

Documentation & Project Diary

Innovation Lab 3
Year 2023/2024

Project: **Animal Certificate**

Team: **39**

1. General Information

Project name: Animal Certificate

Supervisor: Rohatsch Lukas, MSc

Innovation Lab 3 *winter term 2023*

Project-team:

Tolan Christopher, if22b198@technikum-wien.at, project manager

Cvetkovska Maja, if20b072@technikum-wien.at

(Slomka Philip, if19b217@technikum-wien.at), left team during term

Zeilingner David, if22b143@technikum-wien.at

Management Summary of the Project

The animal certificate project is all about connecting people - connecting them better with their beloved animals through validation of heritage. We will enable an easy solution for knowing a pet's owner, breed, animal breeder and parents by saving this information to the Sepolia blockchain.

After completion of this project people should be able to scan a pet's NFC-tag, and be presented with all available information about the animal on an easy to use, user oriented web-app.

Framework Conditions and Project Environment

The architecture of the project is divided into three main components. On one hand there is the hardware component, as we want to store some kind of linking data on NFC-tags. On the other hand we want to create the project's backend on the blockchain via a smart contract. Last but not least a frontend in the form of a React web-app.

In the hardware scope we are already provided with a python program, which is able to read and write data to and from NFC-tags. Project specifications request us to run this software on a Raspberry Pi computer, because of its easy to use general purpose IO pins used to attach the mentioned NFC-reader.

React is the tool of choice for creating the frontend. It is a simple js framework that should make it easy to create complex web apps. Since this app is designed for the average consumer, there is no accessibility target greater than the AA standard. In this module we will use unit tests to test the logic. However, since it is a user interface, tests are also necessary to test the comprehensibility of the app. Sufficient usability tests must also be carried out.

The critical point of our project is actually the smart contract. A fault in the contract not only could result in forgery of pet certificates, but even in the wiping of entire accounts. Therefore we need to handle ownership handling with special precaution. So, sufficient unit tests are required. Tests for Solidity (the blockchain programming language of choice) can be generated using the "Remix-Tests" function inside the Remix IDE.

It was advised we may run the contract on the Bloxberg blockchain, but due to technical limitations it was currently not possible to do so, so the Sepolia test-chain was chosen instead.

Since all parts of the project are dependent on each other, but do not build on each other and no modules are to be delivered ahead of time, the same completion period applies to all modules.

Semester-Roadmap

This semester, we're focusing on making things better, faster, and getting ready for a bigger launch. By the time we wrap up this term, we want to have a top-notch product that's ready for everyone to use, making sure pet owners can easily get all the info they need about their pets through our platform.

In our blockchain endeavors, we will concentrate on migrating our contract to the Bloxberg network, since we had some issues last term. Further improvements regarding functionality and code quality are also planned.

After the big UI overhaul in the last term, we will build upon that, refining everything and adding the last major feature, the family tree to the frontend. Adding some more functionality like removing and adding diseases, breeding and marking animals als dead, are also in scope.

Given the challenges experienced with the NFC reader in the previous semester, this semester will emphasize ensuring its seamless integration and functionality too. We aim to resolve compatibility and operational issues, ensuring that data can be reliably stored and retrieved from NFC chips.

Team roles remain the same as last semester: David will lead the web development initiatives, Chris will focus on blockchain-related developments and assist all other team members with their tasks, and Maja will be responsible for all NFC-related tasks.

Last semester, we welcomed Philip Slomka to our team. Philip took the time to dive into our project's tech side and got the hang of the tools and methods we use. Now, with that experience under his belt, he's ready to help us push the project forward this semester. As we move ahead, every team member, Philip included, will stick to their roles, ensuring we keep things smooth and specialized.

Note: Philip left the team during the 2023/24 winter term, thus not all the goals have been reached during this term. However the most significant ones are implemented and only minor ones are still missing.

Collaboration & Tooling

- Atlassian Jira - Collaboration Tool
 - <https://animalcertificate.atlassian.net/jira/software/projects/ANIMALCERT/boards/1> (permission request required)
- Atlassian Bitbucket - Code Repository (Deprecated)
 - <https://bitbucket.org/animalcertificate/animalcertificate/src/main/>
- Frontend Github - Code Repository
 - <https://github.com/Morplson/animalcert-dev>
- Hosting dev server
 - <https://animalcert-dev.vercel.app/>
- Discord - Communication
 - <https://discord.gg/jmxCbwgjvk>

Since the beginning of InnoLab1, we have used Jira as our collaboration tool and Discord as our communication platform. We will continue to do so in InnoLab3.

The React frontend is maintained in a separate GitHub repository due to our hosting choice. Vercel makes it easy to publish web apps and is mostly free. Gitbucket would require a subscription of \$9.99 per month. So, we decided against it. Vercel is a hosting service that hosts for free and can be built directly from GitHub. The main branch represents the above listed link, but there are also individual subpages for all feature branches, and you can even call individual commit builds yourself. Vercel also provides EU-compliant analytics.

2. Brief Project-Description

The project AnimalCertificate aims to transform the world of animal ownership with its blockchain-based registry, ensuring authentic ownership certificates that cannot be tampered. A standard for recording the very important details of a pet's origin, family tree, health-data and more using the technology of NFTs on Ethereum based blockchain.

