Bilkent University

Department of Computer Engineering

Senior Design Project

HelPet

Low Level Design 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/

Table of 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. 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 by the help of the deep-learning pet owners receive the notification if the system notices their animal among the dataset of the animal photos which the other users share. 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 low-level properties and the design of HelPet will be described. Firstly, the design trade-offs of the system and the engineering standards will be discussed. Secondly, packages of the system will be described briefly with diagrams according to the model-view-controller architecture pattern. Thirdly, the class interfaces in all packages will be shown according to model-view-controller architecture.

1.1. Design Trade-Offs

1.1.1. Availability vs. Internet Access

HelPet is an application which is designed for help the animals. Thus, the application must be available all the time in the case of an emergency. However, internet access is the most important part of our system to work and internet may not be accessible in any location.

1.1.2. Complexity vs. Usability

We wanted that HelPet meet all the needs of animal lovers. However, to meet all the needs of animal lovers would increase the complexity of the application and decrease the usability of the application. Thus, we decided that it is more important to provide users with more specific and usable platform so we decided to decrease the complexity and increase the usability of the application.

1.1.3. Performance vs. Memory

The application requires a lot of usage of memory to keep all the information of the application which includes user information, animal information, and dataset of the deep-learning. This excessive usage of memory may cause performance problem.

1.1.4. Reliability vs. Performance

To make the application more reliable, the dataset of deep-learning part must be wide enough which cause a lot of memory usage. This situation may cause some problems with performance. However, we want our application to be more reliable to make the users more satisfied.

1.2. Interface Documentation Guidelines

Class *ClassName*
Description of Class
Properties
AccessModifiers Type AttributeName Example: private String[] keywords;
Methods
AccessModifiers ReturnType MethodName (Parameters) Example: public void updateSearch();

1.3. Engineering Standards

To meet engineering standards, this report is written according to IEEE format. This format is standard and official according to Institute of Electrical and Electronics Engineers for technical texts, articles, and research papers especially about computer science[8]. Also, we used Unified Modeling Language(UML) diagrams to describe the design and working principles of the application. UML is the best and easiest way to describe the classes, actors, roles, actions and the subsystem of the software.

1.4. Definitions, acronyms, and abbreviations

UI : Abbreviation of User Interface.

MVC : Abbreviation of Model-view-controller which is an architectural pattern which divides software into three parts according to the jobs of the subsystems.

SQL : Abbreviation of Structured Query Language which is a programming language for managing the database.

GUI : Abbreviation of Graphical User Interface.

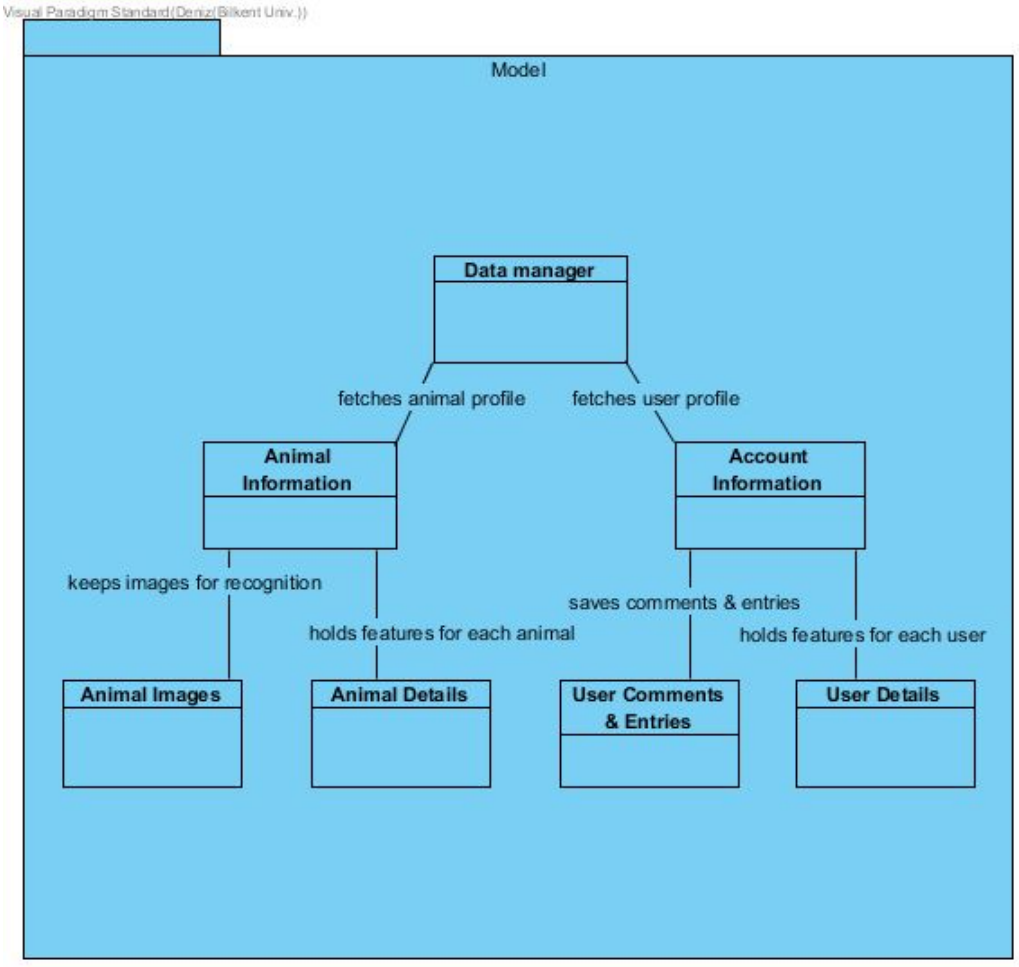
API : Abbreviation of Application Programming Interface.

