Bilkent University

Department of Computer Engineering

Senior Design Project

HelPet

Analysis Report

Berat Tuna Karlı - 21400505
Berke Deniz Başaran - 21400996
Doğa Zeynep Germen - 21201974
Irmak Tural - 21301099
Numan Mertcan Cankara - 21201743

Supervisor: Halil Altay Güvenir
Jury Members: Çiğdem Gündüz Demir, Mustafa Özdal
Innovation Expert: Burcu Coşkun Şengül
Website: https://mcankara.github.io/HelPet.io/

Analysis Report
March 19, 2018

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.

Contents

1. Introduction

For the maintenance of a balanced ecosystem, we cannot consider a world without animals. However, as the cities becomes bigger and wider, some of the animal species such as dogs, cats, birds, etc. start to live in the cities with us 'humans'. Some of us prefer to live with the animals and own them as pets but some of us do not prefer that. For the pet owners, having pet requires too much responsibility and attention. In other respects, without any human support, a lot of animals try to live in the crowded and dangerous cities. To help both pet owners about their pets and the stray city animals, we will develop an android application HelPet.

According to American Veterinary Medical Association, only in U.S.A. there are almost 44 millions of household dog owning, 36 millions of household cat owning and 3 millions of household bird owning [1]. For each household, veterinary visits per year is almost 3[1] .Thus, there is a huge need and market for pets and their care. By the help of HelPet, pet owners can easily manage their pets care. An android application HelPet can be reached by anybody who have the Internet access and smart phones to get help for their pets. By the help of the HelPet, pet owners can see the top rated veterinary services near their current location and rate the veterinary services regarding their performance. HelPet send notifications to the pet owners about their pets veterinary service appointments, vaccine dates and feeding times. Also HelPet users can have a profile for their pets if they want to breed their pets with others which they want to pair according to their profile. In addition, according to statistics of American Society for the Prevention of Cruelty to Animals, the %15 of the pets gets lost [2]. Thus, as a solution for that issue, HelPet users also can find their lost pets by sending their photo and location where pet is lost and receive the photo and the location of the suspicious lost animal by the other users. Pet owners also can find a temporary caretaker for their pets if they have to leave their pets for a while and report the caretakers performance by rating them. Lastly, HelPet users can ask question about animals, get answers about users and get tips about the animal care as notification.

On the other hand, there are numerous city animals which have not got any house or owner who support them to live a healthy and happy life. According to presentation of Prof. Dr. Tamer Dodurka, there is no information about the number of pet dogs and stray dogs in Turkey [3]. However, according to Dodurka, in the world %75 of the dog population consists of the stray dogs and only in Italy, %25 of the pet dogs is abandoned by their owners into the city streets [3]. To help the stray animals, animal lovers can report an animal which needs any

help by taking the photo of it, specify the location and send it to the HelPet. This reporting will be sent to the nearby animal care volunteers as a notification and make them to be aware of the animals that need help. Also users can report the dangerous animals to protect people and also the other animals.

In the following parts of this report, the brief description of the application will be given. Also, under the title of constraints, there will be several approaches about restrictions and limitations of the application will be discussed. Thereafter, professional and ethical issues will be argued. Lastly, functional and non-functional requirements of the application will be clarified.

2. Current Systems

- **Petsbook** is a mobile application that helps pet owners in various ways [4].
    - Petsbook is multi-platform application. It is available in both IOS and Android.
    - It lets you create a profile for your pet.
    - In includes animal adoption system.
    - It lets user watch expert videos about their pets.
    - It also includes pet-guides, where useful information can be read by users.
- **PiP** is designed to help pet owners to find their lost animals [5].
    - The PiP is a mobile application for iPhone and Android.
    - Application uses facial recognition technology to re-unite lost pets with their families.
    - However, the score of the PiP in Google Play Store is 2,3.
    - According to comments in Google Play Store, some essential functions of application does not work properly.
- **11pets**
    - It has 2 major functionalities which are *Remember*, *Monitor*.
    - It has reminders on major aspects of the pet such as food tracking, medication, internal de-worming, bathing, nail clipping, ear cleaning, etc.
    - It also allows the pet owners to keep track of pet's height, weight, temperature, pet allergies, genetic tests, lab documents, etc. [6]

- **Pati Birliği:** Another application that's decicated to the street animals. [7]
    - It is available for both iOS and Android users
    - It has 3 main functionalities which are adoption, help the animal and check the location of the animal. Users are directed to the social media profiles for the help and adopt options.
    - Users can visualize the animals that are currently helped at their own profile page

Every system has beneficial features. However, all applications have different specific aspects for pets. PetsBook is interested in social features for pet owners, 11pets cares about health of pet and PiP has one single focus on finding lost animals via image processing.

This project aims to be a comprehensive application composed of the features above and some additional features. Moreover, one essential part of the project is finding lost animals. Even there is an application (PiP) for this purpose, current application does not work properly and did not updated since 2015. Therefore, current application is not able to satify the need.

3. Proposed System

3.1. Overview

HelPet is a mobile application that targets to animal lovers. It can be considered as a platform that provides help, consulting to owner of pets and animals that is wounded or lost.

When people encounter an animal that is wounded or indigent, even they want to help them they may not help in some cases. Because they may be in a situation that they have to go somewhere else or do not have enough time or equipment. With the help of HelPet they have a chance to notify people by sharing the picture and location of the animal. This feature does not require registration.

Pet owners have difficulties about finding proper partner for their animals to mate. But HelPet offers a feature to solve this problem. Pet owners add their pet and its features to their profile. Users could make their search by filtering the features of pet acording to their wishes. Then users are able to contact with the owner of the pet they want to mate with their pet.

Pet owners may forget their pets' vaccine time. While adding a pet to their profile, they may add time periods of vaccine of their pets. HelPet can remind pet owner on the vaccine time.

Also people who can not be able to look after their pet anymore need to find a safe place to give their pet. HelPet has a category which people can add their animals with their features and location for adoption. People who want to adopt a pet can search an animal from that category by filtering acording to their personal demand. Hence adoptation is very sensitive issue for both who gives the pet and who adopt the pet, HelPet provides an opportunity user to see and comunicate with candidate owner of their pets.

It is very difficult to take pets to everywhere. For example people who need to travel for two weeks may not be able to take their pets with themselves. Therefore, they need to leave their pets to private pet hotels but they may not be affordable or suitable. Via HeLPet users can find people easily to look after their pet. Because HelPet includes a category which consists of announcements from people who want to be a host to pets for a limited time. Also hosts declare which animal tand which time period they can host so that people who search for a host can find the proper caretaker and they can communicate with host. In additon, at the end of the time period of takecare, pet owner can rate the host and comment to his/her profile.

Searching lost pets is really challenging issue even in our times. There is no easy way to find a pet but HelPet supports a category for lost animal announcements. People can post their lost animals with their picture and probable province. If someone sees the picture of any lost animal, they can communicate with the owner of announcement owner. Also they can check the category that shows indigent animals with their picture and location, and they can be absolutely sure from the picture.

In addition HelpPet includes a forum that users can ask questions, share information and consult each other about their pets.

Furthermore, the pictures are out of purpose will be reported by users and then admin can remove the picture or ban the user if it is essential.

3.2. Functional Requirements

3.2.1. User Functionality Requirements

- The user should be able to sign in after filling the sign up form with him/her information.
- In case of emergency, the user should be able to use essential features such as posting a wounded animal or posting lost animal without creating an account.
- The user should be able to create a profile(s) for him/her pets after creating an account.
- The user should be able to select which information will be shown in his profile.

- User can search a partner to his/her animal to mate by filtering based on the features that is demanded.
- User will be able to take picture of a wounded animal or through the HelPet and transmit to other users with location of animal.
- User will be able to take notifications which inform users about the wounded animal that is located nearby.
- User can search a volunteer to look after his/her animal temporarily, user can declare his/her expectations and the time.
- User can announce that he/she could look after an animal temporarily.
- User can rate and comment the hosts and they can see the ratings.
- User can search pets and select to adopt.
- User can report his/her pet as missing pet an.
- User can be reminded by HelPet about his/her animal's vaccine time.
- User will be able consult each other through provided Q&A.
- Users will be able to report irrelevant posts.

3.2.2. System Functionality Requirements

- The system should collect data from user's inputs.
- The system should show the average rate of a user in their profile.
- The system should present the pictures, brief description and the location of the indigent pet.
- The system will check the similarty between images of pets that found in street and images of pets that are posted as lost.
- The system will list the result of search based on users' preferences such as location, breed.

3.3. Non-Functional Requirements

3.3.1. **Reliability**

The provided services must be accessible 98% of the time

3.3.2. **Performance**

During the process "Update Customer" system responses should be no more than 1 second. System respond time shall be at most 1.5 seconds.

3.3.3. Precision

In case of any changes made to the system entities, the time of change must be recorded with detail.

3.3.4. Configurability

The website will be configurable enough to make the necessary or arbitrary changes. It may also allow small changes such as the ability of a user to change his/her credentials.

3.3.5. Capacity

Up to 500-600 of users can use the site at the initial state. up to 100,000 unique user can be recorded to the system.

3.3.6. Concurrency

All online users can use the emergency services for pets and street animals.

3.3.7. Backup and Recovery

In any case of unscheduled system crash, data on the website will be recovered from the backup database whichs daily updated. All processes can be made within 1 work day after the crash.

3.3.8. Availability

The platform is available 7/24. Users will be able to log-on and update the social feed any time.

3.3.9. Auditing and Reporting

User activities will be recorded regularly by the system admins. The recorded activities are going to be accessible with reports. This will allow site admins to watch over users who tend to use the app for abusive purposes.

3.4. Pseudo-Functional Requirements

3.4.1. Issue Tracking

GitHub will be used for version control and issue tracking.

3.4.2. Deployment Platform

HelPet needs to have mobility so it has to be mobile application. Application will be published for Android platforms.