The project scope will is roughly defined as following:

- In scope
 - NFTs can be minted with extended properties(vaccinations, color, tbd)
 - Family tree visualization extension(seeing siblings, cousins up to 3rd degree, parents up to great-great grandparents)
 - Integration with NFC read/write functionalities
 - Smart contract will run on an ethereum-based blockchain
 - Frontend will look modern and minimalistic
 - Frontend will have more features reflecting the smart contracts new features
 - Adding/Removing diseases to existing animals
 - Declaring animals as dead
 - Contract will be deployed on the Sepolia network
- Deliverables
 - Smart Contract (as solidity script)
 - Python script for reading NFCs over a microcontroller
 - Website to access the smart contract
- Out of scope
 - Expansion on other blockchains
 - Monetizing the smart contract
 - React native mobile app
- Acceptance criteria
 - Project will be accepted when it is agreed with the supervisor that all requirements are met
- Constraints
 - Illnesses of team members may affect the deadlines of user stories in sprints
 - Team members failing semester may result in reduction of personnel thus reducing available resources
 - The project must be done by the end of the 3rd term
- Assumptions
 - All promised tools (microcontroller, licenses) are provided

3. Specification of the Solution

The Animal Certificate Project aims to provide a secure, reliable, and efficient system for managing animal data as NFTs. It is an existing project that we as a team took over from and are going to improve and enhance over the course of the next semesters. Since the technology stack was new for all the team members, this semester was focused more on learning the technologies and accommodating them. The following is a brief summary of what is already implemented:

- System environment

The solution provides a digital platform for the management of animal certificates. The system has the capability to create, store, and track animal certificates.

It only covers the issuance of certificates for many common domestic animals such as dogs and cats. Wild animals or livestock are not included.

- Features (functional requirements)

- User Management

- Ability to connect with a (metamask) wallet

- Certificate Management:

- Ability to issue certificates
- Ability to revoke certificates
- Ability to view and search certificates by animal, owner, and certificate number
- Ability to track the status of animal certificates
- Ability to update certain data of the certificate

- UI/UX:

- Clean website design
- Create animal certificate form
- View animal certificate details
- Update data
- Visualize family tree

4. Effort Estimation

For the upcoming semester, our approach to effort estimation remains consistent with our methodologies from the previous term.

We used the in the moodle provided excel sheet for estimating our efforts for the user stories we have defined for this term so far following the guidelines of the Delphi method we learned during the first lecture of the last term and the provided slides.

Each user story has an estimated effort, a pessimistic and an optimistic one. All the user stories created in the Jira board are added and updated regularly and added to corresponding sprints during the phase of the project.

We consider 8h the maximum estimation for a user story in a sprint, since it is the approximate workload we have per team member in one iteration. Therefore we cannot assign more hours to it to stay in guidelines with finishing user stories within one sprint. This will result in roughly 140h of workload for the team during this term. We have also considered the time it takes us to write documents and have meetings in this estimation.

(see end of the document for Chapter estimations)

5. Delivery

This semester's scope of delivery of our solution contains the following:

- Extended Animal Certificate Smart Contract as Solidity File (.sol)
- React Frontend
- Python Script for interacting with NFC chips

Since the contract used is stored on the Sepolia blockchain, we will not use any form of “local” persistence, so no database or anything alike. The frontend acts as a Web3 application and interacts with the blockchain, more specifically with the smart contract deployed on it.

Specifically for the NFC reading part, our lecture borrowed the following device to help us out with implementation.

- Raspberry Pi equipped with an RFID NFC reader

The Frontend is currently not accessible over the internet; however, it can be run using a localhost server to run the React app, following the usual procedure for running React apps locally. Additionally, we are actively working on implementing the capability to read NFC tags directly within the web app.

6. Our Project Diary

Term 1:

- **10.10.2022**

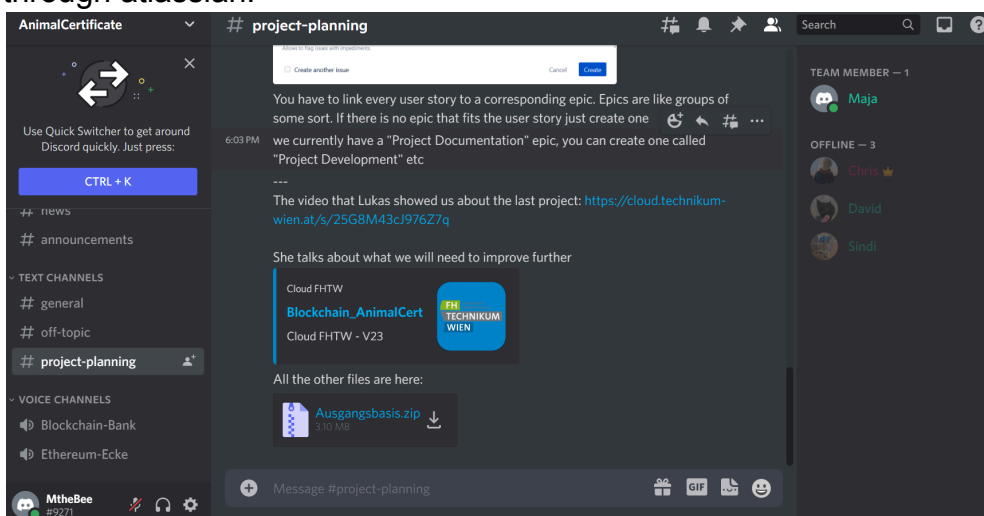
We contacted the supervisor via email to schedule our first meeting.

- **11.10.2022**

We received a return email from the supervisor with times to set up a meeting. By mutual agreement with the team, we have decided that the meeting will be on Thursday, October 13 via Zoom.

