

Software Requirements Specification

08.04.2022

Elif Nur Kemiksiz 100217006

Emirhan Uslu 150119730

Mehmet Alper Karabay 150119044

Muhammet Eren Atala 150119904

Nesrin Şimşek 150119664

Prepared for

CSE3044 Software Engineering Term Project

**Table of Contents**

**1. INTRODUCTION 3**

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

1.4 References 3

1.5 Overview 3

**2. GENERAL DESCRIPTION 4**

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Characteristics 4

2.4 General Constraints 4

2.5 Assumptions and Dependencies 4

**3. SPECIFIC REQUIREMENTS 4**

3.1 External Interface Requirements 5

*3.1.1 User Interfaces 5*

*3.1.2 Hardware Interfaces 5*

*3.1.3 Software Interfaces 5*

*3.1.4 Communications Interfaces 5*

3.2 Functional Requirements 5

*3.2.1 Login Feature 5*

*3.2.2 Registration Feature 5*

*3.2.3 Forum Feature*

*3.2.4 Profile Feature*

*3.2.5 Browse Feature*

*3.2.6 Missing-Ad. Feature*

*3.2.7 Pet Adoption Feature*

*3.2.8 Filter Feature*

*3.2.9 Animal Help Feature*

*3.2.10 Post Sharing Feature*

3.3 Non-Functional Requirements 5

*3.3.1 Performance 6*

*3.3.2 Reliability 6*

*3.3.3 Availability 6*

*3.3.4 Security 6*

*3.3.5 Maintainability 6*

*3.3.6 Portability 6*

3.4 Inverse Requirements 6

3.5 Design Constraints 6

3.6 Logical Database Requirements 6

3.7 Other Requirements 6

**4. UML DIAGRAMS 6**

4.1 Use Cases 6

*4.1.1 Use Case #1 6*

*4.1.2 Use Case #2 6*

4.2 Classes / Objects 6

*4.2.1 <Class / Object #1> 6*

*4.2.2 <Class / Object #2> 6*

4.3 Sequence Diagrams 7

4.4 Data Flow Diagrams (DFD) 7

4.5 State-Transition Diagrams (STD) 7

**A. APPENDICES 7**

A.1 Appendix 1 7

A.2 Appendix 2 7

**1. Introduction**

In this Software Requirements Specification(SRS), a mobile application 'Petiverse' will be introduced. In this document, the necessary materials and information about Petiverse for SRS will be collected and presented. In addition, many departments will be introduced, for example General Constraints, necessary information will be given.

**1.1 Purpose**

Defining and describing the functions and specifications of the Petiverse mobile application is the primary goal of this Software Requirements Specification.This Software Requirements Specification illustrates, in clear terms, the system’s primary uses and required functionality as specified by our customer. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

**1.2 Scope**

The software system being produced is called Petiverse. It is being produced for term project of Marmara University CSE3044 lecture. This system is designed to provide a social network among animal lovers and pet owners to support animals in need and create a network to share and learn new information.

The Petiverse mobile application will allow any user to create an account to become a member. This creation process optionally includes sub creation of pet profile which users have. The system will allow members to browse, search, filter and share new posts to the network. The system will have different types of posts to be posted which are pet friends, missing ads, animals in need (with pictures and location). That way users will be able to communicate and aid each other easily. The forum (blog) side of the system will have different categories such as type of animal, topics etc. that way users will be capable of finding what they need more quickly.

**1.3 Definitions, Acronyms, and Abbreviations**

**Android:** Android is a Linux-based operating system developed for mobile devices.

**iOS:** iOS is an operating system for running iPhone, iPad and iPod Touch devices.

**Flutter:** Flutter is a SDK that enables mobile application development for platforms such as Android and iOS.

**Firebase:** Firebase is a platform where backend operations such as data hold/fetch, realtime database, and authentication operations can be performed.

**Dart:** Dart is a programming language in which web, server, mobile applications and IoT devices can be developed.

**User Account:** A user account is allocated on a network server used to store a computer username, password, and other information. A user account allows or does not allow a user to connect to a network, another computer, or other share. Any network that has multiple users requires user accounts.

**Mobile Operating System:** A mobile operating system, also called a mobile OS, is an operating system that is specifically designed to run on mobile devices such as mobile phones, smartphones, tablet computers and other handheld devices.

**Database:** A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content***.***

**Browse:** In general, browse refers to reading and scanning through data. Browsing is commonly used to describe when a user reads through web pages on the Internet and is also often referred to as surfing. For example, you're browsing the Internet with an Internet browser now if you're reading this page on your computer. If you have visited more than one page on Computer Hope, you are browsing our website.

**Search:** Sometimes referred to as seek, a search is a function or process of finding letters, words, files, web pages, or other data. Many operating systems, software programs, and websites contain a search or find feature to locate data. For example, the search box for Computer Hope is at the top and bottom of each of our pages. When on the Internet, you use a search engine to search through web pages on the Internet.

**1.4 References**

<https://www.computerhope.com/>

<https://flutter.dev/>

<https://dart.dev/>

<https://jisajournal.springeropen.com/track/pdf/10.1186/s13174-015-0033-5.pdf>

<https://www.cse.msu.edu/~chengb/RE-491/Papers/SRS-BECS-2007.pdf>

<https://tr.wikipedia.org/wiki/Android>

<https://smartpro.com.tr/ios-nedir/>

<https://www.kumsalajans.com/blog/web-ve-mobil-yazilim/flutter-nedir>

<https://www.mobiler.dev/post/flutter-da-firebase-storage-kullanimi>

<https://tr.wikipedia.org/wiki/Dart_(programlama_dili)>

**1.5 Overview**

*This subsection should:*

*(1) Describe what the rest of the SRS contains*

*(2) Explain how the SRS is organized.*

**2. General Description**

This section includes details about what is and is not expected of the Petiverse Mobile Application in addition to which cases are intentionally unsupported and assumptions that will be used in the creation of the application.

**2.1 Product Perspective**

Although there are similar applications for the use of animal lovers, Petiverse distinguishes itself from other applications by gathering features for both socializing and meeting the needs of animals under a single platform and providing ease of use as it is a mobile application.

**2.2 Product Functions**

Users create a profile for themselves by becoming a member of Petiverse for free. Then, after users log in to the application with their credentials, they can view the pages containing the posts for their needs through the directions in the application or they can share on these pages themselves. These posts can be for animal adoption, mating, missing ad, food supply, veterinary needs or exchange of ideas.

**2.3 User Characteristics**

Petiverse users are sensitive to both their own pets and stray animals. Users share their own problems related to animals or voluntarily support each other in solving other users' problems. They also voluntarily come together to meet the instinctive needs of animals, such as mating or socializing.

The typical Petiverse user is simply by anyone that has access to a smart phone. It is assumed that the user is familiar enough with operating a smart phone.

**2.4 General Constraints**

Security concerns? passwords encryption etc.

*This subsection of the SRS should provide a general description of any other items that will*

*limit the developer’s options for designing the system. (See the IEEE Guide to SRS for a partial list of possible general constraints).*

**2.5 Assumptions and Dependencies**

Assumption:

* All of the user’s mobile phones are in proper working condition.
* The user is capable of operating these phones’s basic functions including but not limited to being able to login using the application.