3.4.3. **Databases**

Firebase will be used to keep the all data such as user profiles, pets information, ratings, comments, etc.

3.4.4. **External Tools**

Google Maps Android API will be used to add location.

3.5. System Models

3.5.1. Scenarios

Scenario 1

Use Case Name: CreateNewAccount

Actors: User Armağan

Entry Conditions:

- User Armağan is on LogIn screen.

Exit Conditions:

- User Armağan is on Homepage screen.

Main Flow of Events:

1. User Armağan presses ‘Create Account’ button.
2. HelPet displays SignUp screen.
3. User Armağan enters his name as ‘armagan’.
4. User Armağan enters his surname as ‘sari’.
5. User Armağan enters his username as ‘a.sari’.
6. User Armağan enters his e-mail address as ‘armagan@sari.com’.
7. User Armağan enters his phone number as ‘+905358823308’.
8. User Armağan presses ‘Submit’ button.
9. HelPet sends a verification e-mail to Armağan’s e-mail address.
10. User Armağan verify his account by clicking ‘Continue’ link.
11. HelPet displays the homepage for Armağan.

Scenario 2

Use Case Name: LogIn

Actors: User Armağan

Entry Conditions:

- User Armağan is on LogIn screen.

Exit Conditions:

- User Armağan is on Homepage screen.

Main Flow of Events:

1. User Armağan enters his e-mail as ‘armagan@sari.com’.
2. User Armağan enters his password.
3. User Armağan presses ‘Login’ button.
4. HelPet checks if credentials are correct.
5. HelPet gives a warning that says ‘Wrong password’
6. User Armağan enters his password again.
7. HelPet checks if credentials are correct.
8. HelPet confirms that email and password are correct.
9. HelPet displays homepage for Armağan.

Scenario 3

Use Case Name: AnimalInNeed

Actors: Guest

Entry Conditions:

- Guest is on Homepage screen.

Exit Conditions:

- Guest is on Homepage screen.

Main Flow of Events:

1. Guest presses the Emergency button
2. Guest presses ‘Animal in Need’ button
3. He uploads three pictures of the animal that he found.
4. System displays a success message.
5. Guest sets the location.
6. Guest adds his phone number.
7. Guest presses ‘POST’ button.

Scenario 4

Use Case Name: FoundLostAnimal

Actors: Guest

Entry Conditions:

- User Armağan is on Homepage screen.

Exit Conditions:

- User Armağan is on Homepage screen.

Main Flow of Events:

1. Guest presses the Emergency button
2. Guest presses ‘I Found an Animal’ button
3. Guest uploads a pictures of the animal.
4. System displays a success message.
5. Guest sets the location.
6. Guest adds his phone number.
7. Guest presses ‘POST’ button.

Scenario 5

Use Case Name: EditAccount

Actors: User Armağan

Entry Conditions:

- User Armağan signed up and decided to change his account informations.

Exit Conditions:

- User Armağan is on Settings page with his new settings.

Main Flow of Events:

1. Armağan opens his account settings.
2. Armağan changes his email information that was used for signup process.
3. Armağan adds his phone number.
4. Armağan saves all changes.
5. System shows success message to Armağan.

Scenario 6

Use Case Name: AddPetToProfile

Actors: User Armağan

Entry Conditions:

- User Armağan signed up and decided to add his pet ‘Max’ to his profile.

Exit Conditions:

- User Armağan is on his own profile page.

Main Flow of Events:

1. Armağan presses ‘Add Pet’ button.
2. Armağan enters his pet’s name as ‘Frank.
3. Armağan selects pet’s as ‘dog’.
4. Armağan selects breed of his dog as ‘Frank.
5. Armağan uploads 2 pictures of Frank.
6. Armağan enters vaccine period as ‘Once in four months’
7. Armağan fills a description part as ‘My cute little boy ☺’.
8. Armağan enable the option for make his pet visible in mating section.
9. Armağan clicks the ‘Add Pet’ button.
10. System shows a success message.

Scenario 7

Use Case Name: LookAfterVoluntarily

Actors: User Armağan

Entry Conditions:

- User Armağan signed up and decided to look after somebody’s pet.

Exit Conditions:

- User Armağan is on homepage.

Main Flow of Events:

1. User Armağan presses the dropdown menu button.
2. User Armağan presses the ‘Look After’ button from dropdown.
3. User Armağan presses the ‘I am a Volunteer’ button.
4. User Armağan selects a time period as ‘14 March – 24 March’.
5. User Armağan chooses a ‘dog’ to look after.
6. User Armağan enters the daily price as 50₺.

7. User Armağan sets the location as Bilkent 3 Ufuk Sitesi.
8. Armağan presses the ‘Post’ button.
9. System displays a success message.

Scenario 8

Use Case Name: FindPlaceForMyPet

Actors: User Armağan, User John

Entry Conditions:

- User Armağan decided to give his pet to someone for some period.

Exit Conditions:

- User Armağan is on homepage.

Main Flow of Events:

1. User Armağan presses the dropdown menu button.
2. User Armağan presses the ‘Look After’ button from dropdown.
3. User Armağan presses the ‘Search Volunteers’ button.
4. User Armağan selects a time period as ‘25 March – 30 March’.
5. User Armağan presses the ‘Search’ button.
6. System will display available people.
7. User Armağan decided to select John to give his pet for 25-30 March.
8. User Armağan goes John’s profile.
9. User Armağan sends a direct message to John.

Scenario 9

Use Case Name: PostForAdoption

Actors: User Armağan

Entry Conditions:

- User Armağan decided to adopt his pet ‘Max’ to someone else.

Exit Conditions:

- User Armağan is on his own profile page.

Main Flow of Events:

1. Armağan presses ‘Adopt Your Pet’ button.
2. Armağan chooses the pet ‘Max’ he wants to adopt.

3. Armağan fills a description part as ‘I need to adopt my lovely friend Max to someone who will care about him because I got an job offer in Thailand’.
4. Armağan clicks the ‘POST’ button.
5. System shows a success message.
6. Armağan is currently on homepage.

Scenario 10

Use Case Name: FindPetToAdopt

Actors: User Armağan, User Taner

Entry Conditions:

- User Armağan decided to adopt an animal.

Exit Conditions:

- User Armağan is on the profile page of the owner of the pet he wants to adopt.

Main Flow of Events:

1. Armağan presses ‘Adoption Posts’ button.
2. Armağan enters ‘Filter results’.
3. Armağan selects pet as ‘cat’.
4. Armağan selects location as ‘Bangkok’.
5. Armağan clicks ‘Apply Filters’.
6. Armağan finds a cat named Minik clicks on ‘Owner’s Profile’.
7. Armağan clicks the ‘Message’ button.
8. Armağan writes ‘Dear Mr. Taner, I want to adopt your Cat Minik’.

Scenario 11

Use Case Name: FilterHomepage

Actors: User Armağan

Entry Conditions:

- User Armağan decided to filter his own homepage.

Exit Conditions:

- User Armağan is on homepage.

Main Flow of Events:

1. Armağan presses ‘Filtering’ button.

2. Armağan chooses ‘Cat’ and ‘Dog’ for animals.
3. Armağan restricts location by choosing ‘Bangkok’.
4. Armağan clicks the ‘Apply Filter’ button.
5. System shows a success message.

Scenario 12

Use Case Name: FindMate

Actors: User Armağan, User Sarp

Entry Conditions:

- User Armağan decided to find a partner for his cat ‘Minik’.

Exit Conditions:

- User Armağan is on the profile page of the owner of the pet he wants to mate with his cat.

Main Flow of Events:

1. Armağan presses ‘Find Mate’ button and goes to list of animals which are visible for mating for his dog.
2. Armağan presses ‘Filtering’ button.
3. Armağan chooses ‘Cat’ as animal.
4. Armağan chooses ‘Male’ as gender.
5. Armağan restricts location by choosing ‘Bangkok’.
6. Armağan clicks the ‘Apply Filter’ button.
7. System shows a success message.
8. System shows candidate pets.
9. Armağan finds a cat named Mimi clicks on ‘Owner’s Profile’.
10. Armağan clicks the ‘Message’ button.
11. Armağan writes ‘Dear Mr. Sarp, I want to mate my cat ‘Minik’ with your cat’.

Scenario 13

Use Case Name: HelpIndigentAnimal

Actors: User Armağan

Entry Conditions:

- User Armağan decided to help one indigent animal.

Exit Conditions:

- User Armağan is on page of indigent animal's profile.

Main Flow of Events:

1. Armağan presses 'Help Animal' button.
2. Armağan presses 'Filtering' button on list of animals page.
3. Armağan chooses 'Cat' as animal.
4. Armağan restricts location by choosing 'Bangkok'.
5. Armağan clicks the 'Apply Filter' button.
6. System shows a success message.
7. Armağan finds a lovely helpless cat and clicks on 'Cat's Profile'.
8. Armağan clicks the 'Show Location' button.
9. Armağan clicks 'Write Comment' button and writes 'I will find you and feed you today'.

Scenario 14

Use Case Name: Question/Answer

Actors: User Armağan

Entry Conditions:

- User Armağan decided to ask a question about his pet 'Max'.

Exit Conditions:

- User Armağan is on Q/A page.

Main Flow of Events:

1. Armağan presses 'Q/A' button.
2. Armağan clicks 'New Topic'.
3. Armağan selects keyword as 'Cat', 'Stratching Problem'.
4. Armağan writes 'My cat is stratching my arms, what should I do?' to the question part.
5. Armağan uploads two pictures of his strached arm.
6. Armağan clicks the 'Submit' button.
7. System shows a success message.

Scenario 15

Use Case Name: LostPetNotification

Actors: User Armağan, User Taner

Entry Conditions:

- User Taner took a picture of a dog that he thinks lost then upload it.

