



VILNIUS UNIVERSITY  
FACULTY OF MATHEMATICS AND INFORMATICS  
INSTITUTE OF COMPUTER SCIENCE  
INFORMATION TECHNOLOGIES STUDY PROGRAM

Problem-Based Project

**Lost animal platform**

Done by:  
Edvinas Gerdvila  
Renaldas Narbutas  
Arsenij Nikulin

Supervisor:  
dr. Agne Brilingaitė

Vilnius  
2024

# Contents

<b>Abstract</b>	<b>3</b>
<b>Santrauka</b>	<b>4</b>
<b>Introduction</b>	<b>5</b>
<b>1 Related Works</b>	<b>6</b>
<b>2 Analysis</b>	<b>6</b>
2.1 Color histograms . . . . .	6
2.2 Texture histograms . . . . .	6
2.3 Edge detection . . . . .	7
2.4 Structural similarity index measure . . . . .	7
<b>3 Design</b>	<b>7</b>
3.1 Use cases . . . . .	7
3.2 System requirements . . . . .	8
3.2.1 Functional requirements . . . . .	8
3.2.2 Non-functional requirements . . . . .	10
3.3 Architecture . . . . .	10
3.4 Database . . . . .	14
3.5 Authentication . . . . .	15
3.6 Real world example . . . . .	15
<b>4 Implementation</b>	<b>16</b>
4.1 Tools . . . . .	16
4.2 Database . . . . .	16
4.3 Map . . . . .	17
4.4 Text processing . . . . .	17
4.5 Automating installation . . . . .	17
<b>5 Testing</b>	<b>17</b>
<b>Conclusions and Future Work</b>	<b>23</b>
<b>Appendices</b>	<b>24</b>

## Abstract

Abstract is in the main report language Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

# **Santrauka**

## **Darbo pavadinimas kita kalba**

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

# **Introduction**

# 1 Related Works

## 2 Analysis

We have looked into many different ways we could use our user provided data to find the best match for the pets.

### 2.1 Color histograms

Color histograms provide an approach to summarize the color distribution of an image, which helps in distinguishing and comparing the visual characteristics of different pets.

A color histogram represents the frequency of occurrence of various color values in an image. For the purpose of our platform, we generate histograms for the RGB (Red, Green, Blue) color model. Each image uploaded to the platform is processed to create a three-dimensional histogram. This process involves:

- Converting the image from its original color space to a normalized RGB color space to mitigate variations in lighting and exposure.
- Dividing each color channel into a predefined number of bins to capture the frequency of each color intensity.
- Flattening the three-dimensional histogram into a one-dimensional vector for easier comparison. This vector contains the concatenated histograms of all three color channels.

To determine the similarity between two images, we employ the Bhattacharyya distance, which measures the overlap between two statistical samples.

A smaller Bhattacharyya distance indicates a higher degree of similarity between the two histograms.

### 2.2 Texture histograms

It matches images based on textural features, such as fur patterns and unique markings. Texture histograms summarize the distribution of local texture patterns in an image, which can distinguish pets even in images where color information alone is insufficient.

From the extracted texture features, we construct texture histograms. Each texture metric (contrast, correlation, etc.) contributes to a separate histogram. The histograms are normalized and combined into a single texture histogram that represents the overall textural signature of the image.

For comparing the texture histograms of different images, we use methods such as the Chi-square statistic.

A lower Chi-square value indicates a greater similarity between the textures of two images, facilitating more accurate matches based on textural features.

## 2.3 Edge detection

It identifies pets by their structural outlines, such as shapes of ears, tails, and fur patterns. This method focuses on detecting boundaries and edges in the images of pets.

We utilize the Canny edge detector, a popular method for edge detection, which provides high sensitivity in capturing edges with precision. The process involves several steps:

- Applying a Gaussian blur to the image to reduce noise and detail, focusing on significant transitions.
- Calculating the gradient of the image intensity at each pixel, determining the direction and intensity of edges.
- Applying non-maximum suppression to thin out the edges, ensuring that only the most pronounced edges are detected.
- Using double thresholding to distinguish between strong and weak edges, and linking them to detect complete edges.

After processing with the Canny edge detector, the result is an edge map that highlights the boundaries and outlines of objects within the image. Each detected edge is represented by white lines on a black background, clearly delineating the structural features of the pet.

To compare edge maps from different images, we employ a technique similar to template matching, where the edge map of a lost pet is used as a template to search for similar structures in the edge maps of found pets.

## 2.4 Structural similarity index measure

The SSIM is a method for measuring the similarity between two images, which is particularly effective in assessing the quality of visual structures between the images of pets.

SSIM is designed to improve on traditional methods like mean squared error, which may not correlate well with visual perception.

In our platform, SSIM is utilized to compare the uploaded image of a lost pet with the database of images of found pets. The SSIM index provides a measure from -1 to 1, where 1 indicates perfect similarity. This method allows us to detect high levels of visual similarity in terms of brightness, contrast, and structure.

# 3 Design

## 3.1 Use cases

This "Lost and found animal platform" may be used for the following purposes:

- **Searching for lost pets:** Users can use the platform to search for their missing pets by browsing through reports of lost animals, filtering based on various attributes, and get notifications if similar report is near to him.
- **Searching for the owner of pets:** People (or shelters) who have found lost pets can use the platform to search for the owners by creating reports with descriptions and images of the found animals.

- **Spreading the word:** The web platform offers several options for spreading information related to pets, such as publishing posts on social networks, automatically generating posts and notifying nearby users.
- **Volunteering:** People who not directly involved with lost or found pets can freely join in the search for lost pets to increase the chance of finding them.
- **Support and socialisation:** The platform provides a support contacts, tips related for this topic, possibility to write comments below the posts.
- **Raising reputation:**

To achieve these goals, but also to maintain order on the platform, a system was developed that demonstrates the rights and capabilities of the users 1.

There are 4 actors in the diagram. They represent the roles on the website.

