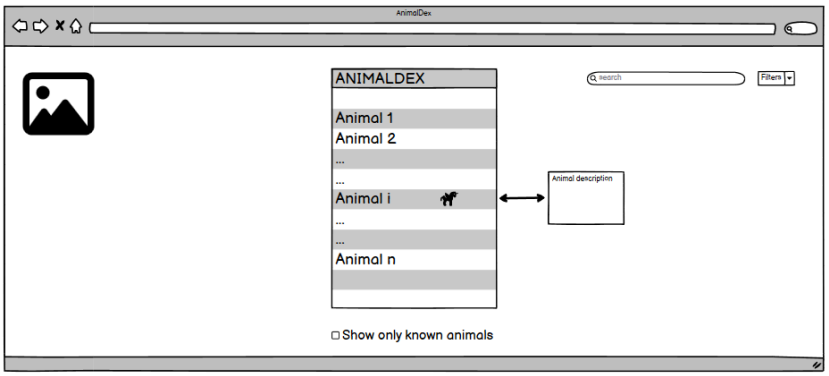
Certificates

Certificates mockup

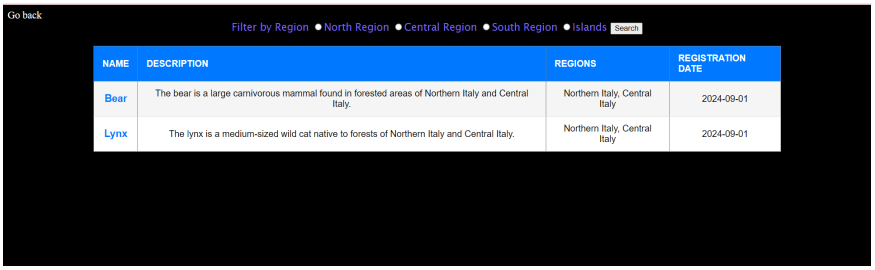


Final version

Animaldex

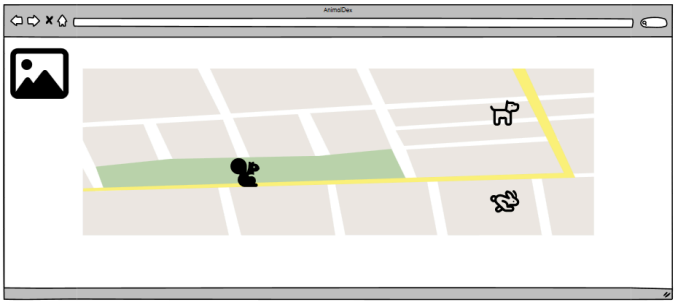


Animaldex mockup



Final version

Map