Dependencies:

* The user must have an internet connection to access the application.
* Phone or tablet must have either an android or IOS operating system to run this application.

*This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.*

**3. Specific Requirements**

*This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project’s software design, implementation, and testing.*

*Each requirement in this section should be:*

* *Correct*
* *Traceable (both forward and backward to prior/future artifacts)*
* *Unambiguous*
* *Verifiable (i.e., testable)*
* *Prioritized (with respect to importance and/or stability)*
* *Complete*
* *Consistent*
* *Uniquely identifiable (usually via numbering like 3.4.5.6)*

*Attention should be paid to the carefully organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.*

**3.1 External Interface Requirements**

**3.1.1 User Interfaces**

The user interface for the software shall be compatible to any android smart phone that has Internet connection which user can access to the system. Because this is a mobile application, the user interface design ought to be highly user-friendly. This mobile based application would have a user-friendly interface providing the users with an easy-to-navigate and convenient social experience.

**3.1.2 Hardware Interfaces**

Since the application must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for e.g Modem, WAN-LAN.

(Android system specifications, requirements)

**3.1.3 Software Interfaces**

Divide into to parts

Front-End Technologies

Flutter / Dart

Back-End Technologies

Operating systems: Android / Ios

Database: Firebase

Coding Language: Dart

Framework: Flutter

Editor IDE: Visual Studio Code / Emulator

**3.1.4 Communications Interfaces**

To access the mobile application, the Client must be connected to the Internet.

**3.2 Functional Requirements**

*This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.*

**3.2.1 Login Feature**

3.2.1.1 Introduction / Description

Every user can sign in using their email address. But if they are new user, then they must register in the register feature.

3.2.1.2 Inputs / Display

Users need to enter their email address and password.

3.2.1.3 Processing

User’s password and email address will be checked from firebase.

3.2.1.4 Outputs

If user’s password and email address is correct then the user will enter the system successfully.

3.2.1.5 Constraints

User’s password and email address must match with existing data in firebase.

Password and email parts cannot be empty.

3.2.1.6 Error/Data Handling

If any of the constraints is not satisfied, a related error message will be shown by the system.

**3.2.2 Registration Feature**

3.2.2.1 Introduction / Description

Every user must sign up to enter the system.

3.2.2.2 Inputs / Display

Name, surname, birth date, email address, password.

3.2.2.3 Processing

System saves the data in the firebase.

3.2.2.4 Outputs

If user satisfies the constraints, the system gives a successful message.

3.2.2.5 Constraints

Mandatory sign-up details must be provided and valid.

New accounts with existing account details cannot be created.

User must agree to all terms and conditions laid by Petiverse if the user cannot use the application.

3.2.2.6 Error/Data Handling

If any of the constraints is not satisfied, a related error message will be shown by the system.

**3.2.3 Forum Feature**

3.2.3.1 Introduction / Description

Every user can create a topic and view the posted topic.

3.2.3.2 Inputs / Display

System displays the last 10 topics that were created.

Users can access the filter feature with a button.

Vertical swipe movements.

3.2.3.3 Processing

System gets the last 10 posted topics via firebase. When the user swipes down, this gesture is validated and finally brings new posts to screen.

3.2.3.4 Outputs

View new or past posts.

3.2.3.5 Constraints

If the user at the top of the page swipe down, the gesture will finalize. Also if the user is at the bottom of the page swipe up gesture will not finalize.

3.2.3.6 Error/Data Handling

If any of the constraints is not satisfied, a related error message will be shown by the system.

**3.2.4 Profile Feature**

3.2.1.1 Introduction / Description

All users will have an individual profile that contains user's information like name, age, photo etc.

3.2.1.2 Inputs / Display

User profiles have some mandatory inputs that are name, surname, birthday and email. Users will have to give those inputs while registering. Also, there will be some non-mandatory fields like profile picture, favorite pets. Users will always be able to edit them.

3.2.1.3 Processing

System gets all the information needed as inputs from user and save them in database and creates a profile page to shown

3.2.1.4 Outputs

There will be a profile page that contains all the user information.

3.2.1.5 Constraints

User must be registered to have a valid profile

3.2.1.6 Error/Data Handling

Mandatory information like name and birthday will be checked while registering.

**3.2.5 Browse Feature**

3.2.5.1 Introduction / Description

This feature enables users to browse the corresponding page among posts.

3.2.5.2 Inputs / Display

Vertical swipe movements.

3.2.5.3 Processing

When the user swipes up, this gesture is validated and finally brings past posts to the screen. When the user swipes down, this gesture is validated and finally brings new posts to screen.

3.2.5.4 Outputs

View new or past posts.

3.2.5.5 Constraints

If the user at the top of the page swipe down, the gesture will finalize. Also, if the user is at the bottom of the page, the swipe up gesture will not finalize.

3.2.5.6 Error/Data Handling

If corresponding constraint is not satisfied an error message will be shown such as “There does not exist new post”

**3.2.6 Missing-Ad. Feature**

3.2.6.1 Introduction / Description

This feature shows posted missing-Ads.

3.2.6.2 Inputs / Display

Filter feature.

3.2.6.3 Processing

System gets the last 10 posted missing ads via firebase. When the user swipes down, this gesture is validated and finally brings new posts to screen.

3.2.6.4 Outputs

View new or past ads.

3.2.6.5 Constraints

If the user at the top of the page swipe down, the gesture will finalize. Also, if the user is at the bottom of the page, the swipe up gesture will not finalize.

3.2.6.6 Error/Data Handling

If corresponding constraint is not satisfied an error message will be shown such as “There does not exist new missing ads”

**3.2.7 Pet Adoption Feature**

3.2.7.1 Introduction / Description

This feature will show users the animals they can adopt.

3.2.7.2 Inputs/ Display

Filter feature.

3.2.7.3 Processing

It compiles the information specified by the user and brings the animals according to these characteristics.

3.2.7.4 Outputs

Sees new or past animal adaptation posts according to the specified features.

3.2.7.5 Constraints

The user must specify an animal type.

3.2.7.6 Error/Data Processing

If there is no animal with the specified characteristics. No pet adoption post found.

**3.2.8 Filter Feature**

3.2.8.1 Introduction / Description

Users will be able to filter the outputs according to some information.

3.2.8.2 Inputs / Display

Users can filter the ads and forum posts by giving pet type and-or ad location as inputs.  
3.2.8.3 Processing

System will get a filtering parameter as input from user and filter the results using that parameter

3.2.8.4 Outputs

The output list that is filtered will be shown to the user.

3.2.8.5 Constraints

User must give a valid pet type or valid location as filtering input

3.2.8.6 Error/Data Handling

There will be a default checkbox list for filtering parameters and the user will have to choose one of the valid inputs from that list.

**3.2.9 Animal Help Feature**

3.2.9.1 Introduction / Description

This feature enables users to navigate the animal help page. Users can see nearby animals in need on this page.

3.2.9.2 Inputs / Display

User clicks the animal help page button on the navigation bar.

3.2.9.3 Processing

When the user clicks the button, it will navigate to the page and load the existing animal help posts.