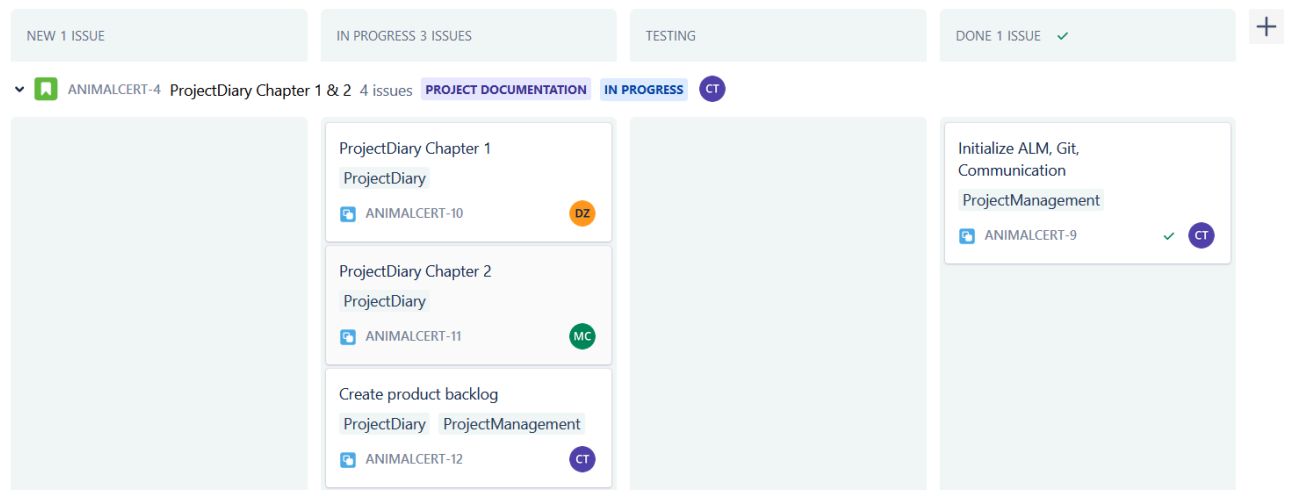
- **13.10.2022**

We held a meeting with the supervisor. First he explained to us what exactly is happening in the Project, how far did the previous group go and provide us with the necessary documents. We decided to communicate with each other through a discord channel and the assignments will be shared and the issues will be solved through atlassian.

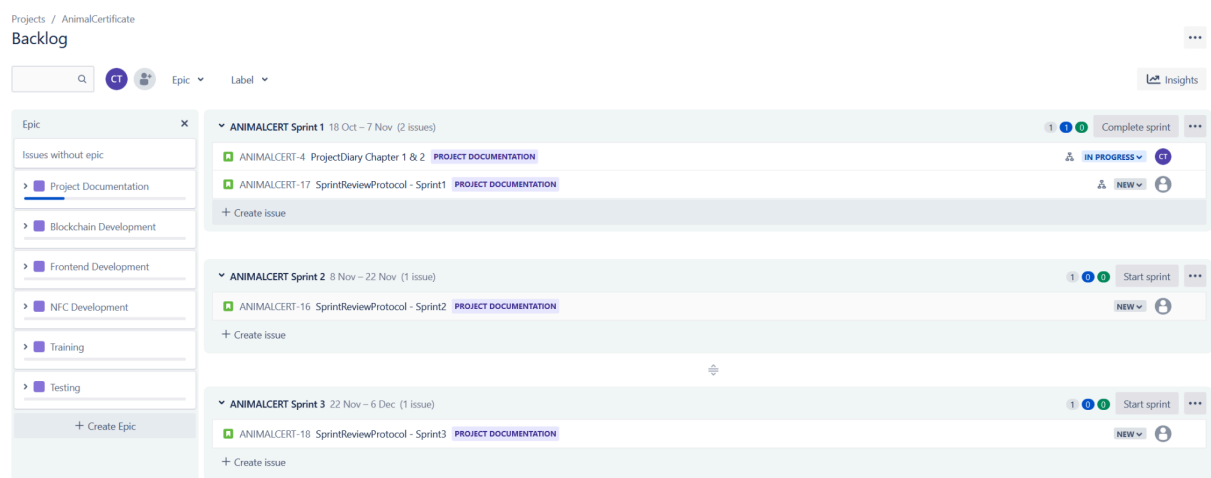


- **18.10.2022**

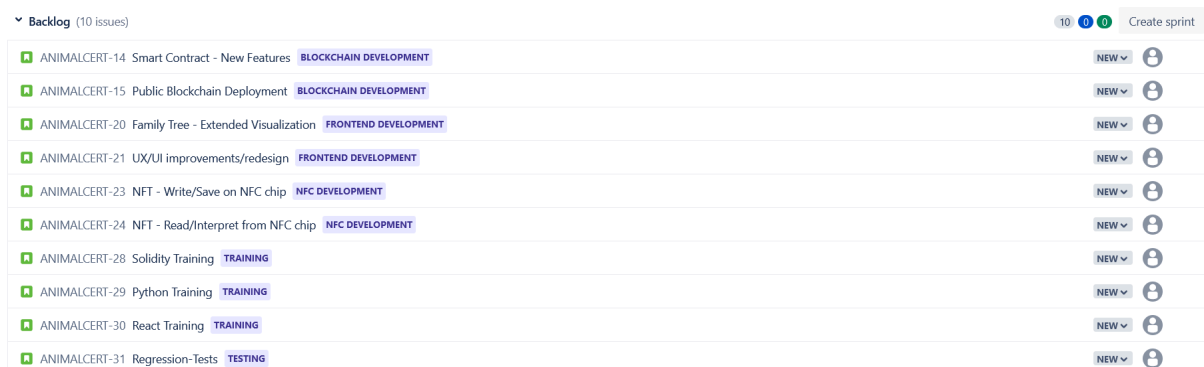
With the help of atlassian, everyone was given a specific issue to perform. Each of us successfully solved the assigned task. We talked to each other about the software we will use in the project and how the project will proceed.