2. Packages

For Helpet, the chosen architectural pattern is Model-View-Controller(MVC). This pattern separates and classifies the software to three main components that are UI elements to be visualized, data to be hold and requested, and control elements that will pull/push requests to the other components in order to ensure the data flow. MVC pattern provides sufficient and genuine features and it's in a harmony with the design decisions made by us, developers.

Each package is visualized separately. Model package allows queries run by the controller package to access the related animal or user data. Another package View manages the user interface elements. It allows user to navigate through the application by providing the related page views for each panel requested by the user. The third package controller provides the system's main functionality with a collection of the subsystems. Controller acts as a bridge between model and view packages, getting the data from model, processing it, and demonstrating the results with the view package functionalities.

2.1 Model



This package hosts the system data which is requested by controller. Both animal and user data are stored with the help of this package.

Data Manager: Classifies the data as two main parts which are animal and account data for user. Responses to queries run by the controller package.

Animal Information: Holds the information of pets and lost animals enlisted to the system by the registered or unregistered users.

Animal Details: Keeps the detailed, descriptive features about the animals.

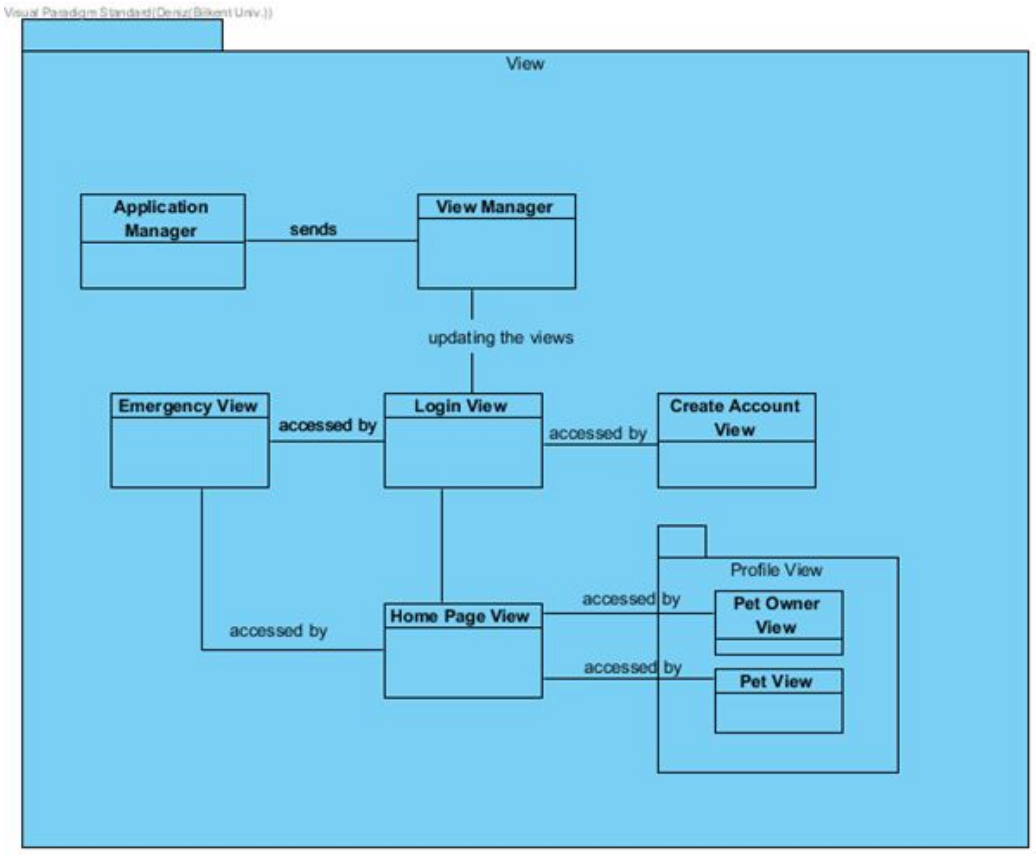
Animal Images: The image data kept for the animal matching algorithm which is dedicated to find the lost animals. Besides, animal images are used by the pet owners, allowing them to show them in their animals' profile page.

