CS 491
Senior Design Project
2021 Fall

High-Level Design Report
PolliVidis

Ömer Ünlüsoy - 21702136
Elif Gamze Güliter - 21802870
İrem Tekin - 21803267
Ece Ünal - 21703149
Umut Ada Yürüten - 21802410

Supervisor: Ercüment Çiçek
Jury Members: Shervin Arashloo and Hamdi Dibeklioglu

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Table of Contents

1 Introduction

The process of identifying pollens is considered as a tedious job since it is currently done manually. An image containing pollen requires manual labor to process since the samples of the pollens are collected manually. Furthermore, identifying pollen from irrelevant noise or content as well as distinguishing pollen variants from one another must be done through trained eyes. Additionally, training new students to identify pollens requires further effort, especially distinguishing pollens with similar granular cell structures.

Along with the developments in machine learning, most systems are transforming into automated models. The advancements of image processing and image classification highly inspired an automated system that could improve the manual process of pollen classification. Although there have been attempts to create such a machine learning model in order to identify pollen in a given data, currently there is no widely available model to identify pollen types for palynology, the branch of biology which examines pollen. Hence, with the system we will develop, we aim to fill the role of a publicly available web application for pollen identification in Turkey which would help academics to easily classify the pollen they research and furthermore aid students trying to learn the details of palynology. PolliVidis would allow users to easily upload an image to the web application and get the results. Apart from being widely accessible, PolliVidis also aims to create a database for palynologists to share pollen data across the country.

1.1 Purpose of the System

Palynologists, scientists that study pollen, collect pollen samples from the air and after observing them under the microscope, label them. Furthermore, after labeling those pollen they take the pollen photo and store them in databases to use them in further studies. Until now, this process was done manually. After communicating with some palynologists [1] we learned that this job takes a lot of time and automatizing the process can benefit the scientist in their studies. Moreover, detecting pollen from the samples and labeling them needs trained people and training people takes a lot of effort too. Therefore, we propose a system that automatically detects pollen from samples, sorts them into their species, and optionally, stores them in a database.

Additionally, based on our database idea, we thought of a system where people can store their sample photos but also share them and examine other people's samples. Using this system, scientists can use other samples from different areas in their studies and also contribute to the database. We saw that no such system exists, and we aim to build this database for all palynologists in the world. Moreover, as showing the contents of the database as a map, we thought that non-academic people can also benefit from this idea: Since in Turkey there is not a system where allergic people can see pollen density on a map and take precautions according to this data, we realized that with our pollen map we can also fulfill this need.

In conclusion, we have three goals:

1. Designing and training a Convolutional Neural Network to classify pollen types.
2. Forming a database where scientists can share and store their labeled photos.
3. Showing the contents of this database as a pollen map so that people that have pollen allergies can take precautions against it.

1.2 Design goals

1.2.1 Design Trade-offs

The only design trade-off in the system is shown below.

1.2.1.1 Security vs Efficiency

Since using hashing algorithms to store passwords needs extra time for calculations, it may cause delay in the system's response time.

1.2.2 Design Criteria

The design criteria presented below will be followed to ensure the quality of the system.

1.2.2.1 Usability

The system should

- yield an analysis with the pollen samples taken under a light microscope under 2 seconds.
- allow pollen analysis without any registration.
- allow users to use the Pollen Map without any registration.
- allow users to upload pollen samples and analyze them in four clicks.
- have understandable guidelines

1.2.2.2 Reliability

The system should

- not allow non-academic people to add any pollen sample analysis data to the database.
- ensure results that have more than %90 percent accuracy for the pollen classification.
- not lose any pollen data unless the user deletes it.

1.2.2.3 Privacy and Security

The system should

- require passwords that contain uppercase and lowercase letters, and has at least 8 characters with a mixture of both numbers and letters.
- ensure that the user's data is safe by not storing their password directly but hashing it with the Google recommended hashing algorithm SHA-256[2].
- get permission from academic users to share the location and the date of the samples with other users.
- not share any personal information with the third parties.

- not require any personal information from unregistered users to analyze pollen and look through the pollen map.
- not allow any unauthorized access to the server.

1.2.2.4 Efficiency

The system should

- have a loading time that doesn't exceed 2 seconds which is the maximum loading time recommended by Google [3].
- conduct the analysis of the sample on the server rather than the user's computer to decrease memory usage.
- respond to user clicks with a maximum delay of 1 second for a seamless experience.

1.2.2.5 Accessibility

The system should

- be able to work on most of the web browsers such as Safari, Chrome, Firefox and Microsoft Edge.

1.2.2.6 Extensibility

- The Pollen Map should be extendible since this map might contain global data in the future.
- The ML model should be improvable since more datasets will be available in the future.

1.3 Definitions, Acronyms, and Abbreviations

Palynology: Branch of biology studying pollens.

Academic: User with a pollen or biology related background such as palynologists, biologists, and palynology or biology students.

Allergenic Pollen: Specific pollen types that humans can develop allergies to.

Sample / Pollen Sample: Pollen image shot by a light microscope containing a few pollens.

Noise: All other shapes in the sample rather than pollen itself such as spores.

Sample Analysis: Procedure of identifying pollen types, classification, from a sample using machine learning.

Analysis Report: Report containing pollen information generated by the pollen analysis.

Pollen Map: Google Maps supported map showing pollen information and distribution of Turkey.

Pollen Extraction: Process of extracting a single pollen image from the sample with few pollens using the Pollen Extraction Algorithm coded by us.

