CS 491
Senior Design Project
2021 Fall

Analysis 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

Table of Contents

1. Introduction....3
2. Current System....4
3. Overview....4
4. Functional Requirements....6
    4.1 System Functionality.... 6
    4.2 User Functionality.... 6
5. Nonfunctional Requirements....7
    5.1 Usability.... 7
    5.2 Reliability.... 7
    5.3 Privacy and Security.... 7
    5.4 Efficiency.... 8
    5.5 Accessibility.... 8
    5.6 Extensibility.... 8
6. Pseudo Requirements....9
    6.1 Version Control and Management.... 9
    6.2 Implementation.... 9
7. System Models....10
    7.1 Use Case Model.... 10
    7.2 Object and Class Model.... 19
    7.3 Dynamic Models.... 24
    7.4 User Interface.... 33
8. Other Analysis Elements....42
    8.1 Consideration of Various Factors in Engineering Design.... 42
    8.2 Risks and Alternatives.... 44
    8.3 Ensuring Proper Teamwork.... 45
    8.4 Ethics and Professional Responsibilities.... 45
    8.5 Planning for New Knowledge and Learning Strategies.... 45
    8.6 Project Plan.... 46
9. References....50

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.

PolliVidis aims 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.

In this report, an analysis of the system is provided and explained. Firstly, the currently available systems are provided and then the proposed system is explained along with the differences with the current systems. Next, the functional, non-functional, and pseudo requirements are listed. Followingly, the scenarios, common use case models and class models are presented. Afterwards, dynamic models are described. Finally, the user interface mock-ups are presented and explained.

2. Current System

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 [1]. Furthermore, there are some successful models that work with other microscope types but they lack a functional end product that is publicly available [2].

3. Overview

PolliVidis is a pollen classification platform for academics and students alike. It can also be utilized by people with pollen allergies. PolliVidis will be widely accessible as it will be a web application. It will provide a pollen map built by the academic community. Furthermore, it will allow data sharing between academics which will contribute towards building a vast pollen database and help research in the palynology field in the process.

In PolliVidis, there are three aspects that will be offered: classification and analysis of given pollen samples with a trained ML model, creation of a pollen database for further research, and construction of a pollen map of Turkey. After the project is finalized, any given pollen sample with allergenic pollen types will be classified autonomously. In addition, together with the pollen map, palynologists and academics will be able to exchange academic information. Moreover, people with pollen allergies will track the frequency dates, and locations of these pollen.

In the initial stage, there is a need for a large labeled pollen dataset which the ML model will be trained with. Unfortunately, all available datasets are designed for students and professors to learn the details of each pollen species. Thus, 2-5 images per pollen type are available [1, 3]. We will constitute a light microscope pollen dataset for ML models from scratch. After communication with Ankara University, Researcher Aydan Acar decided to help us constitute the dataset [4]. She has ready-to-be-photographed and purified preparate of most common allergenic pollens in Turkey. We will take 200-300 photos of 25-35 pollen species to obtain 500-700 pollen photos per species. This photography procedure takes considerable time using a light microscope. Unfortunately, there is no available light microscope allergenic pollen dataset and we have to construct this dataset from scratch.

After the constitution of the dataset, we will design a neural network model to classify these allergenic pollens. The neural network will be a Convolutional Neural Network (CNN) type and it will be trained with our dataset. Several methods such as transfer learning and data augmentation will be used to increase the accuracy. After the training procedure, the model will be available through PolliVidis website for free pollen analysis for everyone.

Afterwards, the PolliVidis website will be built. The pollen map which will use Google Maps API will be added to the main page. We will design an interface so that everyone, without an account, will be 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 example via the website. However, only palynologists and academics can update the pollen map with their samples and uploading data to the map will require an academic account. Login will be required to protect the accuracy and the reliability of the map.

The analysis we plan to supply with our model will consist 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 and palynologists agree to share their sample, the pollen map will be updated with their analysis. Hence, everyone will be able to reach academic and statistical information using this platform and pollen map. Moreover, students will also be able to use the PolliVidis for educational purposes, such as uploading pollen photos and learning their type without consulting their instructor.

4. Functional Requirements

4.1 System Functionality

The system should:

- detect and classify allergen pollen types found in Turkey.
- should count the pollen types and density in the uploaded pollen samples.
- display the sample analysis report after the sample upload.
- require academic registration for academic usage.
- not require registration for pollen analysis.
- not require registration for allergy map usage.
- allow any user to upload pollen sample to analyze.
- should store the pollen samples that are uploaded by academic users.
- display pollen map supported by Google Maps.
- display pollen samples and their analyses reports on the map.
- allow academics to look through the pollen analyses with its publisher information.
- allow academics to view communication information of other academics.

4.2 User Functionality

The academic user should:

- register to the system with required information.
- login in order to contribute to the Pollen Map.
- view other pollen samples and their analysis with its publisher information.
- upload pollen samples to analyze.
- able to see the count, density, and the species of the sample from the analysis.
- look through the pollen analyses with its publisher information.

The anonymous user should:

- not register to the system to analyze any pollen samples.
- not register to the system to look through the Pollen Map.

- be able to see the count, density, and the species of each pollen sample from its analysis report.

5. Nonfunctional Requirements

5.1 Usability

The system should

- be able to work on most search engines such as Safari, Chrome, Firefox, and Mozilla.
- yield an analysis with the pollen samples taken under a light microscope within 2-3 seconds.
- allow pollen analysis without any registration or login.
- allow Pollen Map usage without any registration or login

5.2 Reliability

The system should

- ensure that the pollen map data that it offers is reliable and obtained from a palynologist or academic's samples.
- ensure reliable results (more than %90 accuracy) for the pollen classification.
- should not lose any pollen data unless the user deletes or doesn't let data to be added to the database

5.3 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[5]

- get permission from academic users to share the location and the date of the samples with other users.
- not require any personal information from unregistered users to analyze pollen and look through the pollen map.

5.4 Efficiency

- The webpage’s loading time should not exceed 2 seconds which is the maximum loading time recommended by Google [6].
- Analysis will be conducted on the server rather than the user's own computer which should decrease memory usage.

5.5 Accessibility

The system should be

- available on most used internet browsers (Safari, Chrome, Firefox, Microsoft Edge, and Mozilla).

5.6 Extensibility

- The ML model can be improved in the future with more datasets available.
- The allergenic pollen in Turkey will be examined to scale the project. However, the pollen types can be increased in the future.
- Pollen Map will be based on Turkey where the examined pollen is located. This map can be extended as global in the future.

6. Pseudo Requirements

6.1 Version Control and Management

- GitHub and Git will be used for version control.
- Google Docs will be used for project management and source sharing.
- GitHub pages will be used as project website.

6.2 Implementation

- PolliVidis will be implemented as a website.
- The system will have server-client architecture.
- Python will be used in the back-end.
- For the Machine Learning model, CNN will be used.
- PyTorch package will be used for implementing convolutional neural networks.
- Python Django and React frameworks will be used for building the website.
- MySQL will be used in database design.
- VGG-16 or VGG-19 pre-trained models are planned to be used as transfer learning strategy.
- All dataset will be collected by us, in Ankara University Science Faculty labs.
- Dataset will be stored in the database of the project written in MySQL.
- Object Oriented Programming (OOP) paradigms will be followed during the implementation.

7. System Models

7.1 Use Case Model

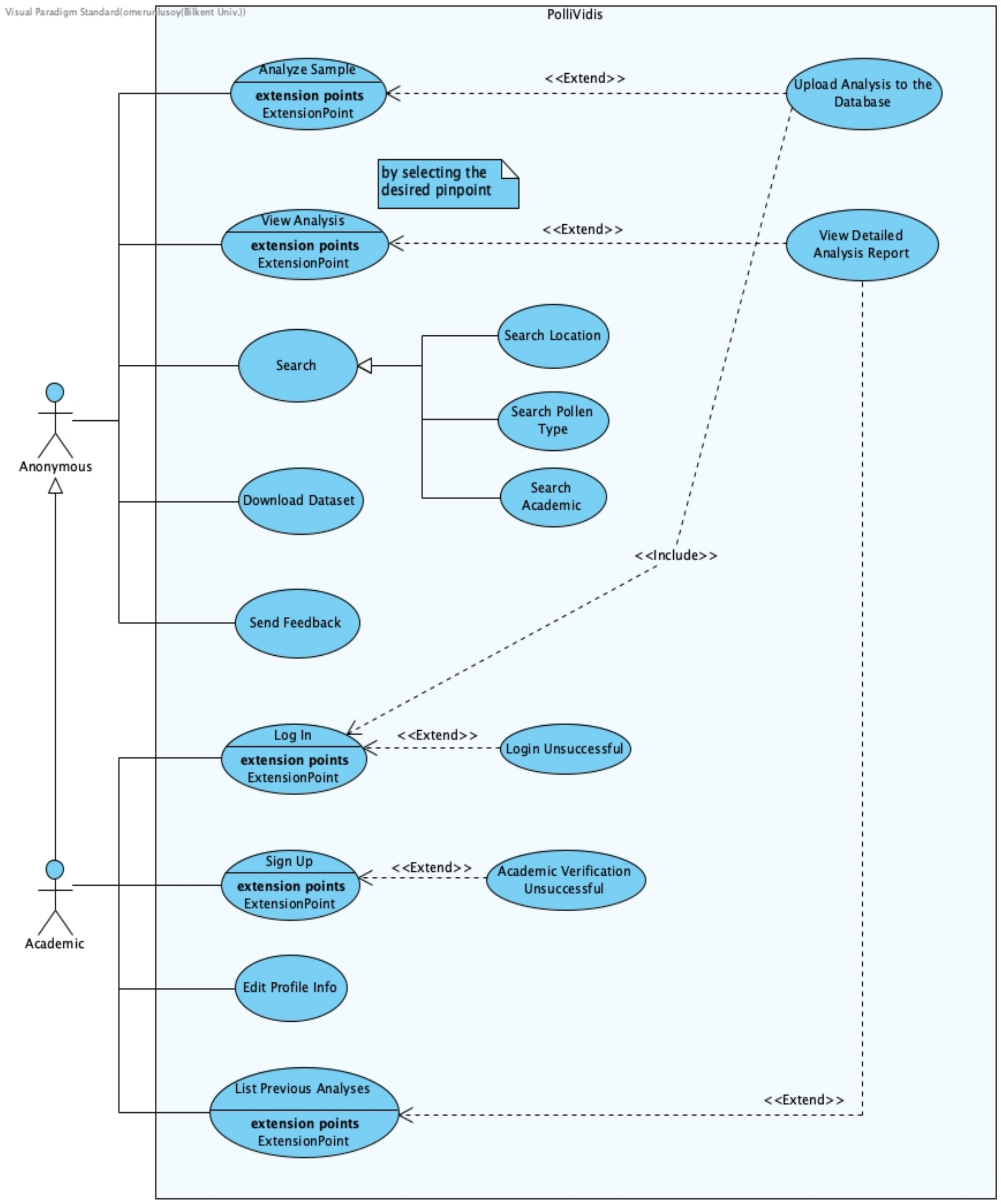


Figure 1: Use Case Diagram of PolliVidis

<u>Use Case#1</u>

Use Case: Analyze Sample

Primary Actor: Anonymous

Stakeholders and Interests:

- A user wants to upload a sample to be analyzed

Pre-conditions:

- User is in Upload Sample Page

Post-conditions:

- User will have the analysis
- User will be in Report Page of the analysis

Entry conditions:

- User uploads image sample or samples

Exit conditions:

- No exit condition.

Main Event Flow:

1. In Upload Sample Page, user uploads the sample.
2. Image is analyzed.
3. User is redirected to Report page of the sample.

<u>Use Case#2</u>

Use Case: View Analysis

Primary Actor: Anonymous

Stakeholders and Interests:

- User wants to view an analysis of a sample represented by a pinpoint in the map.

Pre-conditions:

- User must be in Map Page.

Post-conditions:

- User will see the analysis of the selected sample.

Entry conditions:

- User clicks on a pinpoint on the map.

Exit conditions:

- No exit conditions.