A first sprint was created and the next 2 sprints are also predefined.



A backlog of stories were created to keep track of what needs to be done. It will get more refined with time.



- 9.11.2022

This week we lost our team member Kastrati due to study restriction because of failed exams in the previous semester.

The goal for the past week(s) was to train ourselves in various technologies. The basic concepts of Python were understood, including complex techniques. Practice in programming with Python was completed through exercising. The basics of React were also learned, including the use of JSX and the creation of React components. The communication between components in React was also studied, as well as the

use of React Hooks.

- **22.11.2022**

This week, progress was made in learning React and blockchain development. First progress was being made on the side of the smart contract, as we tested the abilities of Solidity using Remix, as well as the basics of web3.py.

- **6.12.2022**

This week, progress was made in Ethereum and Solidity development. While reading NFC in web was attempted, it was determined to not be applicable with the current web API. However, success was achieved in using web3.js to interact with the Ethereum blockchain.

- [Home](#)
- [nft](#)
- [Show all](#)
- [Animal with ID 0](#)
- [Mint a token](#)

Connect Wallet

Current Block Height: 8409651

The basics of Solidity were also covered, including types, the message global variable, arrays, mappings, structs, and the overview of variable scope and special variables.

- **20.12.2022**

This week, progress was made in the research and development of a blockchain project involving Ethereum and React. Testnet research was conducted and Goerli was chosen as the preferred testnet. A demo contract was created and deployed, and current smart contracts were tested. The deployment of a smart contract was also tested. React Router was installed and a simple application was created to understand React Routing. A Python program was developed that can read and write data to and from NFC.

- **17.1.2022**

This week, progress was made in the development of a project involving NFC technology.

But first off, a basic version of the smart contract was deployed to the Goerli-blockchain and is accessible for testing under the address [0xfe05a9212e9fb33e68b9864c52d2029e2ea04ce6](#).

Overview [ERC-721]

Max Total Supply: 3 ANIMAL_CERTIFICATE ⓘ

Holders: 2

Transfers: 3

We also set our sights on finally purchasing a specific NFC reader for the project. The product we settled on is the [Neuftech USB 13.56 MHz kontaktlos Kartenleser](#) and was subsequently purchased. Understanding of Python libraries for reading NFC over a microcontroller was also gained.

- 24.1.2022

Finally, the functionalities of the react app could fully be migrated to the new project. A user is now able to create new tokens on the page, as well as able to view other tokens.

Term 2:

- 14.03.2023

Estimations and documents are prepared for the continuation of the Animal Certificate project. User stories and epics were created in Jira and estimations were added in the provided excel file.

Delphi Verfahren mit PERT - Unsicherheit ausdrückbar durch Schätzbereich: Breite, p(range)									
ID	Themes / Areas / Arbeitspakete TopLevel	Epics / User Stories / Arbeitspakete TopLevel	User Stories / Detail Level / Beschreibung	Optimistisch (Sp)	Wahrscheinlich (Sp)	Pessimistisch (Sp)	Dokumentation von Annahmen, Überlegungen, Risiken, Bedingungen und Diskussionsergebnissen aus dem Schätzmeeting	Erwartet nach PERT 1:4:1 (Sp)	Kalibrierter erwarteter Aufwand (Ph)
				14,6	30,0	44,6		29,83	29,83
72	Animal Certificate	Smart Contract Extension	Add vaccinations	2	4	5		3,83	3,83
73	Animal Certificate	Smart Contract Extension	Read vaccinations	1	3	4		2,83	2,83
74	Animal Certificate	Smart Contract Extension	Duplicate vaccinations	1	3	4		2,83	2,83
75	Animal Certificate	Smart Contract Extension	Disease breeding restriction	2	4	6		4,00	4,00
76	Animal Certificate	Smart Contract Extension	Set pet (fur) color	1	2	3		2,00	2,00
78	Animal Certificate	Smart Contract Extension	Read pet color	0,5	1	1,5		1,00	1,00
77	Animal Certificate	UI Extension	Website access	1	1	2		1,17	1,17
79	Animal Certificate	UI Extension	Edit certificate	1	3	4		2,83	2,83
80	Animal Certificate	UI Extension	Website help resources	1	3	5		3,00	3,00
81	Animal Certificate	NFC Integration	Scan pet's NFC	2	3	5		3,17	3,17
82	Animal Certificate	NFC Integration	Save data to NFC	2	3	5		3,17	3,17

▼ Backlog (11 issues)

ANIMALCERT-72	Add vaccinations	SMART CONTRACT EXTENSION
ANIMALCERT-73	Read vaccinations	SMART CONTRACT EXTENSION
ANIMALCERT-74	Duplicate vaccinations	SMART CONTRACT EXTENSION
ANIMALCERT-75	Disease breeding restriction	SMART CONTRACT EXTENSION
ANIMALCERT-76	Set pet (fur) color	SMART CONTRACT EXTENSION
ANIMALCERT-78	Read pet color	SMART CONTRACT EXTENSION
ANIMALCERT-77	Website access	UI EXTENSION
ANIMALCERT-81	Edit certificate	UI EXTENSION
ANIMALCERT-83	Website help resources	UI EXTENSION
ANIMALCERT-82	Scan pet's NFC	NFC INTEGRATION
ANIMALCERT-85	Save data to NFC	NFC INTEGRATION