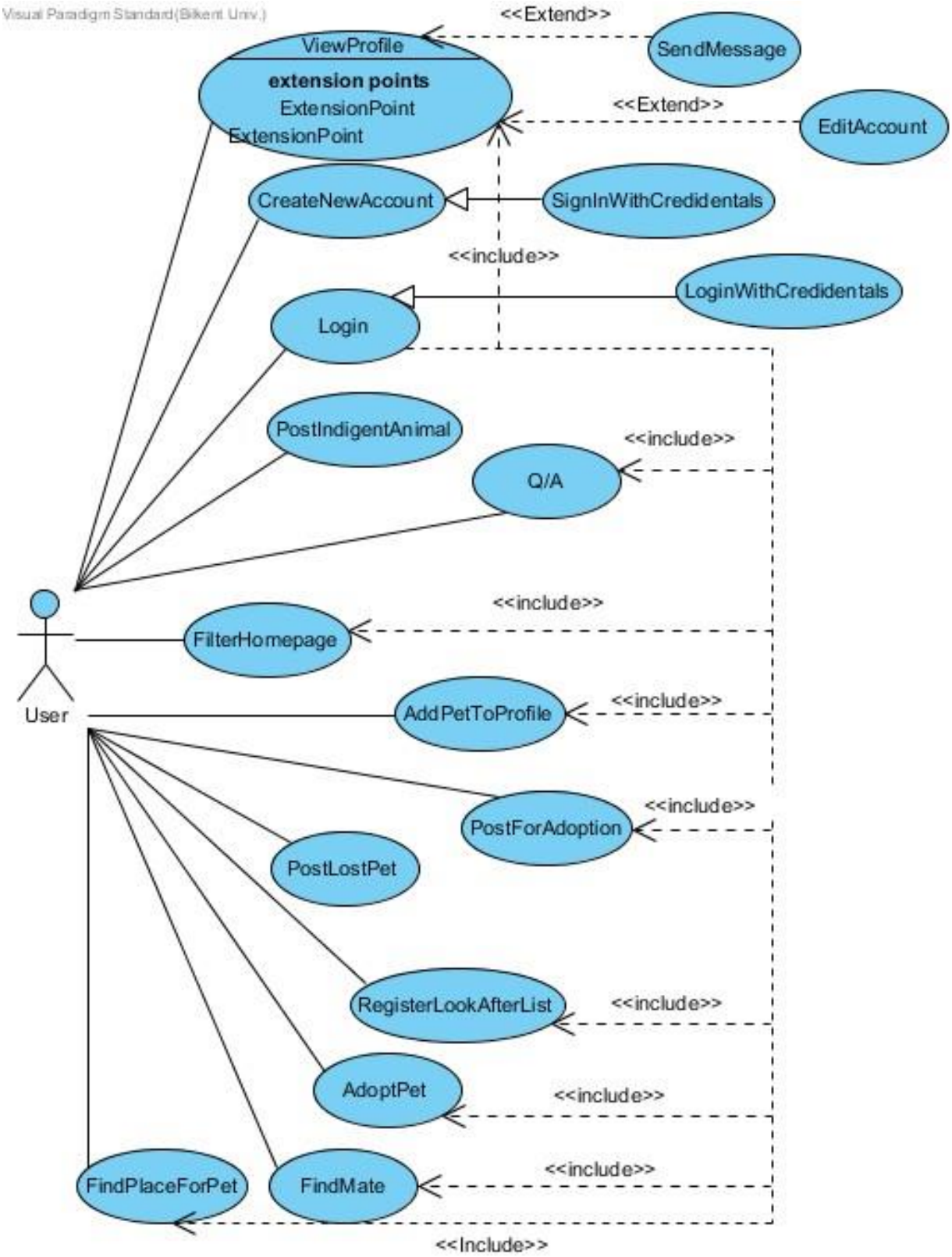
Exit Conditions:

- User Taner is on homepage.

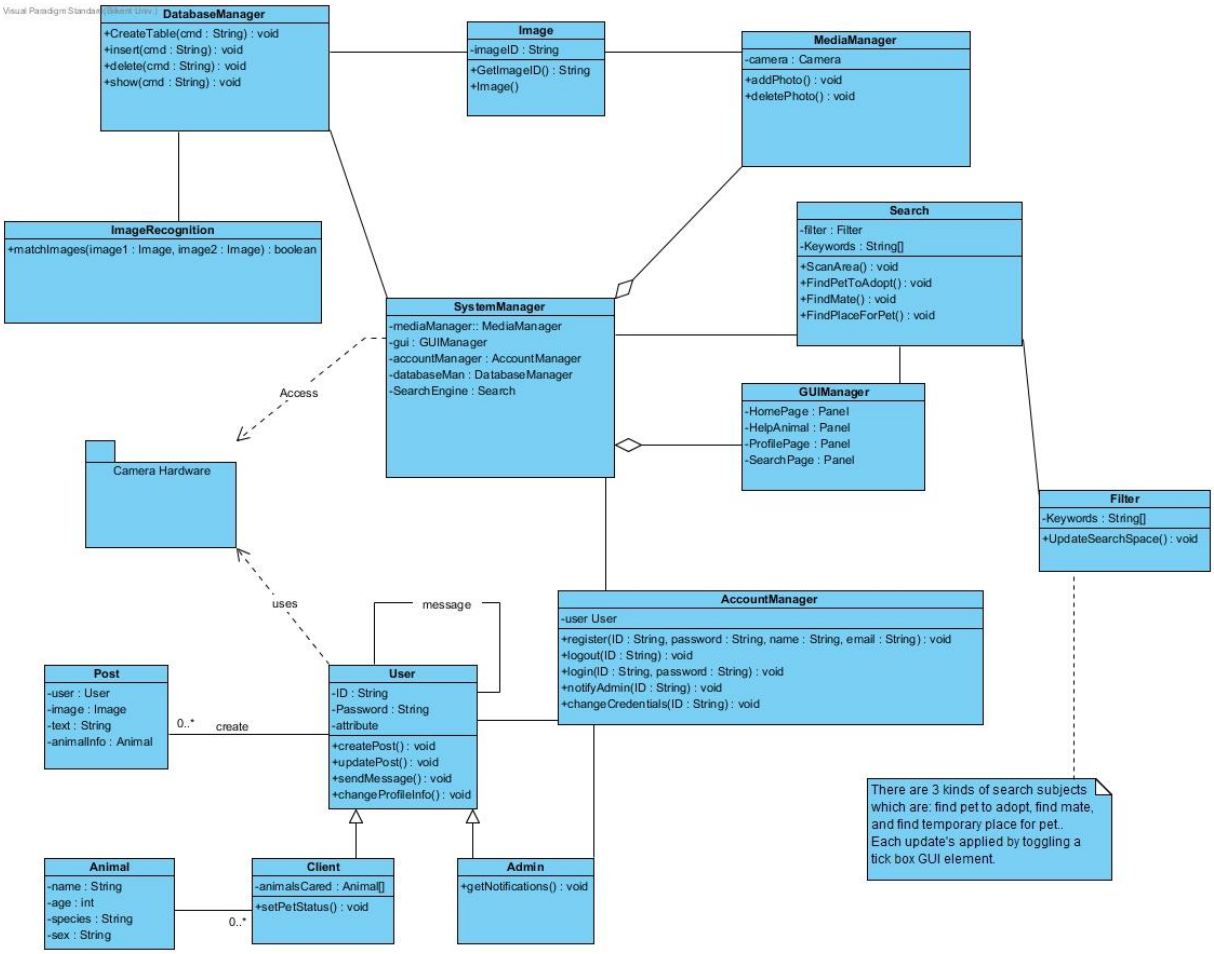
Main Flow of Events:

1. Armağan posts his lost dog's picture into system.
2. Taner presses 'I found a dog' button.
3. Taner uploads 3 pictures of the dog that he found.
4. System will check the similarity between Armağan's lost dog and uploaded pictures.
5. System find the similarty rate as %65.
6. System will notify Armağan saying 'Check this out! It might be your dog'.
7. Armağan opens the application.
8. Armağan sees his lovely dog 'Max'.
9. Armağan sends a private message to Taner.

3.5.2. Use Case Model



3.5.3. Object and Class Model



- *DatabaseManager*

In this class, each method works with the query parameter. The methods essentially obey the SQL language in order to provide the database functionality.

void CreateTable(String cmd) : This method is used to create a table so that SQL queries can be used to retrieve the relevant data. The data is considered to be anything from user credentials to the image IDs which are used to get the actual image.

void insert(String cmd) : With the insert method, new tuples can be added to the existing tables.

void delete(String cmd) : Delete is used to remove the existing tuples from identified table.

void show(String cmd) : The existing entries can be visualized with this method.

- *Image*

Since the application's main functionality is image recognition, image is necessarily declared as one of the core components.

String GetImageID() : It returns the image id so that image can be pointed easily.

- *MediaManager*

The system records or deletes images with a manager class called MediaManager.

void addPhoto() : Image is added to the system with addPhoto function.

void deletePhoto() : Existing image is deleted from the system with this function.

- *ImageRecognition*

The missing pets are identified with image recognition technique. This class will perform a matching algorithm supported by auxiliary functions in order to compare two images and give the desired result.

boolean matchImages(String imageID1, Image imageID2) : Two imageIDs taken as parameters will be compared with this function. The IDs are taken in order to point the images easily. A boolean value will return indicating if two images match or not.

- *SystemManager*

All the manager classes will perform their tasks under the system manager class, which is the main director of our application. This class will function with singleton pattern.

MediaManager mediaManager() : An instance of media manager class is created so that its methods and attributes can be accessible from system manager.

GUIManager gui() : gui instance is initialized in order to control the GUI elements via system manager.

AccountManager accountManager() : All accounts are managed by account manager. It's initialized in order to access the accounts and manipulate their data from system manager when it's necessary.