(Ankara) Dataset: Pollen dataset, collection of pollen images, created by us in Ankara University.

PolliVidis Database: MySQL based database to store all (allowed) uploaded pollen samples with their analyses in order to construct the Pollen Map.

CNN: Convolutional Neural Networks

Transfer Learning: Using pre-trained networks such as AlexNet or VGG-19 to boost the classification.

Data Augmentation: Manipulating dataset to avoid overfitting and increasing accuracy.

Google Maps API: used API for the Pollen Map of PolliVidis.

Django: Python package for website backend.

React: JavaScript library for website frontend.

MySQL: Relational database management system for SQL.

PyTorch: Python package for Machine Learning and Deep Learning.

1.4 Overview

PolliVidis is a pollen classification platform for academics, students and people with pollen allergies. It is a web application which makes it widely accessible to interested users. PolliVidis provides a pollen map built by the academic community's pollen analyses. Moreover, it allows data sharing between academics which will contribute to building a vast pollen database and help palynology research. Furthermore, PolliVidis has three main promises: classification and analysis of given pollen samples with a trained ML model, creation of a PolliVidis database which can help further palynology research and academic information exchange, and construction of a pollen map of Turkey. In addition, using the pollen map people with pollen allergies can track the frequency dates, and locations of these pollen and schedule their visits. In PolliVidis everyone, without the need of an account, is able to access the pollen map and learn our analysis, allergenic pollen information based on time and location. Moreover, everyone can use our model to analyze their own pollen sample via the website. However, only academics can update the pollen map with their samples and uploading data to the map requires an academic account. Login into these academic accounts is required to protect the accuracy and the reliability of the map. Furthermore, the pollen analysis consists of pollen classification and counting. After uploading a sample, the user can learn the pollen types and their ratio (number) in their samples. If academics agree to share their sample analysis, the pollen map will be updated with their analysis. Moreover, students are also able to use the PolliVidis for educational purposes, such as uploading pollen photos and learning their type without consulting their instructor.

2. Current Software Architecture

Although there are several research papers that focus on pollen classification and identification via CNN models and other machine learning algorithms, there isn't any proper ML model that works with light microscope images. The one popular dataset POLEN23E that consists of 805 images of 23 pollen types, is used in most current models and it is very limited, it has 35 images for a pollen type on average [4]. Furthermore, there are some successful models that work with other microscope types such as but they lack a functional end product that is publicly available [5].

With PolliVidis we aim to create a large pollen dataset and machine learning model for academic community’s use. PolliVidis provides a public web application that analyzes pollen samples, classifies and counts the pollen types in the sample images. In addition, with the help of the pollen map which will be updated by the academic user’s pollen analyses, PolliVidis offers a map to check for people with pollen allergies. Furthermore, PolliVidis provides educational help for the students that work on pollens.

3. Proposed Software Architecture

3.1 Overview

In this section, the software architecture of PolliVidis is explained by examining the subsystem decomposition, hardware/software mapping, data management, access control and security, global software control and boundary conditions of the system. The system is decomposed into subsystems in order to develop and maintain the software easier. The subsystems and their relationships with each other are explained using diagrams. The hardware/software mapping is shown with a deployment diagram. In persistent data management, the database objects and database system information are given. Authorization and security measures are explained in access control and security. The general control organization of the software is explained in global software control and at last the initialization, termination and failure boundary conditions of the system are explained.

3.2 Subsystem Decomposition

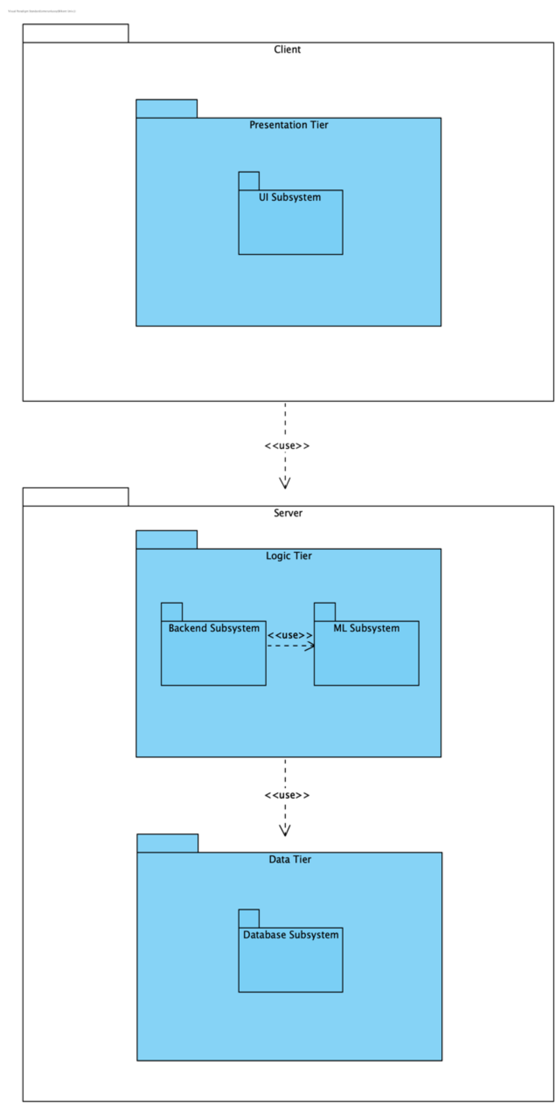

Figure 1: Subsystem Decomposition

System composition of PolliVidis mimics 3-Tier Client/Server Architecture as PolliVidis is a web application. This architecture allows PolliVidis to be used by multiple users simultaneously while protecting the performance of the server. The communication is supplied by the queries between the client and server sides.