Account Information: Information which belongs to the registered users are stored in the database of the application.

Users Comments & Entries: Each entry or comment posted by the application users are stored in the database system.

User Details: Personal information of the users are stored in the system as well. This information is available in the profile view of any user.

2.2. View



UI elements are collected under the package called View.

Application Manager: Linking the controller package to the view package as the main functionality, in addition, controls the element called view manager that directs two distinctive views referred as Login View and Home Page View.

Login View: The first page that user comes across. Allows user to enter the system by evaluating the username and password inputs of the user. It's bound to the Create Account View.

Create Account View: Navigates an unregistered user, allowing the user to register in system.

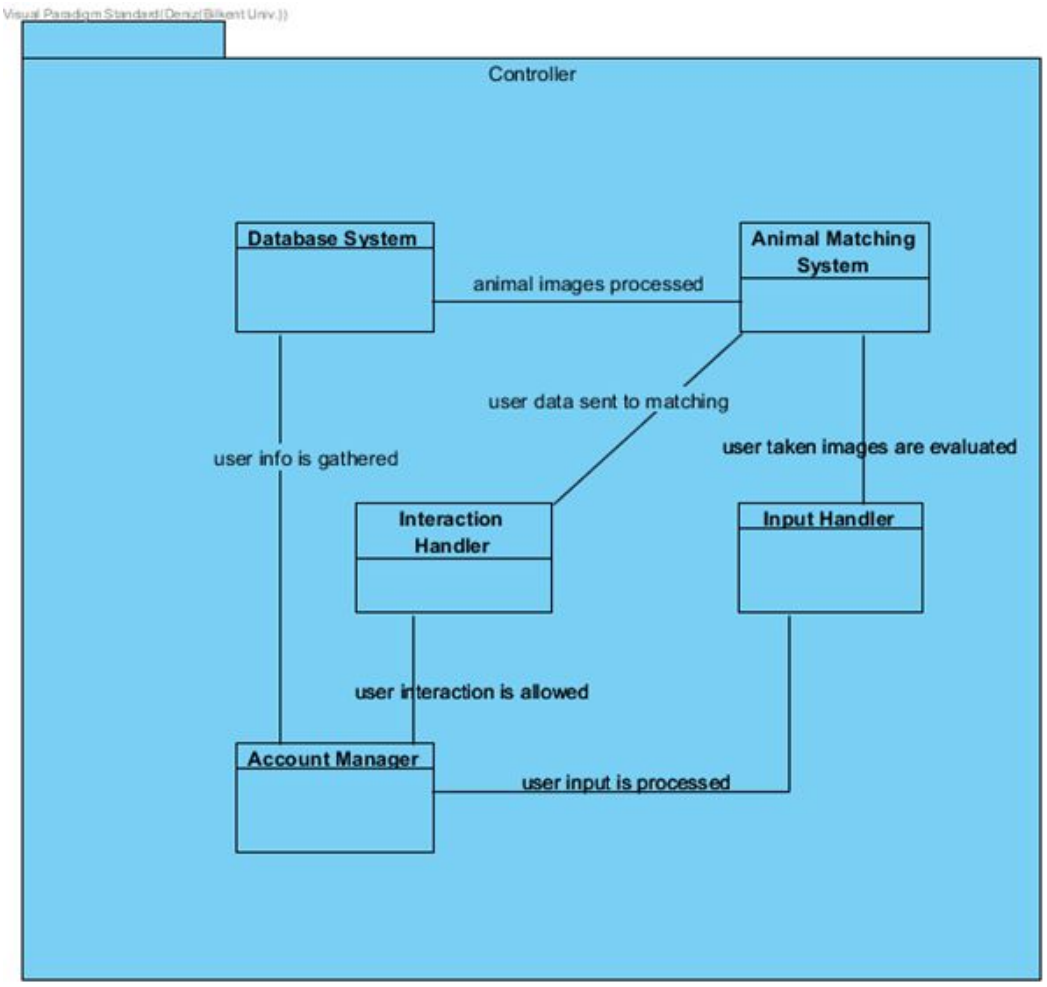
Home Page View: Hosts the Profile View and Emergency View, which direct the user
through event driven functions.

Profile View: Gives characteristic information about pet owner and his/her pets. Can be accessed by the other users.

Emergency View: Allows both registered and unregistered users to access Animal in need and I found a lost animal functions. The second function relies on the image and location inputs fed to the system by the user.