+ Create issue

- 18.03.2023

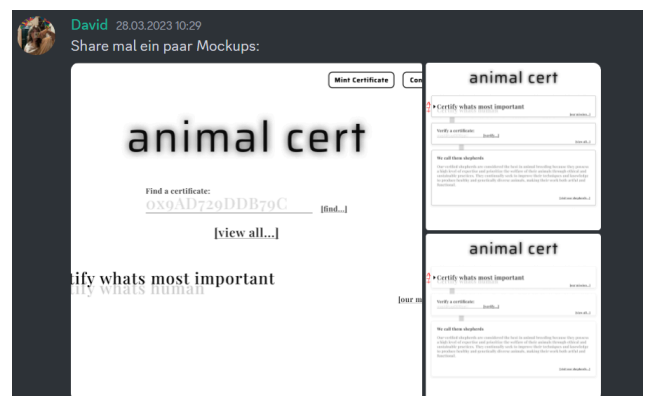
A new team member, Philip Slomka, joined our team and started getting involved in our project. He joined our Discord server, got introduced to the team and the project and started learning our technology stack to be able to start working on the project himself.

- 26.03.2023

Extended the smart contract to support fur-color and deployed and connected it to our frontend.

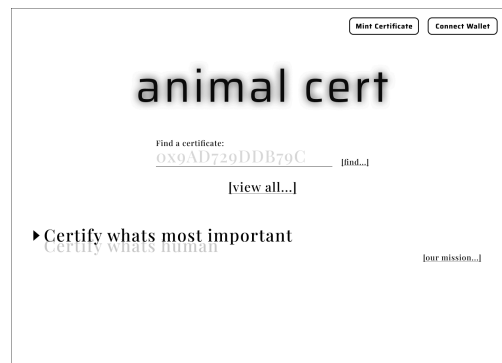
- 28.03.2023

Our team successfully created the first sprint for the Animal Certificate project. During this session, we discussed and agreed upon the tasks and objectives to be accomplished in the upcoming sprint. We held a vote on which design proposal (wireframe) we should commit to moving forward. There were 3 to vote on: ->



- 11.04.2023

In today's meeting we unilaterally agreed on the first design. Remarks that it looks like an obituary have been made.



- 14.04.2023

Started work on the mockups for the other pages. More to follow.

<https://www.figma.com/file/RS63jX7FBgw2tNXrHeAP8a/Animal-Certificate?node-id=0%3A1&t=cM0BQ518iVFJ4O6-1>

- 16.04.2023

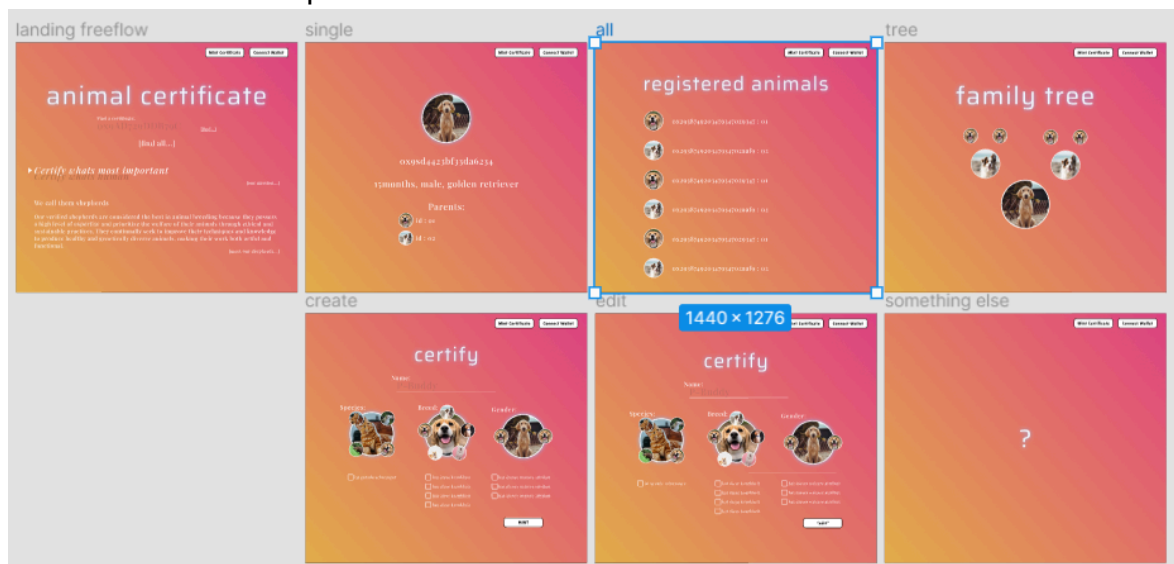
Added missing attributes for the minting page that were added with the contract and repaired the minting functionality.

- 20.04.2023

We discovered that the NFC reader we had ordered, the Neuftech USB 13,56 MHz kontaktlos Kartenleser 14443A-Protokoll IC Kartenlesegerät mit 5pcs Schlüsselanhänger, was designed only for reading tag IDs and not for writing data on NFC tags. This realization came after testing the hardware and evaluating its capabilities. As a result, the Python script we had developed for interacting with NFC chips would not be able to perform the desired functionality of writing data to NFC tags.

- 24.04.2023

Finished all the mockups.