On the client side, PolliVidis implements the Presentation Tier which holds the UI Subsystem. This subsystem is implemented with React and manages user interface components such as pages and page interactions. This subsystem sends queries to the server to handle user requests and shows the results of the queries to the user.

On the server side of the system, PolliVidis implements Logic and Data Tiers. In the Logic Tier, there are two subsystems, namely Backend Subsystem and ML Subsystem. Backend Subsystem is implemented with the Python Django Framework. It directs queries coming from the client side to the Data Tier and handles user requests. This subsystem uses the ML Subsystem to extract pollen images from the sample image and analyze them one by one. It returns the analyses to the user.

ML Subsystem is the core of PolliVidis and implements two main functionalities; pollen extraction from the sample image, and pollen classification with PyTorch library of Python. It is driven by the Backend Subsystem. The classification is done by a Convolutional Neural Network and pollen extraction uses Image Processing with the SCikit-Image package of Python.

In the Logic Tier, a single subsystem named Database Subsystem handles database interactions of PolliVidis. It implements the MySQL database and all queries within it and supplies Python Model classes for ease of use. This subsystem is driven by the Backend Subsystem by its pre-implemented query functions. The database of PolliVidis is mainly for samples uploaded by users, user information, and pollen information.

3.3 Hardware/Software Mapping

In our project, the client will use a web application in order to utilize the full extent of the services. Therefore, the client machine is a web application. The client machine has a TCP connection with the database server, which is built upon a relational database model. Furthermore, the client also has a connection with the Web Server via HTTPS

Connection. The back end of the web server is mainly built on REST API and since the location of the client will be used, Google Maps API will also be used.

The Server Controller utilizes the Session Manager for other services, mainly the Machine Learning Model artifact and the Database Manager. The Machine Learning Model artifact uses the training data set provided by the program, meanwhile Database Manager communicates with the Database Server.

The corresponding deployment diagram of PolliVidis can be seen in the Figure2.