Main Event Flow:

1. User enters Map Page
2. User clicks on a pinpoint
3. Information about the sample is presented.

Alternative Event Flow:

1. User enters Map Page
2. User clicks on a pinpoint
3. Information about the sample is presented.
4. User clicks to see further information about sample.
5. User is redirected to Report Page of the sample.

Use Case#3

Use Case: Search

Primary Actor: Anonymous

Stakeholders and Interests:

- User wants to search a sample, location, or academic

Pre-conditions:

- User must be in Map Page

Post-conditions:

- A list of search results will be presented to the user.

Entry conditions:

- User enters parameter to the search bar in the Map Page

Exit conditions:

- No exit conditions.

Main Event Flow:

1. User enters Map Page
2. User enters a parameter to be searched
3. Relevant information is listed to user

Use Case#4

Use Case: Download Dataset

Primary Actor: Anonymous

Stakeholders and Interests:

- User desires to download the dataset used to train the ML model of the application

Pre-conditions:

- User has opened the right panel

Post-conditions:

- User has access to the dataset

Entry conditions:

- User clicks the download button on the right panel

Exit conditions:

- No exit conditions

Main Event Flow:

1. User opens the right panel
2. User clicks the “Download Dataset” button

Use Case#5

Use Case: Send Feedback

Primary Actor: Anonymous

Stakeholders and Interests:

- User wants to provide feedback about the website or the system.

Pre-conditions:

- User must be in Send Feedback Page

Post-conditions:

- User’s feedback will be sent

Entry conditions:

- User has entered appropriate information and requested to send feedback

Exit conditions:

- No exit conditions

Main Event Flow:

1. User enters Send Feedback Page.
2. User enters relevant information of name, email contact, and feedback message.
3. User clicks on “Send” button.
4. User is redirected to home page.

Use Case#6

Use Case: Login

Primary Actor: Academic

Stakeholders and Interests:

- An academic user wishes to login

Pre-conditions:

- User must not be logged in previously
- User must be in the Login Page

Post-conditions:

- User is prompted about the login attempt
- User is redirected to appropriate page according to login attempt success

Entry conditions:

- User enters email and password then clicks on “Login” button

Exit conditions:

- Login attempt is successful OR
- Login attempt is unsuccessful

Main Event Flow:

1. User enters Login Page
2. User enters email and password
3. User clicks on “Login” button
4. Attempt is successful
5. User is redirected to home page

Alternative Event Flow:

1. User enters Login Page
2. User enters email and password

3. User clicks on “Login” button
4. Attempt is unsuccessful
5. User is prompted to enter valid email and password

<u>Use Case#7</u>

Use Case: Sign Up

Primary Actor: Academic

Stakeholders and Interests:

- A user wishes to sign up to the system

Pre-conditions:

- User has not signed up before
- User has not logged in beforehand.
- User must be in Signup Page

Post-conditions:

- User is prompted about the success of the signup attempt
- User is redirected to the appropriate page according to attempt success

Entry conditions:

- User is on Signup Page
- User has entered name, appellation, email, institution, and password
- User clicked on “Signup” button.

Exit conditions:

- Signup is successful OR
- Signup is unsuccessful

Main Event Flow:

1. User enters Signup Page
2. User enters entered name, appellation, email, institution, and password
3. User clicks on “Signup” button
4. Information is checked to be valid

5. User is prompted about success
6. User is redirected to home page

Alternative Event Flow:

1. User enters Signup Page
2. User enters entered name, appellation, email, institution, and password
3. User clicks on “Signup” button
4. Information is checked to be invalid
5. User is prompted about invalidity and asked for valid information.

Use Case#8

Use Case: Edit Profile Info

Primary Actor: Academic

Stakeholders and Interests:

- Academic user wishes to edit their profile information

Pre-conditions:

- Academic is logged in

Post-conditions:

- Information about the user is updated

Entry conditions:

- User must be in Edit Profile Page

Exit conditions:

- User completes the update

Main Event Flow:

1. User enters Edit Profile Page
2. User changes profile information

3. User saves the update
4. User is redirected to Profile Page

<u>Use Case#9</u>

Use Case: List Previous Analyses

Primary Actor: Academic

Stakeholders and Interests:

- Academic user wants to see previously uploaded samples analysis by himself/herself.

Pre-conditions:

- Academic user is logged in.

Post-conditions:

- User sees previously uploaded samples

Entry conditions:

- User enters Previous Analyses Page

Exit conditions:

- User navigates to another page

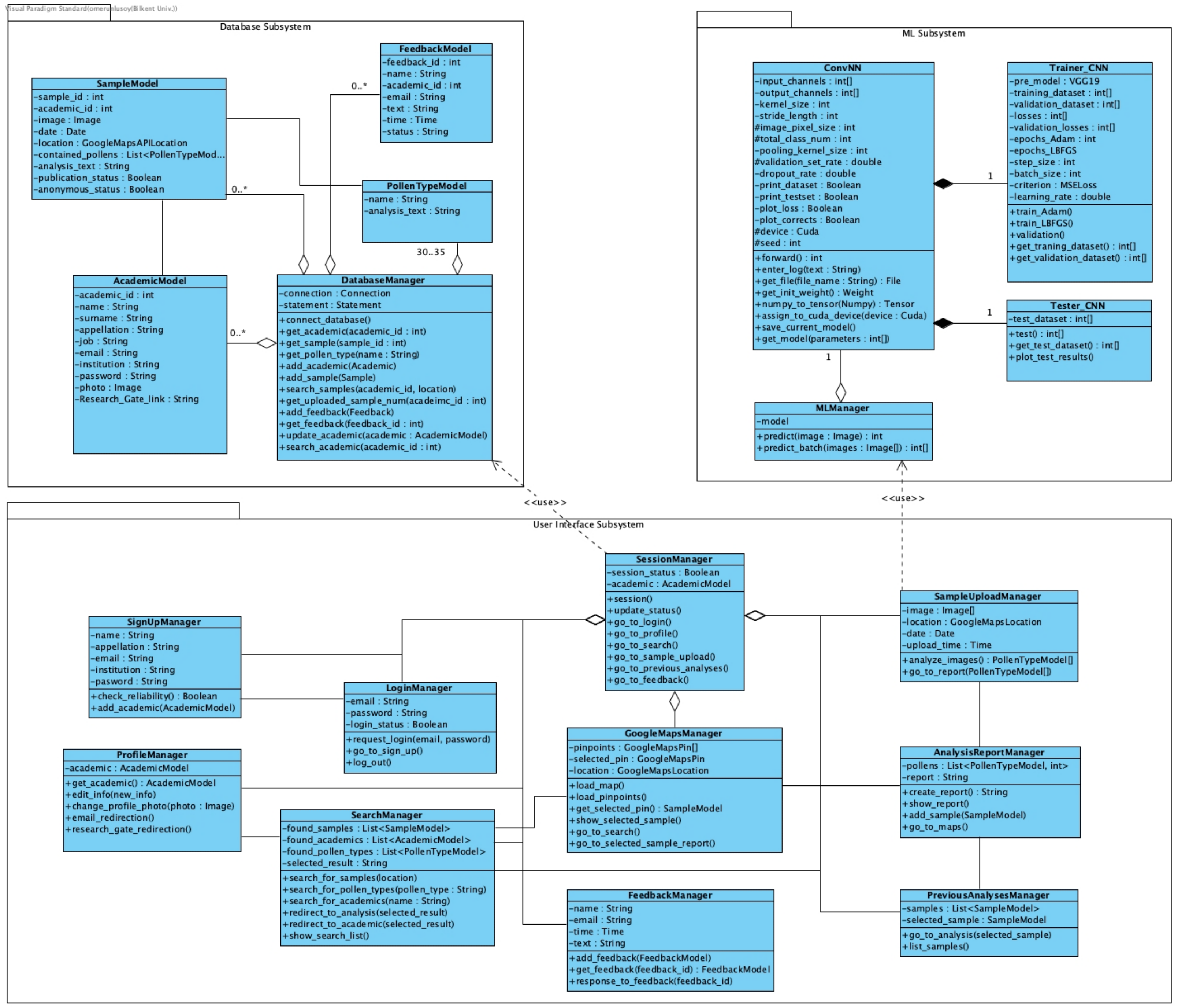
Main Event Flow:

1. User enters Previous Analyses Page
2. User scrolls through the history
3. User clicks on one of the samples
4. User is redirected to Report Page of the clicked sample

Alternative Event Flow:

1. User enters Previous Analyses Page
2. User scrolls through the history
3. User navigates to another page

7.2 Object and Class Model



Visual Paradigm Standard(omerunlusoy(Bilkent Univ.))
Database Subsystem
FeedbackModel
-feedback_id : int
-name : String
-academic_id : int
-email : String
-text : String
-time : Time
-status : String
SampleModel
-sample_id : int
-academic_id : int
-image : Image
-date : Date
-location : GoogleMapsAPILocation
-contained_pollens : List<PollenTypeMod...
-analysis_text : String
-publication_status : Boolean
-anonymous_status : Boolean
0..*
0..*
PollenTypeModel
-name : String
-analysis_text : String
30..35
AcademicModel
-academic_id : int
-name : String
-surname : String
-appellation : String
-job : String
-email : String
-institution : String
-password : String
-photo : Image
-Research_Gate_link : String
0..*
DatabaseManager
-connection : Connection
-statement : Statement
+connect_database()
+get_academic(academic_id : int)
+get_sample(sample_id : int)
+get_pollen_type(name : String)
+add_academic(Academic)
+add_sample(Sample)
+search_samples(academic_id, location)
+get_uploaded_sample_num(acadeimc_id : int)
+add_feedback(Feedback)
+get_feedback(feedback_id : int)
+update_academic(academic : AcademicModel)
+search_academic(academic_id : int)
ML Subsystem
ConvNN
-input_channels : int[]
-output_channels : int[]
-kernel_size : int
-stride_length : int
#image_pixel_size : int
#total_class_num : int
-pooling_kernel_size : int
#validation_set_rate : double
-dropout_rate : double
-print_dataset : Boolean
-print_testset : Boolean
-plot_loss : Boolean
-plot_corrects : Boolean
#device : Cuda
#seed : int
+forward() : int
+enter_log(text : String)
+get_file(file_name : String) : File
+get_init_weight() : Weight
+numpy_to_tensor(Numpy) : Tensor
+assign_to_cuda_device(device : Cuda)
+save_current_model()
+get_model(parameters : int[])
Trainer_CNN
-pre_model : VGG19
-training_dataset : int[]
-validation_dataset : int[]
-losses : int[]
-validation_losses : int[]
-epochs_Adam : int
-epochs_LBFGS
-step_size : int
-batch_size : int
-criterion : MSELoss
-learning_rate : double
+train_Adam()
+train_LBFGS()
+validation()
+get_traning_dataset() : int[]
+get_validation_dataset() : int[]
1
Tester_CNN
-test_dataset : int[]
+test() : int[]
+get_test_dataset() : int[]
+plot_test_results()
1
1
MLManager
-model
+predict(image : Image) : int
+predict_batch(images : Image[]) : int[]
<<use>>
<<use>>
User Interface Subsystem
SessionManager
-session_status : Boolean
-academic : AcademicModel
+session()
+update_status()
+go_to_login()
+go_to_profile()
+go_to_search()
+go_to_sample_upload()
+go_to_previous_analyses()
+go_to_feedback()
SampleUploadManager
-image : Image[]
-location : GoogleMapsLocation
-date : Date
-upload_time : Time
+analyze_images() : PollenTypeModel[]
+go_to_report(PollenTypeModel[])
SignUpManager
-name : String
-appellation : String
-email : String
-institution : String
-pasword : String
+check_reliability() : Boolean
+add_academic(AcademicModel)
LoginManager
-email : String
-password : String
-login_status : Boolean
+request_login(email, password)
+go_to_sign_up()
+log_out()
GoogleMapsManager
-pinpoints : GoogleMapsPin[]
-selected_pin : GoogleMapsPin
-location : GoogleMapsLocation
+load_map()
+load_pinpoints()
+get_selected_pin() : SampleModel
+show_selected_sample()
+go_to_search()
+go_to_selected_sample_report()
AnalysisReportManager
-pollens : List<PollenTypeModel, int>
-report : String
+create_report() : String
+show_report()
+add_sample(SampleModel)
+go_to_maps()
ProfileManager
-academic : AcademicModel
+get_academic() : AcademicModel
+edit_info(new_info)
+change_profile_photo(photo : Image)
+email_redirection()
+research_gate_redirection()
SearchManager
-found_samples : List<SampleModel>
-found_academics : List<AcademicModel>
-found_pollen_types : List<PollenTypeModel>
-selected_result : String
+search_for_samples(location)
+search_for_pollen_types(pollen_type : String)
+search_for_academics(name : String)
+redirect_to_analysis(selected_result)
+redirect_to_academic(selected_result)
+show_search_list()
FeedbackManager
-name : String
-email : String
-time : Time
-text : String
+add_feedback(FeedbackModel)
+get_feedback(feedback_id) : FeedbackModel
+response_to_feedback(feedback_id)
PreviousAnalysesManager
-samples : List<SampleModel>
-selected_sample : SampleModel
+go_to_analysis(selected_sample)
+list_samples()

7.2.1 ML Subsystem

7.2.1.1 MLManager Class

MLManager class allows PolliVidis to use our trained machine learning model with two simple functions one for single analysis and the other one for the batch analysis. This class holds the model as its instance.