3.2.9.4 Outputs

View animal help page.

3.2.9.5 Constraints:

3.2.9.6 Error/Data Handling:

**3.2.10 Post Sharing Feature**

3.2.10.1 Introduction / Description

This feature exists for the user to share the posts.

3.2.10.2 Inputs / Display

When the user clicks the + sign in the navigation bar, he/she selects the desired category among the displayed missing ad, adoption and help categories.

3.2.10.3 Processing

If the user provides all the necessary steps, the post is shared in the section with the relevant posts.

3.2.10.4 Outputs

After the user shares the post, a message is printed that he/she has successfully shared it.

3.2.10.5 Constraints

• For missing ad categories, the user is required to fill in the animal’s name, species, gender, photo and last location information before the pet is lost.

• For the adoption category, it is obligatory to share the photo, age, gender, species and, if any, diseases of the animal.

• For the help category, it is obligatory to share the information about what the help is needed for and where it will be done.

3.2.10.6 Error/Data Handling

If any of the constraints is not satisfied, a related error message will be printed by the system.

**3.3 Non-Functional Requirements**

*Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).*

**3.3.1 Performance**

Application must be lightweight and must send load posts instantly.

**3.3.2 Reliability**

It is very important that the app is reliable for users who use Petiverse simultaneously. All data collected by Petiverse shall be preserved safely and should follow data hiding.

**3.3.3 Availability**

**3.3.4 Security**

Keep your password safe and don’t share it with any other people, applications, or websites under any circumstances. We also suggest using a different password for every service you use. Besides from Petiverse, we will take all the important measures to make our application safe for our users.

**3.3.5 Maintainability**

Petiverse is a new developing app and hence has maintenance on loop. Petiverse will release all new updates first and look for acceptance from its customers. Petiverse aims to surprise its users by releasing fresh updates with new features.

**3.3.6 Portability**

Petiverse can be used on any mobile device operating on android.

**3.4 Inverse Requirements**

Users will not be able to see the ads without registering to the app Users will not be able to chat with someone directly.

**3.5 Design Constraints**

* Our app will support Android Jelly Bean, v16, 4.1.x or newer, and iOS 8 or newer.
* Flutter currently does not support building for x86 Android directly, however apps built for ARMv7 or ARM64 run fine (via ARM emulation) on many x86 Android devices.

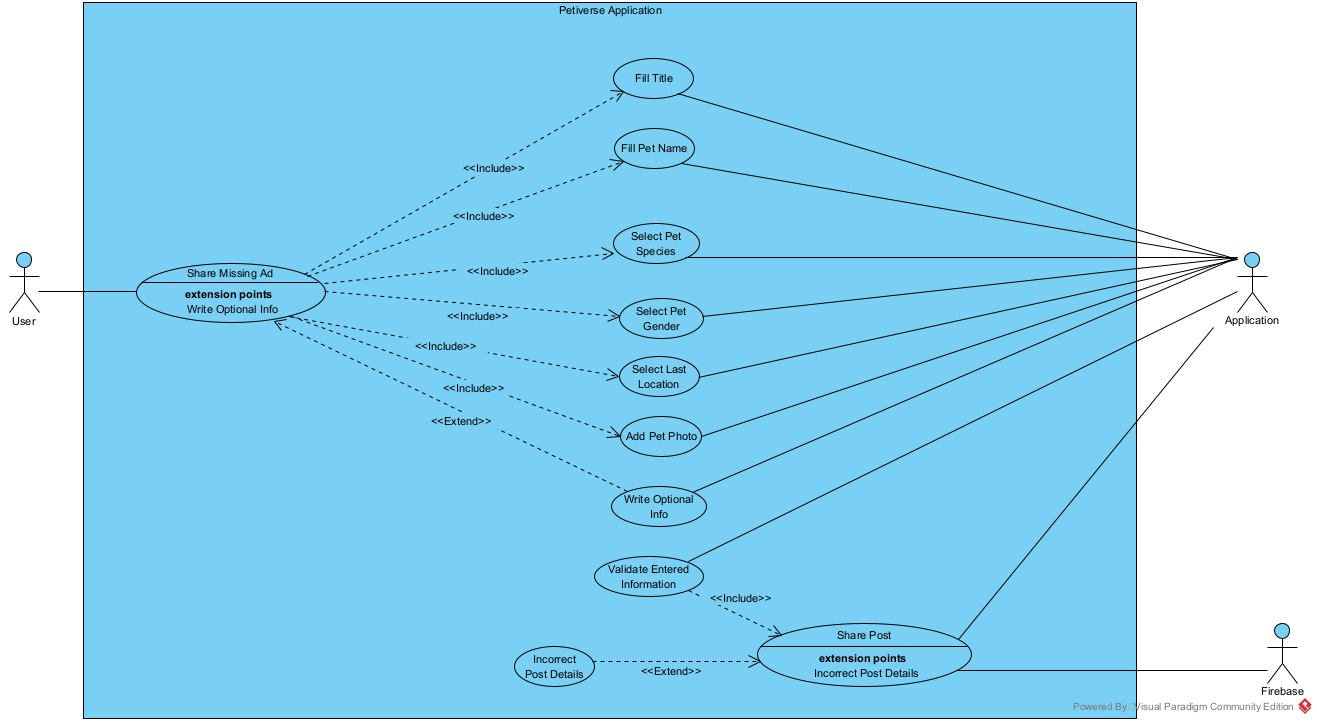
**3.6 Logical Database Requirements**

* Firebase-Firestore, which is a NoSql database, will be used for our project. • All the datas will be stored in JSON type.
* In our database there will be individual collections for user infos and all types of proclamations.
* User’s password will be stored in Firebase as encoded.

**4. UML Diagrams**

**4.1 Use Cases**

**4.1.1 Use Case Share Missing Ad.**



**Use Case: Share Missing Ad**

**Actors:** User, Application

**Description:** This feature exists for the user to share missing ads.

**Includes:** Fill Title, Fill Pet Name, Select Pet Species, Select Pet Gender, Select Last Location, Add Pet Photo

**Extends:** None

**Use-Cases:** None

**Use Case: Fill Title**

**Actors:** User, Application

**Description:** This feature exists for the user to fill title.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click share missing ad button.

**Use Case: Fill Pet Name**

**Actors:** User, Application

**Description:** This feature exists for the user to fill in their pet name.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click share missing ad button.

**Use Case: Select Pet Species**

**Actors:** User, Application

**Description:** This feature exists for the user to select pet species.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click share missing ad button.

**Use Case: Select Pet Gender**

**Actors:** User, Application

**Description:** This feature exists for the user to select pet gender.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click share missing ad button.

**Use Case: Select Last Location**

**Actors:** User, Application

**Description:** This feature exists for the user to select the last location before pet is lost.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click share missing ad button.

**Use Case: Add Pet Photo**

**Actors:** User, Application

**Description:** This feature exists for the user to add pet photo.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click share missing ad button.