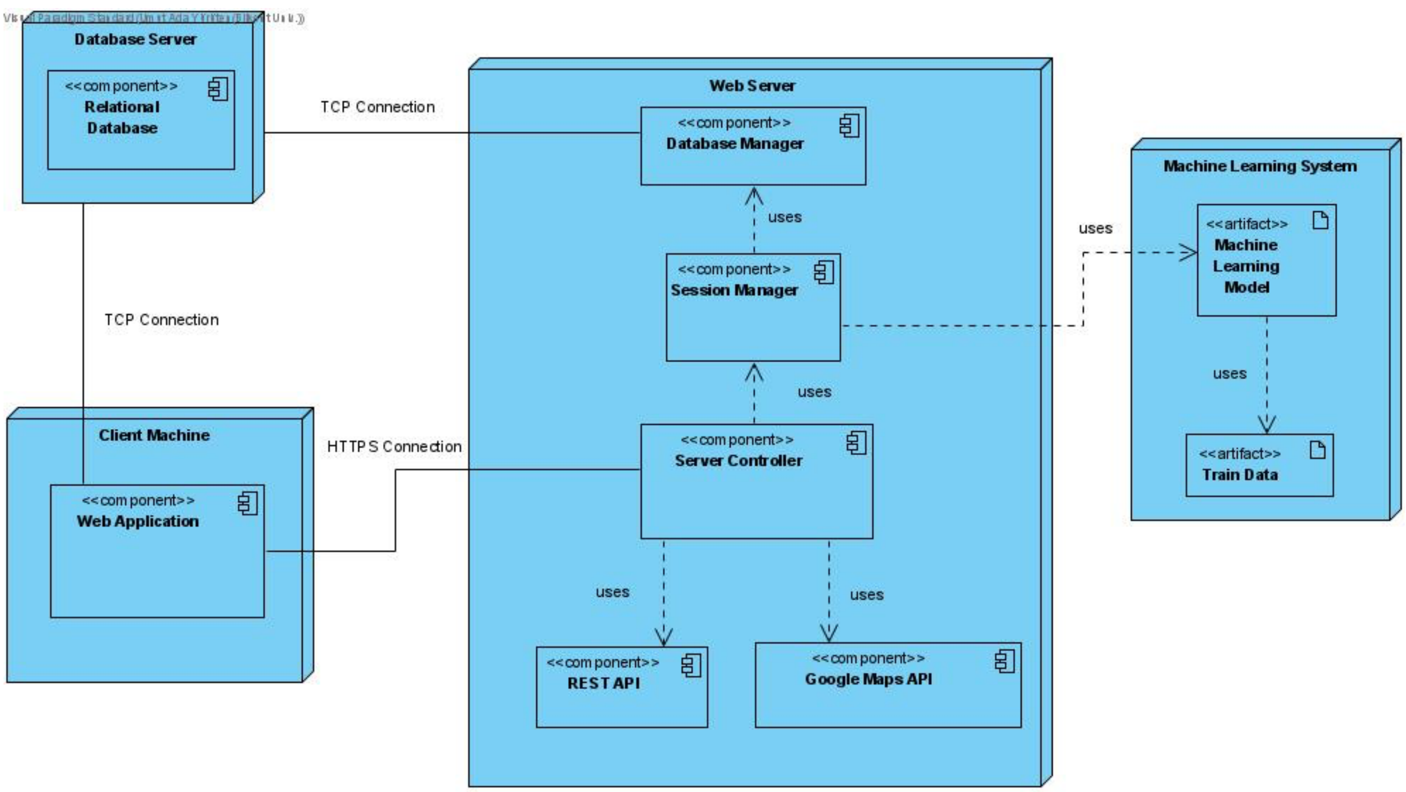

Figure 2: Deployment Diagram

3.4 Persistent Data Management

In PolliVidis, some operations require user specific services while some do not. For example, it offers each user to see the pollen map where the pollen density of Turkey is shown or analyze their pollen data without requiring any registration. However, in order to share analysis data with the application and allow PolliVidis to update the pollen map with analysis data, one should register the website and login as an academic. Therefore, we took into consideration that some operations require user specific services while some do not while designing our database.

As user specific data, information such as name, password and email will be kept in our database. Passwords will be encrypted. Furthermore, the pollen map data, the location and density of pollen data that the academician uploaded, and the photo of the pollen will also be kept in the database. As the database system, we use MySQL database since as developers we are familiar with SQL, SQL tables are well fitted for our data, it is a well-known database for large projects and it can be easily integrated with the tools we use such as Django, REST, React JS.

3.5 Access Control and Security

In PolliVidis, there are two types of users: registered academic users, and unregistered users. Users will register using their names, email addresses, valid passwords, appellation and institution. In order to login, users will use their email and password. In the case of forgetting the password, we will send a link to their email, and they will be able to change their password. The password of the registered users will be saved in the database in encrypted form. A registered academic user will be able to share data in a pollen map, therefore for security reasons, an email address can only be used by a single user. Furthermore, there will be a validation system to verify that the user is actually an academic user.

Users will be able to see/edit their credentials and personal data. Any user can see the pollen map data, however to share academic data, users need to login to the website.

3.6 Global Software Control

PolliVidis is a user-driven event and session-based web application. While most functionality is supplied according to the user’s queries, requests; user account requires a session-based functionality as well.

PolliVidis supports both anonymous usage and academic, login-based functionality. Thus, most of the anonymous usage relies on the user’s queries rather than a session-based usage. However, academic usage with extensive functionalities supplied by an account login requires additional session-based software control. This extension is supplied by the Session Manager in the Backend Subsystem which holds academic user information. This information also allows us to check whether the user should be able to view some data or not such as other academic’s sample analysis.

This should be noted that the most functionality of the PolliVidis Pollen Map with Google Maps API is supplied by the user’s queries rather than a session-based software control.

In the case of session-based software control; several extra functionalities are supplied such as a Profile Page, Previous Analyses, and ability to view detailed Analysis. In order to determine if PolliVidis should supply these; Session Manager passes the session information, user information to the corresponding managers.

3.7 Boundary Conditions

3.7.1 Starting PolliVidis

PolliVidis is a web application which the users should access via a browser. Users should enter the proper URL to access PolliVidis and they will be welcomed with the Pollen Map. As the application supports anonymous users, PolliVidis welcomes its users with its core functionality rather than a Login Page. If the user is an academic and wants to access extra functionalities, s/he should login via the Login Page with proper credentials. If it is the first time the academic user visits the application, s/he can sign-up to PolliVidis using the Sign-Up Page with required information and has to wait for the confirmation. Either case, the user can view the Pollen Map and all the samples it contains in the Pollen Map Page or directly start to analyze his/her samples using the Sample Analysis Page by uploading his/her samples.

3.7.2 Termination

If the user did not login-in to the application and views it anonymously, no termination procedure is required rather than closing the website. However, if the academic user logged in to his/her account; proper log-out from the options panel is suggested.

3.7.3 Failure

There are some possible sources of failure during a session and the most probable ones are internet loss, database failure, and unresponsive server.

In the case of internet loss, an anonymous user might lose his/her sample analysis report since PolliVidis does not save anonymous analyses. In this case, the user can just re-analyze the sample later with proper internet connection. If the user is an academic, the analysis will be saved into the Previous Analyses Page where the user can view it later.

Since all procedures involving Sample Analysis happen on the server side, data loss is not expected in the case of connection loss for academic usage.

In the case of database failure, both user types will be affected similarly. Possible effects are unable to view the Pollen Map as all samples that Pollen Map uses stored in the database and unable to save the Pollen Analysis to the database for the academic usage.

In the case of unresponsive server, no communication between the users and PolliVidis is possible and possible data loss depends on the time the server stops working.