7.2.1.2 ConvNN Class

ConvNN class is the main class of our convolutional neural network implementation. It holds several variables such as input and output channel numbers, kernel size, dropout rate, and some print control as most CNN implementations have. Moreover, it has some functions to create log files for us to track the model during its development. Finally, it has save and load model functionalities to save some time during its development process.

7.2.1.3 Trainer Class

Trainer class trains our CNN implementation with its two train functions, one for Adam optimizer and one for LBFGS optimizer. This class will hold the training and validation datasets we created. Several hyperparameters of the model such as step size of the optimizer and learning rate will be tuned with validation process.

7.2.1.4 Tester Class

Tester class will hold the test dataset we created. It will evaluate the model and calculate each evaluation matrix.

7.2.2 Database Subsystem

7.2.2.1 DatabaseManager Class

DatabaseManager class allows PolliVidis to communicate with the database of the website. This class implements all SQL query functions so that no other class has to implement queries. All User Interface Subsystem classes communicate with the database using DatabaseManager class. The class has a function to connect with the SQL database. It also has getter functions for each table. When a getter is called; DatabaseManager runs a query to get the desired table and row, then it places the returned information into the corresponding greeter model class so that the caller function can retrieve and modify the information easily without any SQL query. The class also has functions to add academic, sample, and feedback. Finally, the class has two advance search functions to search through the samples and academics.

7.2.2.2 PollenTypeModel Class

This model class may hold a pollen type when the DatabaseManager class queries its corresponding SQL table. While the number of pollen types our model will use does not change on the SQL table, active model class number can change during a session.

7.2.2.3 AcademicModel Class

This model class may hold an academic account when the DatabaseManager queries its corresponding SQL table. The class may hold all of the academic profile related variables such as name, job, institution etc. This model class can be modified as the academic edits his/her profile page. Moreover, this model class will be used when a user tries to access the communication information of an academic.

7.2.2.4 SampleModel Class

SampleModel class may hold a sample image with its analysis report when the DatabaseManager retrives a sample from the database. This model will be used when a user tries to view a sample or when an academic wants to upload his/her sample with its analysis to our database.

7.2.2.5 FeedbackModel Class

This model class holds feedbacks before they push to the database.

7.2.3 User Interface Subsystem

7.2.3.1 SessionManager Class

SessionManager class handles each session created by a user. As PolliVidis is a website rather than an application, its backend will be codded as session-based. When a user enters PolliVidis website, SessionManager will create a session for him/her. It holds the session information with the logged in academic. This class also handles page redirections.

7.2.3.2 SampleUploadManager Class

SampleUploadManager class handles the sample upload of a user when s/he wants to analyse a sample. A sample model will be created and the image(s) will be sent to the MLManager. When the MLManager returns with the result, this class will direct user to the Analysis Report page to view the report.

7.2.3.3 AnalysisReportManager Class

This class handles the analysis report of a sample. When the user analyzes a sample or wants to view an analysis of someone else, this class generates or retrieves the report. It also uploads the sample object with its report to the database using DatabaseManager.

7.2.3.4 PreviousAnalsesMangerClass

This manager class retrieves all previous analyses of an academic and lists them when s/he wants to review them. It can redirect academic to the Analysis Report page when s/he wants to review one of the analyses.

7.2.3.5 GoogleMapsManager Class

This manager handles Google Maps API and Maps page, it allows users to use pinpoints and basic map functionalities. When the user wants to review a sample by clicking a pinpoint, this manager handles the sample retrieval and Sample Info panel.

7.2.3.6 SearchManager Class

SearchManager class handles sample, pollen type, and academic info searches with the help of the DatabaseManager class's search functions. It lists all the related results within the Search panel.

7.2.3.7 LoginManager Class

This class handles academic logins to the website. It checks the database for the given e-mail and password. It also ends the session when the academic logs out.

7.2.3.8 SignUpManager Class

SignUpManager handles the academic sign-ups to PolliVidis with proper reliability check as PolliVidis is allows academic accounts only. When the academic signs up, this class adds his/her model object to the database.

7.2.3.9 ProfileManager Class

This manager class allows academics to view and edit their profiles. Moreover, it authorizes the users who want to view the communication information of the academic.

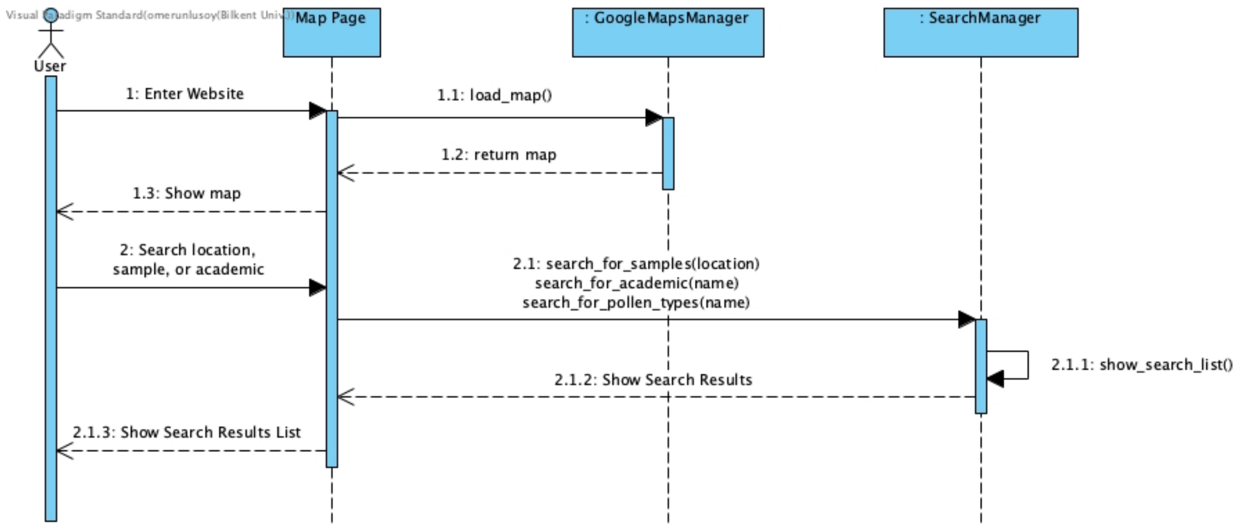
7.2.3.10 FeedbackManager Class

FeedbackManager class allows all users to send feedback to us and get help when it is needed. It creates a feedback model object and puts it into the database using DatabaseManager. As it also takes users communication information, we can return to the user with proper answer. Finally, any academic can ask us questions about the website and the model using Feedback page.

7.3 Dynamic Models

7.3.1 Sequence Diagrams

7.3.1.1 Search Sequence



Visual Paradigm Standard(omerunlusoy(Bilkent Univ.))
User
Map Page
: GoogleMapsManager
: SearchManager
1: Enter Website
1.1: load_map()
1.2: return map
1.3: Show map
2: Search location, sample, or academic
2.1: search_for_samples(location) search_for_academic(name) search_for_pollen_types(name)
2.1.1: show_search_list()
2.1.2: Show Search Results
2.1.3: Show Search Results List

Figure 2: Search Sequence Diagram

Scenario: A user searches in the map page

When a user first enters the website, they are redirected to main page, which is the Map Page. Map Page invokes load_map() method of GoogleMapManager and then presents the results form the manager to user. Afterwards, user may search various locations, samples, oracademics. Afterwards, appropriate method among search_for_samples(loc), search_for_academic(name), and search_for_pollen_type(name) of SearchManager is called. SearchManager searches the requested info in the database and converts the results into a list via show_search_list() method. This list is return to the Map Page which shows the user the results of the search.

7.3.1.2 Login Sequence

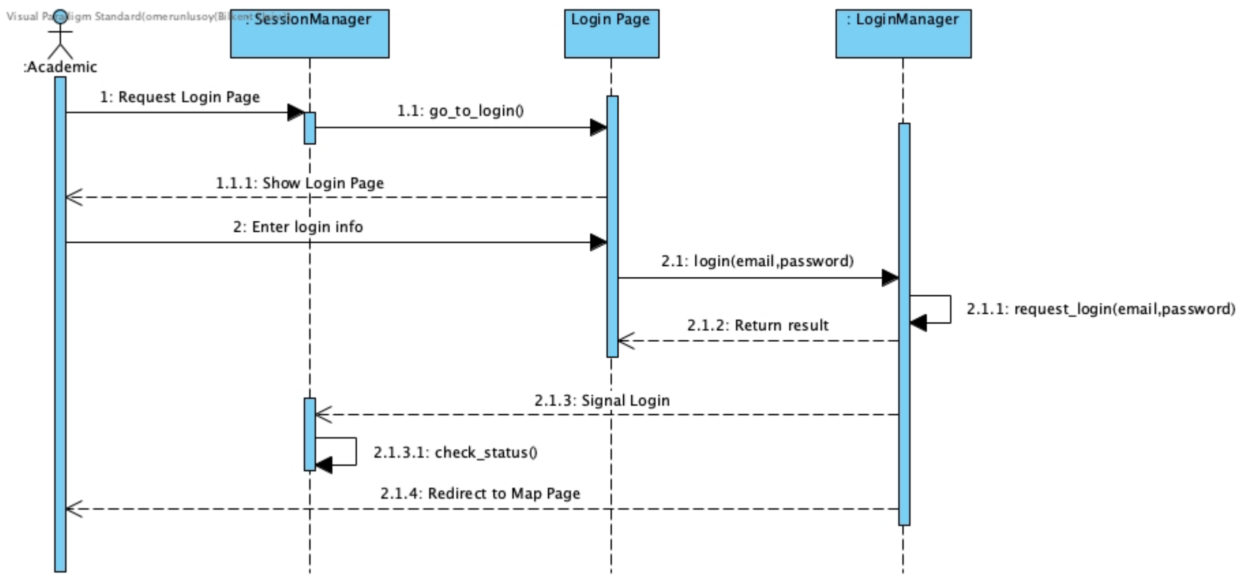


Figure 3: Login Sequence Diagram

Scenario: An academic user login to their account

Firstly, the user requests to enter the login page to login into their account. This request is taken by SessionManager which then calls go_to_login() function to redirect user to the login page. Afterwards, login page is opened and showed to user. Followingly, the user enters their login info, which is the email and password parameters, and this information is sent to the LoginManager instance via the login(email,password) function. Inside this instance, request_login(email,password) function is called that checks whether the given login info is valid in the database. After database acknowledges the login info, LoginManager firstly signals SessionManager that a successful login attempt was made. This allows the SessionManager to update its status and allows the academic user to enter pages restricted only to those who logged in. Meanwhile, LoginManager also redirects user to the Map Page which is the main page of the application.

