

Attachment A: Work plan

A.1 Work plan for phase DP1

Week of the semester	Activity plan
1.week	Familiarization with the topic
2.week	Study of the problem
3.week	Reading articles on the topic and collecting literature
4.week	Reading articles on the topic and collecting literature
5.week	Reading articles on the topic and collecting literature
6.week	Reading articles on the topic and collecting literature
7.week	Thesis writing
8.week	Thesis writing
9.week	Thesis writing
10.week	Thesis writing
11.week	Thesis writing
12.week	Final touches

A.2 Evaluation of the work plan for the DP1 phase

Evaluation of the first part of the thesis DP1 in the summer semester

In the summer semester, I worked on the first steps of my thesis project. The aim of this part was to thoroughly familiarize myself with the topic and to gain the necessary knowledge to solve the problem. This was followed by a schedule of activities that I gradually followed:

In the first week I devoted myself to familiarizing myself with the topic. I studied

the relevant information on the problem, its context, and the goals I had to achieve. This phase was important in gaining a broader overview and basic understanding of the issue.

In the second week, I devoted myself intensively to the study of the problem. I analyzed its nature in more detail, identified the main challenges, and explored the related areas in depth. This step allowed me to gain deeper knowledge and a basis for future procedures.

Weeks 3 to 6 were focused on reading articles on the topic and collecting literature. I thoroughly studied relevant studies, scientific articles, and publications in the relevant fields. The aim was to gain a more comprehensive overview of the problem and to identify the most up-to-date knowledge.

In the seventh week, I worked on writing the thesis. Based on my researched materials and knowledge, I began to formulate the content of the thesis. I gradually developed the structure and wrote parts of the document in order to follow a logical progression and to ensure coherence of the text.

Weeks 8 to 11 were devoted to writing the document and gradually expanding it. At this stage, I included all relevant information, analysis, and results of my studies in the thesis. I tried to achieve a higher level of detailed analysis and to present my findings and results clearly.

In the 12th week, I dedicated myself to the final editing of the document. I reread the entire text and did the necessary proofreading and editing. I ensured that the document was complete, clear and grammatically correct.

Throughout the work, I regularly consulted with my supervisor. Together, we discussed my progress, discussed my results, and evaluating the shortcomings and benefits of my work. His feedback provided guidance for my next steps and helped

me better navigate the problem.

In general, I can conclude that I have successfully met the objectives set out in the schedule. I familiarized myself with the topic, analyzed the problem, studied the relevant literature, and wrote a coherent document. These steps provided me with a solid foundation for the next stages of the thesis and allowed me to better understand and plan a solution to the problem I set out to solve.

A.3 Work plan for phase DP2

Week of the semester	Activity plan
1.week	Setup of frontend, backend and database
2.week	Creating login and register logic and pages
3.week	Creating admin page for role changes
4.week	Creating find you dataset page
5.week	Creating selected dataset page
6.week	Creating annotations page
7.week	Creating export
8.week	Creating testing datasets
9.week	Creating database seeders
10.week	Thesis writing
11.week	Thesis writing
12.week	Final touches

A.4 Evaluation of the work plan for the DP2 phase

Evaluation of the second part of the thesis, DP2, during the winter semester mainly focused on the development of a prototype application. During the course of the semester:

During the first week, attention was devoted to configuring the front-end, back-end, and database. This involved the creation of a Firebase project that incorporates Firebase authentication, Firestore for data storage, and Firebase functions. Also, the .env file was used for database credentials.

The second week was dedicated to formulating the logic and pages for the login and registration processes. Firebase authentication was used, and Firestore was used to manage user roles. Access to specific pages was restricted according to roles, ensuring, for example, that only administrators could access the admin page, and only logged-in users could access the “find a dataset“ page.

Continuing into the third week, the focus shifted to the creation of an admin page with the capability to modify user roles. This feature streamlined the role-changing process, eliminating the need for direct interactions with the Firestore database.