4. Subsystem Services

4.1. Client Side

4.1.1. UI Subsystem

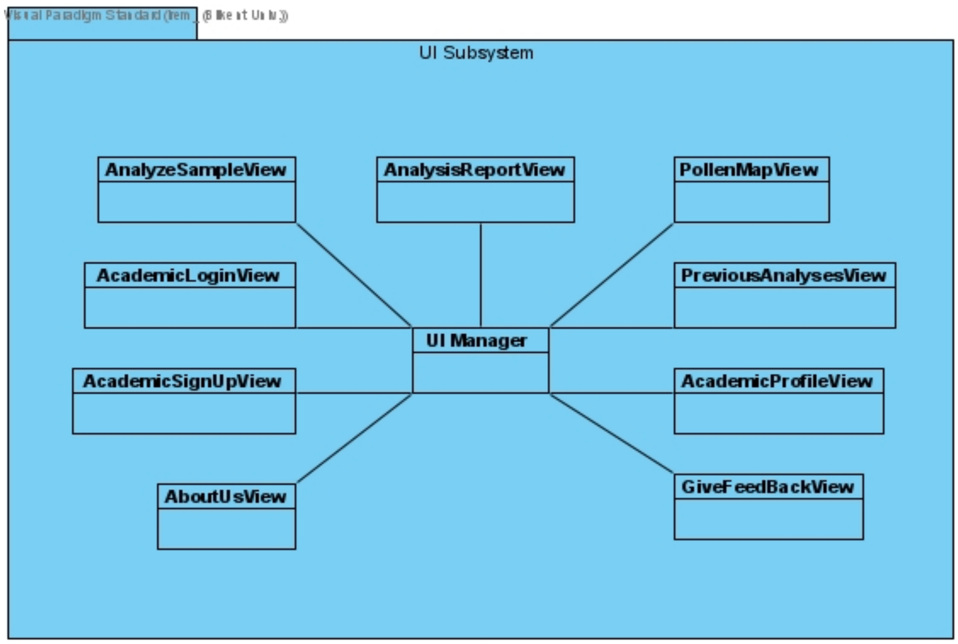

Figure 3: UI Subsystem

4.1.1.1 UI Manager

UI Manager handles the navigation and main structure of the frontend. It controls the presentation tier.

4.1.1.2 AnalyzeSampleView

AnalyzeSampleView handles the user interface of the screen in which users upload sample images and request an analysis.

4.1.1.3 AnalysisReportView

AnalysisReportView handles the user interface of the screen in which the analysis report of the users' samples is shown.

4.1.1.4 PollenMapView

PollenMapView handles the user interface of the Google Maps pollen map which contains the pollen analyses as markers. When clicked on one, the analysis report of the analysis is shown.

4.1.1.5 PreviousAnalysisView

PreviousAnalysisView handles the user interface of the screen in which an academic's previous analysis reports are shown.

4.1.1.6 AcademicLoginView

AcademicLoginView handles the user interface of the screen in which an academic can login.

4.1.1.7 AcademicSignUpView

AcademicSignUpView handles the user interface of the screen in which a user can sign up as an academic.

4.1.1.8 AcademicProfileView

AcademicProfileView handles the user interface of the screen that shows the profile information of an academic.

4.1.1.9 AboutUsView

AboutView handles the user interface of the screen that shows PolliVidis developers' information.

4.1.1.10 GiveFeedBackView

GiveFeedBackView handles the user interface of the screen in which users can send feedback about PolliVidis.

4.2. Server Side

4.2.1. Backend Subsystem

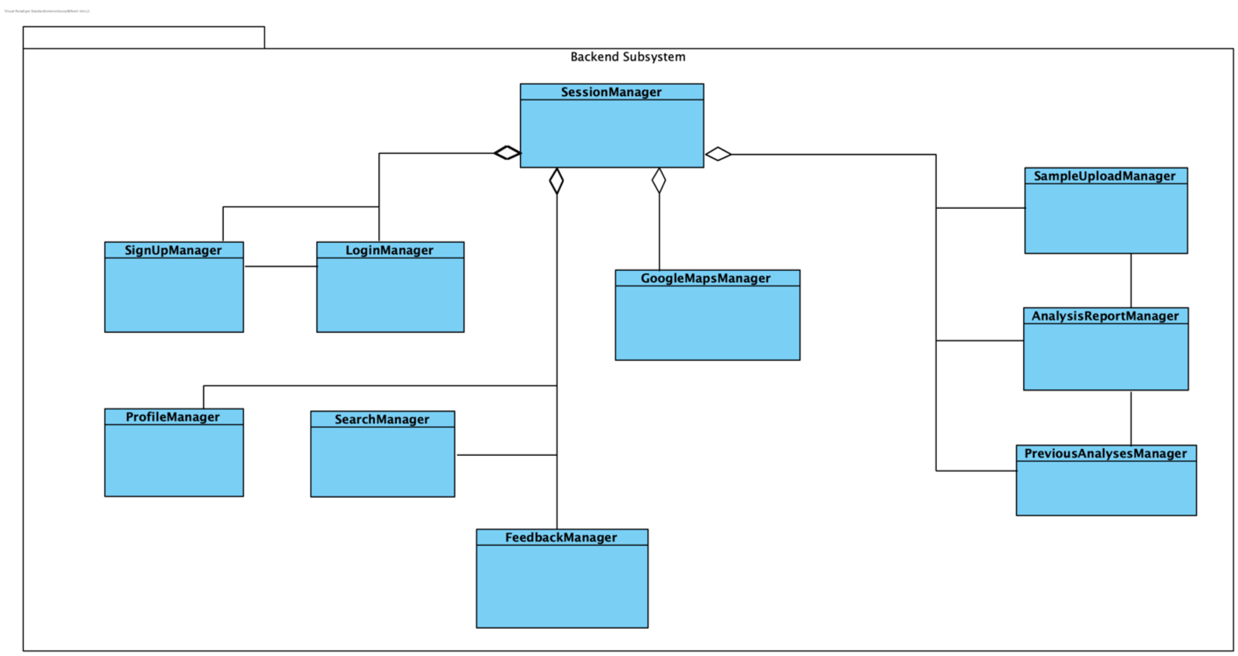

Figure 4: Backend Subsystem

4.2.1.1 SessionManager

Session Manager welcomes all queries coming from users with different sessions. It then redirects the queries to the corresponding manager for an evaluation. This manager also handles the sessions of each logged-in academic user and page redirections.