7.3.1.3 Sign-Up Sequence

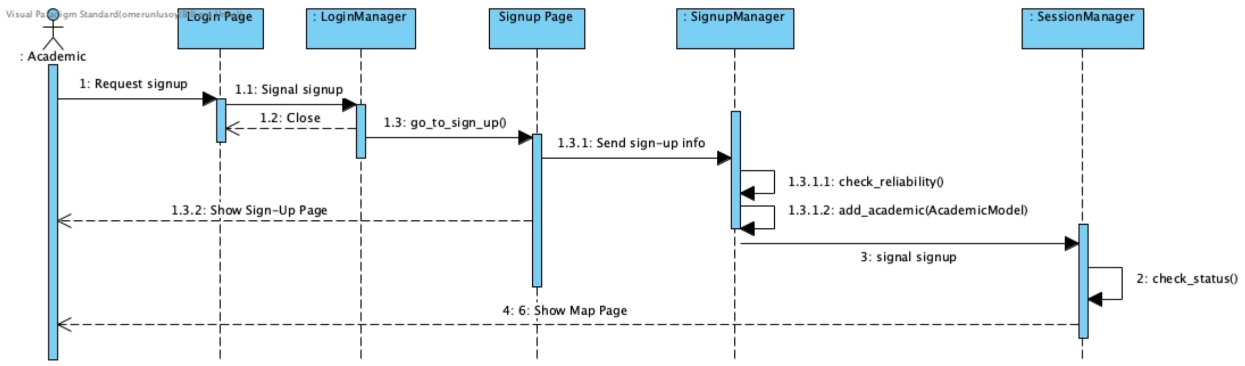


Figure 4: Sign-Up Sequence Diagram

Scenario: A user sign ups to the application

After user clicks to sign up button on login page, LoginManager closes the Login Page and then opens the sign-up page by calling go_to_sign_up() method. After taking information from the user, the appropriate sign-up information is sent to the SignupManager. Followingly, SignupManager checks the reliability of the information, such as checking the existence of the academic in question, with the check_reliability() function . After validating the reliability of the given signup info, SignupManager uploads the new academic profile to the database via add_academic(AcademicModel) method and then signals SessionManager there was a successful signup. SessionManager updates its status to allow user to enter academic-only restricted pages and finally redirects the user to the main Map Page.

7.3.1.4 Upload Sample Sequence

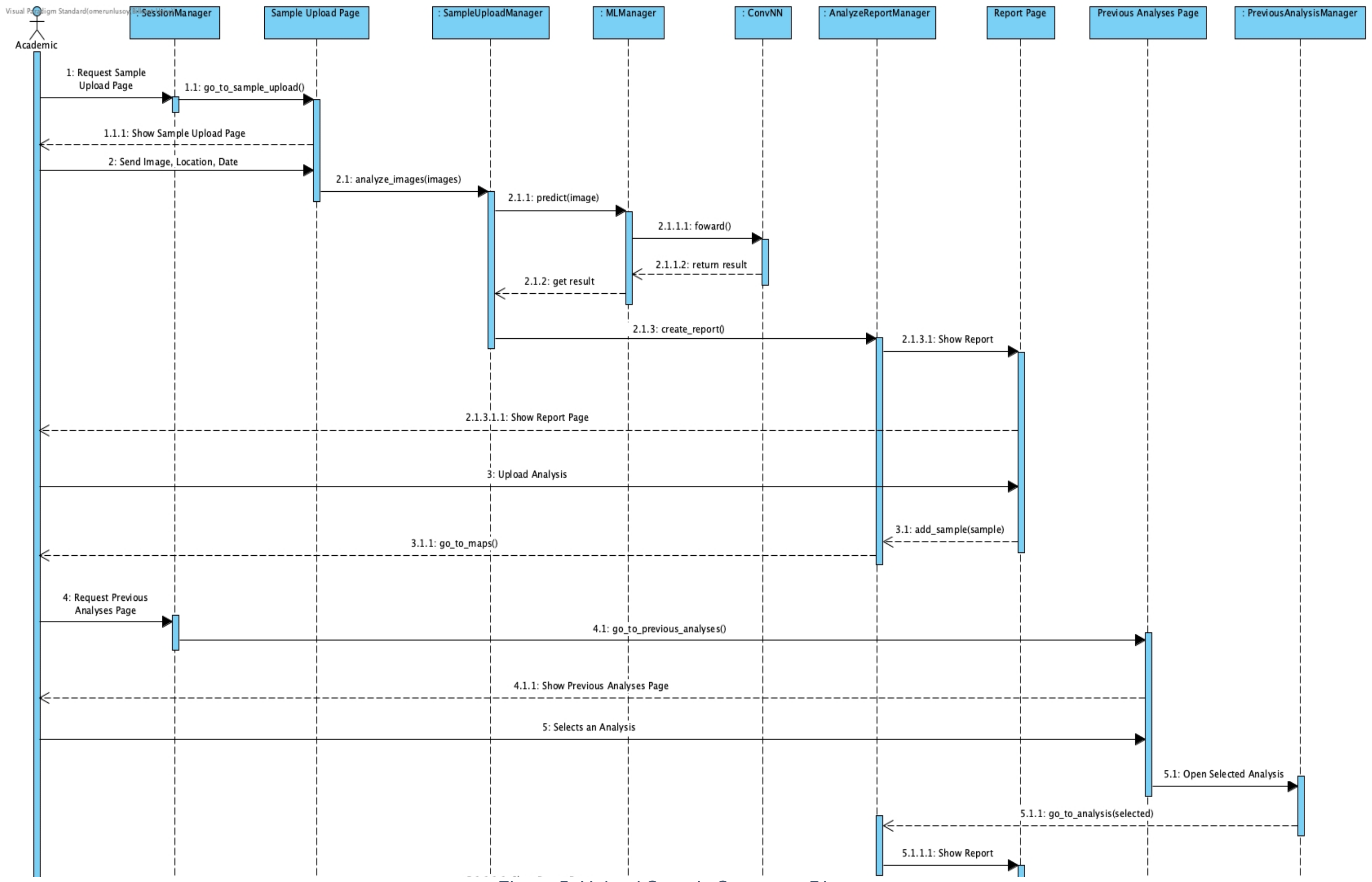

Visual Paradigm Standard(omerunlusoy(Bilkent Univ.))
Academic
: SessionManager
Sample Upload Page
: SampleUploadManager
: MLManager
: ConvNN
: AnalyzeReportManager
Report Page
Previous Analyses Page
: PreviousAnalysisManager
1: Request Sample Upload Page
1.1: go_to_sample_upload()
1.1.1: Show Sample Upload Page
2: Send Image, Location, Date
2.1: analyze_images(images)
2.1.1: predict(image)
2.1.1.1: foward()
2.1.1.2: return result
2.1.2: get result
2.1.3: create_report()
2.1.3.1: Show Report
2.1.3.1.1: Show Report Page
3: Upload Analysis
3.1: add_sample(sample)
3.1.1: go_to_maps()
4: Request Previous Analyses Page
4.1: go_to_previous_analyses()
4.1.1: Show Previous Analyses Page
5: Selects an Analysis
5.1: Open Selected Analysis
5.1.1: go_to_analysis(selected)
5.1.1.1: Show Report

Figure 5: Upload Sample Sequence Diagram

Scenario: An academic user uploads an image sample to get the analysis and then uploads this sample along with the analysis to the database. Then the user looks upon their previous analyses and see the report of one of their previously uploaded analyses.

Firstly, the academic user requests to enter Sample Upload Page, which is processed by the SessionManager. Followingly, SessionManager redirects the user to the desired age via go_to_sample_upload() method. After the Sample Upload Page is shown, user uploads an image along with the location and date of the upload to be processed. Sample Upload Page invokes analyze_image(images) of SampleUploadManager. SampleUploadManager then calls predict(image) of MLManager class, which internally uses the forward() function of ConvNN, the convolutional neural network model. Afterwards, the results are returned back to SampleUploadManager. Then, SampleUploadManager by create_report() function invokes AnalyzeReportManager. During this process, academic user is redirected to Report Page where the resulted report of the uploaded sample is shown. Followingly, the academic user requests this report be uploaded to database, and hence the Report Page calls add_sample(sample) function of AnalyzeReportManager, which uploads the sample to the database. Meanwhile, the user is redirected back to the Map Page.

After a while, the user requests to see their previously uploaded analyses. This request is processed by the SessionManager, which redirects the user to Previous Analyses Page by using go_to_previous_analyses() function. The Previous Analyses Page is shown to the user, where the user able to scroll down and search through their uploaded samples. Then, the user clicks on one of the sample analyses. This request is processed by PreviousAnalysisManager, which uses show_report_function() of AnalyzeReportManager to get the report details of the specific sample. Then the user is redirected to Report Page again.

7.3.1.5 Pinpoint Sequence

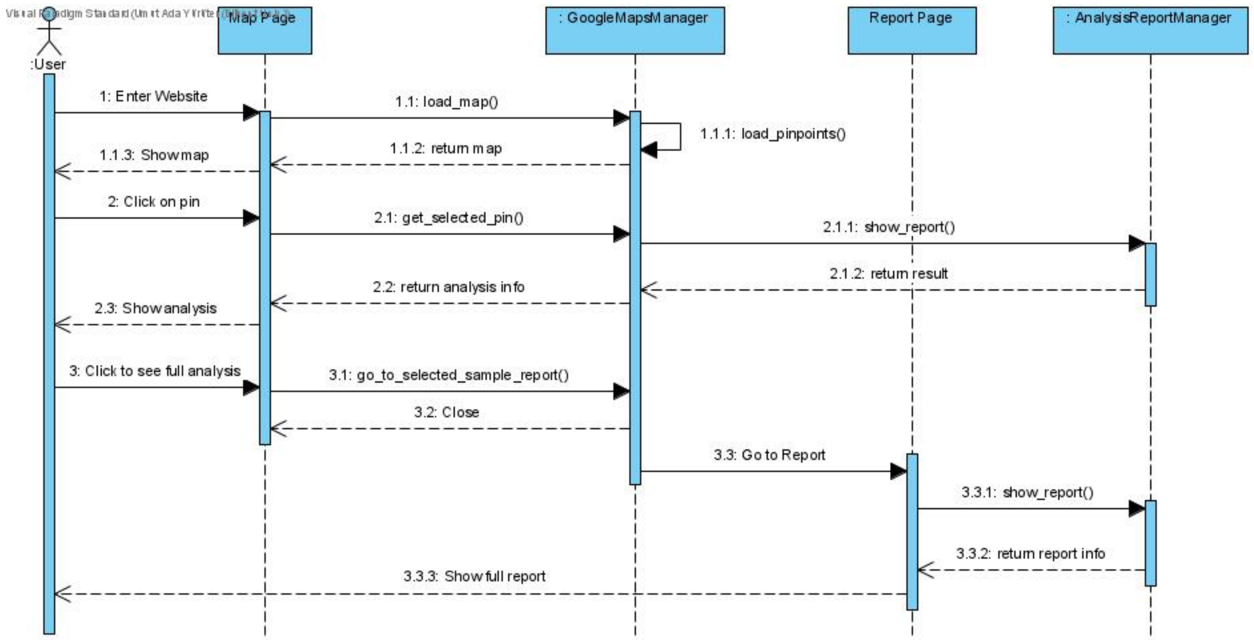


Figure 6: Pinpoint Sequence Diagram

Scenario: A user clicks on a pinpoint in map and then proceeds to see the full report.

Firstly, user enters the website and is redirected to the main page, the Map Page. Then, Map Page invokes the load_map() function of GoogleMapsManager which also calls load_pinpoints() method internally. Next, the loaded map and the pinpoints are presented to the user. Afterwards, user clicks on one of the pinpoints on the map corresponding to a sample upload from an academic user. Map Page calls the get_selected_pin() method of GoogleMapsManager, which itself calls the show_report() of AnalysisReportManager to get the proper analysis report of the relevant sample. Followingly, a part of the sample report is represented to the user in the sidebar of the Map Page. User wants to see more information, and requests to see the full report. Map Page invokes go_to_selected_sample_report() function of GoogleMapsManager, which closes the Map Page and redirects user to the Report Page of the sample in focus. Report Page again uses show_report() method of the AnalysisReportManager to get the report information, but now presents the whole result to the user.