DatabaseManager databaseMan() : Database manager is a class which aids to account manager to store the user data and additionally store the image ids. It's initialized just like the others in the system manager class.

Search SearchEngine() : Search engine is a component that allows user to navigate through application easily. It correlates with the filter class for better search accuracy. It's connection is assembled with other crucial components in the system manager class.

- *Search*

Users navigate in the application with search functionality. Search allows user to access the desired page. It's supported with the class called filter to specify the search options.

void ScanArea() : This method is used to locate the animals which are closest to the user at that time.

void FindPetToAdopt() : User may filter the search options for better accuracy. Once the find pet to adopt option is chosen, user will see the relevant results.

void FindMate() : Another filtering option is find mate. It allows user to find an appropriate mate for his/her pet.

void FindPlaceForPet() : Last filter option is find place for pet. User may search for a suitable place for his/her pet.

- *GUIManager*

GUI elements of the application are initialized and managed in this class. Main GUI elements are the panels in our application.

Panel HomePage() : This panel is used to display the main page components and the buttons on home page such as post, emergency, search, etc.

Panel HelpAnimal() : Each operation involving animal care is handled in different pages. This panel is configured to hold the components of help animal functionality.

Panel ProfilePage() : User profile is visualized with this panel. With this panel, components of user profile are demonstrated neatly.

Panel SearchPage() : For search purposes, user is directed to a different page. GUI components of filtering and searching are handled in the search page panel.

- *User*

Users are defined in two different ways. One is the admin, which has the ability to observe the activities of any client user. The other one is client user which is an ordinary user type that has the ability to post on main page and send messages to other users.

void createPost() : Each user has the ability to create new posts on the main page feed. This method enables users to create new posts.

void updatePost() : Alongside creating new posts, modifying the existing ones can be done by any user as well.

void sendMessage() : Users can send direct messages to each other with help of this method.

void changeProfileInfo() : Users can also change their profile information with change profile info method. Any kind of profile entry except email can be modified by the user.

- *AccountManager*

Any operation related to account management is handled by account manager class.

void register(String ID,String password,String email ,String name) : New users are recorded to application database with this method. User credentials are initially identified with user input and passed to the register method as parameters.

void logout(String ID) : This method allows user to disconnect from the system.

void login(String ID) : User can login the system with this method.

void notifyAdmin(String ID) : Each activity committed by client users are notified to the registered admins with notify admin method.

void changeCredentials(String ID) : When users request changes in profile, the permission for this change is granted and handled by change credentials method.

- *Filter*

  This class supports the search class for more accurate search results.

  void UpdateSearchSpace() : Once the filtering options are identified with tick boxes in search panel, update search space method takes its part to edit the search results.

- *Client*

  Clients are ordinary users of Helpet. Beside the inherited functionalities, clients can update the status of animals they are taking care of.

  void setPetStatus() : Every client can update the current status of the animals they aid or own with set pet status functionality.

- *Admin*

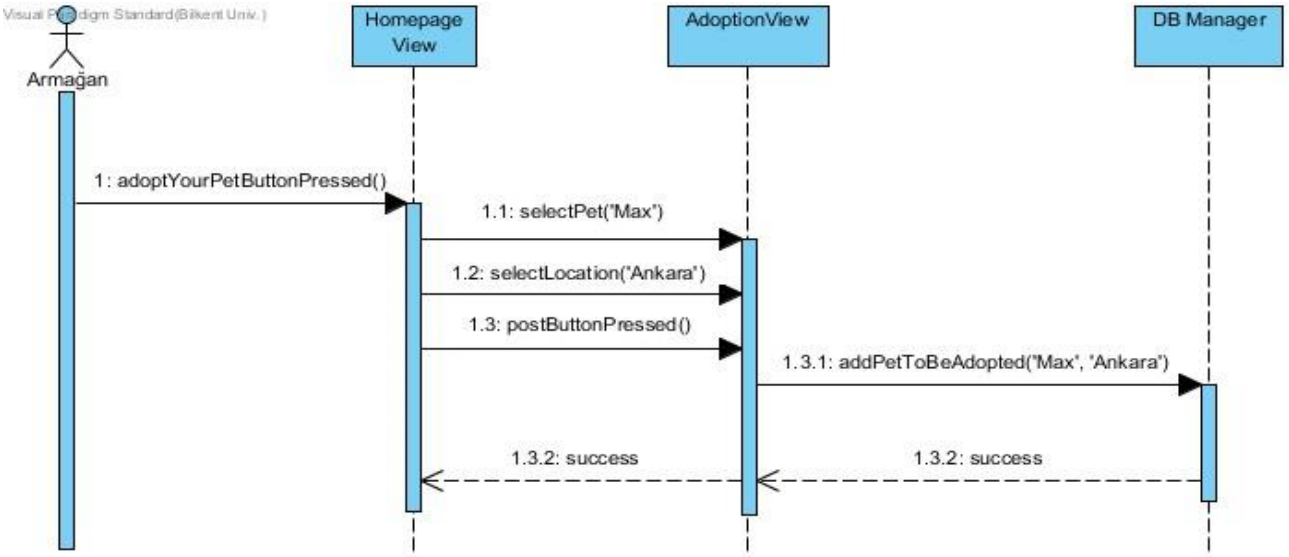
Admins are privilaged users which can observe the activity in system.

void getNotifications() : Each single user activity is retrieved by admin users with get notifications method. The activities are recorded as journal entries periodically.

3.5.4. Dynamic Models

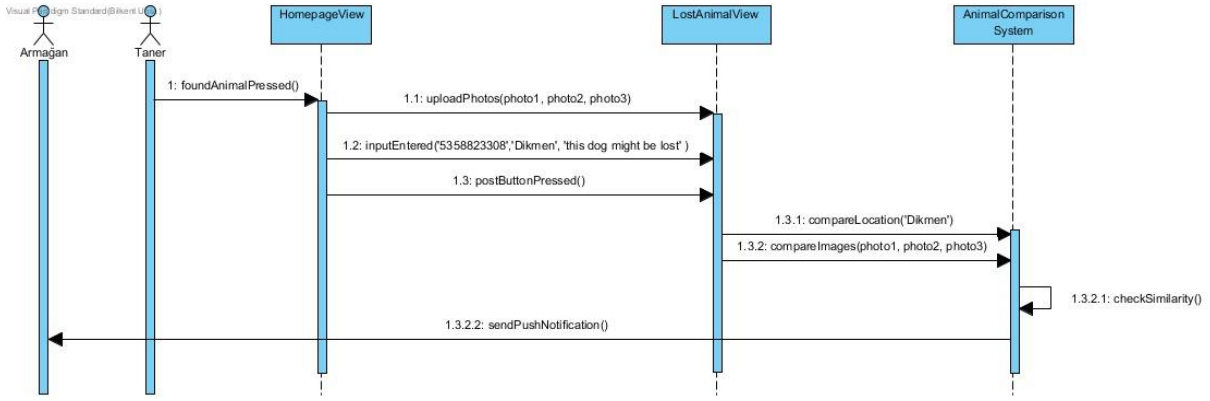
3.5.4.1. Sequence Diagrams

- **Adopt Pet Sequence Diagram**



This diagram shows the flow of actions when the user wants to adopt his pet to somebody else. First of all, *HomepageView* will be displayed, then the *AdoptionView*. After s/he selects his pet and location, '*post'* button will be pressed by the user. Then the system will send these informations to the *DatabaseManager* so that when other users search for a pet to adopt, they will be able to see that dog from the database.

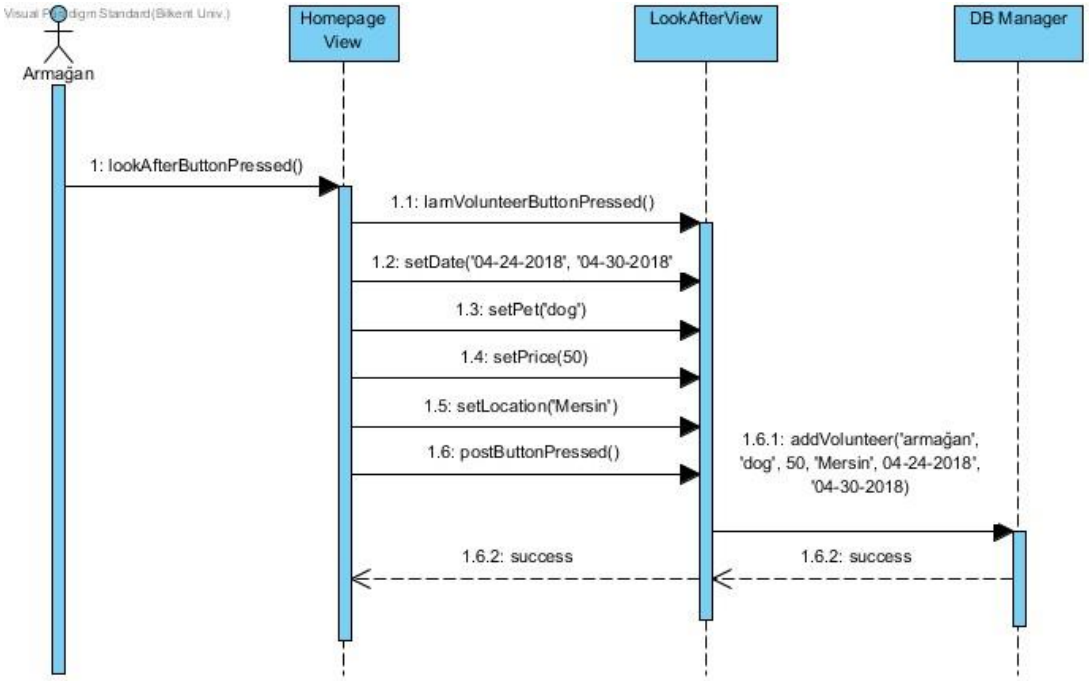
- **Suspicious Lost Dog Post Sequence Diagram**