4.2.1.2 SampleUploadManager

Sample Upload Manager accepts sample image(s) from the user and directs them to the ML Subsystem for pollen extraction and classification. When the ML Subsystem returns the analysis, this manager redirects the user to the Analysis Report page and Analysis Report Manager to view the returned analysis report of the sample.

4.2.1.3 AnalysisReportManager

Analysis Report Manager allows users to view the analysis reports of their samples. This manager handles both fresh analyses redirected from the Sample Upload Manager, and old analyses that users can view from the Pollen Map and Previous Analysis Page. This manager transforms the analysis coming from the ML Subsystem or the database into an analysis report.

4.2.1.4 PreviousAnalsesManger

Previous Analyses Manager retrieves all analyses of the user and lists them as a table. If the user wants to view a specific analysis from the table, this manager retrieves the analysis from the database and passes it to the Analysis Report Manager so that the user can view the entire analysis report.

4.2.1.5 GoogleMapsManager

Google Maps Manager implements the Pollen Map of PolliVidis using Google Maps API. It allows users to interact with the map and retrieve analysis information from different locations. When a user wants to view a specific analysis of a sample, this manager retrieves the specific analysis from the database and passes it to the Analysis Report Manager. It also implements a simple Sample Information Panel for quick look. This manager handles Google Maps pinpoints on the Pollen Map for samples.

4.2.1.6 SearchManager

Search Manager handles the search queries coming from the Google Maps Search Bar. While users can search locations on the map with this search bar, it also allows users to search within samples, academic users, and pollen types. Search Manager uses Database Manager which implements the actual SQL queries within. After obtaining the resultant table, Search Manager lists them in the Search Panel which is the same with the Sample Information Panel.

4.2.1.7 LoginManager

Login Manager operates academic logins to PolliVidis using Database Manager. It checks the database with the typed email and password. This manager also ends the session when the academic user logs out.

4.2.1.8 SignUpManager

Sign Up Manager allows academics to sign up to PolliVidis if they have the proper reliability. Since PolliVidis only allows users with academic background, the reliability of the user is checked by this manager. After a successful sign up, this manager adds the new academic user to the database using Database Manager.

4.2.1.9 ProfileManager

Profile Manager handles academic user profiles with several edit functionalities. Job, email, institution, research gate link, and profile photo can be changed using this manager. It allows users to see their profile in the Profile page. This manager also authorizes academic users who want to view another academic's profile to view the communication information.

4.2.1.10 FeedbackManager

Feedback Manager handles user feedback. This manager allows all users to send feedback to PolliVidis with their communication information so that we can reply back. It saves the feedback to the database using Database Manager.

4.2.2. ML Subsystem

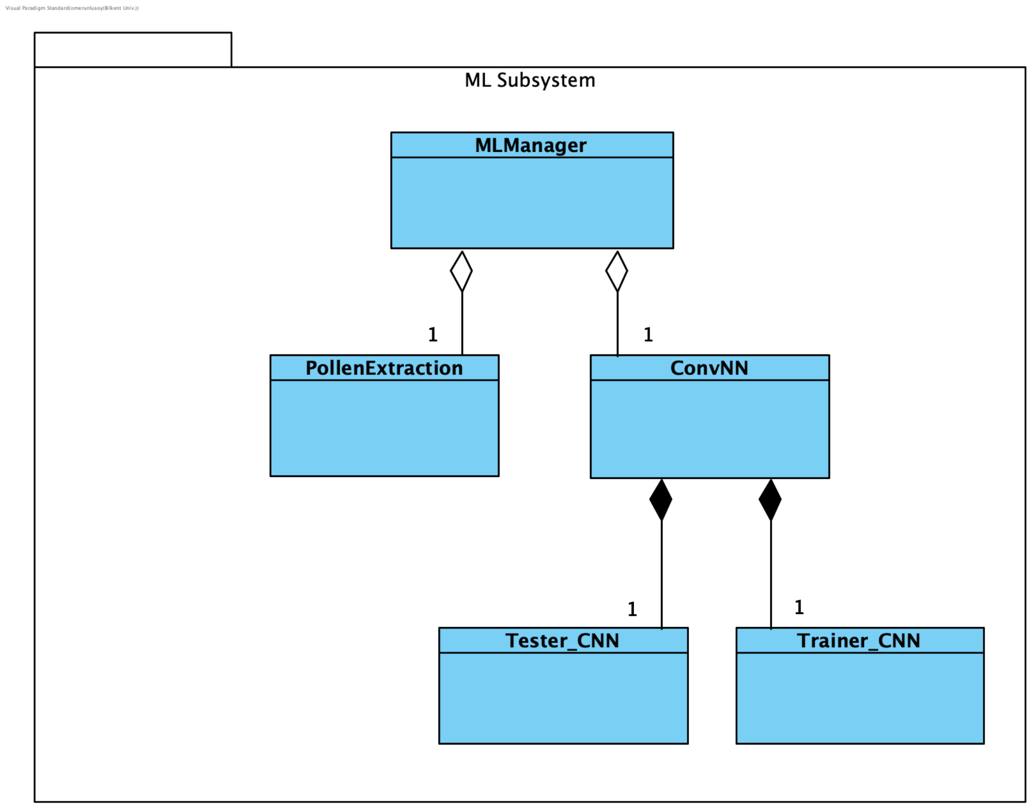

Figure 5: ML Subsystem

4.2.2.1 MLManager

ML Manager handles pollen classification requests coming from the Backend Subsystem. It calls Pollen Extraction to extract pollen images from the given samples image which might include several pollen images. Then, it calls Trainer CNN to predict the classes of each pollen image. Finally, this manager returns all pollen images with their predicted classes to the Backend Subsystem. This manager also handles model training and testing during the model development process.