**Use Case: Write Optional Info**

**Actors:** User, Application

**Description:** This feature exists for the user to write optional info.

**Includes:** None

**Extends:** Share Missing Ad

**Use-Cases:** User must click share missing ad button.

**Use Case: Validate Entered Information**

**Actors:** Application

**Description:** This feature exists for the application to validate entered information.

**Includes:** Share Post

**Extends:** None

**Use-Cases:** User must click share missing ad button.

**Use Case: Share Post**

**Actors:** Application, Firebase

**Description:** This feature exists for the application to share post.

**Includes:** None

**Extends:** Incorrect Post Details

**Use-Cases:** User must click share missing ad button.

**Use Case: Incorrect Post Details**

**Actors:** Application, Firebase

**Description:** This feature exists for the application to print warning messages.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click share missing ad button.

**4.1.2 Use Case Share Pet Adoption**

**Use Case: Share Adoption Post**

**Actors:** User, Application

**Description:** This feature exists for the user to share pet adoption posts.

**Includes:** Add Photo, Type, Age, Gender

**Extends:** None

**Use-Cases:** Users must click the share adoption post button.

**Use Case: Add Photo**

**Actors:** User, Application

**Description:** This feature exists for the user to add a photo into the post.

**Includes:** None

**Extends:** None

**Use-Cases:** Users must click the share adoption post button.

**Use Case: Type**

**Actors:** User, Application

**Description:** This feature exists for the user to fill animal type.

**Includes:** None

**Extends:** None

**Use-Cases:** Users must click the share adoption post button.

**Use Case: Age**

**Actors:** User, Application

**Description:** This feature exists for the user to select the age of the animal.

**Includes:** None

**Extends:** None

**Use-Cases:** Users must click the share adoption post button.

**Use Case: Gender**

**Actors:** User, Application

**Description:** This feature exists for the user to select animal gender.

**Includes:** None

**Extends:** None

**Use-Cases:** Users must click the share adoption post button.

**Use Case: Diseases**

**Actors:** User, Application

**Description:** This feature exists for the user to select diseases of the animal if any.

**Includes:** None

**Extends:** Share Adoption Post

**Use-Cases:** Users must click the share adoption post button.

**Use Case: Breed**

**Actors:** User, Application

**Description:** This feature exists for the user to select the breed of the animal.

**Includes:** None

**Extends:** Share Adoption Post

**Use-Cases:** Users must click the share adoption post button.

**Use Case: Validate Entered Information**

**Actors:** Application

**Description:** This feature exists for the application to validate entered information.

**Includes:** Share Post

**Extends:** None

**Use-Cases:** Users must click the share adoption post button.

**Use Case: Share Post**

**Actors:** Application, Firebase

**Description:** This feature exists for the application to share posts.

**Includes:** None

**Extends:** None

**Use-Cases:** Users must click the share adoption post button.

**Use Case: Incorrect Post Details**

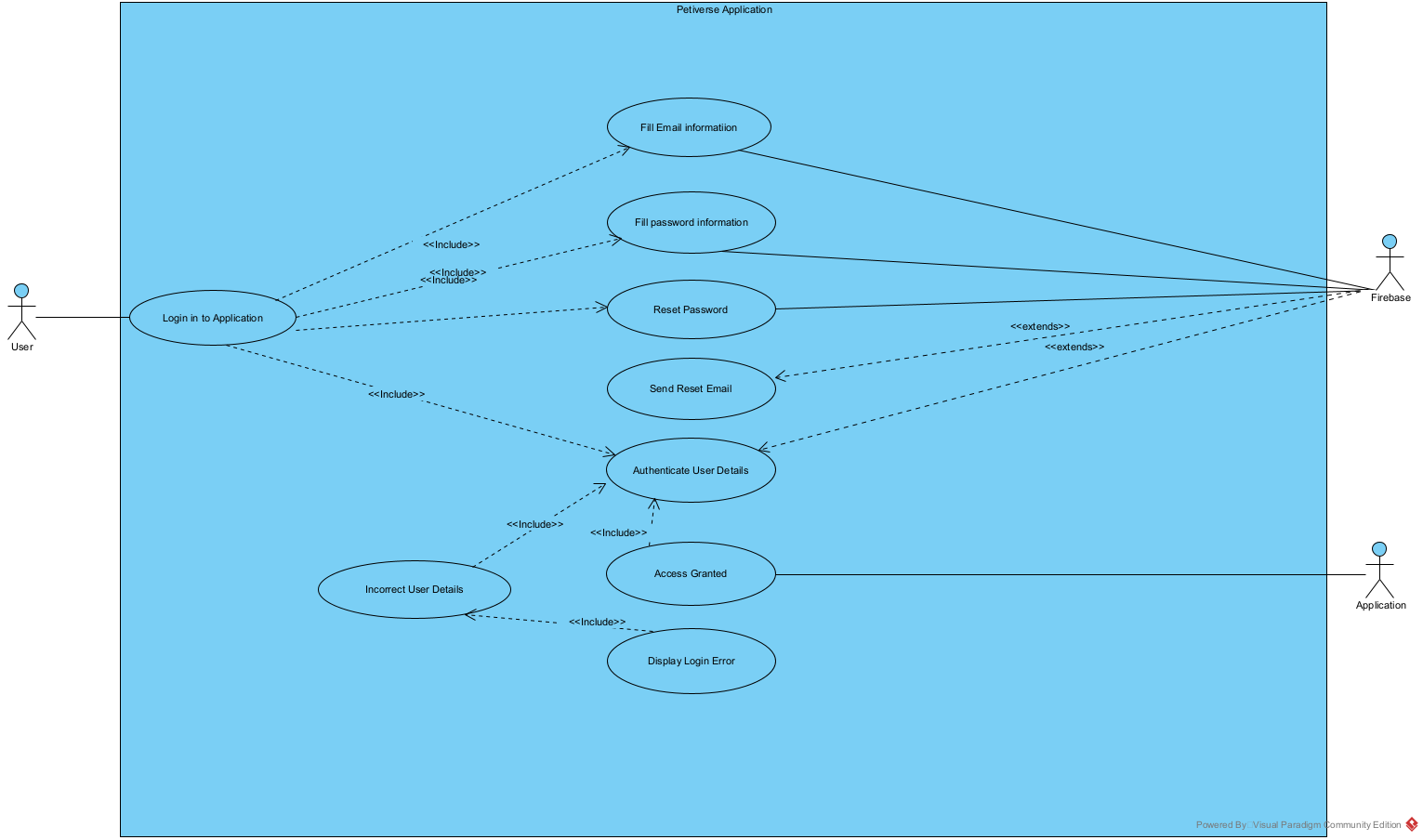
**Actors:** Application, Firebase

**Description:** This feature exists for the application to print warning messages.

**Includes:** None

**Extends:** Incorrect Post Details

**Use-Cases:** Users must click the share adoption post button.

**4.1.3 Use Case User Login **

**Use Case: Login in to Application**

**Actors:** User, Application

**Description:** This feature exists for the user to login to the application.

**Includes:** Fill Email information, Fill password information,Authenticate User details

**Extends:** None

**Use-Cases:** User must click login button

**Use Case: Fill Email information**

**Actors:** User, Application

**Description:** This feature exists for the user to enter their email information.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click login button

**Use Case: Fill Password information**

**Actors:** User, Application

**Description:** This feature exists for the user to fill the password information.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click login button