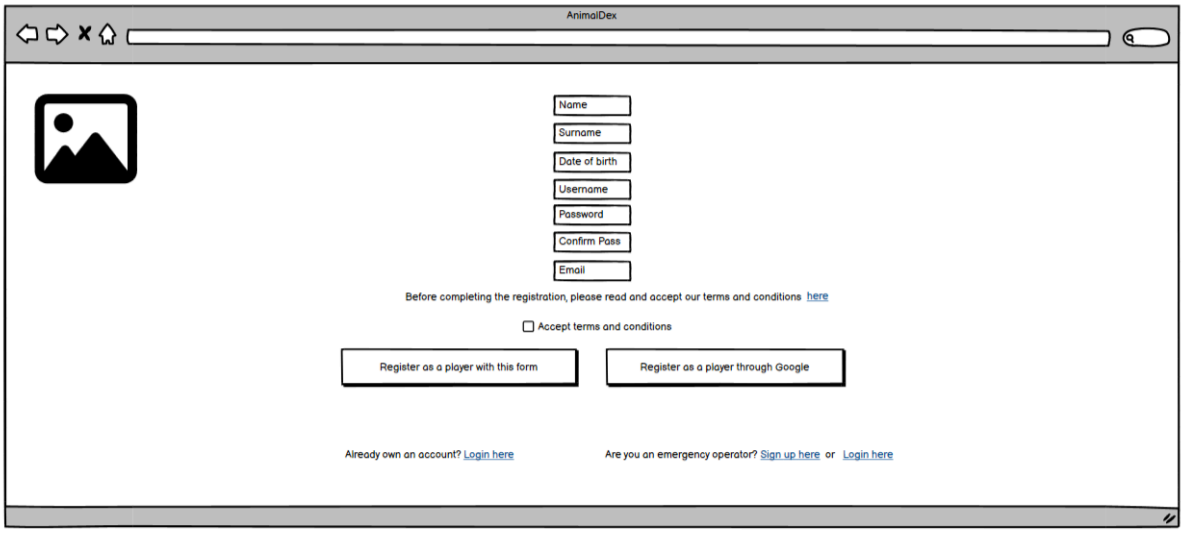


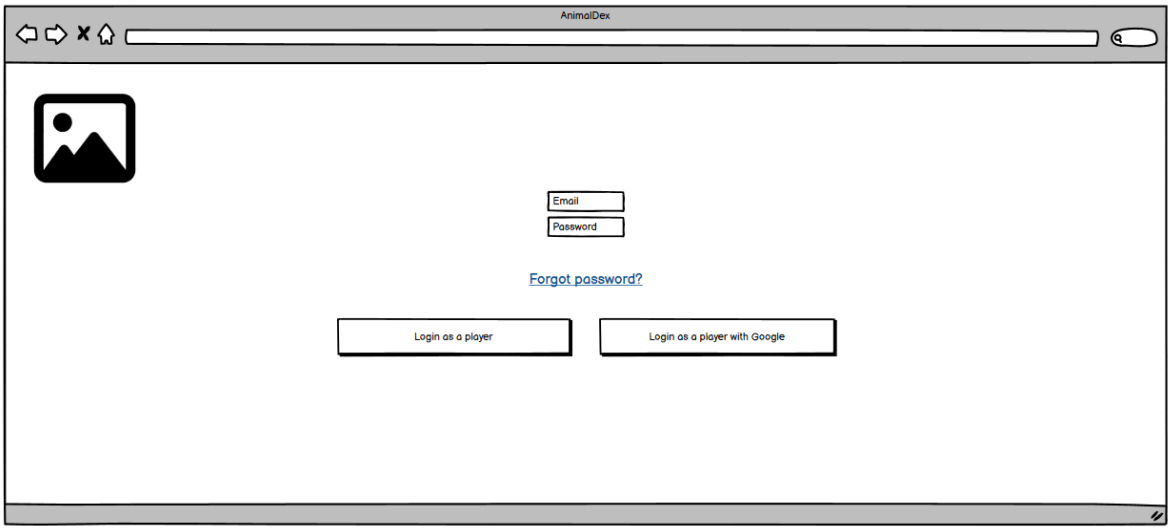
Map mockup

Final version

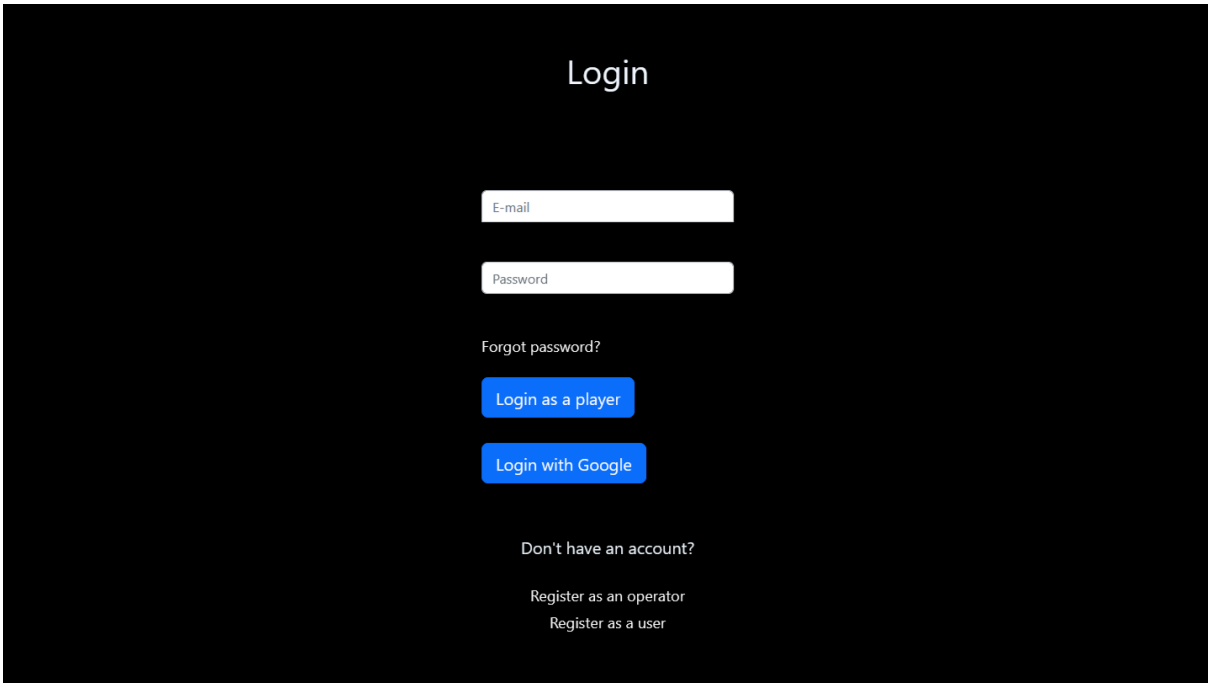
Login and registration

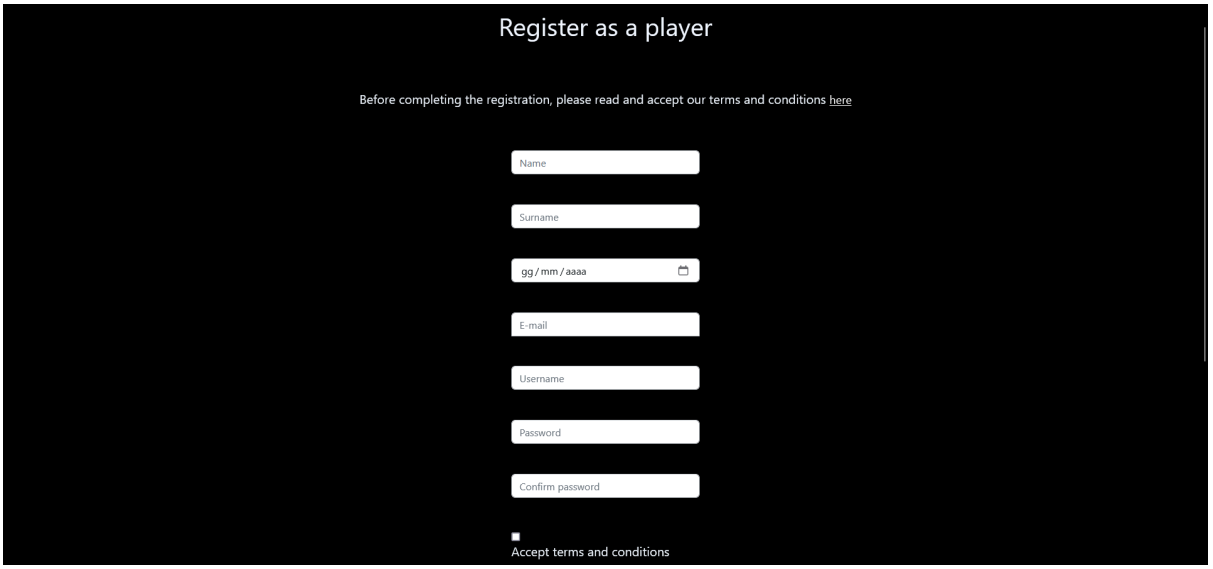
Mockup:





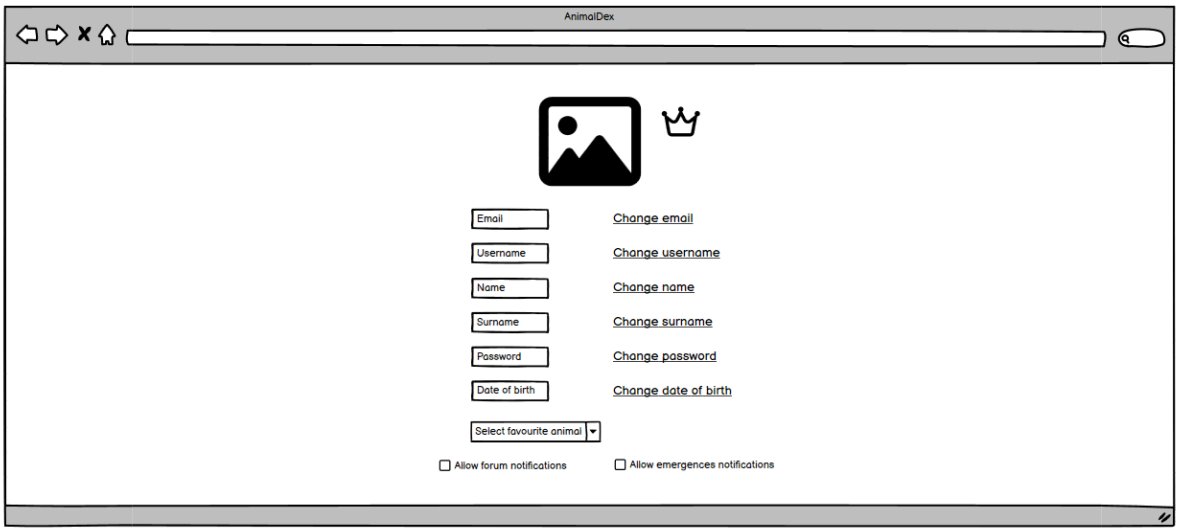
Final version:

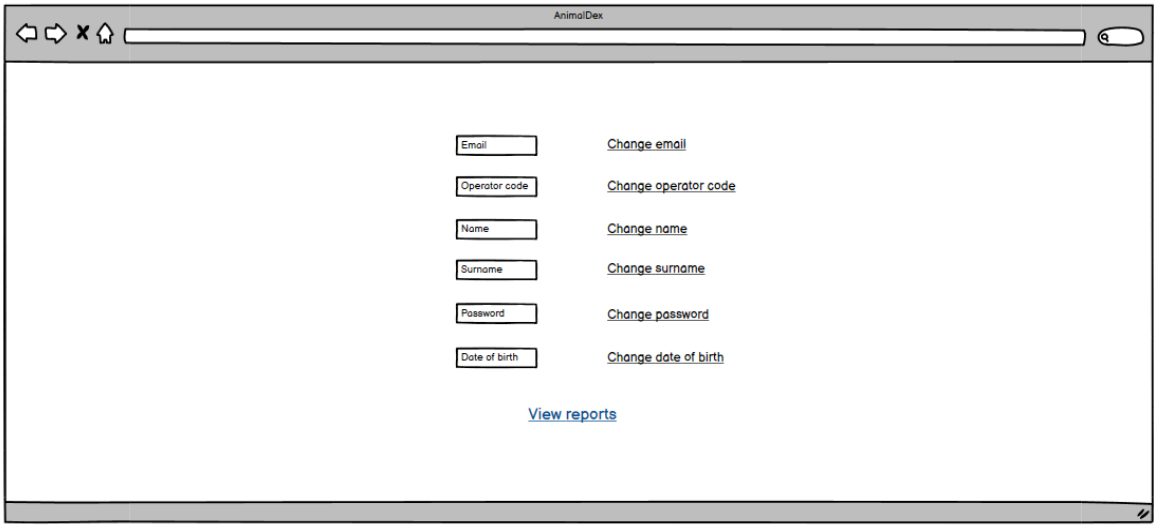




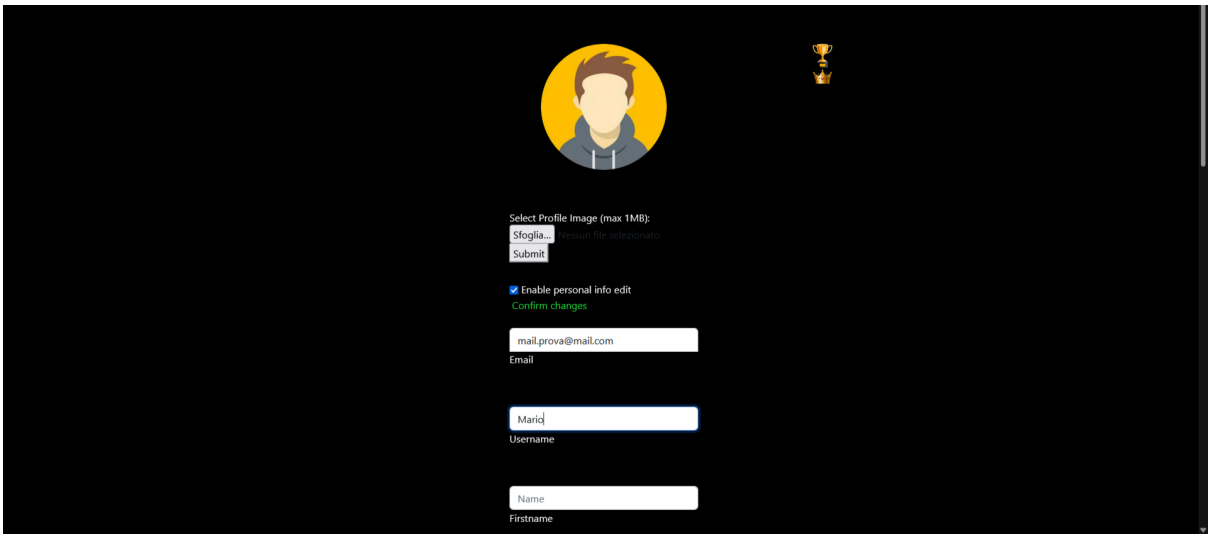
User’s and Operator’s personal pages

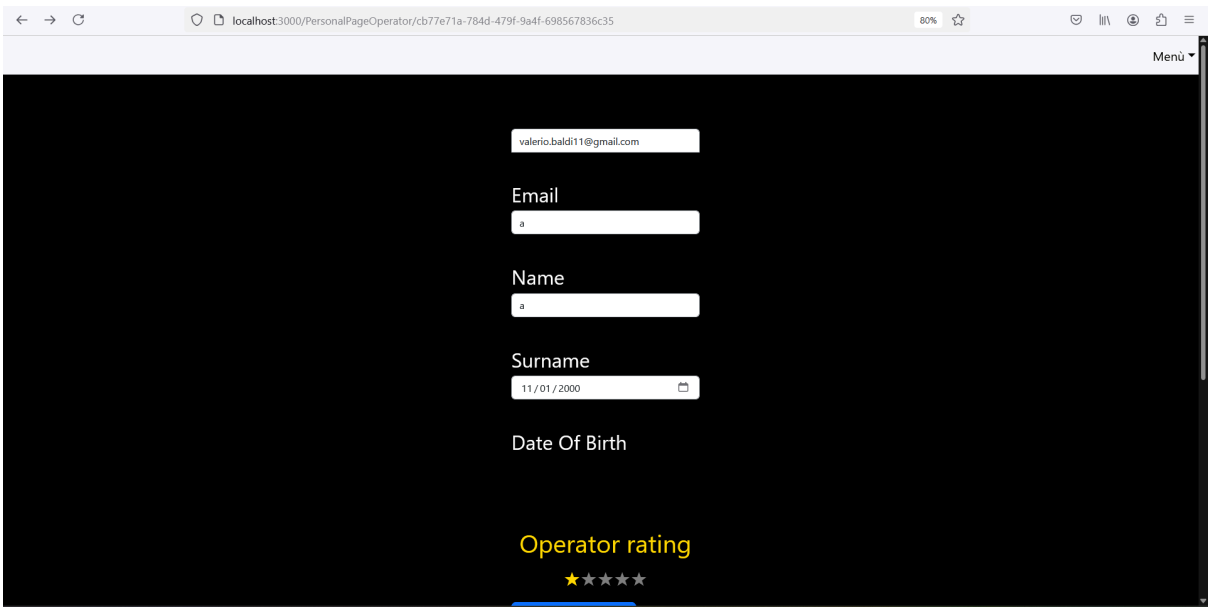
Mockup:





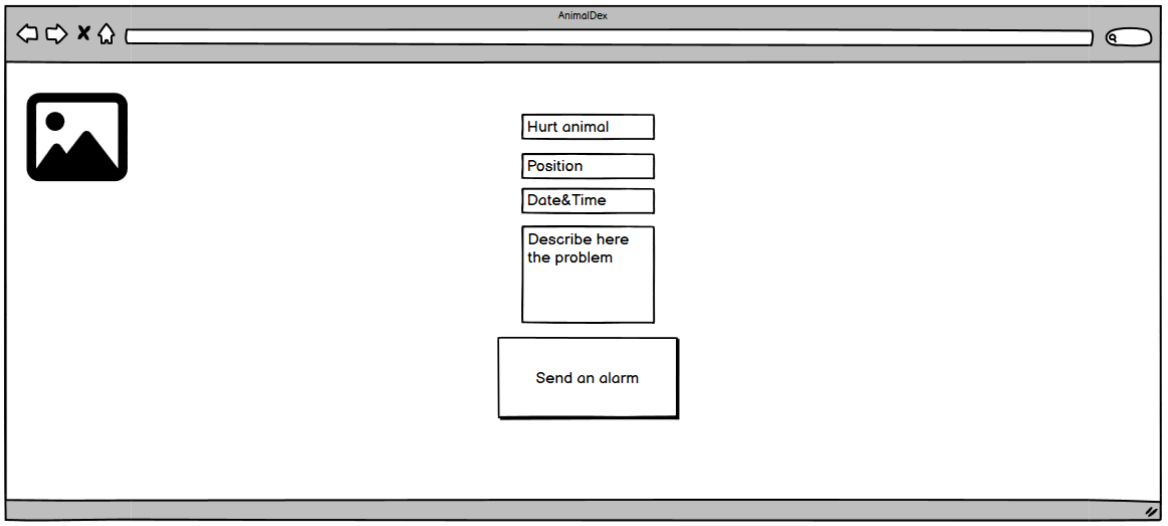
Final version:

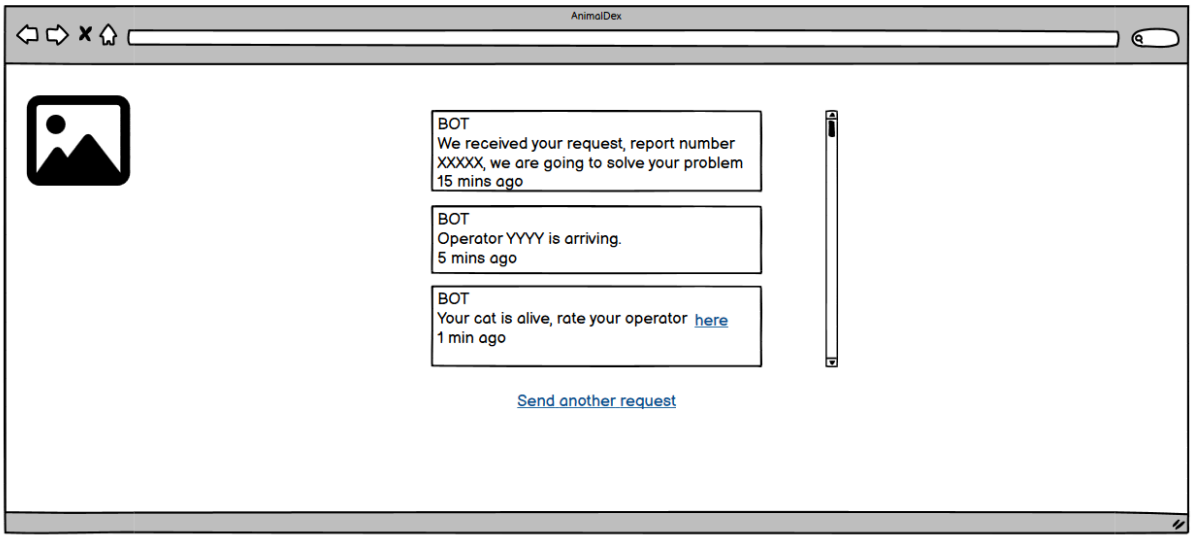


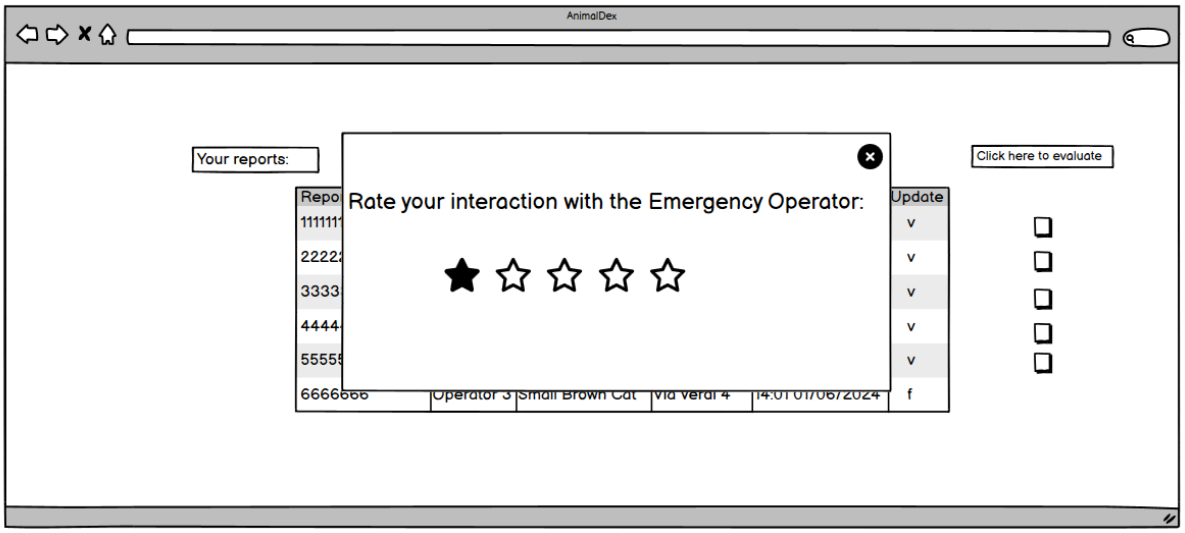


Animal reports

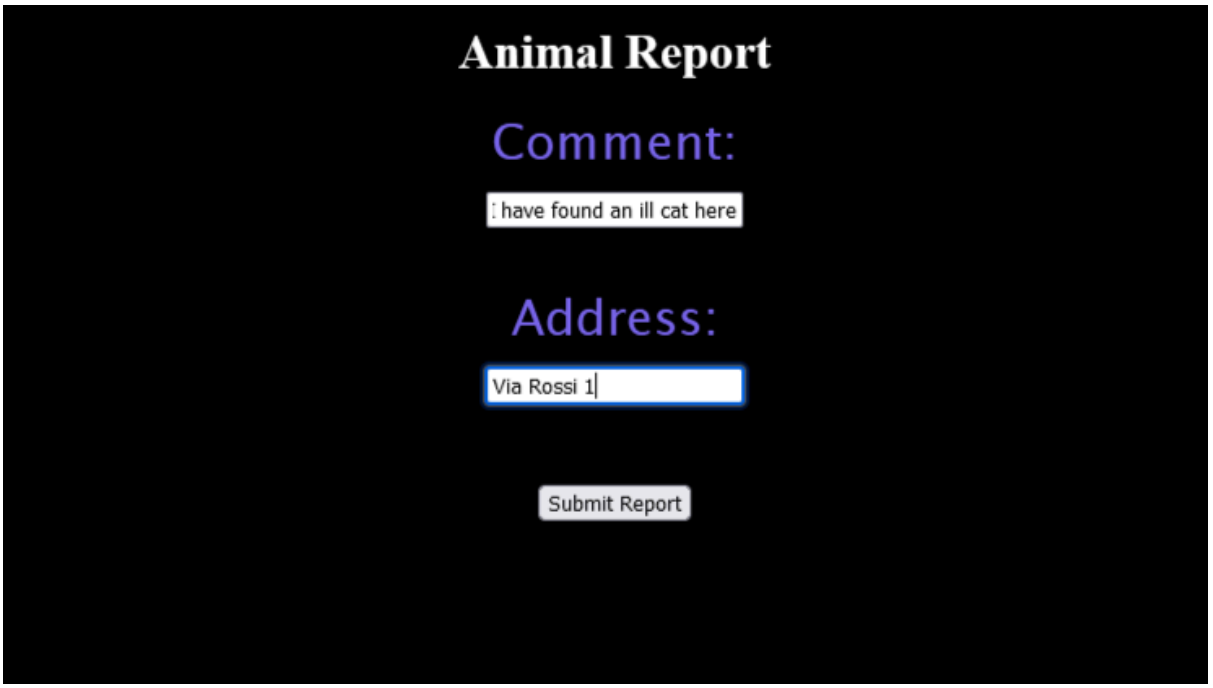
Mockup:

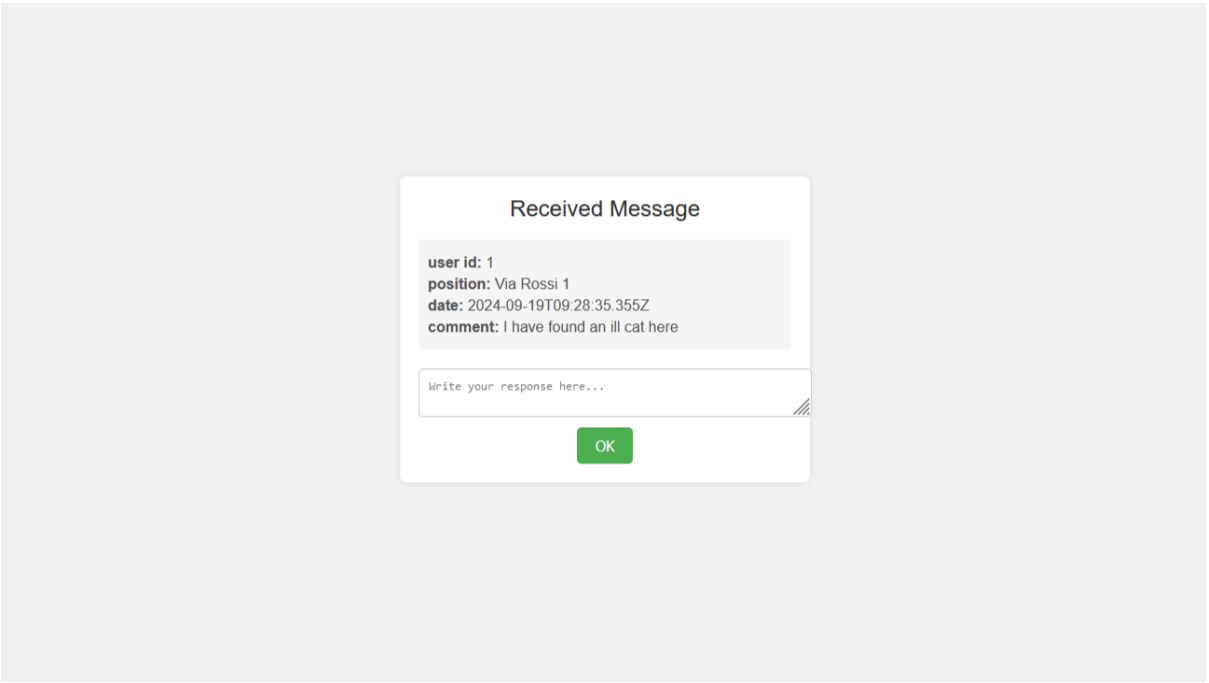


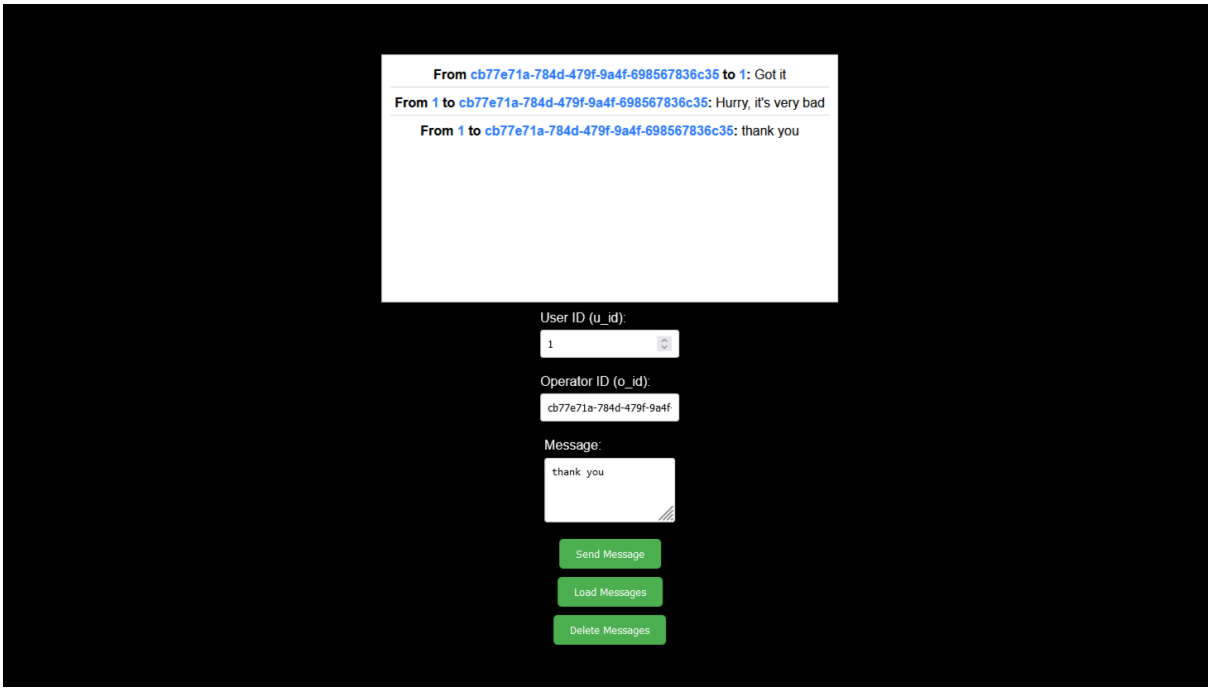


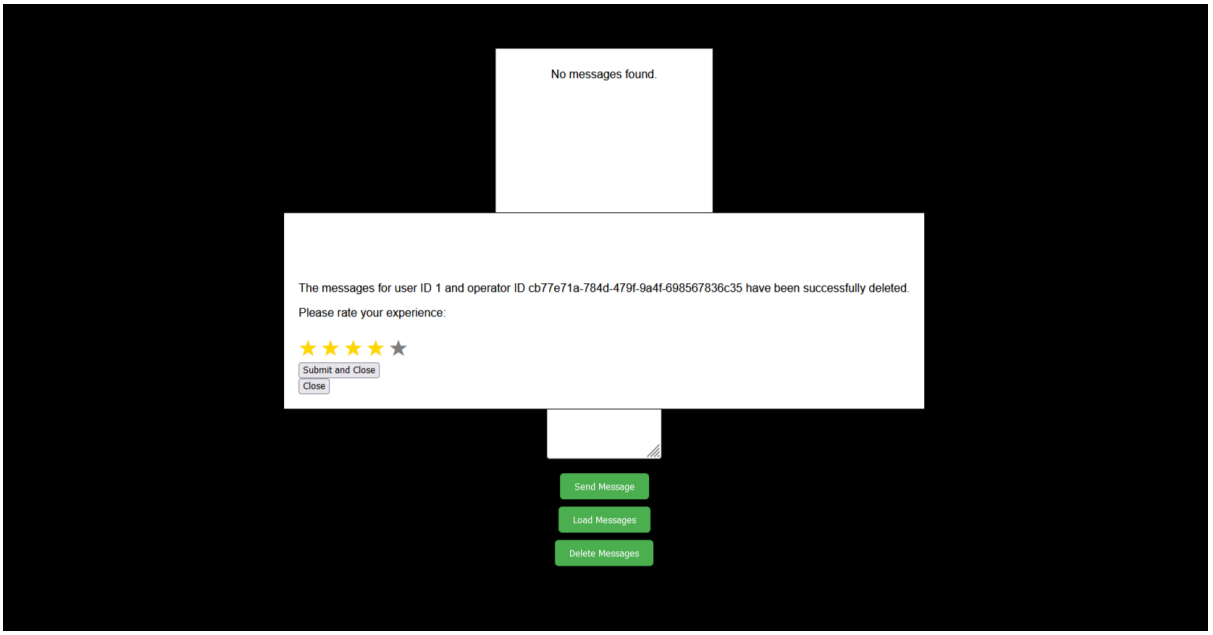


Final version:




Received Message
user id: 1
position: Via Rossi 1
date: 2024-09-19T09:28:35.355Z
comment: I have found an ill cat here
Write your response here...
OK


From cb77e71a-784d-479f-9a4f-698567836c35 to 1: Got it
From 1 to cb77e71a-784d-479f-9a4f-698567836c35: Hurry, it's very bad
From 1 to cb77e71a-784d-479f-9a4f-698567836c35: thank you
User ID (u_id):
1
Operator ID (o_id):
cb77e71a-784d-479f-9a4f
Message:
thank you
Send Message
Load Messages
Delete Messages



8. SCRUM Methodology

The development process was managed using the **SCRUM** methodology, breaking the work into sprints. Each sprint covered different phases of the project, from the initial design to the implementation of key features such as user management, certifications, notifications, the forum, and finally testing and final presentation.

The SCRUM process included the following phases:

1. Design
2. User Management and Certifications
3. OAuth Integration
4. Testing and Optimization

A **burndown chart** was used to monitor progress and ensure timely completion of each sprint.

WORK BREAKDOWN STRUCTURE	TASK TITLE	TASK OWNER	AMOUNT OF WORK IN HOURS			SPRINT	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
			ESTIMATE	COMPLETED	REMAINING					
1	Design of project		15	16	-1	1				107%
