Appendix A

Planning

The project was planned in multiple stages ahead of time. Planning only took place for a certain number of topics ahead and for major checkpoints. Later the major plans @were refined further into smaller segments and the plan table, which can be seen below, was also extended with small chunks of log information as to what topics were covered during major meetings. Overall the project and thesis was on a good track until November 2022, when multiple work and school related events caused it to go off track fairly drastically. It wasn't particularly too hard to catch up to the gap, however the submission is still a bit of a close call.

ID	Date	Activity description	Original proposed date	Date difference	Completed on time	Completed
61	2024-04-15	Meeting (Cancelled), Writing	2024-04-15		No	No
60	2024-04-10	Family Trees, Test Mating adjustments, Print CSS, .htaccess				
59	2024-04-09	Adjustments after meeting				
58	2024-04-08	Meeting	2024-04-08		Yes	Yes
57	2024-03-26	Slovak language				
56	2024-03-24	Authorization, Birthdays				
55	2024-03-23	Design adjustments, Authorization				
54	2024-03-22	Litters, Upadted Cats table				
53	2024-03-21	Photo upload				
52	2024-03-20	Backup help, Test Mating				

51	2024-03-19	Family Tree Additional details				
50	2024-03-18	Meeting	2024-03-18		Yes	Yes
49	2024-03-10	User authorization, Backgrounds, Test mating				
48	2024-03-08	Docker				
47	2024-03-01	Writing				
46	2024-02-28	Dockerfile				
45	2024-02-22	Writing				
44	2024-02-16	Writing				
43	2024-02-08	Custom landing page, Formatting, File upload				
42	2024-02-06	DB Backups, EMS, Breed				
41	2024-02-03	Inbreeding, Fixes, Register edit fields				
40	2024-02-01	Inbreeding, Sire, Dam				

39	2024-01-30	Family Tree				
38	2024-01-29	Family Tree				
37	2024-01-26	CSV file upload				
36	2024-01-23	Cat setup				
35	2024-01-22	Rework begins				
34	2023-05-17	Finish-up and submission	2023-05-17	0	Yes	Yes
33	2023-05-12	Reserved for last-minute changes	2023-05-10	+2	Yes	Yes
32	2023-05-03	Fine-tuning	2023-05-03	0	Yes	Yes
31	Cancelled	Fine-tuning	2023-04-26	-	-	No
30	2023-04-30	Optional: Additional algorithmical and UX extensions, majority of testing done, thesis showcase	2023-04-19	+11	No	Yes

29	2023-04-25	Major functionality finished and functional on a webhosting, Majority of technical documentation	2023-04-05	+20	No	Yes
28	2023-04-20	Majority of implementation chapter written, implementation showcase, beginning of manual tests on webhostings	2023-03-29	+22	No	Yes

27	2023-03-20	Finish writing conceptualization, showcase state of implementation, API extensions	2023-03-15	+5	No	Yes
26	2023-03-12	Build environment and Tooling including tests	2023-03-01	+11	No	Yes

25	2023-03-10	Finishing implementation development	2023-02-20	+18	No	Yes
24	2023-02-15	Meeting	2023-02-15	0	Yes	Yes
23	2023-01-15	Further planning for BP2	2023-01-15	0	Yes	Yes
22	2023-01-02	Rewritten BP1 into LaTeX	2023-12-30	+3	No	Yes
21	2023-01-01	Finished writing BP1	2022-12-25	+6	No	Yes
20	2022-12-30	Wrote the analysis down properly	2022-12-15	+15	No	Yes
19	2022-12-29	Dataset analysis and conversion, Analysis of available algorithms for dataset processing	2022-12-05	+24	No	Yes
18	2022-12-20	Studying materials	2022-11-30	+20	No	Yes

17	2022-12-15	Complex structure of the bachelor's thesis	2022-11-20	+25	No	Yes
16	2022-12-09	Material study, Complex folder structure	2022-11-15	+24	No	Yes
15	2022-12-08	Progress meeting: Further specification breeders' and breeding stations' needs, Structure clarification, Database structure modification needs	2022-11-05	+33	No	Yes
14	2022-11-12	Material studies	2022-10-25	+18	No	Yes
13	2022-10-17	Material studies	2022-10-20	-3	Yes	Yes