7.3.2 Activity Diagrams

7.3.2.1 Map Usage Activity Diagram

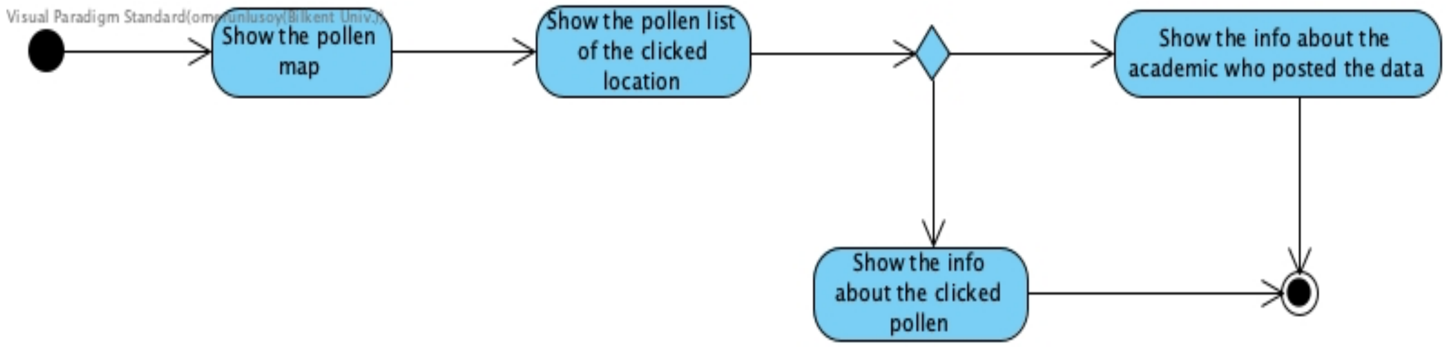

Visual Paradigm Standard(omerunlusoy(Bilkent Univ.))
Show the pollen map
Show the pollen list of the clicked location
Show the info about the academic who posted the data
Show the info about the clicked pollen

Figure 7: Map Usage Activity Diagram

This is the activity diagram of pollen map usage. The pollen map is shown to the user and the user can click at locations. When the user clicks to a location a pollen list for that location is shown. If user clicks on a pollen in the list, pollen information is shown and also information about the academic who posted the data can be seen.

7.3.2.1 Pollen Sample Analysis Activity Diagram

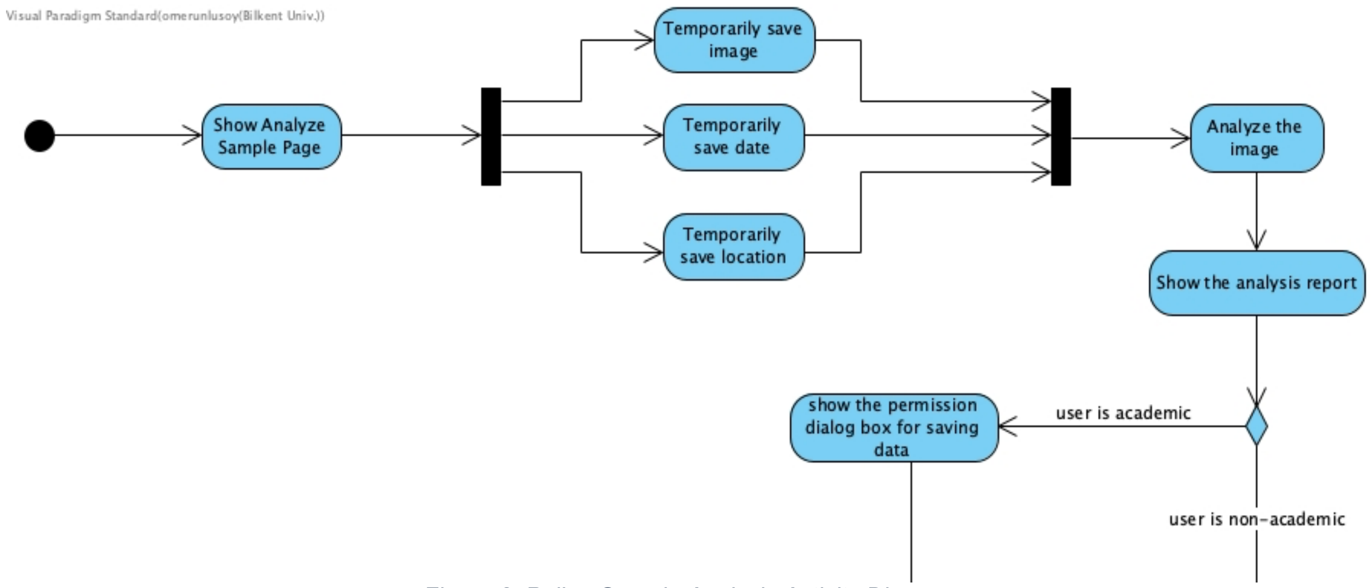


Figure 8: Pollen Sample Analysis Activity Diagram

This is the activity diagram of pollen sample analysis. The application shows the Analyze Sample page then saves (temporarily) the uploaded image, the date the image is uploaded and the location which pollen data in the image is collected. After this the image is analyzed by the system and an analysis report is shown to the user. If the user is logged in as academic the system asks the user if he/she gives permission to publish the data (let data be used to update the pollen map).

7.3.3 State Diagram

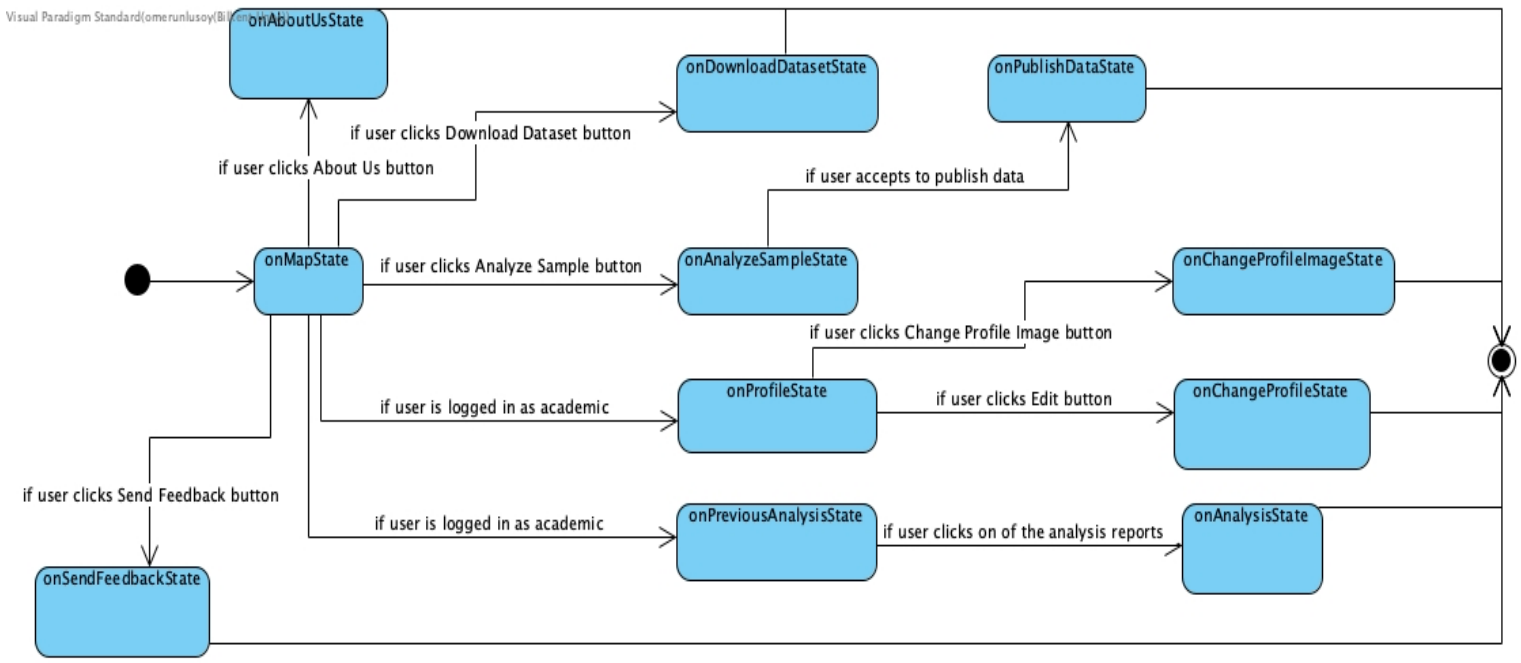
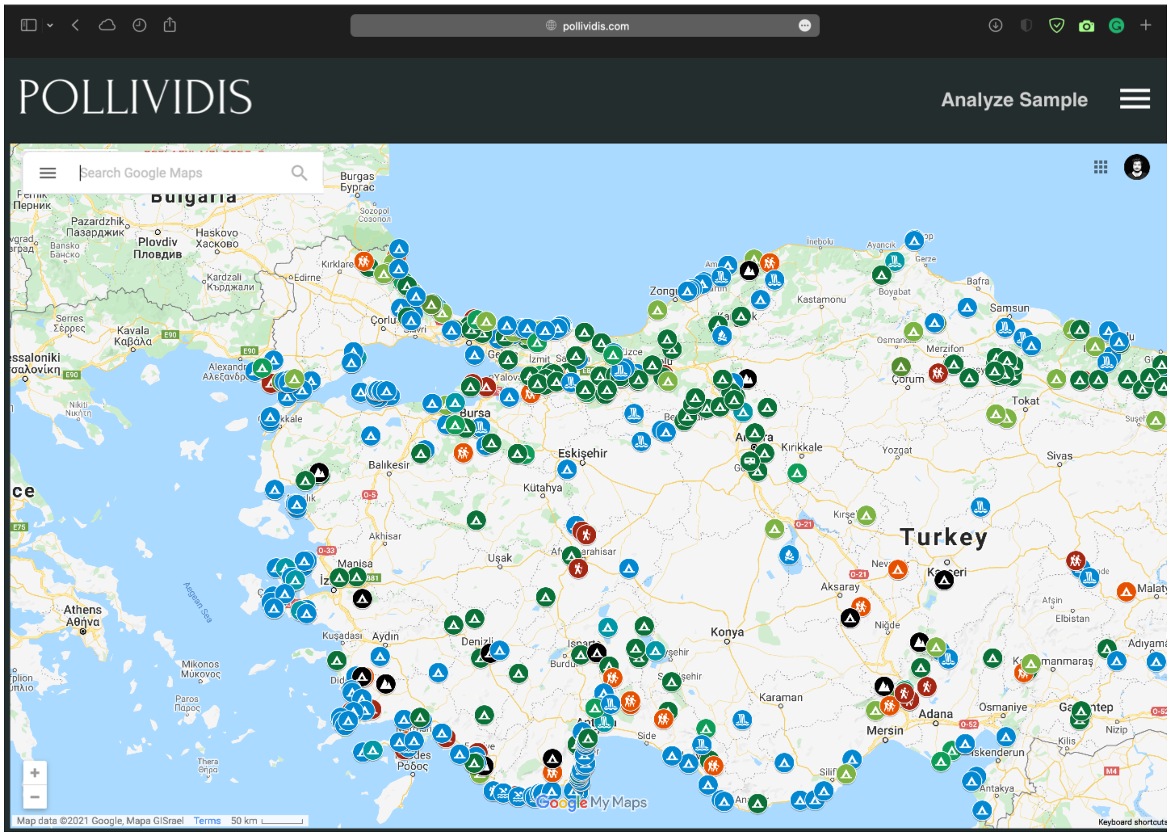

Visual Paradigm Standard(omerunlusoy(Bilkent Univ.))
onAboutUsState
onDownloadDatasetState
onPublishDataState
if user clicks Download Dataset button
if user clicks About Us button
if user accepts to publish data
onMapState
if user clicks Analyze Sample button
onAnalyzeSampleState
onChangeProfileImageState
if user clicks Change Profile Image button
onProfileState
if user clicks Edit button
onChangeProfileState
if user is logged in as academic
if user clicks Send Feedback button
if user is logged in as academic
onPreviousAnalysisState
if user clicks on of the analysis reports
onAnalysisState
onSendFeedbackState

Figure 9: State Diagram

This is the state diagram of the general web application which helps to understand the navigation. The user first sees the pollen map. If he/she clicks the Send Feedback button, Send Feedback page is shown and the user can write and send feedback about the application. If user clicks About Us button, About Us page is shown which includes information about the contributors of this project. User can analyze pollen images by clicking the Analyze Sample button. User can download the pollen dataset that we will create. If the user is logged in as academic, he/she can view his/her profile and look at his/her previous analysis and see the analysis reports. From the profile page the user can change profile image and other profile information. If the user is academic, after the simple pollen analysis the system will ask him/her if he/she wants to publish the analyzed data to contribute to science.