1. The first actor is **Anonymous**, an unauthorised user. He has access only to view content: read articles related to the topic, view lists and maps of lost and found pet reports, and view details of the report, which includes the pet's attributes, comments, and contact details of the report's author. This functionality will help this role not to be able to modify any content on the site, but it will allow access to all the necessary information.
2. The second role is **Default user**. It inherits all the functionality of the previous role, as well as includes CRUD operations with reports. In addition, this role can leave comments under other reports. The most important difference from the previous role is the presence of a profile and the ability to customise it (change avatar, nickname, first and last name, etc.).
3. **Shelter** is a modification of the default user, including all its functionality. The peculiarity of this role is that it has a separate page, as well as a shelter icon in place of its address. This is done because shelters often have quite a large number of pets, so all reports created by the shelter are united under one icon on the site for the convenience of users and to prevent their disentanglement. Thus the shelter can manage its page.
4. **Moderator** is a role that is intended for content moderation. This role can delete incorrect reports and comments, as well as approve reports that failed to pass the automatic AI check.

## 3.2 System requirements

### 3.2.1 Functional requirements

- The system should be deployed on the VU MIF Cloud.
- Automated installation using ansible playbooks.
- The web application will be hosted on a virtual machine with the Apache2 web server.
- The account system should allow users to register, log in, manage their profiles, and recover accounts if needed.
- Users who lost or found a pet must be able to create a report and input necessary data: images, attributes based on keywords, and general location.

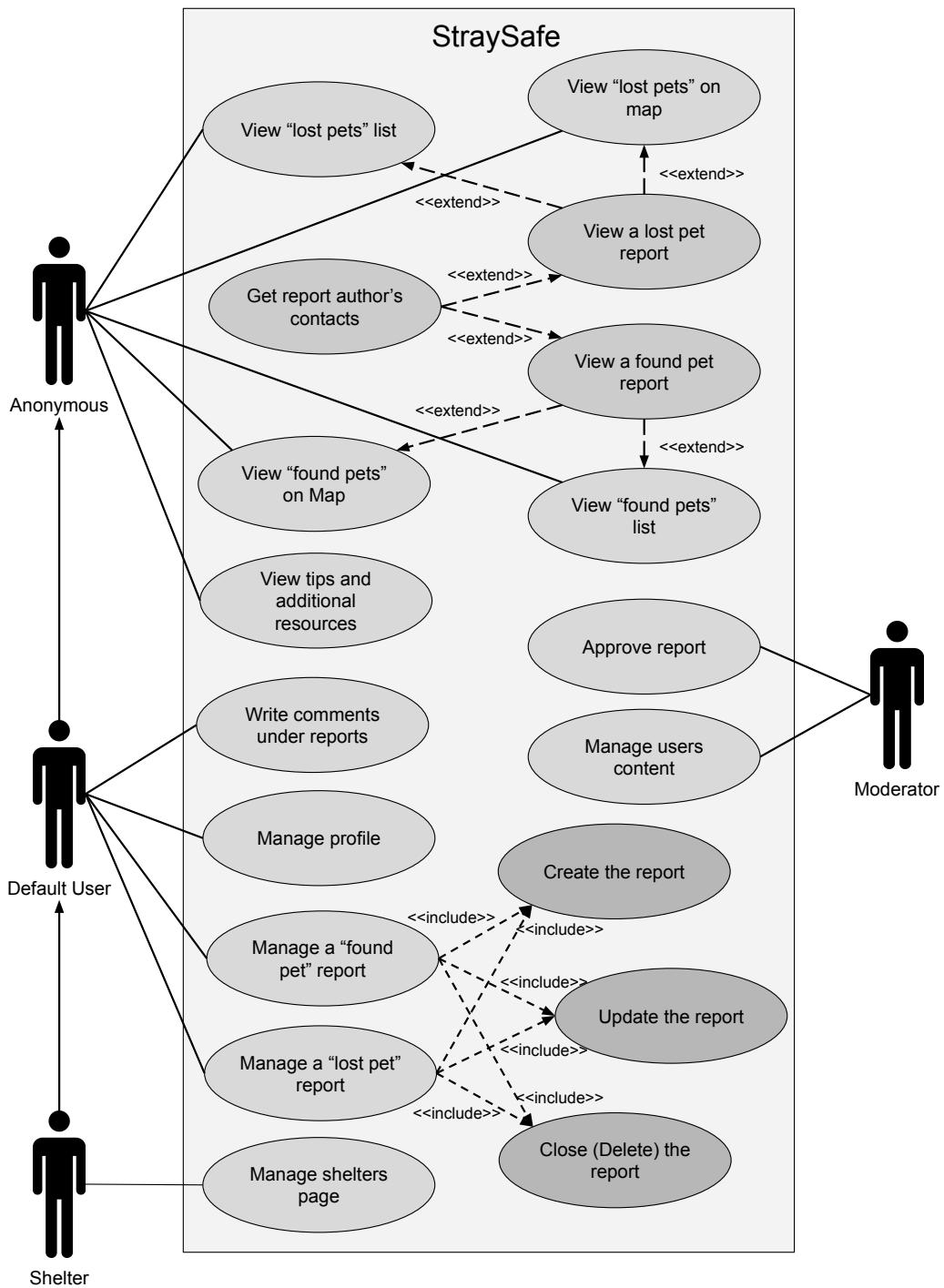


Figure 1. Use Case Diagram

- Each report shall have a comment section where a user can post comments.
- The website shall support multiple languages to cover a diverse user base. Users should have the option to select their preferred language.
- There shall be two views: map and grid list.
- Both views need to be able to filter out information that the user is interested in. It should work by typing keywords in a text area. These include: animal, colour, breed, distinctive marks.

- The map view shall show spots where lost pets often get found.
- Users will get notifications inside the website of found pets based on image similarity and features of the pet.
- The application must have features such as user profiles and password recovery.
- In the account settings, you should be able to change your email, user password, notifications.
- User should be able to manage the report he has created.
- There will be a moderator system where the account is marked as moderator and can edit reports and comments. Ability to add reports if AI filters fail.
- Have pages on how to handle having lost a pet (e.g., “How to find a lost pet”) to ensure accuracy and reliability.
- Images of lost and found pets shall be filtered with AI models. They will be responsible for making sure that it is an actual pet not something else.
- The comments shall be filtered using a keyword-based analysis.
- The interface shall be designed for various devices, including desktops, tablets, and mobile phones.
- The application should have a dark theme.
- Image comparison to find pet similarities.

### **3.2.2 Non-functional requirements**

- Notes of general functionality or terminology should be at the head of every file.
- Pages should load in up to 5 sec.
- The database should follow the first three normalization forms for designing the database.
- Database and Website file system backups must be done automatically every month.
- Security Measures such as spam protection, DDoS attacks, data leaks, Protection from CSRF and XSS attacks, SQL injections.