**Use Case: Reset Password**

**Actors:** User, Application

**Description:** This feature exists for the user to reset his/her password.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click reset password Button

**Use Case: Sent Reset Email**

**Actors:** User,Firebase

**Description:** This feature exists for the system to send email to the user.

**Includes:** None

**Extends:** Authenticate User Details

**Use-Cases:** User must click reset password button

**Use Case: Authenticate User Details**

**Actors:** User, Application

**Description:** This feature exists to verify user information.

**Includes: None**

**Extends: None**

**Use-Cases:** User must click the login button.

**Use Case: Incorrect User Details**

**Actors:** User, Application

**Description:** This feature exists to show incorrect user information.

**Includes:** Authenticate User details.

**Extends:** None

**Use-Cases:** User must click login button

**Use Case: Display Login Error**

**Actors:** User, Application

**Description:** This feature exists to show incorrect user information.

**Includes:** Incorrect User Details

**Extends:** None

**Use-Cases:** User must click login button

**Use Case: Access Granted**

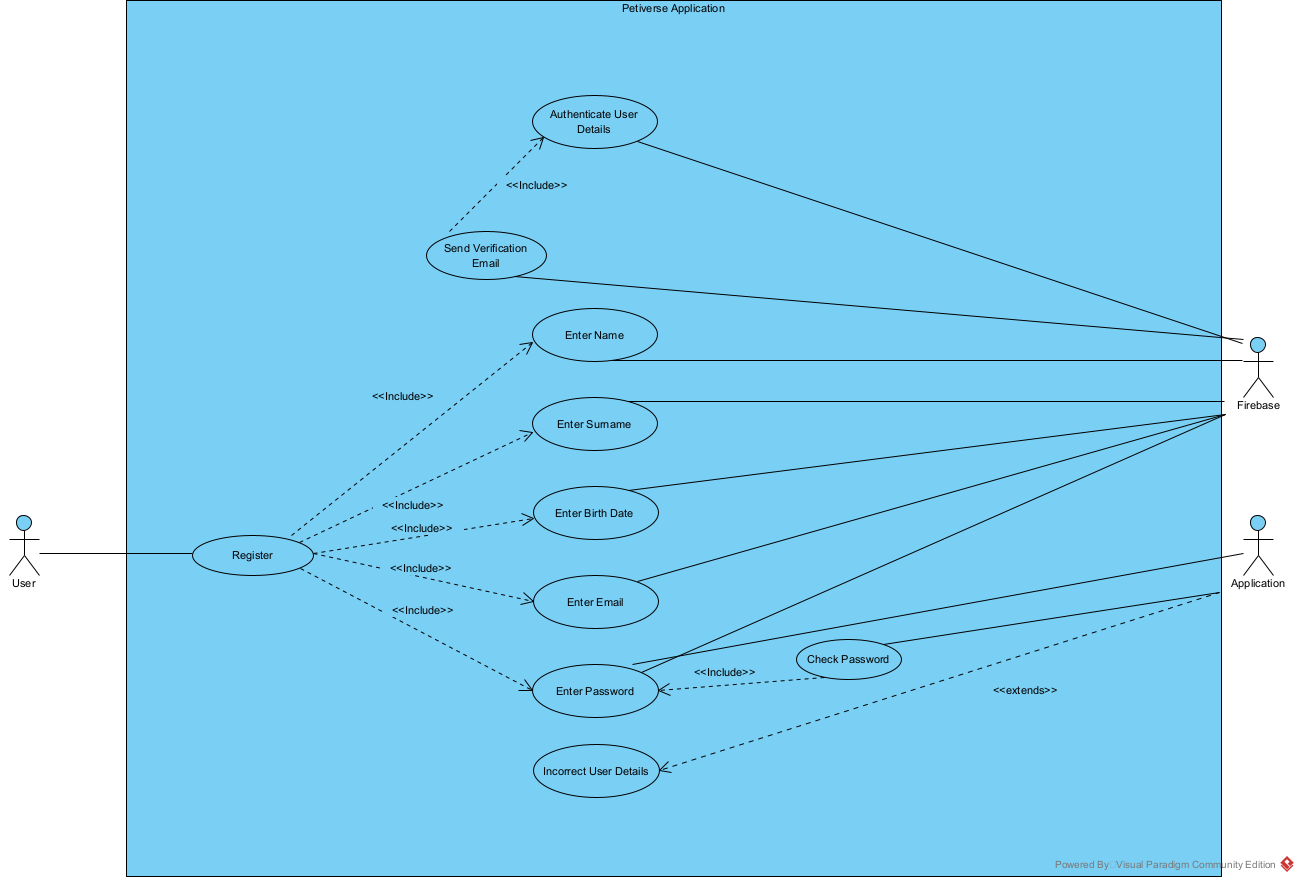
**Actors:** User, Application

**Description:** This feature exists to indicate that the user has entered their information correctly and logged into the application.

**Includes:** Authenticate User details.

**Extends: None**

**Use-Cases:** User must click login button

**4.1.4 Use Case Register**

**Use Case: Register**

**Actors:** User

**Description:** This feature exists for the user to register to the system.

**Includes:** Enter Name, Enter Surname, Enter Birth Date, Enter Email, Enter Password

**Extends:** None

**Use-Cases:** User must click register button

**Use Case: Enter Name**

**Actors:** User, Firebase

**Description:** User fills name field, related data will be stored on firebase.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click register button

**Use Case: Enter Surname**

**Actors:** User, Firebase

**Description:** User fills Surname field, related data will be stored on firebase.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click register button

**Use Case: Enter Birth Date**

**Actors:** User, Firebase

**Description:** User fills Birth Date field, related data will be stored on firebase.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click register button

**Use Case: Enter Email**

**Actors:** User, Firebase

**Description:** User fills Email field, related data will be processed and stored on firebase.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click register button

**Use Case: Enter Password**

**Actors:** User, Firebase,Application

**Description:** User fills Password field, related data will be stored on firebase

**Includes:** Check Password

**Extends:** None

**Use-Cases:** User must click register button

**Use Case: Check Password**

**Actors:** Application

**Description:** Application checks if entered password is valid

**Includes:** None

**Extends:** None

**Use-Cases:** User must click register button

**Use Case: Incorrect User Details**

**Actors:** Application

**Description:** If entered user details are not valid or a given user already exists, the application raises an error.

**Includes:** None

**Extends:** Application

**Use-Cases:** User must click register button

**Use Case: Send Verification Email**

**Actors:** Firebase

**Description:** Firebase send a verification email to user

**Includes:** Authenticate User Details

**Extends:** None

**Use-Cases:** User must click register button

**Use Case: Authenticate User Details**

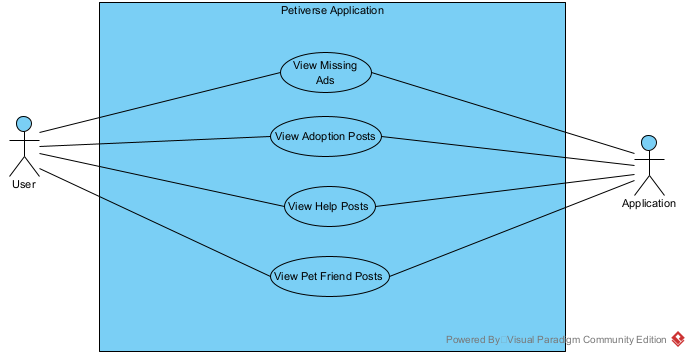
**Actors:** Firebase

**Description:** Firebase verifies given email address is valid

**Includes:** None

**Extends:** None

**Use-Cases:** User must click register button

**4.1.5 Use Case View**

**Use Case: View Missing Ads**

**Actors:** User, Application

**Description:** This feature exists for the user to view missing ads.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click the missing ads post button.