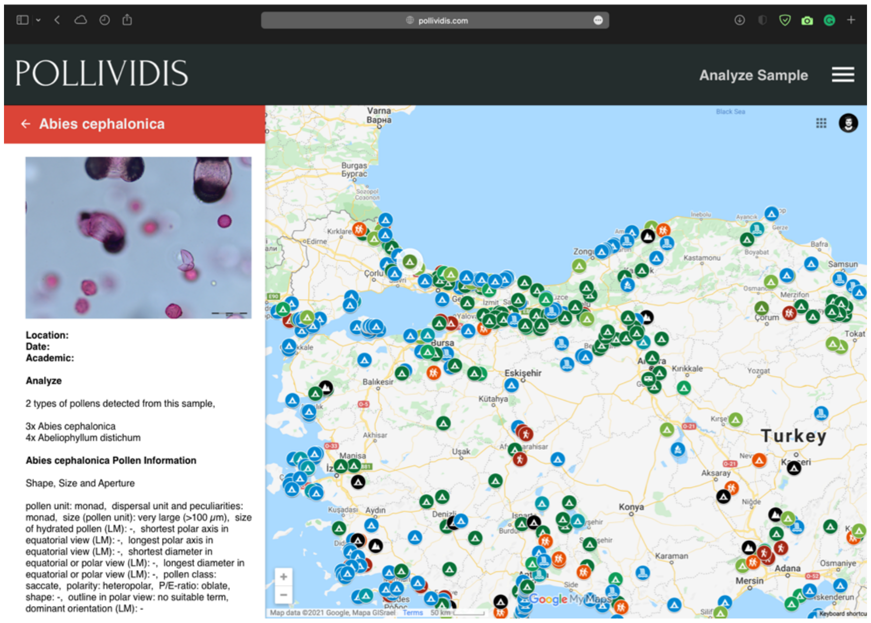
7.4 User Interface

7.4.1 Map Page



Map page of PolliVidis is the first page a user sees. As PolliVidis supports anonymous usage for viewing analyses and analyzing samples, it welcomes users with its Google Maps integrated map page rather than a login screen. Each pinpoint on the map represents a sample analysis and any anonymous user can view them by clicking on them. The map supports basic functionalities of Google Maps such as zoom in and zoom out. A user can go to the Analyze Sample (Upload) Page by just clicking on “Analyze Sample” button. Moreover, the options panel can be opened with “options button” (three lines). Finally, a user can use all PolliVidis search functionalities with the search bar.

7.4.2 Map Page with Sample Info Panel



Location:
Date:
Academic:

Analyze

2 types of pollens detected from this sample,

3x Abies cephalonica
4x Abeliophyllum distichum

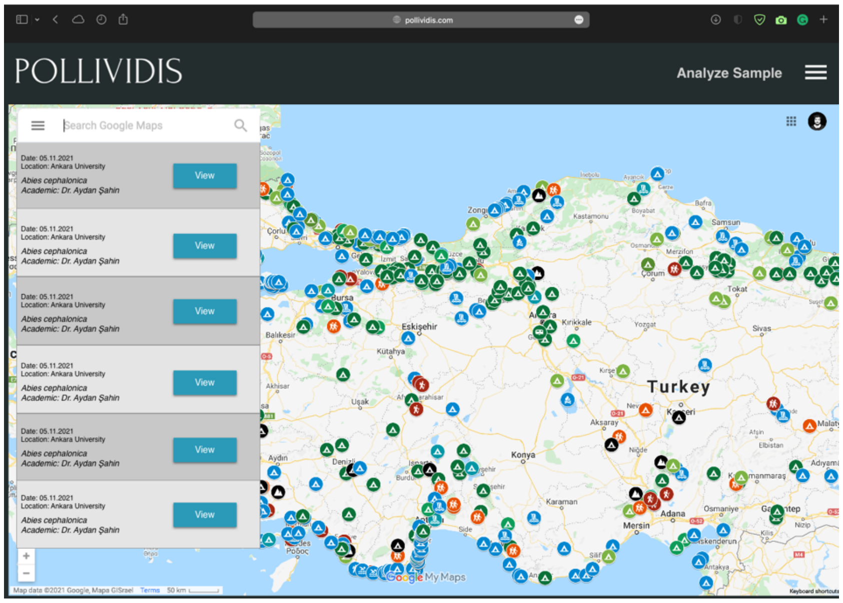
Abies cephalonica Pollen Information

Shape, Size and Aperture

pollen unit: monad, dispersal unit and peculiarities: monad, size (pollen unit): very large (>100 μm), size of hydrated pollen (LM): -, shortest polar axis in equatorial view (LM): -, longest polar axis in equatorial view (LM): -, shortest diameter in equatorial or polar view (LM): -, longest diameter in equatorial or polar view (LM): -, pollen class: saccate, polarity: heteropolar, P/E-ratio: oblate, shape: -, outline in polar view: no suitable term, dominant orientation (LM): -

When a user clicks on a sample (pinpoint) on the map, Sample Info Panel opens at the right side. This panel shows the sample image with its analysis report. This panel allows users to scan the map fast by switching between samples. If the user wants to see the detailed analysis report, s/he can go to the sample's analysis report by clicking on the sample image.

7.4.3 Search Page



Date: 05.11.2021
Location: Ankara University
Abies cephalonica
Academic: Dr. Aydan Şahin

Date: 05.11.2021
Location: Ankara University
Abies cephalonica
Academic: Dr. Aydan Şahin

Date: 05.11.2021
Location: Ankara University
Abies cephalonica
Academic: Dr. Aydan Şahin

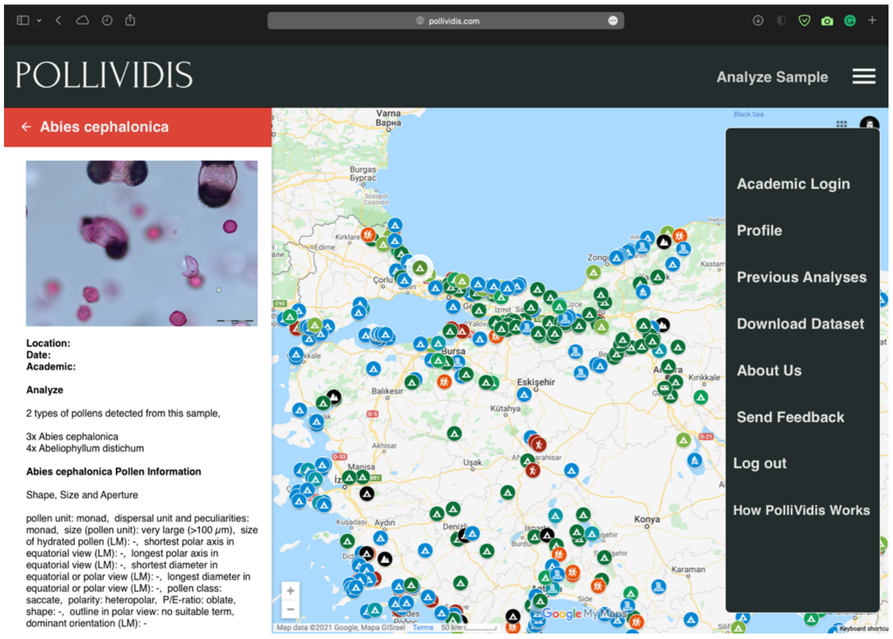
Date: 05.11.2021
Location: Ankara University
Abies cephalonica
Academic: Dr. Aydan Şahin

Date: 05.11.2021
Location: Ankara University
Abies cephalonica
Academic: Dr. Aydan Şahin

Date: 05.11.2021
Location: Ankara University
Abies cephalonica
Academic: Dr. Aydan Şahin

Search bar at the top right corner of the Map Page allows users to use all PolliVidis search functionalities. These search functionalities include searching an academic (for academic accounts only), a location (such as city or building), or a pollen species. These search functionalities relieve user from wandering on the map.

7.4.4 Options Panel Page



Location:
Date:
Academic:

Analyze

2 types of pollens detected from this sample,

3x Abies cephalonica
4x Abeliophyllum distichum

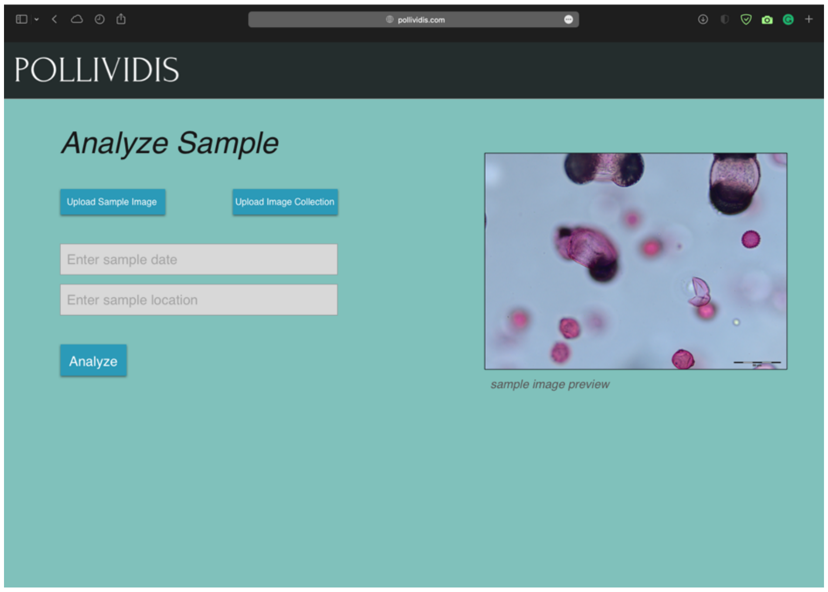
Abies cephalonica Pollen Information

Shape, Size and Aperture

pollen unit: monad, dispersal unit and peculiarities: monad, size (pollen unit): very large (>100 μm), size of hydrated pollen (LM): -, shortest polar axis in equatorial view (LM): -, longest polar axis in equatorial view (LM): -, shortest diameter in equatorial or polar view (LM): -, longest diameter in equatorial or polar view (LM): -, pollen class: saccate, polarity: heteropolar, P/E-ratio: oblate, shape: -, outline in polar view: no suitable term, dominant orientation (LM): -

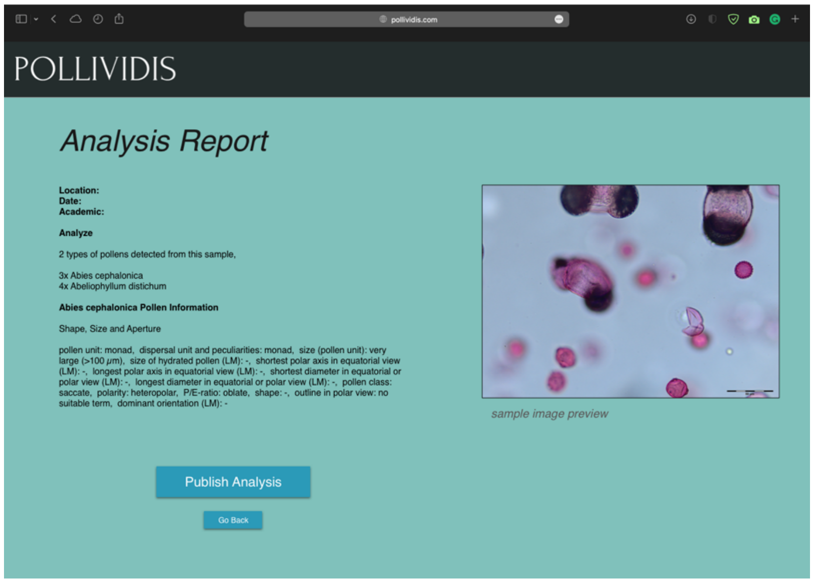
Options panel allows users to login, sign-up, go to their profiles (academic users only), view their previous analyses, send feedback to us, log out, and download the dataset. Although, different pages and users will see different options in the panel, we have included all the options in the mockup to avoid making several mockups just for this page.