## **3.3 Architecture**

The system is built upon a layered architecture designed for scalability, security, and efficiency. It can be seen in **figure 2**. It comprises of four layers distributed across two tiers each with specific responsibilities. Communication flows in a specific pattern: each layer can only send REST API requests downwards to lower layers or respond upwards to requests from higher layers. This design principle ensures a clean separation of concerns.

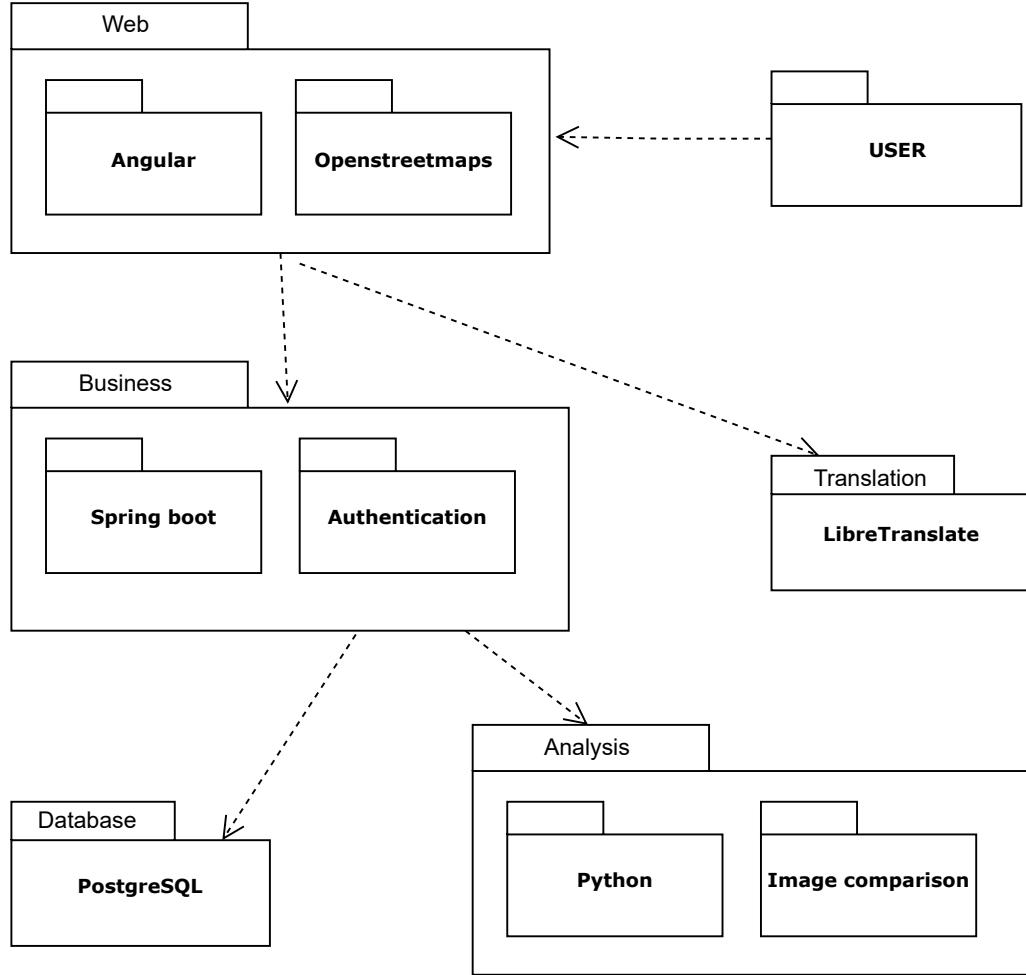


Figure 2. StraySafe Architecture

**Tier 1.** This tier is a unified collection that hosts three critical layers:

**Web Layer:** Utilizes the Angular JavaScript framework, using its server-side rendering ensures that web pages are pre-processed on the server, allowing for faster loading times and improved user experience.

**Business Layer:** Acts as the system's logic center, processing incoming requests from the presentation layer. Tasks such as image analysis and language translation are handled here.

**Analysis Layer:** This layer serves as an intermediary for data access and management. When the business layer requires reading or writing from the database, such as user authentication or fetching reports on lost and found pets, it delegates these operations to the persistence layer, which in turn communicates with the database.

**Tier 2. Database layer:** physically separated from Tier 1 in order to optimize data security. This separation enhances security measures by isolating data. More can be seen in figure 5.

The clustering of three layers into one tier is a choice made for ease of development and deployment. However, this configuration does not impose constraints on scalability or distribution. Given that each layer has been designed with distinct responsibilities and communicates through well-defined interfaces, separating them onto different machines would be a straightforward task.