<https://www.figma.com/file/RS63jX7FBgw2tNXrHeAP8a/Animal-Certificate?node-id=0%3A1&t=cM0BQ518iVFJ4O6-1>

- 01.05.2023

Updated the smart contract compiler version to the newest Solidity version (at that time ^0.8.9) and refactored some of the code because of deprecation. This was done, because the functionality of the diseases that we added last term was limited and to achieve what we have wanted, we had to use a newer compiler.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";

// A contract is a collection of functions and data (its state).
// Once deployed, a contract resides at a specific address on the Ethereum blockchain.
// Learn more: https://solidity.readthedocs.io/en/v0.5.10/structure-of-a-contract.html
contract AnimalCertificate is ERC721
{
    // ...
}
```

- 12.05.2023

Finalized the minting page and added a drop down selection for all attributes that do not have a text input to avoid using numbers representing values in the contract.

Issue Token

Name

Sex

Species

Dog

Gender

Female

Birthdate

Diseases

Arthritis

Chronic kidney disease

Hepatitis

Color

Brown

Also added new pictures for different animals. Not all animals have a picture yet though



- **17.05.2023**

During our meeting with the lecturer, we discussed our fourth sprint progress and addressed the limitations of the NFC device we had ordered. To overcome this obstacle, the lecturer offered to lend us a Raspberry Pi embedded with RFID and NFC chips. This alternative device will allow us to continue working on the project, with the necessary NFC functionality.

- **27.05.2023**

Implemented the functionality to get all pets by owner, which was not available anymore by default because of the solidity version we were using now. So the function was written manually to return all animals related to a user. Further one, some bug fixes were done on the minting page to resolve some issues with inputs.

- **12.06.2023**

Started research about the Bloxberg network we want to migrate our contract to and contacted responsible people over the official Bloxberg contact forms. A reason for migration is that GoerliETH will be deprecated at the end of 2023, since it's still a network based on proof-of-work. The new Sepolia network will take over its place, but since we wanted to have it on the Bloxberg network anyways, we decided this to be a good time to do this.

- **14.06.2023**

Got an answer to the email sent to Bloxberg and started contact with the technical team there. We also successfully set up and tested the Raspberry Pi with RFID and NFC chips, ensuring its proper functionality for our project. On other news we started with a rudimentary ancestral tree.

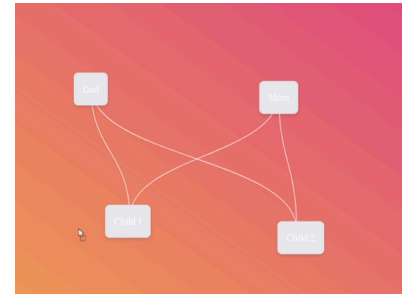
- **16.06.2023**

Tried deploying the contract to the Bloxberg network over Remix, but failed due to a gas estimation issue. Contacted the technical team and waited for further instructions

- **20.06.2023**

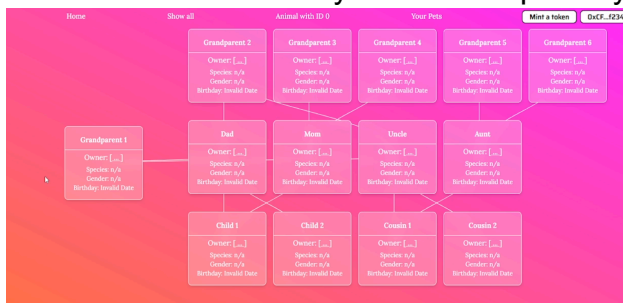
Dear diary, I'm really starting to dislike react. Today I spent 6 hours just trying to pass variables back and forth between splines cards and I tried to make the dots ui happen. THE DOTS ARE SCRAPPED! I don't care anymore.

Users can freely move blocks that are connected by lines ("splines") or more accurately svg paths that bezier curve from one node to its connected nodes. The code I came up with is fairly interesting and worth a look ;)



- **23.06.2023**

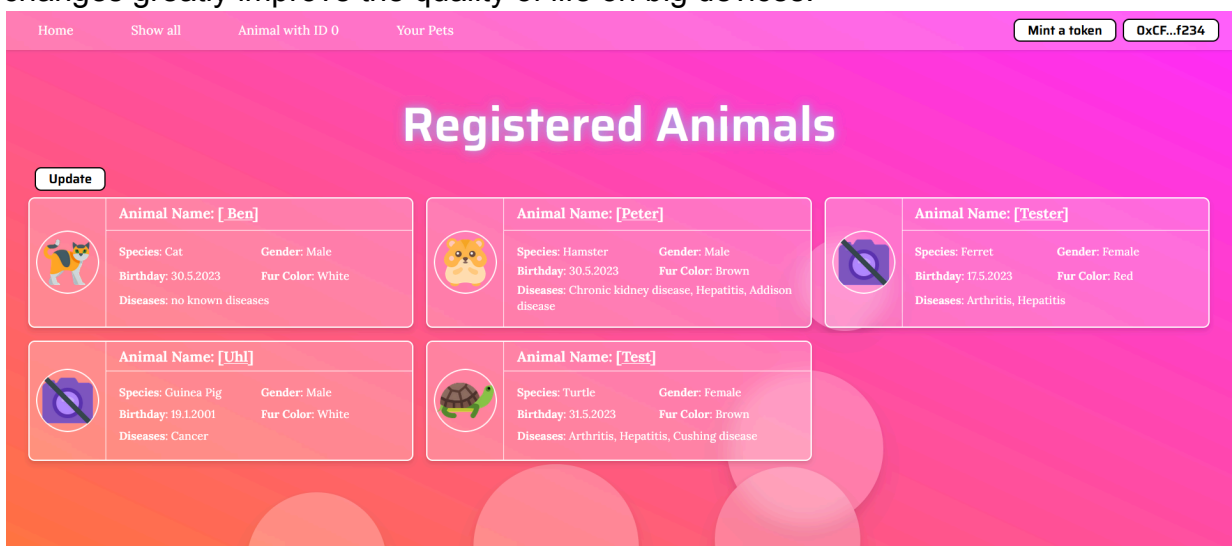
I DID IT! We have a fully functional spline system for the ancestral tree.



Also did some color correction. (the colors pop now!) It's starting to feel like a worthy d-app.

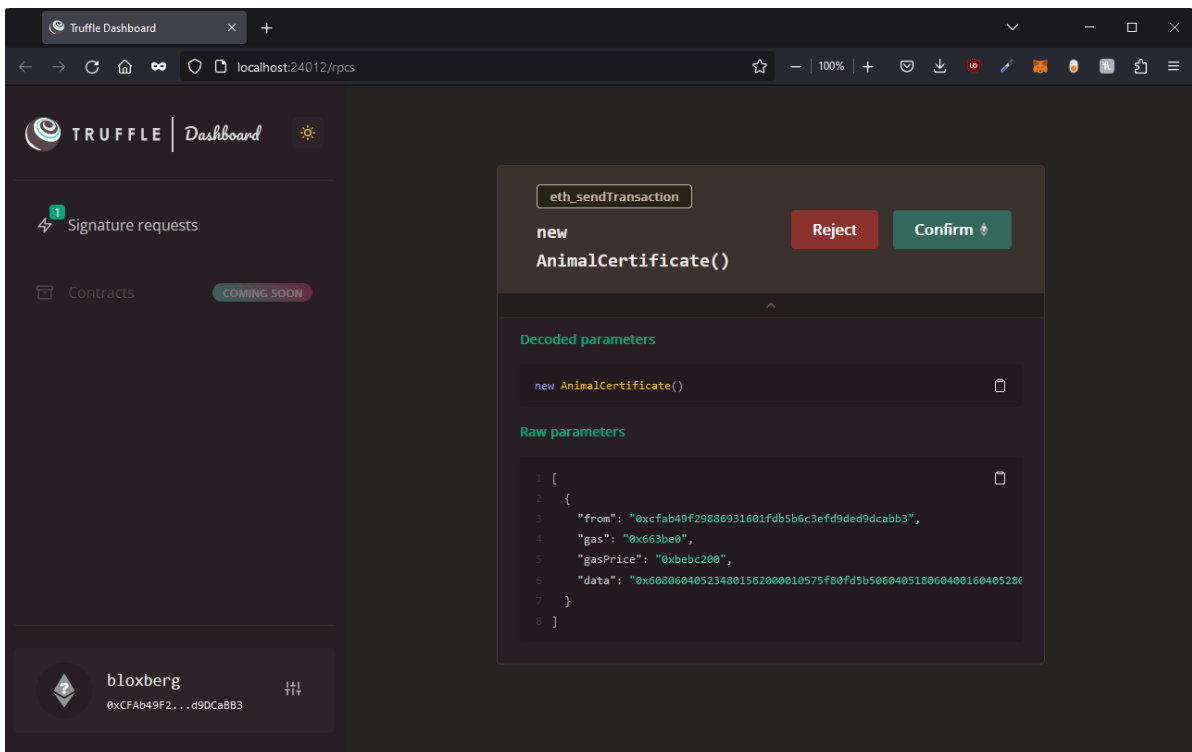
- **24.06.2023**

Started using a new tool called Truffle as suggested by the technical team to deploy the contract. Since this is a library, more time for installing and using it is necessary. The data displayed in the ancestral tree gets loaded from the blockchain, as well as a greatly overhauled design philosophy (grid-like instead of list-like). The performed ui changes greatly improve the quality of life on big devices.



- **25.06.2023**

Tried to deploy the contract with Truffle, but still had issues, about the gas estimations. So further support will be needed from the Bloxberg team.