7.4.5 Analyze Sample (Upload) Page



Analyzing a sample with PolliVidis is available for all users and this page can be accessed directly from the Map Page. A user can upload one sample image to analyze or image collection for collective analysis. The page asks academic users for the date and the location of the sample to upload the sample analysis to the database if they allow it.

7.4.6 Analysis Report Page



Location:
Date:
Academic:

Analyze

2 types of pollens detected from this sample,

3x Abies cephalonica
4x Abeliophyllum distichum

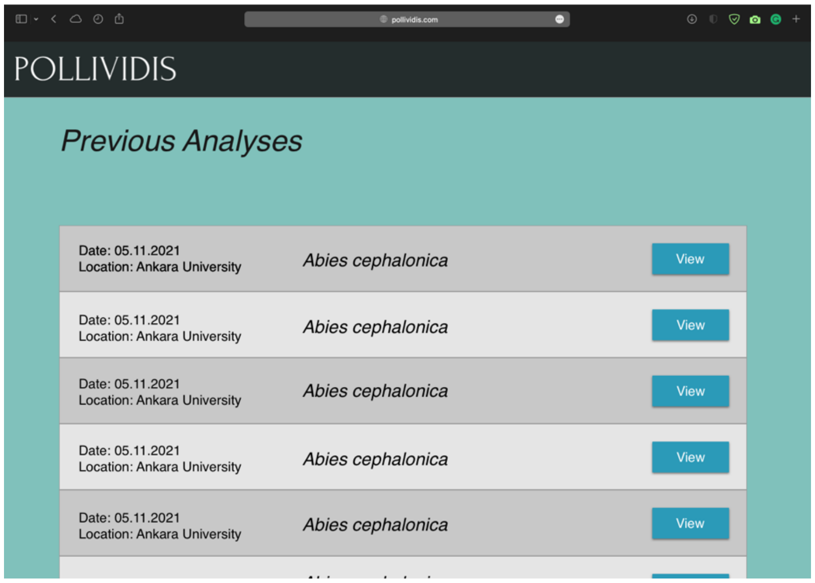
Abies cephalonica Pollen Information

Shape, Size and Aperture

pollen unit: monad, dispersal unit and peculiarities: monad, size (pollen unit): very large (>100 μm), size of hydrated pollen (LM): -, shortest polar axis in equatorial view (LM): -, longest polar axis in equatorial view (LM): -, shortest diameter in equatorial or polar view (LM): -, longest diameter in equatorial or polar view (LM): -, pollen class: saccate, polarity: heteropolar, P/E-ratio: oblate, shape: -, outline in polar view: no suitable term, dominant orientation (LM): -

After uploading a sample image and clicking on the “analyze button”, all users are redirected to the Analysis Report Page to view the detailed analysis of their samples. This report includes the location, date, academic information, pollen species, and detailed analysis of each species of the given sample. If the user analyzing the sample is an academic, this page asks him/her for a permission to upload the analysis to the database. Moreover, any user will be directed to this page when they want to view the detailed analysis of any sample.

7.4.7 Previous Analyses Page

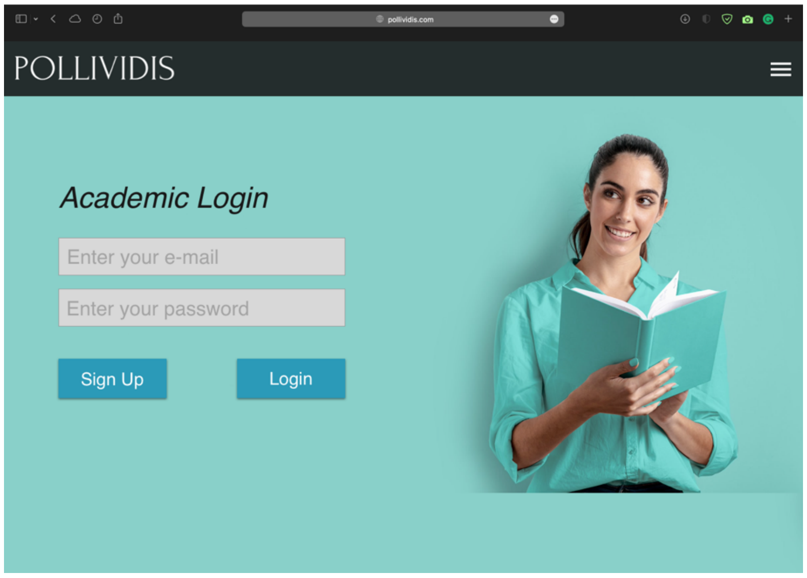


Previous Analyses

Date: 05.11.2021 Location: Ankara University	*Abies cephalonica*	View
Date: 05.11.2021 Location: Ankara University	*Abies cephalonica*	View
Date: 05.11.2021 Location: Ankara University	*Abies cephalonica*	View
Date: 05.11.2021 Location: Ankara University	*Abies cephalonica*	View
Date: 05.11.2021 Location: Ankara University	*Abies cephalonica*	View

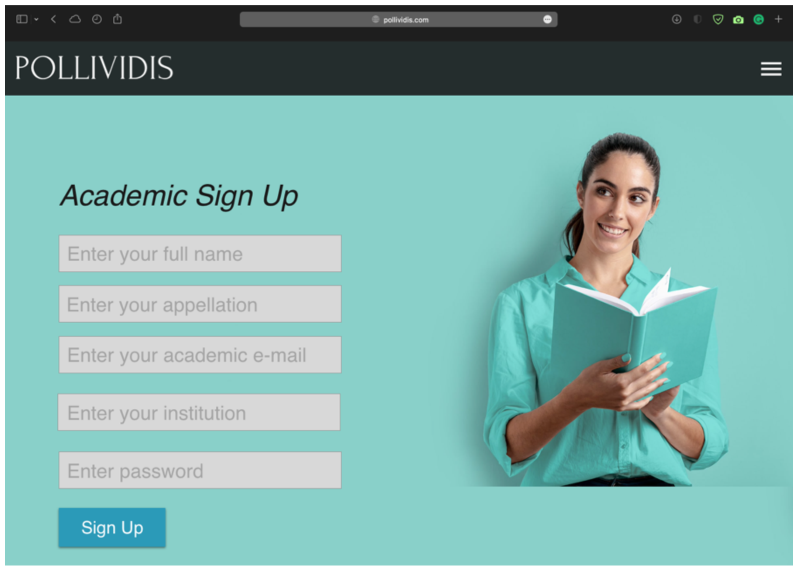
Previous Analyses Page lists all the previous samples and their analyses of the academic. Since we will not store any information for anonymous sample analyses, this page can only be used by academic accounts. However, anyone can view previous analyses of any academic with their Previous Analyses Page if the academic allows it.

7.4.8 Academic Login Page



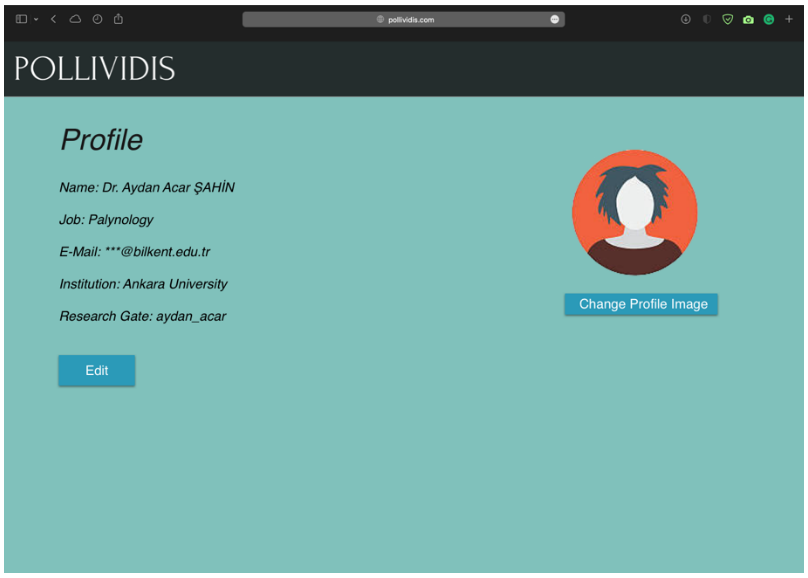
Academic Login Page allows academics to login their accounts for further advantages of an academic account such as Previous Analyses Page and detailed communication information of other academics. Any academic without an account can go the Sign-Up Page using “Sign Up button”.

7.4.9 Academic Sign-Up Page



Any academic can create a PolliVidis account by supplying required information such as appellation, e-mail, and institution to benefit from further advantages of an academic account. A verification procedure will be used to ensure the reliability of the supplied information.

7.4.10 Academic Profile Page



Name: Dr. Aydan Acar ŞAHİN

Job: Palynology

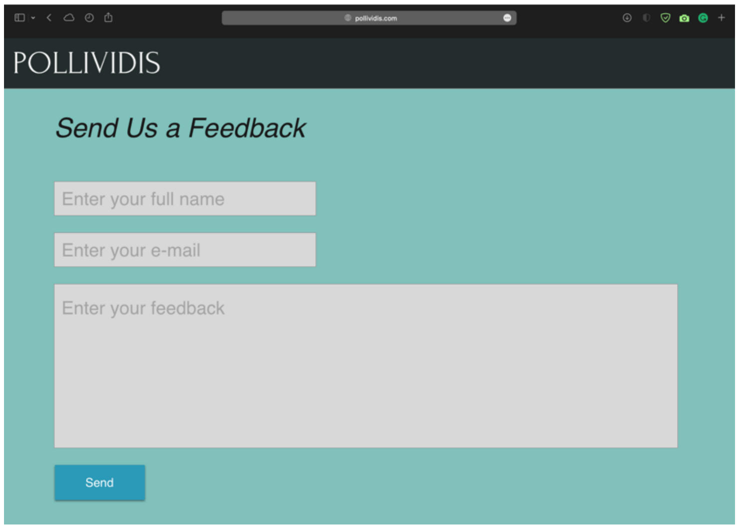
E-Mail: ***@bilkent.edu.tr

Institution: Ankara University

Research Gate: aydan_acar

Each academic account can view their profile and edit the supplied information using Academic Profile Page. This page also includes the communication information and a photo of the academic.