In the fourth week, the “find your dataset“ page was established. Adding a function to add datasets and store them within Firestore. Subsequent weeks involved the creation of the selected dataset page, enabling users to add descriptions to datasets and their respective columns, while also providing a managed view of the dataset.

The sixth week saw the development of an annotations page, allowing users to view annotations. Additionally, functionality was added to annotate columns within the

selected dataset page.

In the seventh week, the implementation of the data set export functionality in both CSV and JSON formats, including annotated columns, was completed.

Week eight was dedicated to the creation of test datasets, while week nine focused on the development of functional seeders for Firestore and Neo4j, incorporating previously generated datasets and annotations.

The last three weeks (including Weeks 10 to 12) were devoted to updating this Overleaf document, which forms an integral part of my diploma thesis.

Throughout the work, I regularly consulted with my supervisor. Together we discussed my progress, discussed my results, and evaluating the shortcomings and benefits of my work. His feedback provided guidance for my next steps and helped me better navigate the problem.

In general, I can conclude that I have successfully met the objectives set out in the schedule. I created a working prototype of my page. This provides me with a solid foundation for the next and final stage of my thesis.

A.5 Work plan for phase DP3

Week of the semester	Activity plan
1.week	Adding PostgreSQL using repository design pattern
2.week	Adding backend logic for PostgreSQL
3.week	Adding vis-network for PostgreSQL visualization
4.week	Deployment of the application
5.week	Integrating Neo4j aura and GCP cloud SQL
6.week	Creating microservices
7.week	Creating CICD pipelines
8.week	User testing
9.week	Application testing
10.week	Fixing bugs and issues found during testing
11.week	Thesis writing
12.week	Final touches

A.6 Evaluation of the work plan for the DP3 phase

During the first week, I focused on restructuring the application to accommodate multiple databases using the repository design pattern, as previously outlined in the thesis.

In the second week, I added logic to support PostgreSQL database integration, ensuring compatibility with the back-end and APIs.

The third week was dedicated to integrating a new visualization tool for PostgreSQL, since the existing tool, Neovis.js, was only tailored for Neo4j databases.

Week four was primarily spent deploying the application to the Google Cloud

Platform (GCP) platform.

During the fifth week, I incorporated databases from the cloud, including Neo4j Aura and Cloud SQL from GCP.

In the sixth week, I transitioned from a monolithic application to microservices architecture.

The seventh week involved implementing Continuous Integration and Continuous Deployment (CI/CD) pipelines for deploying both the front-end and back-end of the application.

Week eight was dedicated to finding bugs and testing the application to identify areas for improvement.

During the ninth week, I conducted extensive application testing, which included determining the optimal region for deployment, comparing the performance of monolithic and microservice architectures, and evaluating the differences in cost and computing power between Neo4j and PostgreSQL.

In week ten, I addressed and resolved bugs and issues uncovered during testing.

Week 11 was allocated to write the thesis document documenting the project's development process, implementation details, and outcomes.

Finally, in week twelve, the last week was dedicated to finalizing the thesis document, ensuring completeness and coherence.

Throughout the work, I regularly consulted with my supervisor. Together, we discussed my progress, discussed my results, and evaluating the shortcomings and benefits of my work. His feedback provided guidance for my next steps and helped me better navigate the problem.

In conclusion, I have successfully achieved the objectives outlined in the schedule. I completed the entire application by integrating PostgreSQL, introducing a new visualization tool, deploying the application, and performing thorough testing. With these accomplishments, I can confidently state that my diploma thesis assignment is now complete.

Attachment B: User Guide

B.1 Program function

I developed a web-based data science environment designed to seamlessly handle the integration, annotation, and distribution of datasets to meet the specific needs of data scientists. The primary objective is to provide a comprehensive infrastructure for efficient data manipulation and evaluation of AI models. This application supports dynamic addition, annotation, and transmission of data sets on demand.

To facilitate user-friendly data interaction, I devised a method to annotate datasets and send them. The chosen annotation format ensures clarity and accessibility for human users through the interface and machine interpretation through the API.