This diagram explains the situation when user wants to post a dog that might be lost. First *HomepageView* will be displayed, then *LostAnimalView* will showed to the user. After system get the necessary information from the user such as photos of the dog, phone number of the user and location, this information will be send to the *AnimalComparisonSystem* that

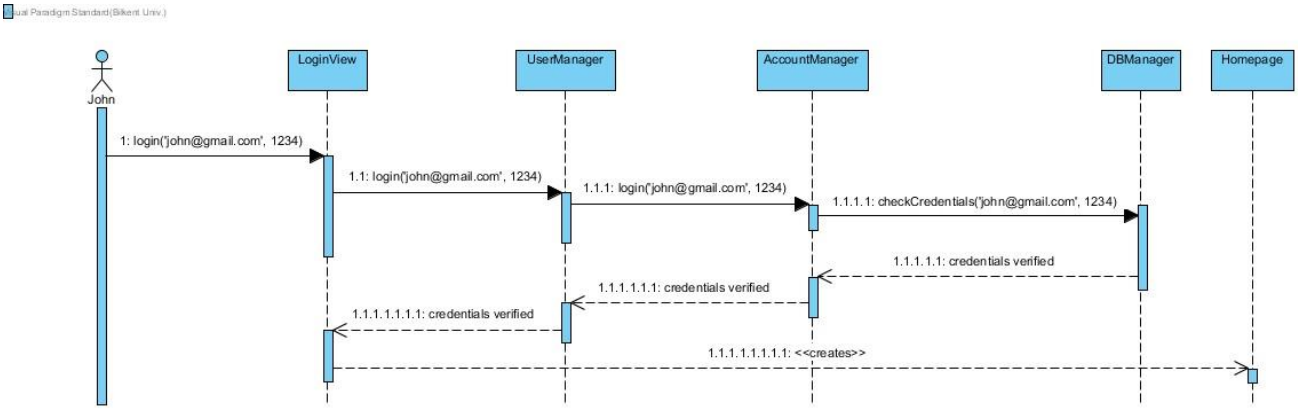
will check the similarty between current entries in database and given informations by the user. If the system finds such similarty in database, it will send a notification to the user who has lost his dog.

- **Look After for a Pet Sequence Diagram**



This diagram explains the situation when the user wants to make a post indicates that s/he will be able to look after someone's pet for a certain period of time. First of all *HomepageView* will be display, then *LookAfterView* will be showed to the user. After s/he fills the required fields of the form, these inputs will be send to the *DBManager* so that other users will be able to find him/her to give their pet for a certaion time of period.

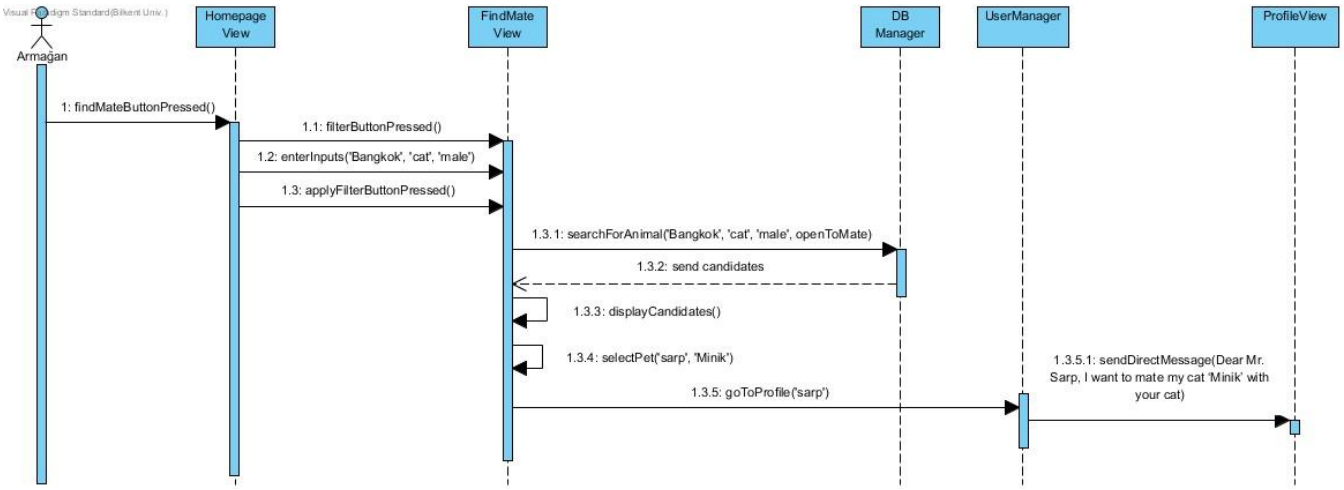
- **Login Sequnce Diagram**



This diagram shows the flow when user successfully login to the system. First system shows *LoginView*. User fills the required fields. This information will be sent to the

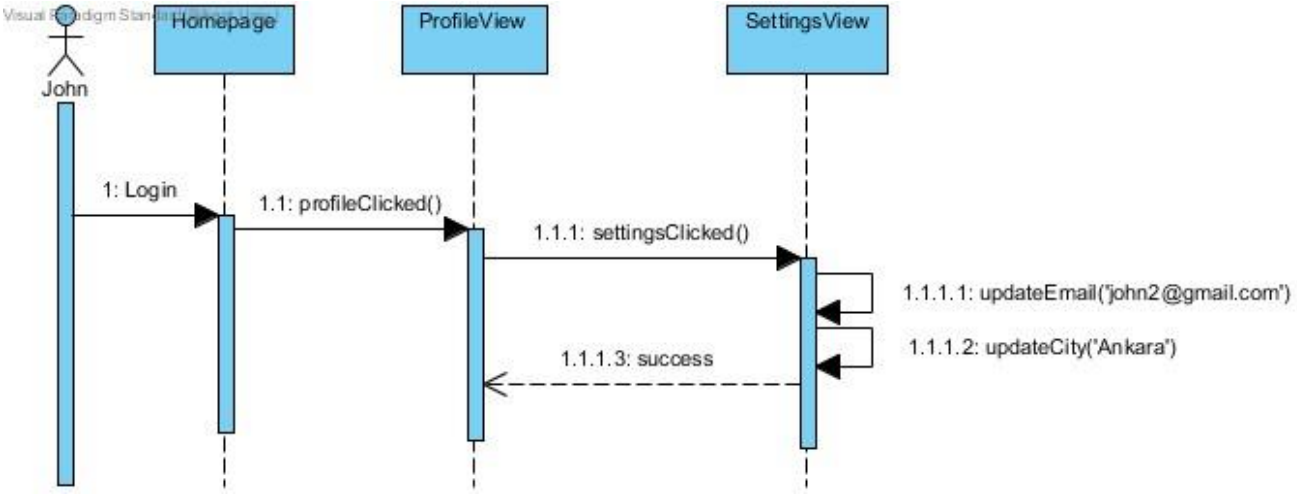
UserManager and *UserManager* sends this to the *AccountManager*. *DBManager* sends the success message to the *AccountManager*, and then *AccountManager* sends another success message to *UserManager*. When the response from *UserManager* comes to *LoginView*, the *HomepageView* will be created.

- **Find mate for Pet Sequence Diagram**



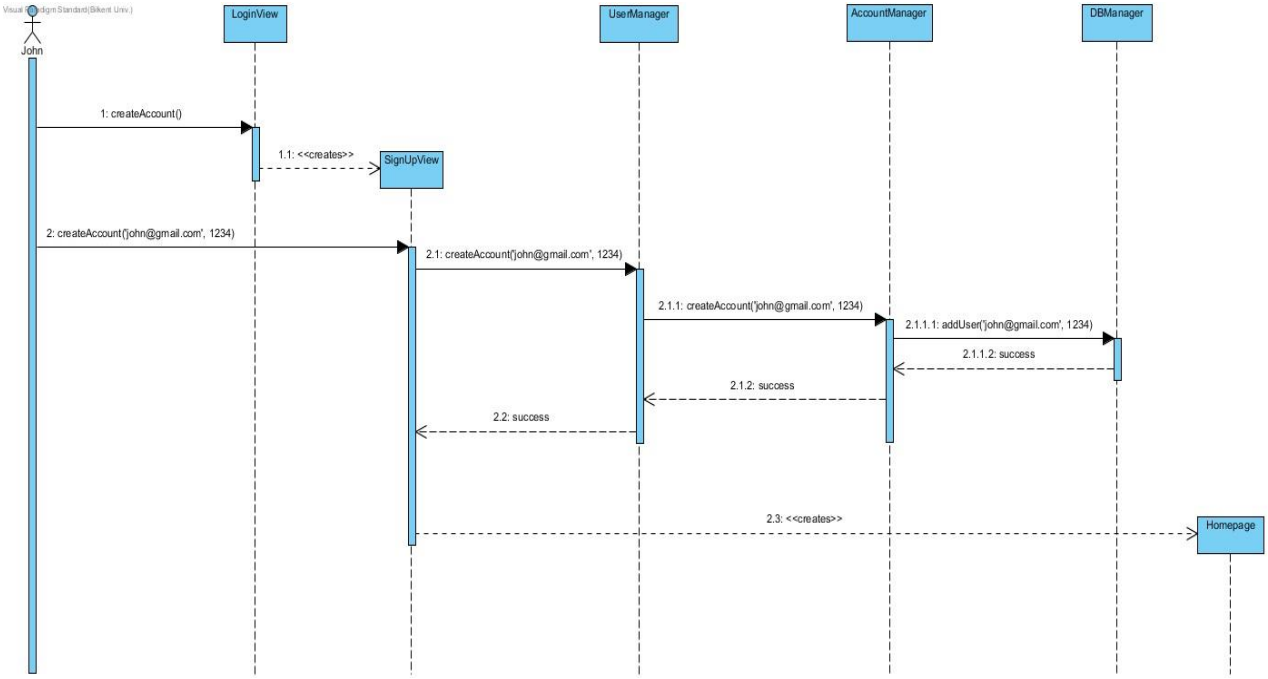
This sequence diagram explains the situation when the user wants to find a partner for his/her pet. First of all, *HomepageView* will be displayed, then *FindMateview* will be showed. After user fills the necessary fields of the form, *DBManager* will check and sends the results of the query to the *FindMateView*. User selects one of the candidates. Then with the help of *UserManager* user will be able to go to other user's profile and in the *ProfileView*, user sends a private message to other user.