4.2.2.2 PollenExtraction

Pollen Extraction implements Image Processing Algorithm in Python SCikit-Image package. This class extracts single pollen images from the given sample image which may include few pollens within.

4.2.2.3 ConvNN

ConvNN is the main class of ML Subsystem which implements the Convolutional Neural Network. It holds all hyperparameters and saves the model after the training procedure. Predictions will be made in this class by loading the trained model and forwarding the given pollen image to the model.

4.2.2.4 Trainer_CNN

Trainer_CNN implements the training functionality of the model. This class trains the designed model for later predictions.

4.2.2.5 Tester_CNN

Tester_CNN implements the test functionalities of the model to obtain the accuracy of the model for evaluation.

4.2.3. Database Subsystem

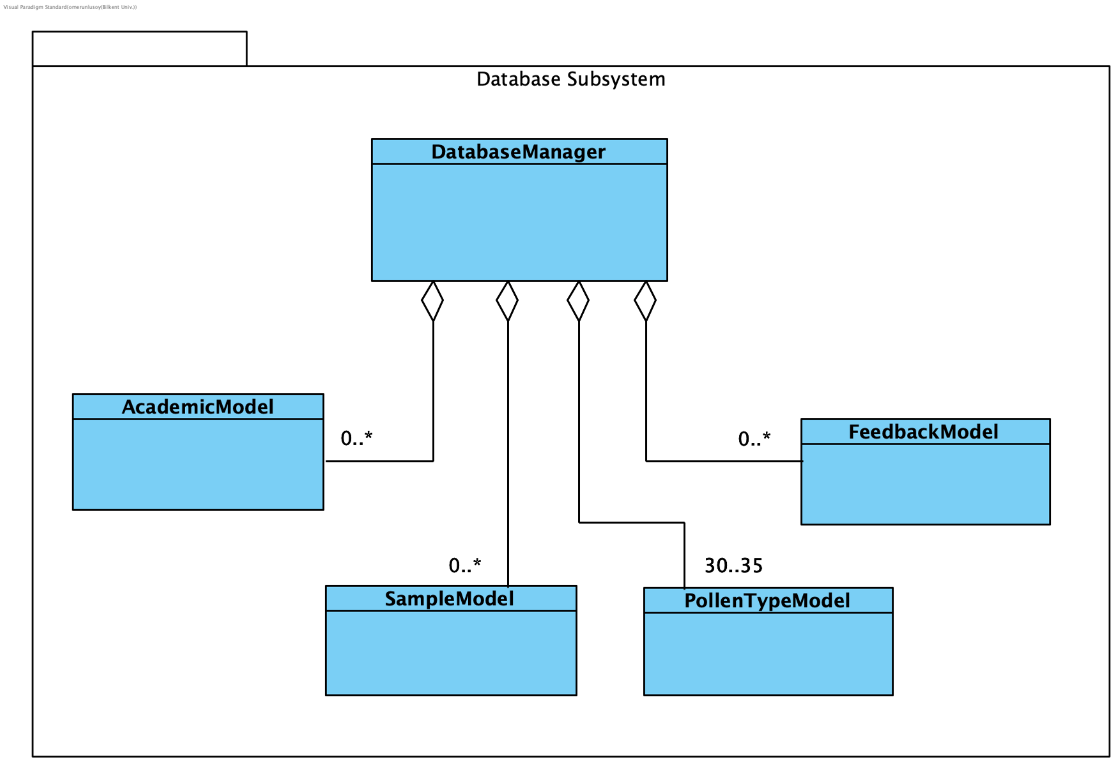

Figure 6: Database Subsystem

4.2.3.1 DatabaseManager

Database Manager handles the communication of the backend and the PolliVidis Database. It implements the schema (definition) of the PolliVidis Database and all the required queries. All queries used by the Backend Subsystem are implemented in this manager. It allows other classes to use Python objects (models) for communication.

4.2.3.2 AcademicModel

Academic Model corresponds to an academic user in the database. It is implemented to ease the communication between Database Manager and other managers. It supplies all academic information such as name, job, institution, and communication information.

4.2.3.3 SampleModel

Sample Model corresponds to a single sample with its analysis in the database. It is used to construct the Pollen Map and view the Pollen Analysis Report. It holds all the information about a single sample such as its location, related academic, and included pollen types with corresponding counts.

4.2.3.4 PollenTypeModel

Pollen Type Model corresponds to a single pollen type in the PolliVidis Database. It holds the name of the pollen with its explanation text. It is used to construct a Pollen Analysis Report from the sample's included pollen list.

4.2.3.5 FeedbackModel

Feedback Model welcomes the user feedback with their communication information and allows Database Manager to upload the feedback to the database.