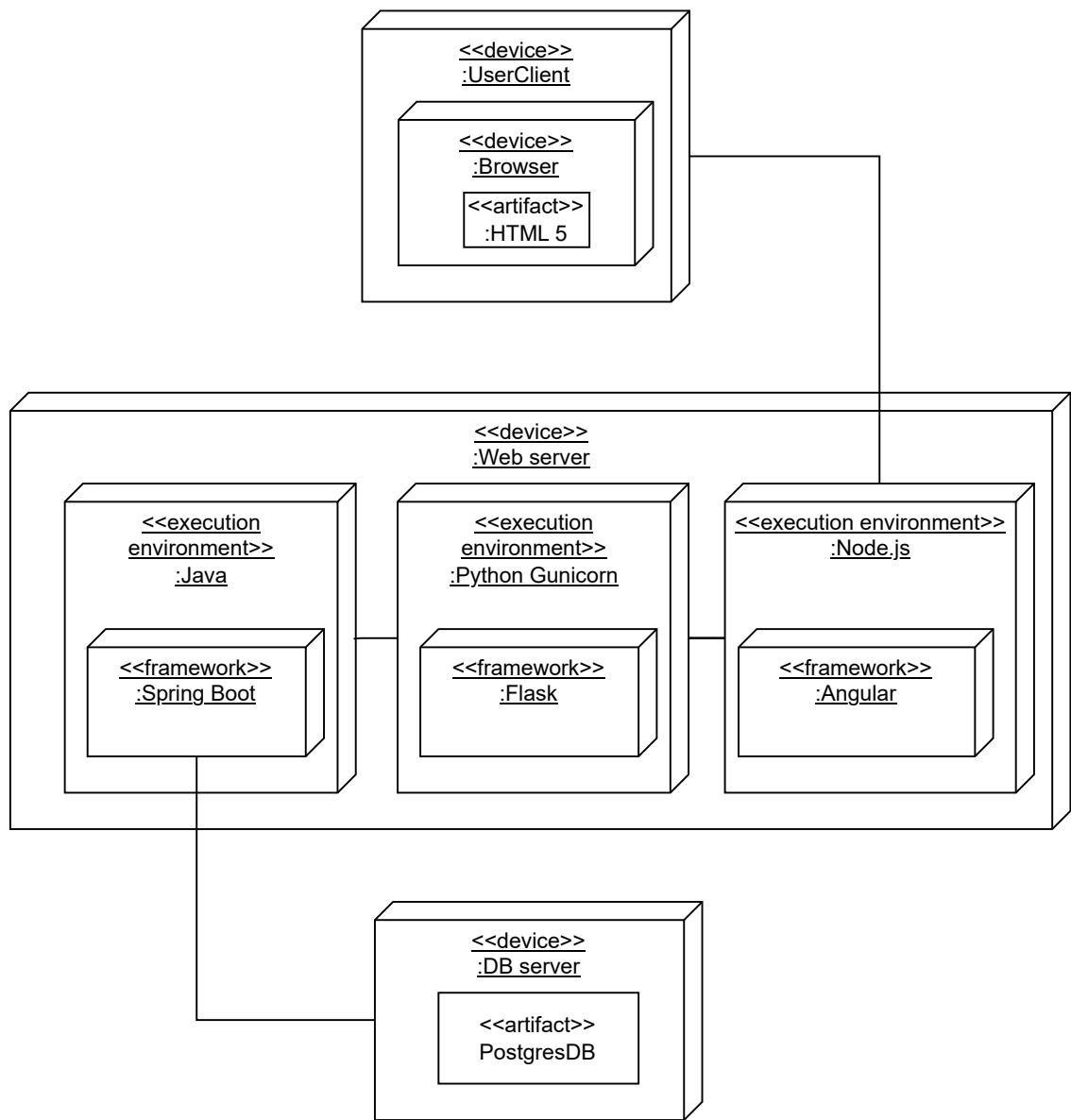


Figure 3. StraySafe Deployment

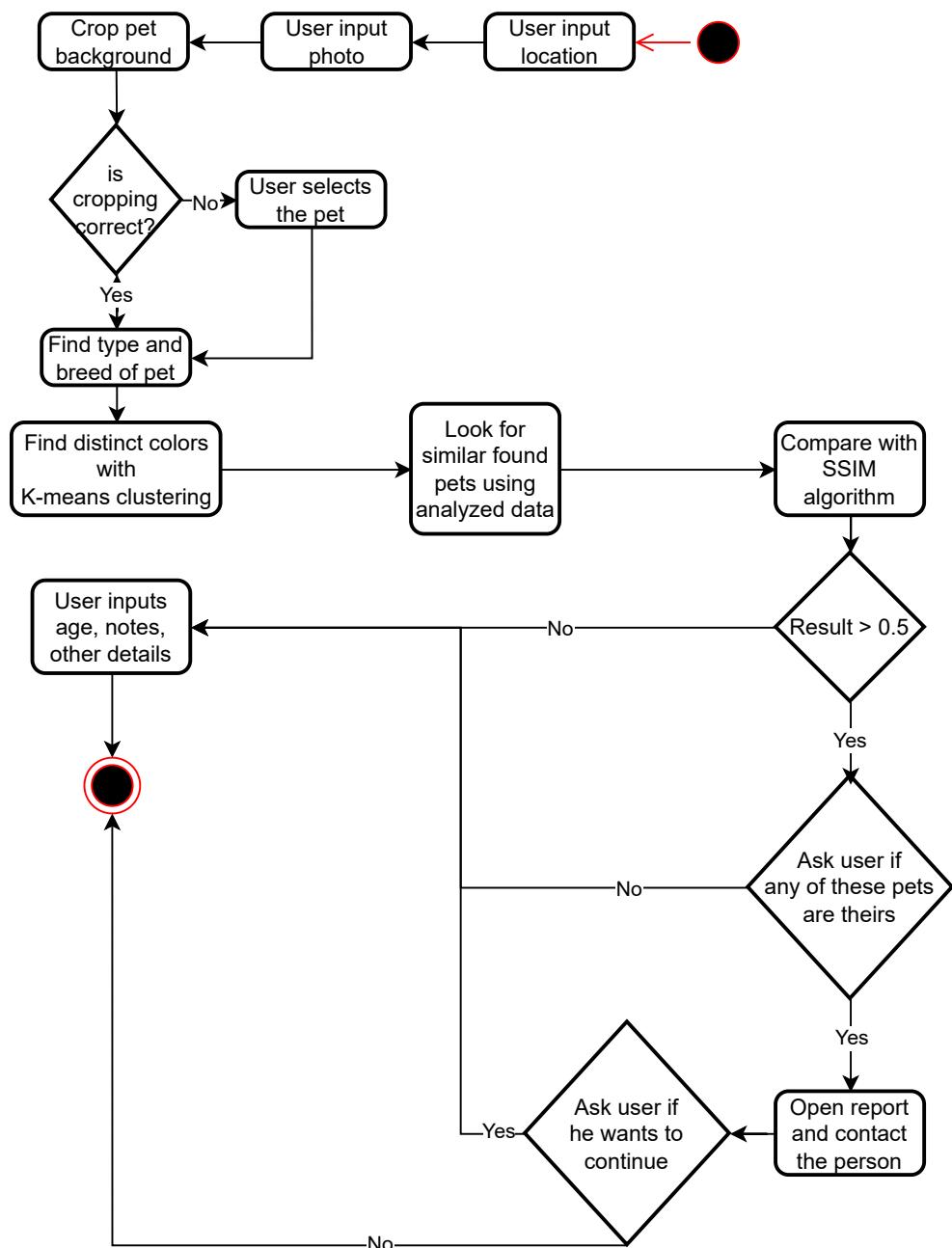


Figure 4. StraySafe Flow Chart

## 3.4 Database

To logically store all system data such as users, pet information, announcements, comments, etc. an Entity Relationship model was developed. The description below describes the ER model presented in 5.