The application features a robust database model to store annotated datasets, allowing users to explore the data representation and understand the relationships between different elements. This dual functionality, both human-centric through the user interface and machine-readable through the API, ensures versatility in data manipulation.

I have set up an admin user for testing purposes with the following credentials:

1. Username: admin@gmail.com
2. Password: admin@gmail.com

To facilitate testing, I also provided a Postman export that contains all available endpoints. You can find the export file in the folder of my diploma thesis, named

“diploma-thesis.postman_collection.json.” This will assist you in efficiently testing and exploring the various functionalities outlined in the project.

Some endpoints require the user to log in, mainly those that change the data. To enable the functionality of the Postman collection, in other words, to log in follow these steps:

1. Initiate a call to either `firebase/login`
2. Extract the “idToken” value from the response.
3. Assign this “idToken” value to a global variable named “firebaseToken.”

Locate “Environment quick look” in the top right corner or refer to this tutorial. for detailed instructions. By completing these steps, you will set the necessary authentication token for Firebase API requests within the Postman environment.

The routes inside API folder use global variable ”api_path” which value depends on what do you want to test you have three option to choose from:

1. 'http://localhost:3000' for local testing
2. 'http://127.0.0.1:5001/diploma-thesis-project-c46d9/europe-west3/API' for local testing using the firebase emulator.
3. 'https://europe-west3-diploma-thesis-project-c46d9.cloudfunctions.net/API' for testing the deployed application

B.2 Using the app

The app is available online on this link. If you want to use it locally you have to install Neo4j desktop create project: diploma-thesis-project with database: diploma-thesis-database with password: 123456789 and version 5.12.0. Then you need to

click start on this database. You also need to create PostgreSQL version 16.2 with port 5432 with user: postgres and password: postgres and database name: diploma-thesis-database. Both the Neo4j and the PostgreSQL databases need to run for you to be able to start the app. The next step is to go into the project folder which I will upload together with my thesis. Go to the back-end folder/functions/API and run commands: “npm i” to install all required dependencies and then run command “npm run dev” to run the back-end of my application. Then go to the front-end and run command “npm i” to again install the required dependencies and run “npm run dev” to run the front-end of my application. You can now visit “http://localhost:5173/” where my application resides. This only starts up the development of API back-end monolit, but it has all the functionalities of the app.

In each firebase emulator, he has to go into each folder inside 'back-end/functions' namely AddUserRole, Annotation, API, Column, Dataset, Seeder, SetUserRole, and run the following commands in each one "npm i", "npm run build" and after that he can use "firebase emulators:start --only functions" inside folder "backend" to start the emulator The functions AddUserRole and SetUserRole need to be deployed on firebase functions, as they work directly with the firebase ecosystem.

Inside postman request to /annotation/postImport sometimes it fails with internal server error the problem is with postman, you need to delete the uploaded files and upload them again and it fixes the problem. The files are found in tab Body of the request inside postman. If the request does not go through, the solution is often as simple as clicking "send" again, which typically resolves the issue.

Attachment C: Description of the digital part

Registration number of the work in the information system: FIIT-182905-103097

Content of the digital part of the thesis (ZIP archive):

Folder/File	Description
\back-end	Back-end of my application
\back-end\functions\API	NestJS monolithic application
\back-end\functions\AddUserRole	Cloud function to add user role
\back-end\functions\Annotation	Microservice for annotations logic
\back-end\functions\Column	Microservice for column logic
\back-end\functions\Dataset	Microservice for dataset logic
\back-end\functions\Seeder	Microservice for seeder logic
\back-end\functions\SetUserRole	Cloud function to set user role
\front-end	Vue.js application and all config files
\front-end\package.json	contains necessary libraries to run the application
\front-end\src	application source files
\front-end\src\main.ts	main application file
\front-end\src\views	views of my application
\front-end\src\components	components of my application
\datasets	example datasets
\datasets\import	Files for postImport endpoint
diploma-thesis.postman_collection.json	postman JSON with all available endpoints
\assets	assets of my diploma thesis

Archive name: DP_PeterPlevko.zip