```
Starting migrations...
=====
> Network name:    'dashboard'
> Network id:     8995
> Block gas limit: 10000000 (0x989680)

1_deploy_ac.js
=====

  Deploying 'AnimalCertificate'
  -----
  > transaction hash: 0xc665330b4a2b4cf0d2d075c5b8a87a1b84a1626ffb98b7af649fc9f52fa06b00sage.
  x Blocks: 0        Seconds: 8
  *** Deployment Failed ***

"AnimalCertificate" -- The contract code couldn't be stored, please check your gas limit..

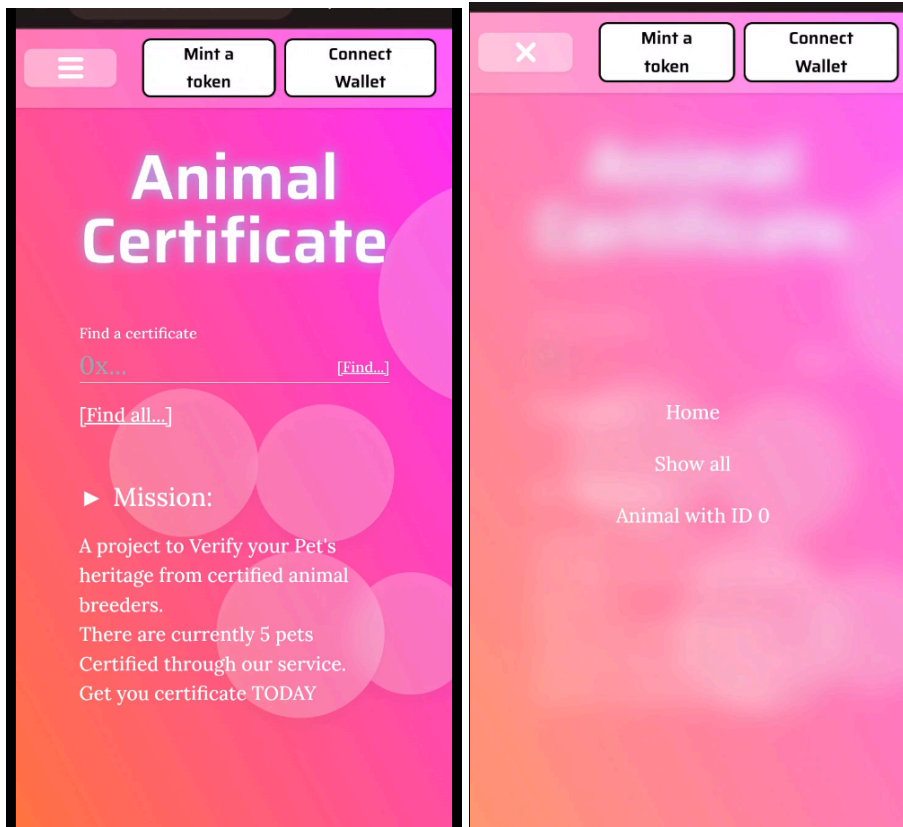
Exiting: Review successful transactions manually by checking the transaction hashes above on Etherscan.