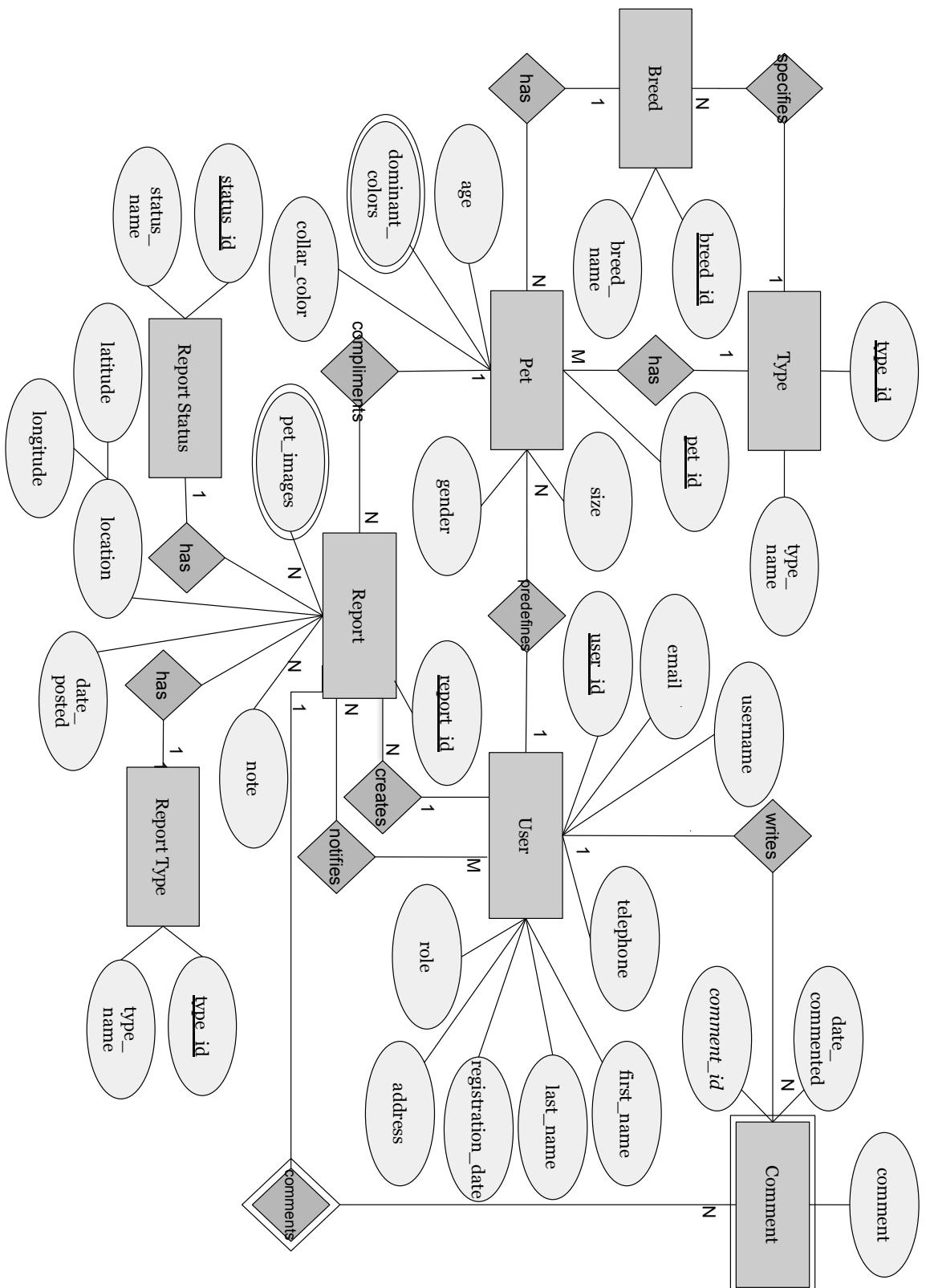


Figure 5. Entity Relationship Model

**User** is by far the most important entity in this model, as only authenticated users have access to create new reports. It has the following list of attributes: user\_id (primary key), username, profile\_image, first and last name, email and phone, registration\_date, address, and role. User has a one-to-many relationship with the **Pet** entity, which means that a single user can specify multiple pets. The **Pet** entity has the following attributes: pet\_id (primary key), name, size, gender, age, collar\_color, and dominant\_color (multi-valued attribute). The **Pet** entity also relates important entities such as **Type** and **Breed** in the many-to-one cordiality, meaning that one pet is associated with only one type and breed. These two entities only have keys (type\_id and breed\_id) and their names (type\_name and breed\_name). In addition, **Type** has a one-to-many relationship with **Breed** because each breed refers to a different type of animal.

Returning to **User**, this entity has a one-to-many relationship with the **Report** entity named "creates", since a report can have only one author. In turn, the report describes a lost or found pet, for which there is a many-to-one relationship with the **Pet** entity, for example, the same pet can be lost several times. The **Report** entity contains the following attributes: report\_id (primary key), pet\_image, date\_posted, note, and a combined location attribute consisting of latitude and longitude. For this entity, there are also 2 additional entities such as **Report Type** and **Report Status** with one-to-many relationships, which means that one report is associated with only one report type and status. These two entities have keys (type\_id and status\_id) and names (type\_name and status\_name).

There is another many-to-many relationship between the **User** and **Report** entities called "notifies". It is designed to notify users if their reports are relatively similar to other reports and are in close enough proximity to each other. The only weak entity, which also relates both **User** and **Report** in this model is **Comment**. It is related to **Report** by a many-to-one relationship, and also each comment has an author from the **User** entity. The comment entity has a comment\_id, a publication date, and the comment itself.

## 3.5 Authentication

The authentication in our system relies on JSON Web Tokens(JWT) which are used for securely managing user sessions. JWTs are made of three parts:

- The header
- The payload
- The signature

The header specifies the token type JWT and the algorithm used for signing the JWT token. The payload contains the claims, which are statements about the user and additional metadata. The information in payload is encoded but not encrypted. Thus, it does not contain any sensitive data because it can be decoded easily. The most crucial part for security is the signature which is generated by applying the signing algorithm specified in the header in our case HMAC-SHA256 to the encoded header, payload, and a secret key that is kept on the server-side. This process verifies that the sender of the JWT token is actually the user is the one he says he is.