1,1	ER Schema and User stories	everyone	4	4	0	1	04/03/2024	06/03/2024	3	100%
1.2	LoFi	everyone	3	3	0	1	06/03/2024	08/03/2024	3	100%
1.3	Function Points	everyone	5	4	1	1	13/03/2024	14/03/2024	2	80%
1.4	COCOMO II	everyone	1	1	0	1	14/03/2024	14/03/2024	1	100%
1.5	System architecture estimation	everyone	1	2	-1	1	20/03/2024	22/03/2024	3	200%
1.6	II Sprint definition	everyone	1	2	-1	1	22/03/2024	22/03/2024	1	200%
2	Implementation of II Sprint features		13	13	0	2				100%
2,1	First drafting of db structure	Alessandro Catalano	2	2	0	2	25/03/2024	25/03/2024	1	100%
2,2	First drafting of frontend	Claudia Di Paolo	3	3	0	2	25/03/2024	25/03/2024	1	100%
2,3	First drafting of certification service	Saverio Dieni	5	5	0	2	25/03/2024	26/03/2024	2	100%
2,4	Database within the docker container	Alessandro Catalano	3	3	0	2	27/03/2024	27/03/2024	1	100%
2,5	Certification service within the docker container	Saverio Dieni	2	2	0	2	29/03/2024	29/03/2024	1	100%
2,6	Signup/login Logic Part 1	Catalano and Di paolo	4	6	-2	2	30/03/2024	12/04/2024	14	150%
2,7	Communication between certification and recognition	Saverio Dieni	2	2	0	2	02/04/2024	02/04/2024	1	100%
2,8	First version of central server	Saverio Dieni	8	10	-2	2	03/04/2024	05/04/2024	3	125%
2,9	Local db (sql file + docker)	Alessandro Catalano	2	2	0	2	10/04/2024	10/04/2024	1	100%
2.10	Certificate Image's Handling	Valerio Baldi	3	10	-7	2	02/04/2024	13/04/2024	12	333%