- **Edit Account Sequence Diagram**



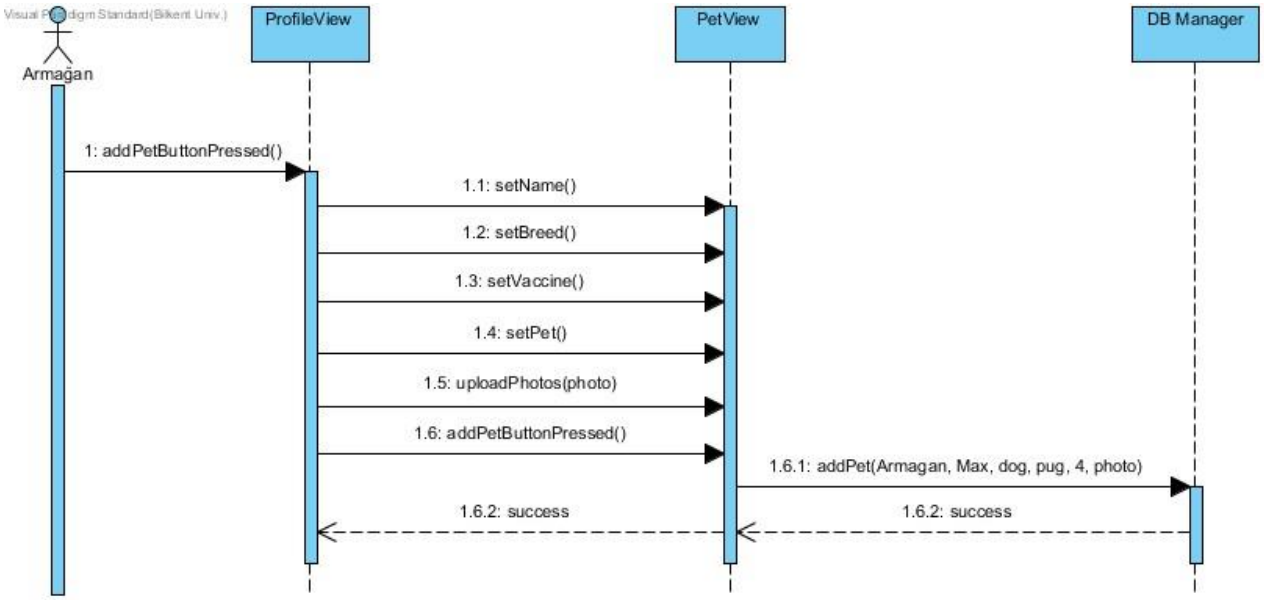
This diagram explains the situation when user wants to edit his profile. First s/he will see the *HomepageView*. The *ProfileView* will be showed after user presses according button. Then s/he will be able to change its personal information with the help of *SettingsView*. After changing fields successfully *SettingsView* returns a success message to the *ProfileView*.

- **Create a Account Sequence Diagram**



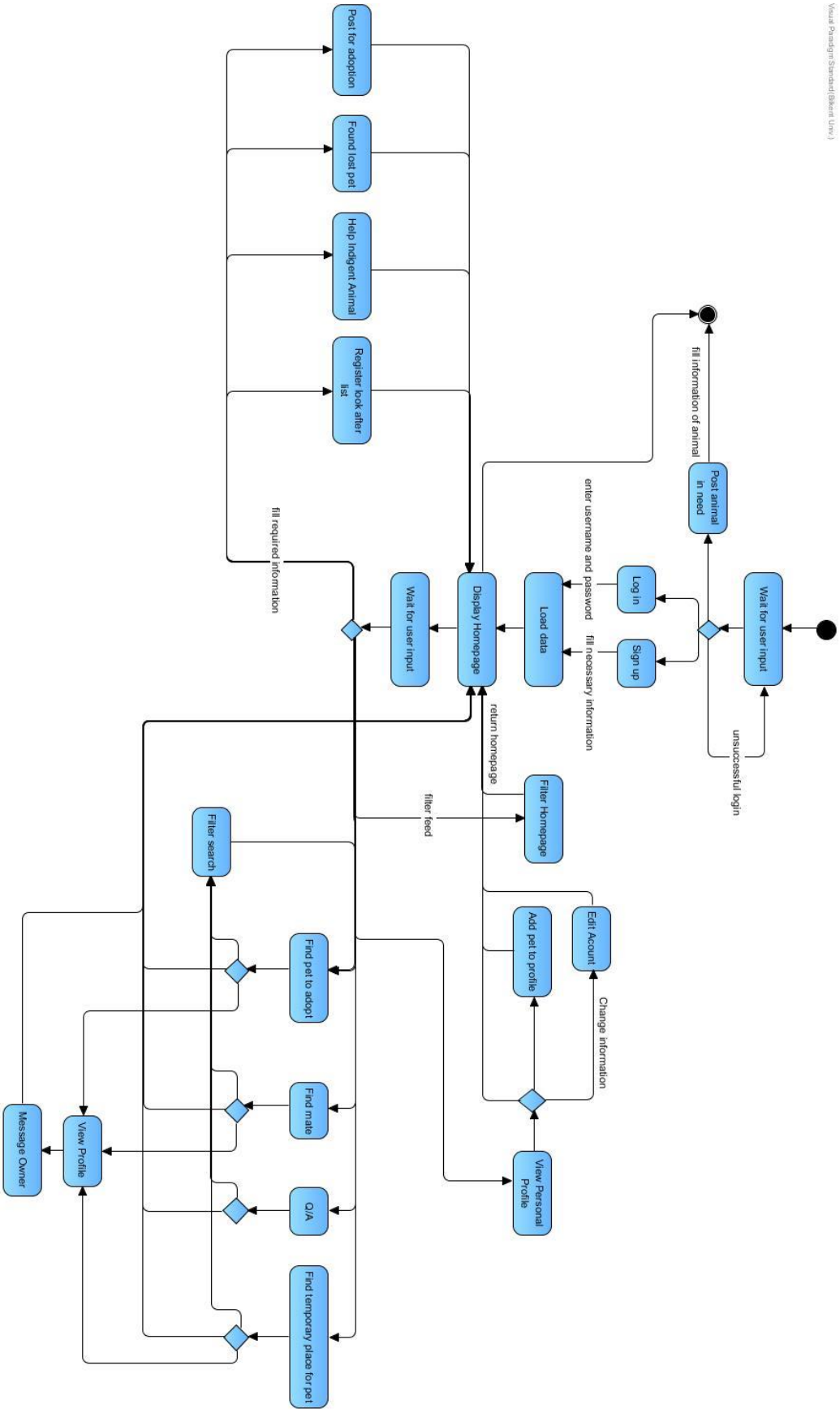
This diagram explains the situation when user creates a new user account successfully. *LoginView* will be displayed first, then the *SignUpView*. S/he fills the required fields. The information, then, is sent to the *UserManager* and then *AccountManager*. *AccountManager* passes this information to the *DBManager* in order to store the new entry. After creating a new user successfully *DBManager* returns a success message to the *AccountManager*, then *AccountManager* sends a success message to the *UserManager*. *UserManager* sends a success message to the *SignUpView*. Last, *SignUpView* creates a *HomepageView* for the user.

- **Add Pet to User Profile**



This diagram explains the flow of the actions when user wants to add the his/her pet to the user's personal profile. User will see the *ProfileView*. After that *PetView* is displayed. User fills the required fields of the form in order to add a pet to his profile. This informations will be sent to the *DBManager* in order to keep this data. Then *DBManager* returns a success message to the *PetView*. *PetView* sends a success message to the *ProfileView*.

3.5.4.2. *Activity Diagrams*



- There are three options when user starts the application; user could log in, sign up or post an animal that is wounded or indigent.
- If user just wants to post an animal without logging in, the only thing that is needed to be done is filling the information of animal (location and description) and then user could exit.
- If user signs up, then the required information needs to be filled. After a successful sign up, data of the account is loaded, homepage appears and system starts to waits user.
- If user logs in and authentication is succeeded, again data of the account is loaded, homepage appears and system starts to waits user.
- After homepage is successfully appeared, user could filter the homepage according to his/her personal choices. User could view and edit his/her profile or (s)he could edit pets to his profile.
- User could search pets to adopt (s)he could filter the search and if (s)he find a pet to adopt (s)he could view the profile of owner and message him/her.
- User could search for a mate for his/her pet. Again (s)he could filter the search, view the profile of chosen pet's owner and message him/her.
- User could search for a temporary place for his/her pet. (S)he is able to filter the search and contact with the host.
- Q/A part is designed for questions of users. User could read from this platform by filtering the results or ask a new question.
- User could register for looking after list. (S)he fills necessary information and (s)he starts to wait to be a host.
- User could want to help an indigent animal if (s)he encounter in his/her homepage. (S)he looks for the information of the animal (location, description and species) and if (s)he is able to help, (s)he helps.
- User could post a pet if (s)he encounters a lost pet in street, (s)he takes photos of the the pet and post with a brief description and location information.
- If post for adoption is selected, user fills required information and post it.

In every state, user is able to exit the system, but it is not shown on the diagram to prevent confusion in the design.