## 3.6 Real world example

Real world example | The average user for our system would begin by having a picture of a dog6 or cat. Next step he would take is go to the homepage 7 , most likely the user is not logged in and

gets redirected to the login page<sup>8</sup> a new user would likely head to the register page<sup>9</sup> and register. After that the user gets redirected back to filling his report, the first page a user sees is the map page 10 where he has to specify the location where he lost, found or seen a pet. After that the user gets redirected to the pet information submission pages 11. For the average user the last stage of submission would be previewing his pet report 13



Figure 6. Image of a found, seen or lost pet

## 4 Implementation

### 4.1 Tools

### 4.2 Database

Relation

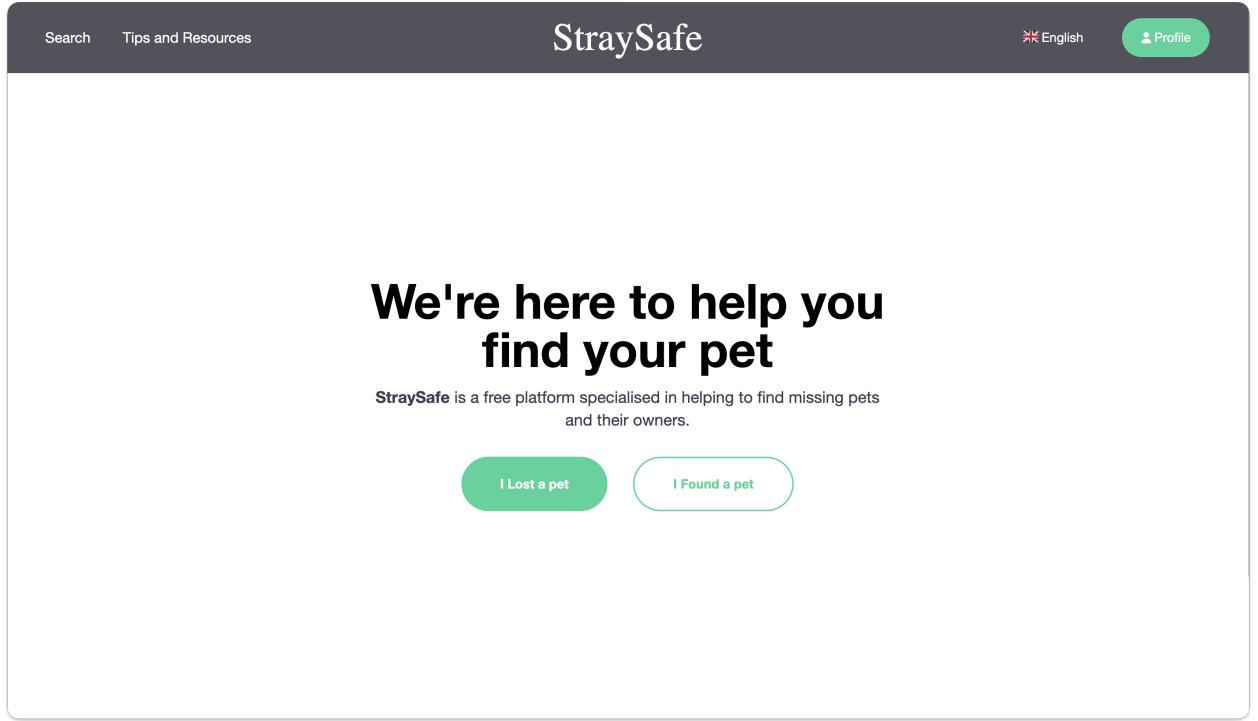


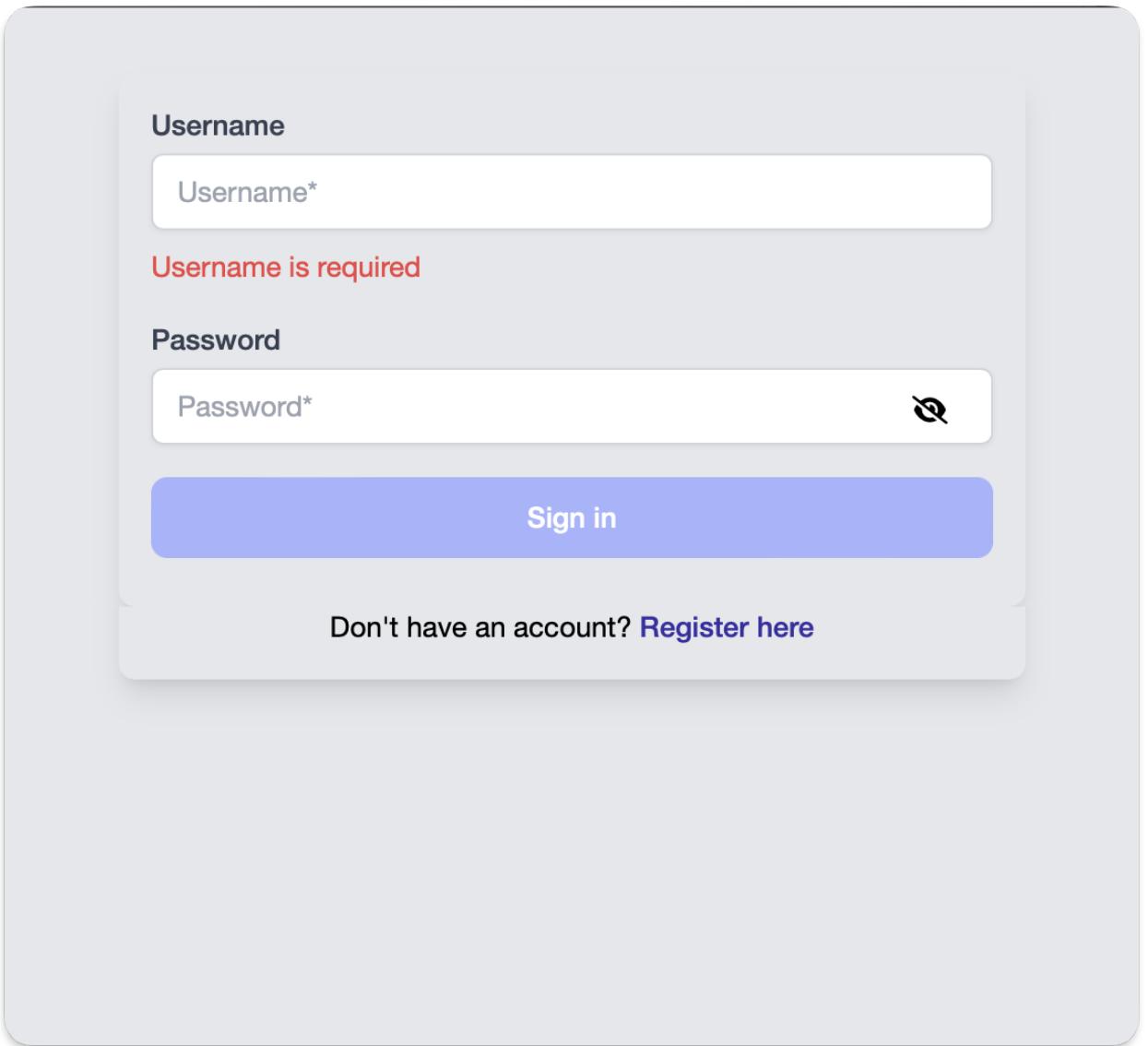
Figure 7. Home page

#### 4.3 Map

#### 4.4 Text processing

#### 4.5 Automating installation

### 5 Testing



The image shows a login page with a light gray background and rounded corners. It features two input fields: a white one for 'Username' and a white one for 'Password'. Below each field is a red error message. A large blue button labeled 'Sign in' is centered below the password field. At the bottom left, there's a link to register.

**Username**

Username\*

Username is required

**Password**

Password\*

Sign in

Don't have an account? [Register here](#)

Figure 8. Login page

# Register

Username

Geedis

Password

.....



Email

egerdvila@gmail.com

First Name

Edvinas