Error: *** Deployment Failed ***

"AnimalCertificate" -- The contract code couldn't be stored, please check your gas limit..

    at C:\Users\Chris\AppData\Roaming\npm\node_modules\truffle\build\webpack:\packages\deployer\src\deployment.js:330:1
    at processTicksAndRejections (node:internal/process/task_queues:95:5)
Truffle v5.10.1 (core: 5.10.1)
Node v18.12.1
```

Overhaul for the Navbar, which now also includes a mobile friendly hamburger menu.



- **26.06.2023**

Extended the function to search for pets from one owner with a basic check and error message.

Term 3

- **11.10.2023**

Created simple python scripts: one for writing some dummy data on NFC Tag and another for reading the data from the NFC Tag.

- **16.10.2023**

Incorporated the React Redux state management system to replace the previous inefficient and error-prone property cascading approach, known as "ducktape." This change aims to improve the overall performance, scalability, and maintainability of the platform.

- **17.10.2023**

Integrated Wagmi for Walletconnect (the ability to connect any wallet to the interface) to replace web3-rpc calls with easy to handle Coposition calls.

- **18.10.2023**

Conducted a thorough refactoring of the codebase, removing deprecated code and making substantial structural improvements to enhance code clarity, maintainability, and overall quality in preparation for code review.

- **30.10.2023**

Fixed a bug in the webkit build pipeline by mismatched dependencies.

- **19.11.2023**

After spending a lot of time trying to migrate the contract from Goerli to Bloxberg and failing, we decided to migrate to the Sepolia Network instead.

We had a longer email conversation with a technician from Bloxberg, but because of some unknown error, we were not able to deploy our contract on the Bloxberg chain, which the FH Technikum is part of.

Sepolia had no issues at all and worked as fine as Goerli before it. The decision to migrate was made, because Goerli was soon to become deprecated, so a switch was necessary.

- **20.11.2023**

Created a new Python Script for Raspberrypi that includes a blockchain integration using abi and contract address.

- **21.11.2023**

Philip left the team due to him not being able to continue studying at FH Technikum Wien. From now on, only we, Maja, Chris, and David are working on the project.

- **02.12.2023**

Updating the script so when we enter the animal ID, it fetches the animal data and retrieves the owners information using blockchain integration

- **14.12.2023**

Updating the Python Script when we enter the animal ID, it fetches both the animal ID from the blockchain. Then it prompts to place an NFC card, and upon doing so, it writes the data onto the card

```
(testenv) raspberrypi@raspberrypi:~/MFRC522-python $ python WriteCardNEW.py
Enter Animal ID for fetching data: 1
Place your NFC tag to write data.
Write Successful
(testenv) raspberrypi@raspberrypi:~/MFRC522-python $ python CardRead.py
Scanning for a card...
To cancel, press Ctrl+C
NFC Tag ID: 358561192025
NFC Tag Data: {"Animal Data": [1, 0, 0, 0, false, 1, "Little F
```

- **19.12.2023**

Enhanced the main NFT View with filtering capabilities, allowing users to narrow down their NFT search based on various criteria.

- **24.12.2023**

Introduced full-text search functionality to the NFT platform, utilizing fuse.js, a powerful search library. This feature empowers users to search for NFTs based on their descriptions, traits, and other relevant information.

- **12.1.2024**

Implement tooltips across the platform to provide concise and informative pop-up descriptions.

- **16.1.2024**

Added a dedicated interface to confirm pregnancy for NFTs.

- **17.1.2024**

Introduced a confirmation interface for births of NFTs, allowing users to officially mark the birth of new Pets in their collection and maintain a comprehensive record of their NFT's lineage and breeding history.

- **30.01.2024**

Held a presentation about Animal Certificate to Prof. Mense and prepared the marketing jingle video as well as all the final documentation for the project.

7. Future Improvements

Since the project has potential to be extended in future by another team, here are some ideas, that might improve and extend the current state of Animal Certificate:

- Removing diseases
- Adding limitations to breeding based on diseases
- More sophisticated breeding system
 - breeding between animals of different owners
 - confirmation system from both parties
- More advanced birth interface
 - being able to set more attributes
 - set attributes per newborn
- Mobile Application
 - Including NFC scanner functionality
- Deployment of smart contract on mainnet or scientific chains like Bloxberg
- Some ideas for the UI:
 - Improve the landing page
 - Unifying the design philosophy
- Add unique pictures to your animals
- Overhauling the naming system
- Overhauling Breeds and Species (basically make there be more options)
- Adding information about vaccines or other medications

8. Special Thanks

We, as the whole Animal-Certificate team would like to especially thank Prof. Rohatsch for his continuous support and engagement throughout the last three terms while working on this project. It has been a pleasure working with him and we wish him all the best! :D

Sincerely,
Maja Cvetkovska
Christopher Tolan
David Zeilinger