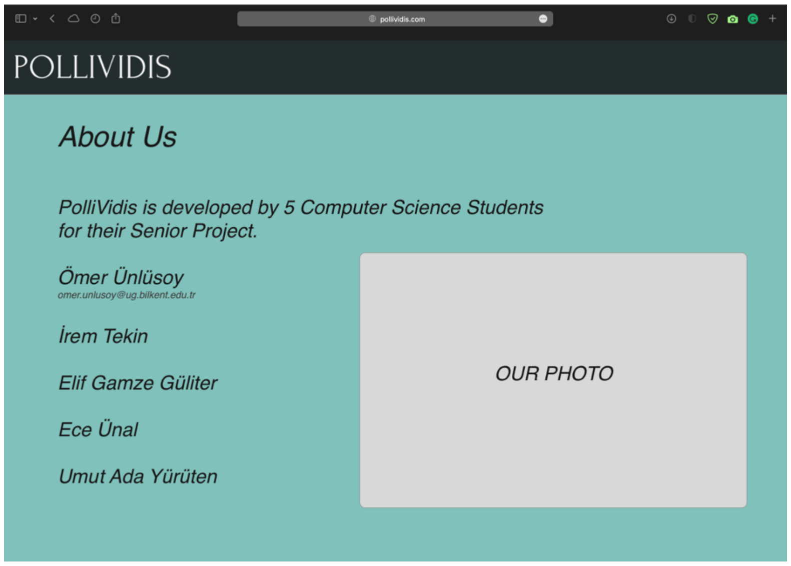
7.4.11 Send Feedback Page


POLLIVIDIS

Send Us a Feedback

Any user can send us feedback using Send Feedback Page. Full name and an e-mail account should be supplied so that we can respond.

7.4.12 About Us Page


POLLIVIDIS

About Us

PolliVidis is developed by 5 Computer Science Students for their Senior Project.

Ömer Ünlüsoy
omer.unlusoy@ug.bilkent.edu.tr

İrem Tekin

Elif Gamze Güliter

Ece Ünal

Umut Ada Yürüten

Some information about each team member with a group photo can be viewed at About Us Page.

7.4.13 How It Works Page

As an extra page, How It Works Page explains the Machine Learning model we will use and the analysis procedure for curious users. Since the page will just include some explanation about the model and a representative image of the model's layer which neither of them ready, we did not design a mockup for this page.

8. Other Analysis Elements

8.1 Consideration of Various Factors in Engineering Design

8.1.1 Public Health

Effect Level: 8
Since PolliVidis gives pollen predictions, directly related to the 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 advices.

8.1.2 Public Safety

Effect Level: 3
Only concern with the public safely 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 want to keep his/her communication information private, s/he can do so.

8.1.3 Public Welfare

Effect Level: 8
PolliVidis wants to increase user's life quality by showing the allergenic pollen information at his/her area so that user can learn the pollen levels and take precautions. Moreover, academics can share sample analyses quickly which may increase collaboration in academic researches.

8.1.4 Global Factors

Effect Level: 3

Although, PolliVidis can be opened worldwide; some adaptations and regulations will be needed as different countries have different allergenic pollen types and single model cannot be trained to classify all of them. However, analysis sharing between academics would help any academic worldwide.

8.1.5 Cultural Factors

Effect Level: 2

Since PolliVidis is related mostly about academic research and scientific information sharing, expected cultural problems or affects are minimal.

8.1.6 Social Factors

Effect Level: 4

Sample and analysis upload to the pollen map will be allowed for academic staff only. Thus, anyone will not be able to upload random images to the map 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.

8.1.7 Environmental Factors

Effect Level: 2

It is not expected that the usage of PolliVidis website causes 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, generated dataset will be published and eliminate the need of collecting the same samples again and again for future researches which saves human and electrical energy.

8.1.8 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.

8.2 Risks and Alternatives

- **Incorrect pollen classification**
  - CNN might mis-classify the taxonomy of the sample uploaded to the system. As all models, our model will have an accuracy rate which means that it might classify some samples incorrectly.
  - **Likelihood**: It will depend on our implementation, expected likelihood is less than %5.
  - **Alternative:** If the user is a student, he/she can double check the data by asking an expert. If the user is an academic, he/she can possibly tell the output is wrong. These types of mis-classifications can be written as a feedback.

- **High running time for the image classification**
  - Classification might take more than expected time
  - **Likelihood:** Moderate
  - **Alternative:** Users can use servers

- **Error rate in Pollen Map**
  - Allergenic pollen data uploaded by academics might not show exact percentage
  - **Likelihood:** Moderate

- **Alternative:** Users will be alarmed about possible mistakes in data

8.3 Ensuring Proper Teamwork

In order to collaborate over the code, we will use GitHub and GitHub issues. We plan to divide group work into three: collating data, building the website and building the machine learning models. However, we plan to switch teams from time to time. One will be working for both web design and ML models since it will be more fair to everyone.

We are using Google docs to write reports together. While learning and investigating algorithms, we share topics and conduct a meeting in which all of us explain what we learn and discuss. We keep important links, data we collected in our GitHub repository and Google Drive.

8.4 Ethics and Professional Responsibilities

PolliVidis aims to provide allergenic pollen classification with the best accuracy. Furthermore, it aims to provide information about the allergenic pollen density to its users therefore the data it provides to its users should not be biased. For registered users, it should ensure the privacy of their personal information and fit the KVKK in Turkey. It should not share or process any personal data without the user's consent [7]. For further improvements, privacy policies might be extended to fit into global regulations. For unregistered users, it should not save or require any usage of personal information.

8.5 Planning for New Knowledge and Learning Strategies

There is only one member in our team who is experienced with Deep Learning, who is Ömer. Therefore, he leads us in the learning process. We watch YouTube tutorials and Udemy Courses on Convolutional Neural Networks. For building the website, we plan to use YouTube tutorials as well. Furthermore, we read academic papers on previous examples about pollen classification.

8.6 Project Plan

WP#	Work Package Title	Leader	Members Involved
WP1	Preparing Pollen Dataset	İrem Tekin	Ömer Ünlüsoy
WP2	Preparing the Pollen Extraction Algorithm	Umut Ada Yürüten	İrem Tekin
WP3	Transfer Learning Process	Ömer Ünlüsoy	Elif Gamze Güliter
WP4	Data Augmentation Process	Elif Gamze Güliter	İrem Tekin Ece Ünal Ömer Ünlüsoy
WP5	Preparing the First CNN Model	Ömer Ünlüsoy	Umut Ada Yürüten Ece Ünal İrem Tekin Elif Gamze Güliter
WP6	Backend of the Pollividis Website	Ece Ünal	İrem Tekin Ömer Ünlüsoy
WP7	Finalizing the CNN Model	Ömer Ünlüsoy	Ece Ünal Elif Gamze Güliter Umut Ada Yürüten İrem Tekin
WP8	Front-end Implementation and Integration with Google Maps API	İrem Tekin	Ece Ünal Umut Ada Yürüten

WP1: Preparing Pollen Dataset			
Start Date: 4 October 2021		**End Date:** 24 December 2021	
Leader:	İrem	Members Involved:	Ömer
Objectives: 30-35 pollen species will be collected. 200-500 photographs will be taken for each species. It will serve for both our model and feature researches.			
Tasks: **Task 1.1:** Photograph each species at Ankara University Palynology Laboratory.			
Deliverables: **D 1.1:** Dataset consisting of 30-35 pollen species with 200-500 photos			

WP2: Preparing the Pollen Extraction Algorithm			
Start Date: 25 October 2021		**End Date:** 25 November 2021	
Leader:	Umut Ada	Members Involved:	İrem
Objectives: To implement a pollen extraction algorithm to extract each pollen photo separately from a given sample.			
Tasks: **Task 2.1:** Search the existing extraction algorithms. **Task 2.2:** Implement a pollen extraction algorithm in Python. **Task 2.3:** Implement a main class for pre-training process which will take samples from file directories, prepare them for the model, and call the model's predict function.			
Deliverables: **D 2.1:** Pollen extraction algorithm to extract pollens from samples before CNN **D 2.2:** A main class for pre-training process			

WP3: Transfer Learning Process			
Start Date: 25 October 2021		**End Date:** 14 March 2022	
Leader:	Ömer	Members Involved:	Elif Gamze
Objectives: Using a pre-trained model to increase our model's accuracy due to the shortage of our dataset.			
Tasks: **Task 3.1:** Search available pre-trained models (like VGG-19), and decide the one to use in our model. **Task 3.2:** Implement the first version of transfer learning for our model in PyTorch. **Task 3.3:** Run several tests to evaluate the compatibility of the pre-trained model. **Task 3.4:** Adjust the hyperparameters of the model to increase accuracy. **Task 3.5:** Finalize the pre-trained model with small adjustments.			
Deliverables: **D 3.1:** Pre-trained model with hyperparameter adjustment			

WP4: Data Augmentation Process			
Start Date: 25 October 2021		**End Date:** 27 November 2021	
Leader:	Elif Gamze	Members Involved:	İrem, Ece, Ömer
Objectives: Finding the applicable Data Augmentation methods for our model to increase the dataset size and decrease overfitting.			
Tasks: **Task 4.1:** Search the Data Augmentation methods and decide which ones are suitable for our model and dataset. **Task 4.2:** Implement the data augmentation (transformation) in PyTorch.			
Deliverables: **D 4.1:** Data Augmentation (transformation in PyTorch)			

WP5: Preparing the First CNN Model			
Start Date: 15 November 2021		**End Date:** 15 December 2021	
Leader:	Ömer	Members Involved:	Umut Ada, Ece, İrem, Elif Gamze
Objectives: Implement the first version of Pollen Classifier CNN model.			
Tasks: **Task 5.1:** Each team member will learn the neural network and PyTorch basics. **Task 5.2:** Search similar CNN model implementations to get an idea about the required structure of the model. **Task 5.3:** Implement the first version of convolutional, max pooling, and fully connected layers with PyTorch Sequential. **Task 5.4:** Implement Trainer class. **Task 5.5:** Implement Tester class.			
Deliverables: D 5.1: CNN class **D 5.2:** Trainer class **D 5.3:** Tester class			

WP6: Backend of the Pollividis Website			
Start Date: 15 October 2021		**End Date:** 14 March 2022	
Leader:	Ece	Members Involved:	İrem, Ömer
Objectives: Implement the project backend for database, ML model, and UI connections in Python Django.			
Tasks: **Task 6.1:** Each involved team member will learn the basics of Python Django framework which we will use to implement PolliVidis website backend. **Task 6.2:** Implement the database with SQL. **Task 6.3:** Connect the Django backend of the project with the initialized database. **Task 6.4:** Implement query functions which will upload and request samples from the datasets. **Task 6.5:** Connect the backend with the ML model. **Task 6.6:** Implement the function that will produce the analysis report using ML model classification. **Task 6.7:** Connect the backend with the user interface of PolliVidis.			

Deliverables: **D 6.1:** Database Implementation with SQL **D 6.2:** PolliVidis Backend

WP7: Finalizing the CNN Model			
Start Date: 15 December 2021		**End Date:** 10 April 2022	
Leader:	Ömer	Members Involved:	Ece, Elif Gamze, Umut Ada, İrem
Objectives: To finalize the implemented CNN model with proper tests, hyperparameter adjustments, and several optimizations.			
Tasks: **Task 7.1:** Implement different versions of the first CNN with different pre-trained models. **Task 7.2:** For each implementation, run several tests and adjust hyperparameters to increment the evaluation matrices including accuracy. **Task 7.3:** Compare each implementation and decide the final structure of the model.			
Deliverables: **D 7.1:** CNN model			

WP8: Front-end Implementation and Integration with Google Maps API			
Start Date: 15 November 2021		**End Date:** 10 April 2022	
Leader:	İrem	Members Involved:	Ece, Umut Ada
Objectives: To implement the user interface of PolliVidis.			
Tasks: **Task 8.1:** Each involved team member will learn HTML, React basics. **Task 8.2:** Implementation of each mockup. **Task 8.3:** Integration of Google Maps API. **Task 8.4:** Final adjustments between user interface, backend, and database.			
Deliverables: **D 8.1:** User Interface of each PolliVidis page			

9. References

[1] 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.

[2] 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.

[3] AutPal, “Palynological Database,” *PalDat*. [Online]. Available: https://www.paldat.org/. [Accessed: 14-Nov-2021].

[4] İ. Tekin, Ö. Ünlüsoy, and A. Acar, “Ankara University and Pollen Project,” 24-Sep-2021.

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

[6] “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].

[7] “Kişisel Verileri Koruma Kurumu: KVKK: Kişisel Verileri Koruma Kurumu Başkanlığı,” *Kvkk*. [Online]. Available: https://www.kvkk.gov.tr/. [Accessed: 10-Nov-2021].