**Use Case: View Adoption Posts**

**Actors:** User, Application

**Description:** This feature exists for the user to view adoption posts.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click the adoption button.

**Use Case: View Help Posts**

**Actors:** User, Application

**Description:** This feature exists for the user to view help posts.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click the help button.

**Use Case: View Pet Friend Posts**

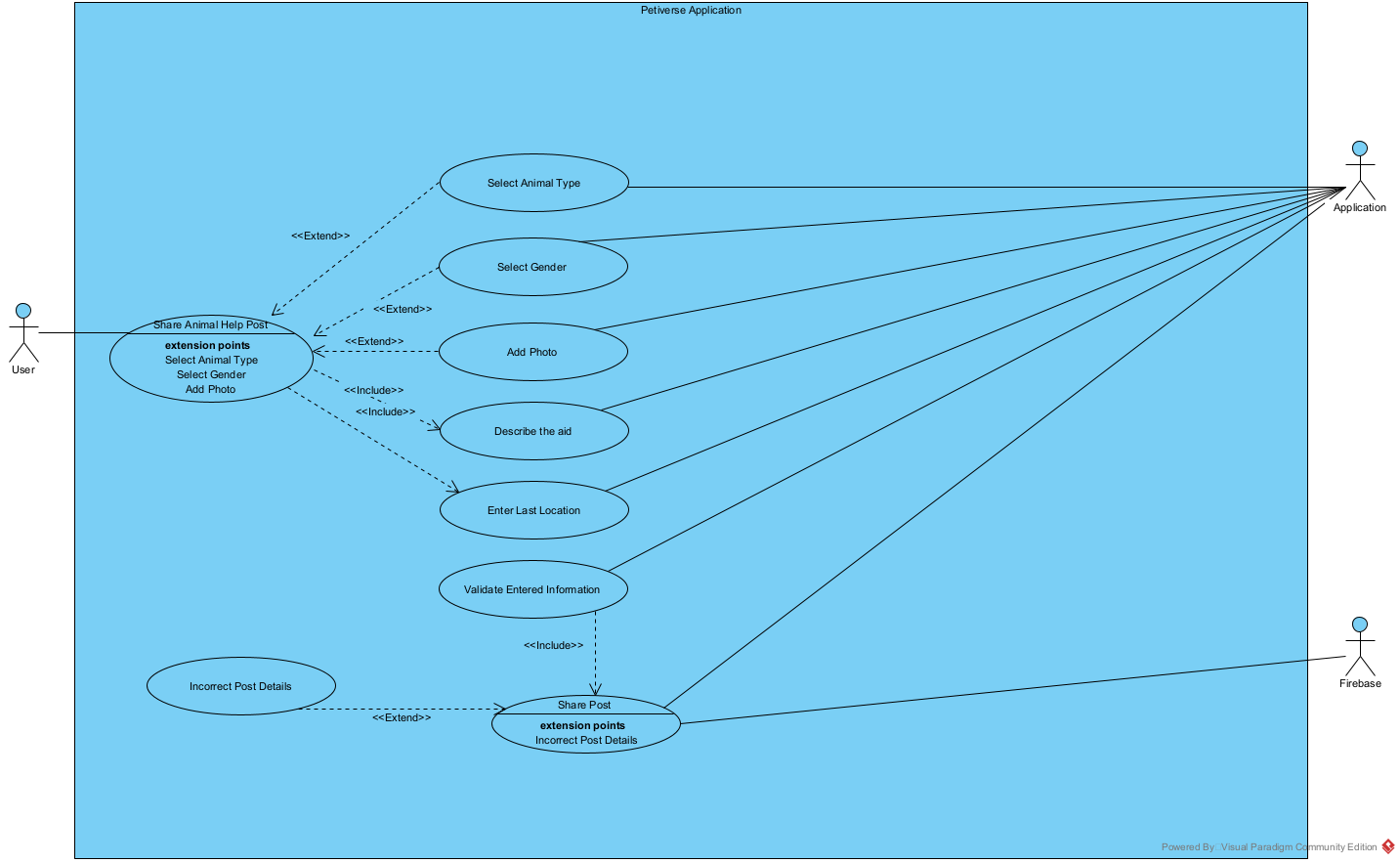
**Actors:** User, Application

**Description:** This feature exists for the user to view pet friend posts.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click the pet friend button.

**4.1.6 Use Case Share Animal Help**

**Use Case: Share Animal Help Post**

**Actors:** User

**Description:** When a user clicks the share animal help post button this feature enables.

**Includes:** Enter Last Location, Describe the aid

**Extends:**

**Use-Cases:** User must click Share Animal Help button

**Use Case: Select Animal Type**

**Actors:** User, Application

**Description:** User selects animal type.

**Includes:** None

**Extends:** Share Animal Post

**Use-Cases:** User must click Share Animal Help button

**Use Case: Select Gender**

**Actors:** User, Application

**Description:** User selects gender of the animal.

**Includes:** None

**Extends:** Share Animal Post

**Use-Cases:** User must click Share Animal Help button

**Use Case: Add Photo**

**Actors:** User, Application

**Description:** User adds a photo of the animal in need.

**Includes:** None

**Extends:** Share Animal Post

**Use-Cases:** User must click Share Animal Help button

**Use Case: Describe the aid**

**Actors:** User, Application

**Description:** User briefly describes the aid that animal needs.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click Share Animal Help button