3.5.5. User Interface

- **LoginPage**

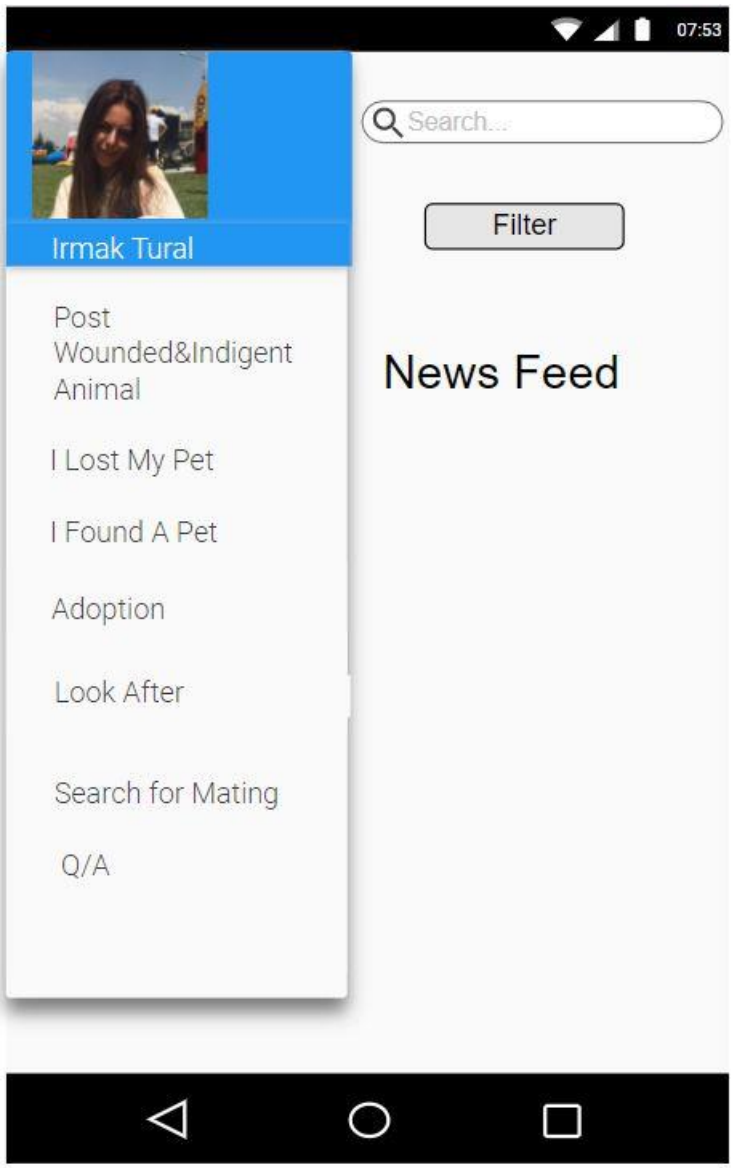
  The user will be able to login to the system with his/her username and password.

- **Emergency Posts**

  User can use core features of the system in case of emergency without signing up. They can post an animal that needs help or s/he can post an animal that might be lost.

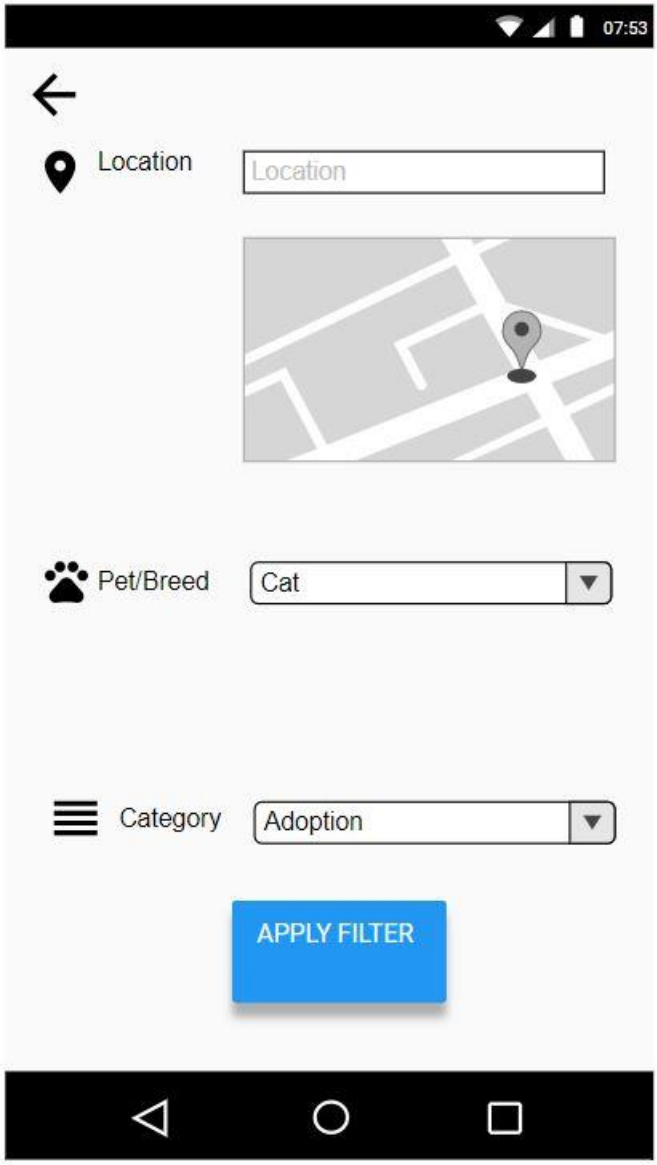
- **Homepage**

  This page is first page when user opens the application after logging in. On this page user can find all post related to adoption, indigent animals, looking after. User can see the details of the posts via clicking on them. Also, user can see the navigation bar via clicking the button located at the top left corner.



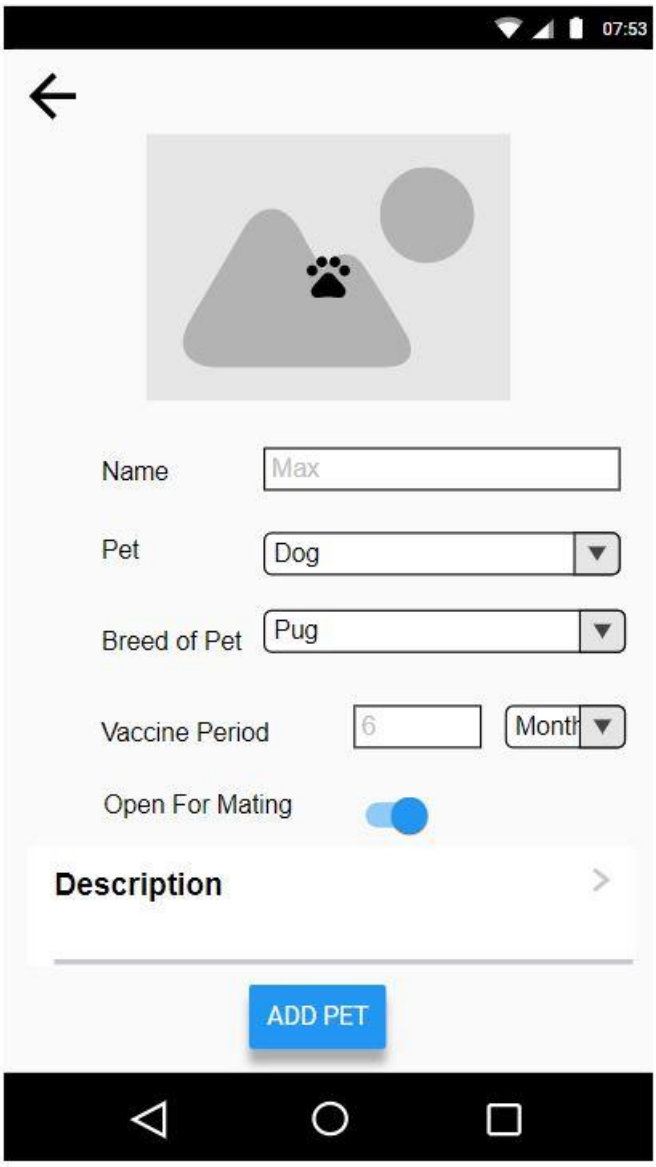
- **Filter Homepage**

  User can filter his/her own homepage according to their preferences. S/he can filter the posts by filterin them according to the location, category or specific animal breed.