Last Name

Gerdvila

Phone Number

+37060491223

Home Address

Didlaukio 59, Vilnius

**Sign up**

Figure 9. Register page

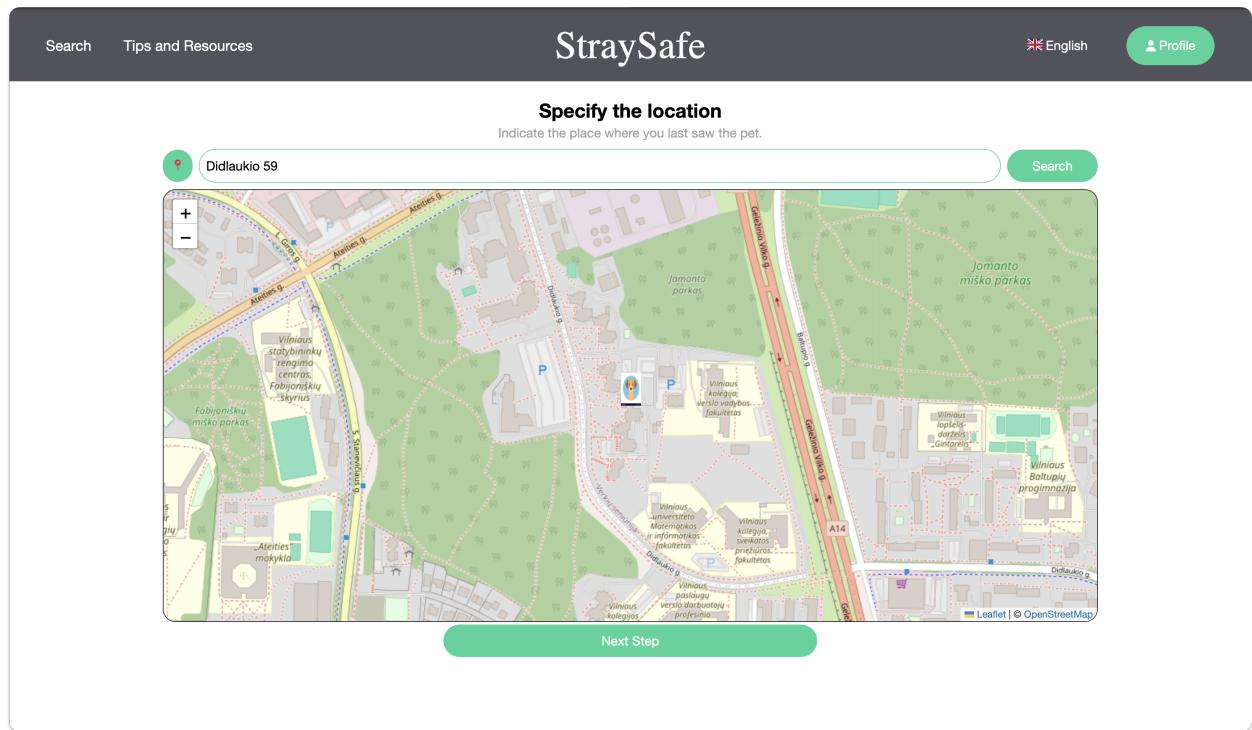


Figure 10. Specify location page for report submission

**Upload a photo of the pet**

Choose the best quality photo possible that clearly shows pet's face.



Upload another photo

**LOST Pet Information**

Give as many exact parameters of the pet as possible.  
\* Indicates required option

Your pet is already in your pet lists? Choose the pet or enter pet id [here](#)

<b>Enter pet name*</b>	<input type="text" value="Bigis"/>	
<b>Type*</b>	<b>Size*</b>	
<input type="text" value="Dog"/>	<input type="text" value="Medium"/>	
<b>Gender*</b>	<b>Age*</b>	
<input type="text" value="Male"/>	<input type="text" value="Adult"/>	
<b>Colors</b>		
Main dominant color	Second dominant color	Collar color
<input style="background-color: #ccc; width: 100px; height: 20px;" type="text"/>	<input style="background-color: #ccc; width: 100px; height: 20px;" type="text"/>	<input style="background-color: #ccc; width: 100px; height: 20px;" type="text"/>
<b>Breed*</b>	<input type="text" value="Beagle"/>	
<b>Additional notes</b>		
<input type="text" value="Lost at iki supermarket."/>		

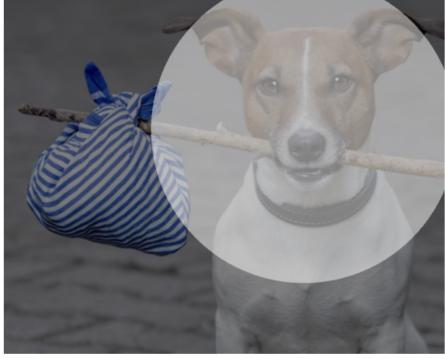
I am agree to share my contacts data located in my profile  
 I am agree with [Privacy policy](#)

[Preview report](#) [Submit the report](#)

Figure 11. Pet information page for lost pet submission

**Upload a photo of the pet**

Choose the best quality photo possible that clearly shows pet's face.