5 Consideration of Various Factors in Engineering Design

5.1 Public Health

Effect Level: 8

PolliVidis aims to increase the life quality of people that have pollen allergies by showing the allergenic pollen information at his/her area so that users can take precautions. Since it is directly related to public health, we have to have some standards which ensures to protect public health, at least not put it in danger. That is the main reason PolliVidis allows only academics to upload their analyses to its database and uses these analyses to give predictions. The system ensures the reliability of each academic. In any case, the predictions of PolliVidis should be taken as advice.

5.2 Public Safety

Effect Level: 3

Only concern with public safety can be the privacy of academics since PolliVidis stores the communication information of each academic. The system gives privacy options to the academic about who can view their information. If an academic wants to keep his/her communication information private, s/he can do so.

5.3 Public Welfare

Effect Level: 8

PolliVidis wants to increase the user's quality of life by showing the allergenic pollen information at his/her area so that the user can learn the pollen levels and take precautions. Moreover, academics can share sample analyses quickly which may increase collaboration in academic research.

5.4 Global Factors

Effect Level: 3

PolliVidis will be designed in a way that it can be opened to the world easily with some adaptations and regulations since different countries have different allergenic pollen types and a single model cannot be trained to classify all of them. Moreover, analysis sharing between academics would help any academic worldwide.

5.5 Cultural Factors

Effect Level: 2

Since PolliVidis is related mostly to academic research and scientific information sharing, effects of cultural factors are minimal.

5.6 Social Factors

Effect Level: 4

Sample and analysis upload to the pollen map will be allowed for academic staff only. Thus, uploading random images to the map will not be possible to prevent potential abuse. The feedback page allows users to send feedback about system problems or specific samples. Finally, PolliVidis does not support direct messaging which reduces the concerns about online abuse.

5.7 Environmental Factors

Effect Level: 3

It is not expected that the usage of PolliVidis website will cause any environmental problems apart from the energy the servers and database will use and this can be seen as the bare minimum damage. Any type of printed or physical material will not be needed. Moreover, the generated dataset will be published and eliminate the need of collecting the same samples again and again for future research, which saves human and electrical energy.

5.8 Economic Factors

Effect Level: 2

All the services PolliVidis uses are free and PolliVidis does not charge its users for the map usage or sample analysis.

	Effect Level	Effect
Public Health	**8**	Showing allergenic pollen density on a map so that people can take precautions is one of the main goals of the system
Public Safety	**3**	System will have privacy options so academics can keep their information private if they would like to.
Public Welfare	**8**	System aims to raise the public welfare with the Pollen Map feature so that users can take precautions but also as letting people share their pollen samples quickly in order to increase collaboration in academic research.
Global Factors	**3**	PolliVidis will be designed in such a way that it can be a world-wide application easily, also sharing analyses with the world would help any academics worldwide.
Cultural Factors	**2**	The effect of cultural factors are negligibly low.
Social Factors	**4**	PolliVidis does not have a direct messaging service and the system will not allow people to upload random images so that online abuse will be prevented.
Environmental Factors	**3**	Publishing the dataset will eliminate the need of collecting the same samples, hence consumption of electric energy will be minimized.
Economic Factors	**2**	The effect of economic factors is negligibly low.

6. Teamwork Details

6.1 Contributing and Functioning Effectively on the Team

We divided the work into team members, and we believe that everyone in the group gave their best while doing their assigned jobs. Furthermore, we created sub-teams of two or three such as ML team, UI team, data team, backend team in order to focus on a certain task better.

We made checklists for the jobs and each team member reported what they did and the difficulties they faced and we tried to overcome them as a team.

6.2 Helping Creating a Collaborative and Inclusive Environment

As group members, we believe that we work in a collaborative and inclusive environment. As a group, we always tried to assign at least two or three people to work on a task since we believe that would create a more collaborative environment. We always try to inform others about the progress of our current task and conduct meetings regularly where we divide work or explain to each other what we have done so far. Furthermore, sometimes there were cases where the learning task was huge. In these cases, we try to divide it and set a meeting where we explain our part to others.

6.3 Taking Lead Role and Sharing Leadership on the Team

For each different job, we chose a leader and this leader made sure that we delivered the job before the deadlines. Also, the people that took the leading role ensured that everyone on the team had collaborated on the job effectively. For the Machine Learning part of the project such as implementing CNN models, collecting data and processing the data part of the project, Ömer is the team leader and he is working with Gamze as a team. For the application part of the project, Irem, Ece and Umut Ada are working together. For the backend side of the PolliVidis website, such as database, and implementations using Python Umut Ada is the team leader. For the UI part and other implementations, such as integrating Google Maps API, İrem is the leader. Both Umut Ada and İrem are working with Ece as a team.

7. References

[1] İ. Tekin, Ö. Ünlüsoy, and A. Acar, "Ankara University and Pollen Project," 24-Sep-2021.

[2] "About the customer matching process," *Google Ads Help*. [Online]. Available: https://support.google.com/google-ads/answer/7474263?hl=en. [Accessed: 08-Nov-2021].

[3] "How fast should my website load?" *Blue Corona*, 20-Aug-2021. [Online]. Available: https://www.bluecorona.com/blog/how-fast-should-website-be/. [Accessed: 08-Oct-2021].

[4] V. Sevillano and J. L. Aznarte, "Improving classification of pollen grain images of the POLEN23E dataset through three different applications of deep learning convolutional Neural Networks," PLOS ONE, vol. 13, no. 9, 2018.

[5] V. Sevillano, K. Holt, and J. L. Aznarte, "Precise automatic classification of 46 different pollen types with convolutional neural networks," PLOS ONE, vol. 15, no. 6, 2020.