12	2022-10-16	E-mail conversation related to Inbreeding conceptualization	2022-10-15	+1	No	Yes
11	2022-10-09	Material studies	2022-10-05	+4	No	Yes
10	2022-10-07	Progress meeting: Basic structure of the bachelor's thesis, Repository structure, Program components, Development environment - Web-hosting preparation, Acquired dataset	2022-09-20	+17	No	Yes
09	2022-09-20	Study of materials	2022-09-10	+10	No	Yes
08	2022-09-05	Study of materials	2022-09-05	0	Yes	Yes

07	2022-08-21	Initial repository commits, Looking for a dataset	2022-08-30	-9	Yes	Yes
06	2022-08-19	List of existing databases and breeds	2022-08-25	-6	Yes	Yes
05	2022-08-18	Write-up of features found in database providers' user interfaces	2022-08-22	-4	Yes	Yes

04	2022-08-16	Meeting: Considering possible topics to be covered in the thesis, Rough data content idea, Keywords, Additional technology suggestions, Analysis continues, Changes in technologies	2022-08-15	+1	No	Yes
03	2022-08-13	Studying the assignment once more, Preparing outlines for the thesis, Initial analysis and technology selection	2022-07-30	+13	No	Yes

02	2022-07-17	Material studies	2022-07-15	+2	No	Yes
01	2022-06	Initial meeting: Topic explanation, Began to acquire materials for study	2022-06	0	Yes	Yes

Appendix B

User Documentation

B.1 Setting up webhosting access

First create a database for the project, copy *.env.example* to *.env*, disable debug and set the database login details in the given file. In case you're going to utilize the Dockerized version (either by using the provided image or by building one yourself from the Dockerfile) don't change the provided database details. Make sure to upload the file onto your webhosting alongside the whole project.

```
APP_DEBUG=false
...

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=homestead
DB_USERNAME=homestead
DB_PASSWORD=secret
```

B.2 Webhosting deployment

The folder hierarchy has to be copied with the exception of *.git* and *node_modules* onto the *public* part of the webhosting by utilizing SFTP or related technologies. Afterwards a database has to be migrated. This can be done by setting up the *.env* file as mentioned above and then using the following commands on your machine in the same folder:

```
php artisan migrate
```

This way all of the required database tables will be created.

B.3 Accessing common functionality

The main functionality of the website is available through the search on the navigation pane. This way any user is able to search the database for cats and any matches he may find are displayable as a profile or a tree. Next to the main functionality users can find test mating. With test mating any user can query the database for male and female cats, optionally of the same breed and compare their inbreeding which is calculated on the frontend. Additionally on the unpublished routes, there is a capability to view charts of the cat's family tree and birthdays. User can also register an account, although this won't allow them to register a new cat nor view backups, because each user account has to be flagged as an administrator manually in the database itself.

B.4 Creating a new user account

A new user account can be created using the *register* route accessible through the main navigation. User is requested to input their e-mail which will be used as the main means of logging in. Additionally the user is requested to input their name and password. The password is hashed on the side of the database.

B.5 Accessing Backups and Cat Registration functionality

The user can access backups and cat registration if their account is flagged as administrator on the side of the database. This requires someone with database access to manually go into the *users* table and flag the *is_admin* boolean as true for the given user.

B.5.1 Setting up an administrator account on a Dockerized build

If you're the administrator, then you will likely wish to create an admin account so you can access cat registration, edit and backup features. In order to do this simply register an account, open the MariaDB console:

To open the console first check which Docker container is yours, then substitute *xxxx* with your container id:

```
docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
ad4951b8d9cd bp:2.0 "sh entrypoint.sh --" 51 minutes ago Up 51
   ↪ minutes 127.0.0.1:8000->8000/tcp
```

In the example above Container ID can be clearly seen. This identification will be used to access the containers shell and from there the MariaDB console without having to expose any additional ports:

```
docker container exec -it xxxx /bin/sh
mariadb
```

Then after getting to the MariaDB console proceed to use the following list of queries. Substitute yyyy for your user's id from the query on the previous line which shows all users:

```
use homestead;
show tables;
select * from users;
update users set is_admin = TRUE where id = yyyy;
```

Now the user with id *yyyy* became an administrator. You can exit the shell and docker container:

```
exit
exit
```

Appendix C

Technical Documentation

C.1 Folder hierarchy

Based on the MVC architecture we can delineate the major folder structure as follows:

- *app* contains the Models

- *app/Http/Controllers* contains the Controllers, which within their methods link to Views

- *resources/views* contain the Views themselves, which get to use Model methods where necessary using the syntax

Additionally:

- *routes* contain the GET, POST and DELETE routes which link to Controller methods

- *test* contains Functional and Unit tests

- *database/migrations* contains the migrations necessary for Models to function

C.2 Dockerfile contents, build and launch order

```
FROM ubuntu:jammy
WORKDIR /app
COPY . /app
ENV DEBIAN_FRONTEND=noninteractive
RUN apt update
RUN apt-get --purge remove php-common
RUN apt install php-common php-mysql php-cli -y
RUN apt install mariadb-server -y
RUN cp -r /app/mariadb.service /etc/systemd/system
RUN service mariadb start && mariadb -u root -e "CREATE DATABASE
    ↪ homestead; GRANT ALL ON *.* TO 'homestead'@'localhost'
    ↪ IDENTIFIED BY 'secret' WITH GRANT OPTION; FLUSH PRIVILEGES;"
RUN mkdir -p /usr/lib/php/20210902
\chapter{RUN cp -r pdo_mysql.so /usr/lib/php/20210902}
RUN cp -r php.ini /etc/php/8.1/cli
RUN php artisan key:generate
RUN service mariadb start && php artisan migrate
ENTRYPOINT ["sh", "entrypoint.sh"]
```

Firstly a Ubuntu image gets used, then all app data get copied onto the */app* folder. Next an adjustment gets made to automate commands without an interactive environment. *apt* follows up with an installation of all relevant packages:

- php-common

- php-mysql

- php-cli

- mariadb-server

Then mariadb gets setup to run on the same image as the main application. Next up an adjustment to the php installation so the maximum upload size is accordingly adjusted. Lastly Laravel key gets generated and MariaDB launched for migrations. All of these steps get done ahead of the runtime of a Docker image in order to build it.

After the Docker image gets built the entrypoint gets called:

```
#!/bin/sh
#ping -t localhost
service mariadb start && php artisan serve --host 0.0.0.0 --port
    ↪ 8000
```

This includes running the mariadb service and web server on the port 8000, which is exposed in the relevant Docker run command:

```
docker build -t bp:2.0 .
docker run -dp 127.0.0.1:8000:8000 bp:2.0 --name bp
```

C.2.1 Ports

Only port opened to the host is 8000. In case you need to expose the mariadb as well (which is not necessary as you can access it using *docker exec -it* followed up with the *mariadb* command) make sure to expose the port 3306 in the *docker run*.

C.3 APIs and Routes

The following routes have been programmed:

GET:

- /

- home

- profile

- profile-search

- cats/cat

- cats/cat/chart (disabled in navigation due to performance issues)

- cats/cat/tree

- test/cat/cat2

- cats/cat/tree/generations

- cats/cat/death

- birthdays (disabled in navigation on webhosting due to query result length issues)

- birthdays/month/day

- register-cat

- cats/cat/edit

- backups/help

- backups/fileName/dl

- register

- login

- forgot-password

- logout

POST:

- register-cat

- family-actions/cat/set-sire

- family-actions/cat/set-dam

- family-actions/cat/set-parent

- cats/cat

- backups/issue

- backups/export

- backups/import

- backups/export_breeds

- backups/import_breeds

- backups/export_ems

- backups/import_ems

- backups/upload

- backups/fileName/restore

- language/change

- register

- login

- forgot-password

- logout

DELETE:

- cats/cat

Appendix D

Additions

Evidence number: FIIT-100241-103151

Digital contents:

- .tar file for Docker

- BP.fig

- Dockerfile

- LICENSE

- Procfile

- SECURITY.md

- app

- artisan

- bootstrap

- breeds.csv

- cats(5).csv

- composer.json

- composer.lock

- config

- database

- ems_values.csv

- entrypoint.sh

- guide.md

- ls.txt

- mariadb.service

- node_modules

- package-lock.json

- package.json

- pdo_mysql.so

- php.ini

- php.ini.bak

- phpunit.xml

- postcss.config.js

- public

- readme.md

- resources

- routes

- server.php

- storage

- tailwind.config.js

- tests

- vendor

- vite.config.js

- webpack.mix.js

- yarn.lock

The digital contents have been saved at the thesis supervisor due to their size exceeding 1GB.