[Upload another photo](#)

**FOUND Pet Information**

Give as many exact parameters of the pet as possible.  
\* Indicates required option

I have only seen the pet.

Type*	Size*
Dog	Medium
Gender	Age
Not selected	Young

Colors

Main dominant color	Second dominant color	Collar color
<span style="background-color: black; width: 100px; height: 10px;"></span>	<span style="background-color: black; width: 100px; height: 10px;"></span>	<span style="background-color: black; width: 100px; height: 10px;"></span>

Breed

Beagle
--------

Additional notes

I saw him at Didlaukio 41 , IKI in Vilnius at 14:00 April 21st.

I am agree to share my contacts data located in my profile  
 I am agree with [Privacy policy](#)

[Preview report](#) [Submit the report](#)

Figure 12. Pet information page for found dog

Author: Viktor Barinov @clotemone  
Posted: 18 of April

**LOST** Pet Information report identifier: [REPORT IDENTIFIER](#)



**Pet name** Bigis

<b>Type</b> Dog	<b>Breed</b> Beagle
<b>Size</b> Medium	<b>Gender</b> Male
<b>Age</b> Adult	

**Colors**

Main dominant color #b0a7a0	Second dominant color
Collar color #b6b6b7	

**Notes** Lost at iki supermarket.

**Telephone** +37062271255

**Report tags**

DOG LOST ACTIVE

Last seen location

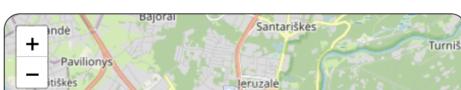


Figure 13. Preview page of pet report

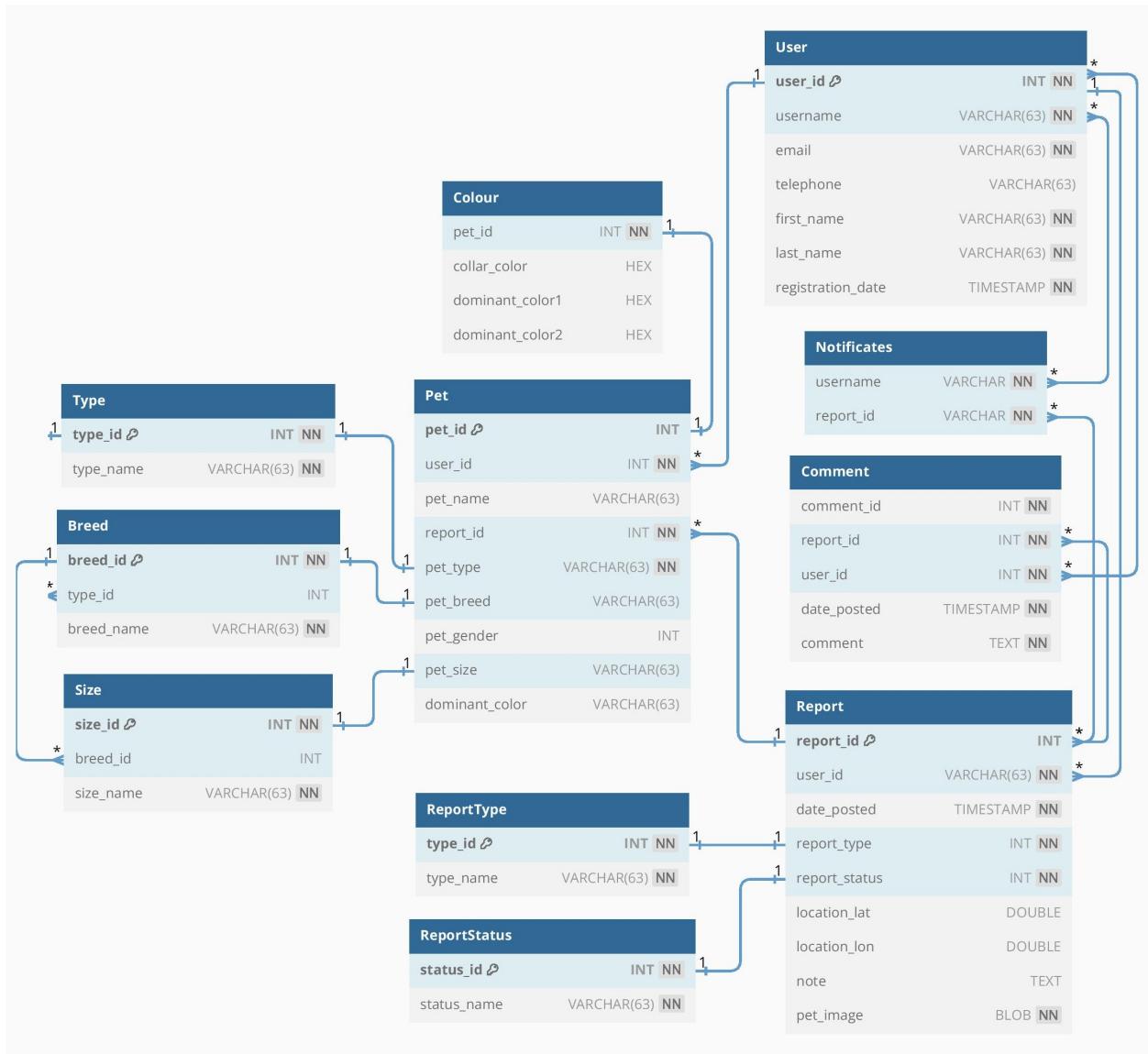


Figure 14. Relational Schema

## **Conclusions and Future Work**

## **Appendices**