**Use Case: Enter Last Location**

**Actors:** User, Application

**Description:** User enters last known location of the animal.

**Includes:** None

**Extends:** None

**Use-Cases:** User must click Share Animal Help button

**Use Case: Incorrect Post Details**

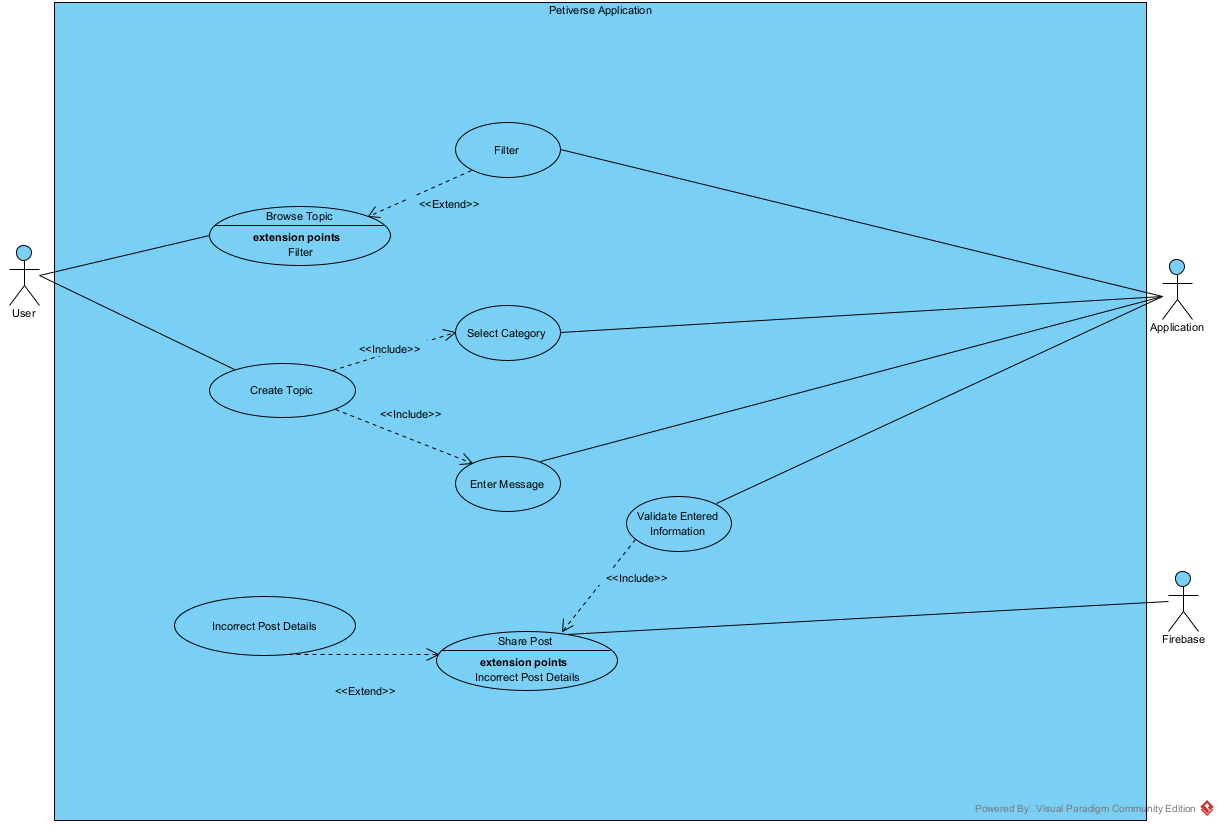
**Actors:** Application

**Description:** If the given informations are not valid application raise an error before sharing post

**Includes:**

**Extends:** Share Post

**Use-Cases:** User must click Share Animal Help button

**4.1.7 Use Case Forum**

**Use Case: Forum**

**Actors:** User, Application

**Description:** This feature exists for creating a topic and viewing the posts.

**Includes:** Select Category, Enter Message, Validate Entered Information.

**Extends:** Incorrect Post Detail, Filter

**Use-Cases:** None

**Use Case: Select Category**

**Actors:** User**,** Application

**Description:** This feature exists for select categories for creating topics.

**Includes:** None.

**Extends:** None.

**Use-Cases:** Users should click the multi choice bar.

**Use Case: Enter Message**

**Actors:** User, Application

**Description:** This feature exists for writing topic content**.**

**Includes:** None

**Extends:** None**.**

**Use-Cases:** Users should enter the text message for the topic.

**Use Case: Validate Entered Information**

**Actors:** Application

**Description:** This feature exists for the application to validate entered information.

**Includes:** Share Post

**Extends:** None

**Use-Cases:** Users must click the share adoption post button.

**Use Case: Share Post**

**Actors:** Application, Firebase

**Description:** This feature exists for the application to share posts.

**Includes:** None.

**Extends:** None.

**Use-Cases:** Users must click the share adoption post button.

**Use Case: Incorrect Post Details**

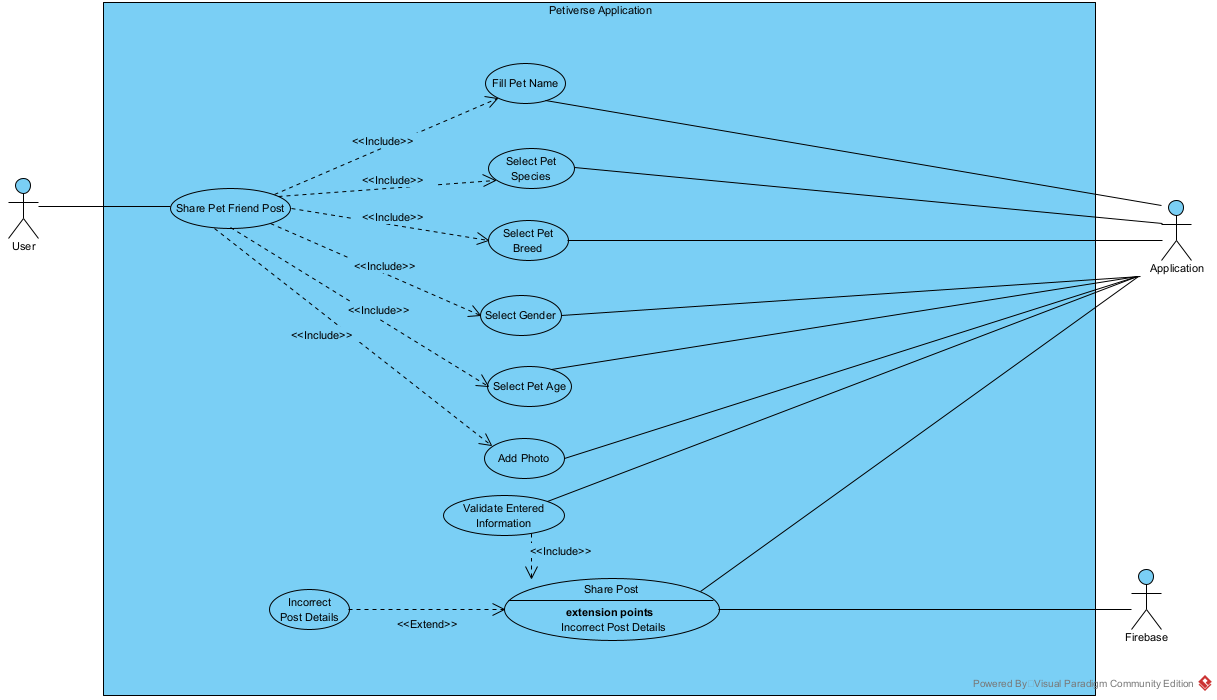
**Actors:** Application, Firebase.