3	Implementation of III Sprint features		23	25	-2	3				109%
3,1	Serving frontend with docker	Alessandro Catalano	2	2	0	3	16/04/2024	16/04/2024	1	100%
3,2	Signup/Login Logic Part 2	Everyone	5	5	0	3	17/04/2024	24/04/2024	8	100%
3.2.1	user animal certificate	Saverio Dieni	1	1	0	3	19/04/2024	19/04/2024	1	100%
3.2.2	User Personal Animaldex	Valerio Baldi Saverio Dieni	2	2	0	3	19/04/2024	19/04/2024	1	100%
3,3	Signup/Login Logic Part 3 + operator table reshape	Alessandro Catalano	4	4	0	3	22/04/2024	24/04/2024	3	100%
3,4	PersonalPage Logic redirect and data handling	Everyone	3	5	-2	3	23/04/2024	24/04/2024	2	167%
3.5	Complete Animaldex	Valerio Baldi	1	1	0	3	25/04/2024	25/04/2024	1	100%
3,6	Google Oauth	Valerio Baldi Saverio Dieni	4	4	0	3	24/04/2024	27/04/2024	4	100%
3.7	Animaldex Sorting	Valerio Baldi	1	1	0	3	27/04/2024	27/04/2024	1	100%
3.8	PersonalPage Logic data displaying	Catalano Di Paolo	3	3	0	3	25/04/2024	27/04/2024	3	100%
3.9	Animaldex Search Bar and Personaldex filter	Valerio Baldi	2	2	0	3	03/05/2024	03/05/2024	1	100%
3.10	Cookie management and integration	Saverio Dieni	4	5	-1	3	02/05/2024	04/05/2024	3	125%
3.11	Delete user account	Catalano Di Paolo	4	3	1	3	03/05/2024	03/05/2024	1	75%
4	Implementation of IV Sprint features		19	17	2	4				89%
4,1	Delete operator account	Catalano Alessandro	1	1	0	4	09/05/2024	09/05/2024	1	100%
4,2	ScoreBoard	Saverio Dieni	2	2	0	4	10/05/2024	10/05/2024	1	100%
4,3	Updating personal info of user and operator	Catalano Alessandro	6	6	0	4	12/05/2024	12/05/2024	1	100%
4.4	Winner announcement	Saverio Dieni	4	5	-1	4	15/05/2024	17/05/2024	3	125%
5	Implementation of V Sprint features		46	40	6	5				87%
5.1	user event interaction + finish special events	Saverio Dieni	5	5	0	5	22/07/2024	03/08/2024	13	100%
5.2	show user list for admin	Alessandro Catalano	5	4	1	5	30/07/2024	02/08/2024	4	80%
5.3	faq page	Catalano/Di Paolo	1	1	0	5	02/08/2024	02/08/2024	1	100%
5.4	operator and user ban by admin	Alessandro Catalano	2	2	0	5	03/08/2024	03/08/2024	1	100%
5.5	Profile image pt2 and centralDB restructuring	Catalano/Di Paolo	3	3	0	5	05/08/2024	05/08/2024	1	100%
5.6	forum add comments and replies	Alessandro Catalano	3	3	0	5	05/08/2024	05/08/2024	1	100%
5.7	forum delete comments and replies by user and adm	Catalano/Di Paolo	2	2	0	5	06/08/2024	06/08/2024	1	100%
5.8	forum improved features	Alessandro Catalano	1	1	0	5	06/08/2024	06/08/2024	1	100%
5.9	modify comments and replies forum and forum filter	Catalano/Di Paolo	3	2	1	5	07/08/2024	07/08/2024	1	67%
5.10	Personal Map	Valerio Baldi	2	3	-1	5	09/08/2024	09/08/2024	1	150%
5.11	animal emergency	Saverio Dieni	8	6	2	5	06/08/2024	09/08/2024	4	75%
5.12	forum now on user side	Alessandro Catalano	3	3	0	5	11/08/2024	11/08/2024	1	100%
5.13	messages between users and operators	Saverio Dieni	8	5	3	5	20/08/2024	20/08/2024	1	63%
6	Implementation of VI Sprint features		9	9	0	6				100%
6.1	Emergency operator evaluation	Baldi/Dieni	3	3	0	6	27/08/2024	27/08/2024	1	100%
6.2	User evaluation	Baldi/Dieni	1	1	0	6	28/08/2024	28/08/2024	1	100%
6.3	update on changing favourite animal and displaying i	Alessandro Catalano	2	2	0	6	28/08/2024	28/08/2024	1	100%
6.4	rest mapping the report and chat microservice	Catalano/Di Paolo	2	2	0	6	28/08/2024	28/08/2024	1	100%
6.5	split change password form change credentials	Catalano	1	1	0	6	28/08/2024	28/08/2024	1	100%
6.6	Testing	Everyone	4	4	0	6	02/09/2024	03/09/2024	2	100%
6.7	Presentation	Everyone	1	1	0	6	07/09/2024	07/09/2024	1	100%
	BURNDOWN DATA	TOTAL HOURS	172	176	-4	56	3,071429			DAY