View Manager: Works in correlation with application manager. Grants access to the of views listed above.

2.3.Controller



Apart from other packages, there is also a controller package which is reserved for main software functionalities. In this package the components are described below:

Animal Matching System: Serves the main purpose of the program. Once the animal images are taken from database system, their matching with user taken images are evaluated with deep learning.

Database System: Manages the queries for getting/removing the data related to both animals and users in view package.

Interaction Handler: User interaction is validated by this handler. Event driven functions are interpreted.

Input Handler: Fetches the account related data and directs it to the matching algorithm.

Account Manager: Allows user to practice the functionalities described above, correlating with database and the handlers.

3. Class Interfaces

In this section, every class of HelPet will be explained in a detailed way including their properties and methods. There will be three sub-sections of this section because our implementation is using Model-View-Controller architectural pattern.

3.1. Model

This section includes interfaces of all classes that are located in the Model Package. The interface includes the hierarchical placement of the classes, their definitions, properties, and all of their methods.

Class *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.
Properties
Public String tableName Public Animal animal Public User user
Methods
void CreateTable(String tableName) void insert(Animal animal) void insert(User user) void delete(Animal animal) void show(Animal animal) Void setPetStatus()

Class *User*
This class includes all relevant functions and information about users.
Properties
Private String id;

Private String password; Private String email; Private Image profilePic; Private String name; Private String msg;
Methods
Public void sendMessage(user, msg) Public void changeProfileInfo(email, name)

Class *Post*
This class includes all relevant functions and information about posts.
Properties
Private String postId; Private User user; Private Image pic; Private String text; Private Animal animalInfo;
Methods
Public Void createPost(post) Public void updatePost(post) Public void sendMessage(user, msg) Public void changeProfileInfo(email, name)

Class *Animal*
This class includes all relevant functions and information about animals.
Properties

Private String name; Private int age; Private String species; Private String sex;
Methods
Public void updateAge(age)

3.2. View

This section includes interfaces of all classes that are located in the Views Package. The interface includes the hierarchical placement of the classes, their definitions, properties, and all of their methods. The classes in the view unit take inputs from the user as well ass gives the final output the user. These inputs are managed by controller unit.

Class *Homepage*
This class includes all relevant functions and information about homepage.
Properties
private Toolbar mainToolbar; private FirebaseAuth mAuth; private DrawerLayout drawer;
Methods
protected void onStart() public void onBackPressed() protected void onCreate public boolean onOptionsItemSelected(MenuItem item) private void logOut() private void sendToLogin() public void onPPClick(View v) public boolean onNavigationItemSelected(MenuItem item)

Class *SetupActivity*
This class includes all relevant functions and information about users profile page..
Properties

private CircleImageView setupImage; private EditText setupName; private StorageReference storageReference; private FirebaseAuth firebaseAuth; private FirebaseFirestore firebaseFirestore; private Bitmap compressedImageFile;
Methods
protected void onCreate(Bundle savedInstanceState) private void BringImagePicker() protected void onActivityResult(int requestCode, int resultCode, Intent data)

3.3. Controller

This section includes interfaces of all classes that are located in the Controller Package. Controller package acts like a bridge between View and Model whereas it is the logical processing unit with complicated algorithms for the application.

Class *ImageRecognition*
This class includes all relevant functions and information about images of pets.
Properties
Private Image image1; Private Image image2;
Methods
Public boolean matchImages(image1, image2);

Class *AccountManager*
This class includes all relevant functions about users actions.

Properties
Private User user;
Methods
Public void register(user) Public void login() Public void logout() Public void notifyAdmin(user) Public void changeCredentials()

Class *Search*
This class includes all relevant functions about searches that users make.
Properties
Private Filter filter; Private String[] keywords;
Methods
Public void scanArea() Public void findPetToAdopt() Public void findMate() Public void findPlaceForPet()

Class *filter*
This class includes all relevant functions about users actions.
Properties
Private String[] keywords;
Methods
Public void updateSearch()

4. Glossary

Image Recognition:Image recognition is the process of identifying and detecting an object or a feature in a digital image or video.A machine learning approach to image recognition involves identifying and extracting key features from images and using them as input to a machine learning model.

Firebase:.Firebase is a mobile and web app development platform that provides developers with a plethora of tools and services to help them develop high-quality apps, grow their user base, and earn more profit. It provides a realtime database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud.

Graphical User Interface: GUI is the interface that interacts with users with dynamic images, buttons and labels. Users can navigate between pages by using the interface with simple operations such as clicking a button. The device responds to the user according to the operation and the given input.

5. 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].

[8] “IEEE Documentation Style”, Wisc.edu,(2012).[Online].

Available:

https://writing.wisc.edu/Handbook/DocIEEE.html.[Accessed:October10, 2018]