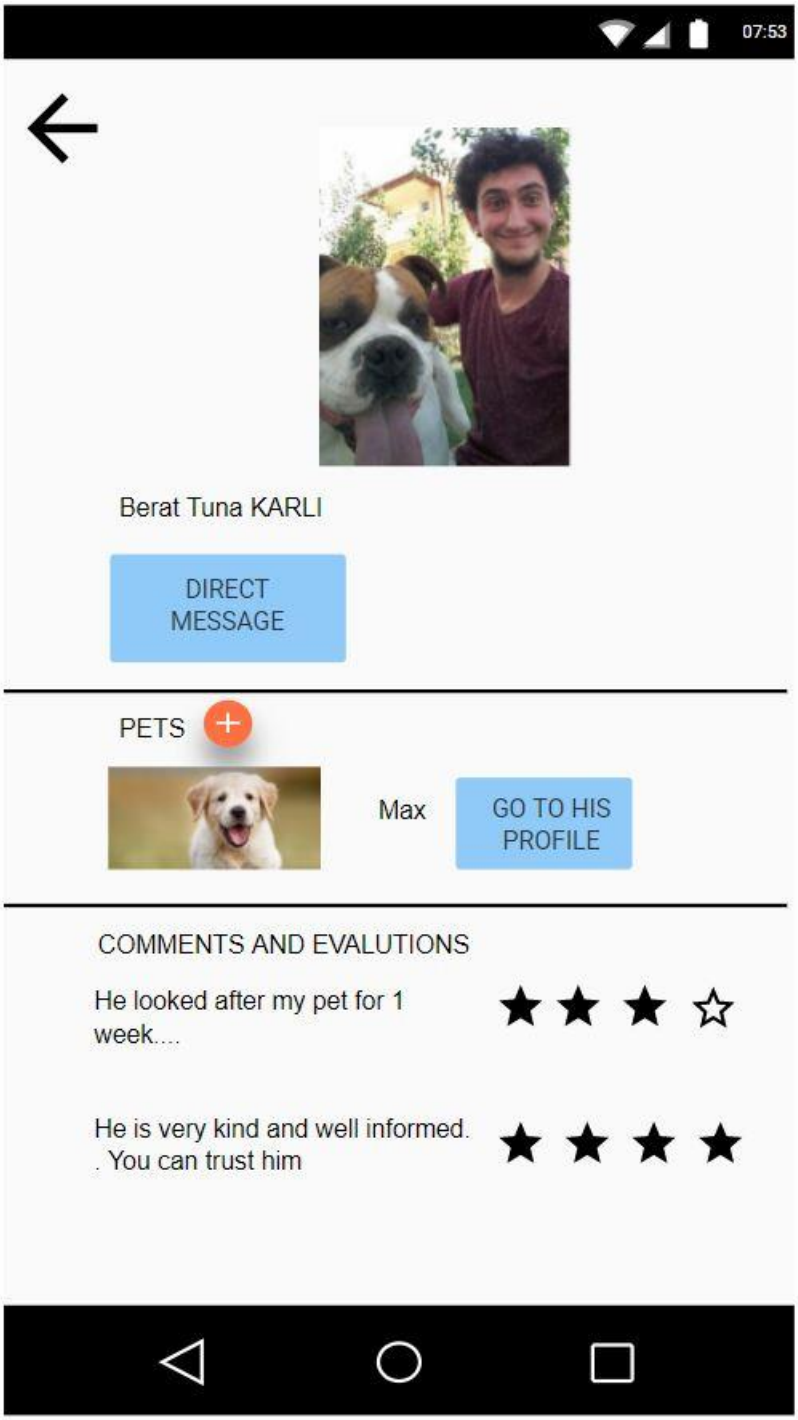
- **Adding a Pet to the Personal Profile**

  User can go to his/her own profile page and add his/her own pet to the profile. User has to fill all required fields to add the pet to their profiles. After adding the pet, other users can see his/her pets on that person's profile.



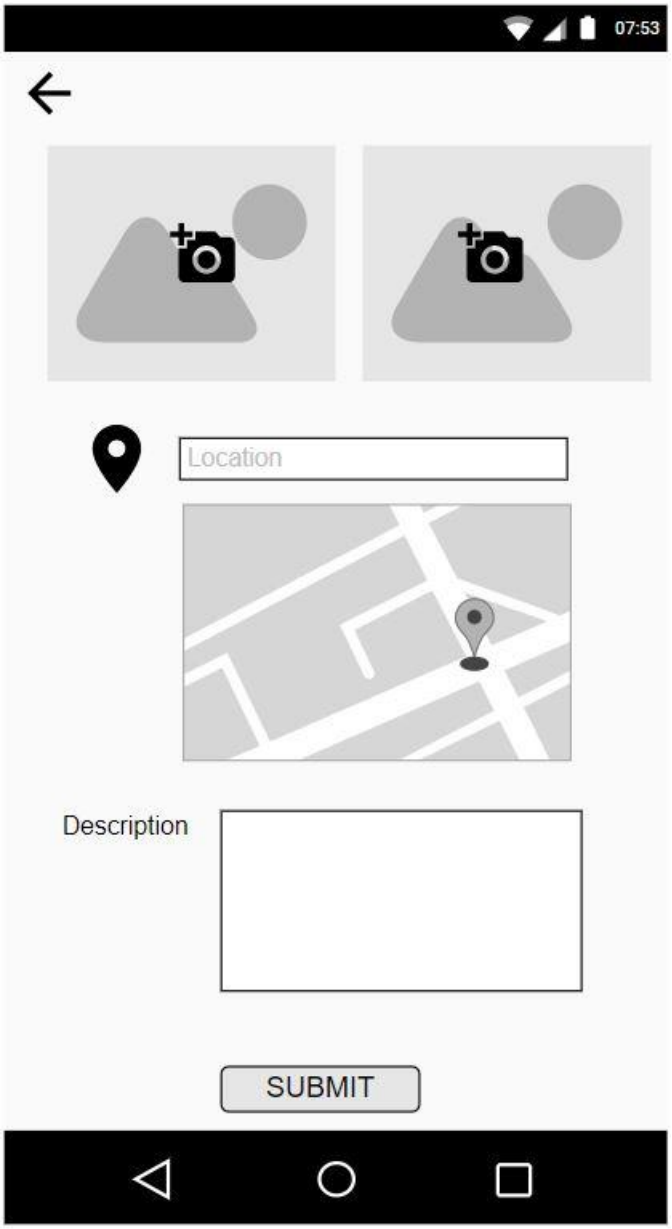
- **User Profile**

  In this page, users can see himself/herself own profile or s/he can see other users' profiles. User can see others' name, surname, pets. Also, comments are located at the bottom of the page, which are written by other users.



- **Submit Lost Pet**

  In this page, users can post a lost animal that s/he found on the street. System will ask user to upload two pictures of the dog, submit the location and optionally description.



- **Post for Adoption**

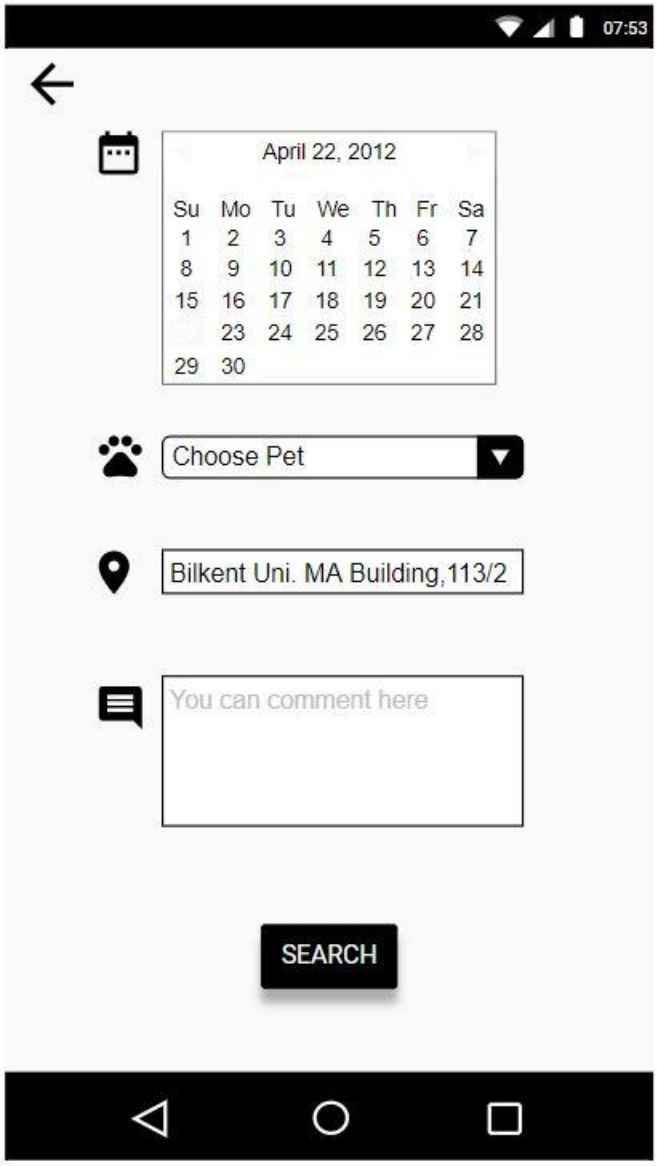
  In this page, user is asked to fill the simple form to submit the adoption post. S/he has to upload a picture. At the same time s/he is asked to enter the breed of the pet, location, gender and a brief description that describes pet's characteristic features.

- **Search to Adopt a Pet**

  In this page, user want to adopt a pet. After s/he search for an adoption based on location, breed, user will see available pets to adopt. If s/he is interested in one of the results, user can see the details of the posts via clicking them.

- **Search for Look After**

  In this page, user search a place for his pet to stay for a certaion period of time. S/he choose a time period from the calendar, a pet, location, and brief description.



4. References

[1] "U.S Pet Ownership Statistics", Avma.org,(2012).[Online]. Available: https://www.avma.org/KB/Resources/Statistics/Pages/Market-research-statistics-US-pet-ownership.aspx. [Accessed: Feb. 17, 2018].

[2] "Lost Pet Statistics: Survey Looks At Likelihood Of Finding A Missing Dog Or Cat" , *The Huffington Post*, November 7, 2012.[Online]. Available: https://www.huffingtonpost.com/2012/07/11/lost-pet-statistics-survey-dog-cat_n_1662860.html. [Accessed: Feb. 17, 2018].

[3] Tamer Dodurka, "Köpek Popülasyonunun Yönetimi", turkvet.biz, May. 2012.[Online]. Available: www.turkvet.biz/yazi/hr_kopek_pop_yonet_dodurka.pps. [Accessed: Feb. 17, 2018].

[4] "Petsbook.com - Evcil dostlarımıza dair aradığınız herşey.", Petsbook.com, 2018. [Online]. Available: https://www.petsbook.com/. [Accessed: March 18, 2018].

[5] "PiP | The Pet Recognition App", Petrecognition.com, 2018. [Online]. Available: http://www.petrecognition.com/. [Accessed: March 18, 2018].

[6] "11pets: Pet Care", 11pets, 2018. [Online]. Available: https://www.11pets.com/. [Accessed: March 18, 2018].

[7] "Pati Birliği", Patibirligi.com, 2018. [Online]. Available: http://patibirligi.com/. [Accessed: 18- Mar- 2018].