**Description:** This feature exists for the application to print warning messages.

**Includes:** None.

**Extends:** Share Post.

**Use-Cases:** Users must click the share adoption post button.

**4.1.8 Use Case Share Pet Friend**

**Use Case: Share Pet Friend**

**Actors:** User, Application.

**Description:** This feature exists for the user’s pets to find their friend for sharing them info.

**Includes:** Fill Pet Name, Select Pet Species, Select Pet Breed, Select Gender, Select Pet Age, Add Photo.

**Extends:** None.

**Use-Cases:** Users must click the share post button.

**Use Case: Fill Pet Name**

**Actors:** User, Application.

**Description:** This feature exists for the user’s pets name text.

**Includes:** None.

**Extends:** None.

**Use-Cases:** Users must click the textbox to write the name.

**Use Case: Select Pet Species**

**Actors:** User, Application.

**Description:** This feature exists for the user’s pets species.

**Includes:** None.

**Extends:** None.

**Use-Cases:** Users must click the textbox to write the pet’s species.

**Use Case: Select Pet Breed**

**Actors:** User, Application.

**Description:** This feature exists for the user’s pet’s breed.

**Includes:** None.

**Extends:** None.

**Use-Cases:** Users must click the textbox to write the pet’s breed.

**Use Case: Select Pet Gender**

**Actors:** User, Application.

**Description:** This feature exists for the user’s pet’s gender.

**Includes:** None.

**Extends:** None.

**Use-Cases:** Users must click the textbox to write the pet’s gender.

**Use Case: Select Pet Age**

**Actors:** User, Application.

**Description:** This feature exists for the user’s pet’s age.

**Includes:** None.

**Extends:** None.

**Use-Cases:** Users must click the textbox to write the pet’s age.

**Use Case: Add Photo**

**Actors:** User, Application.

**Description:** This feature exists for the user’s pet’s photo.

**Includes:** None.

**Extends:** None.

**Use-Cases:** Users must click the textbox to write the pet’s photo.

**Use Case: Validate Entered Information**

**Actors:** Application

**Description:** This feature exists for the application to validate entered information.

**Includes:** Share Post

**Extends:** None

**Use-Cases:** Users must click the share adoption post button.

**Use Case: Share Post**

**Actors:** Application, Firebase

**Description:** This feature exists for the application to share posts.

**Includes:** None.

**Extends:** None.

**Use-Cases:** Users must click the share adoption post button.

**Use Case: Incorrect Post Details**

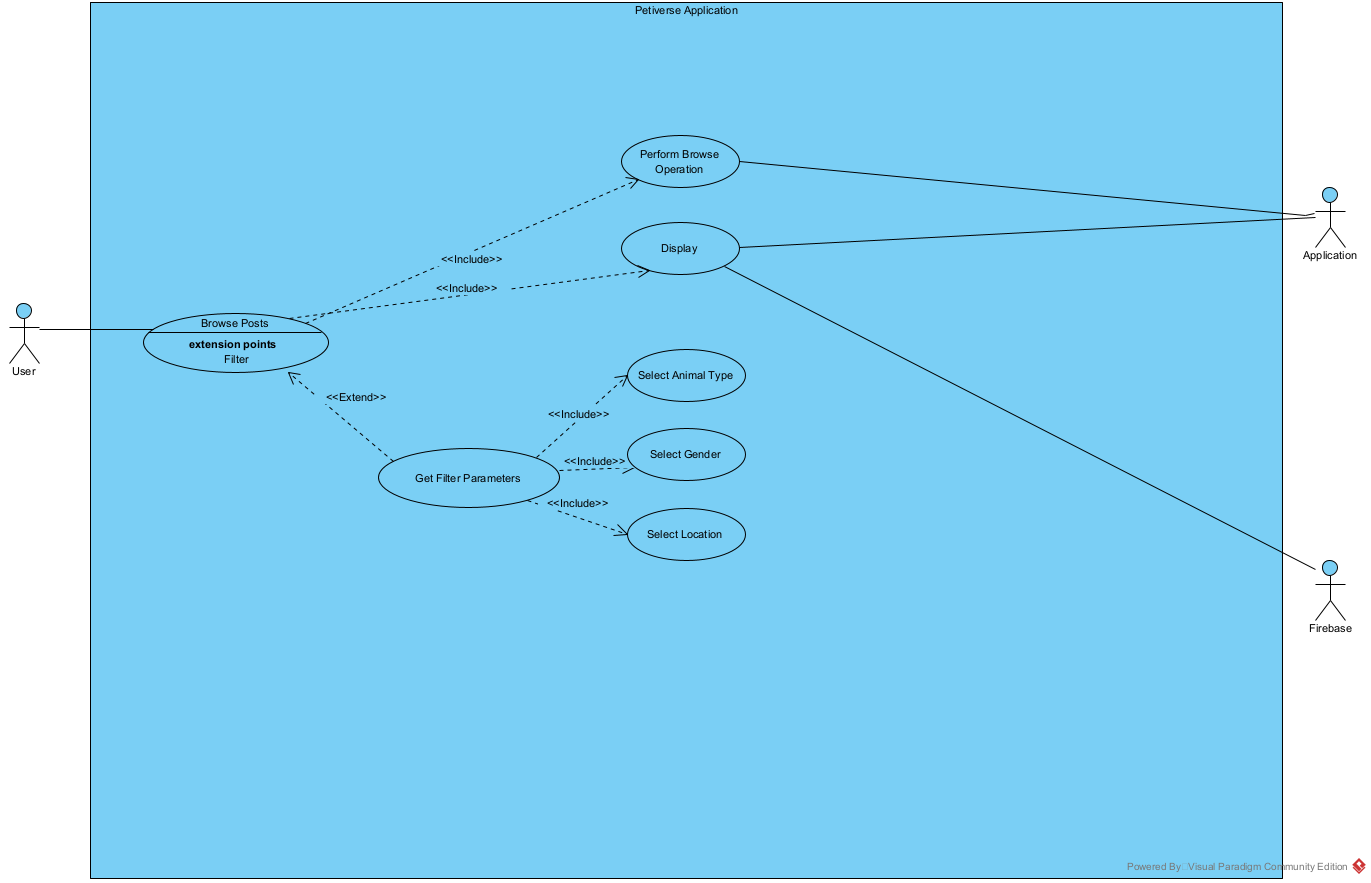
**Actors:** Application, Firebase.

**Description:** This feature exists for the application to print warning messages.

**Includes:** None.

**Extends:** Share Post

**Use-Cases:** Users must click the share adoption post button.

**4.1.9 Use Case Browse And Filter**

**Use Case: Browse Posts**

**Actors:** User, Application, Firebase

**Description:** In order to browse among posts, users must navigate to related pages.

**Includes:** Perform Browse Operation, Display

**Extends:** None

**Use-Cases:** User must navigate to post page

**Use Case: Perform Browse Operation**

**Actors:** User, Application

**Description:** User browses among posts using vertical gestures.

**Includes:** None

**Extends:** None

**Use-Cases:** User must navigate to post page

**Use Case: Display**

**Actors:** User, Application, Firebase

**Description:** When user browsing this operation occurs.

**Includes:** None

**Extends:** None

**Use-Cases:** User must navigate to post page

**Use Case: Get Filter Parameters**

**Actors: User**

**Description:** In order to filter users must enter filtering credentials.

**Includes:** Select Animal Type, Select Gender, Select Location

**Extends:** Browse Posts

**Use-Cases:** User must navigate to post page

**Use Case: Select Animal Type**

**Actors:** User

**Description:** User selects animal type to be filtered.

**Includes:** None

**Extends:** None

**Use-Cases:** User must navigate to post page

**Use Case: Select Gender**

**Actors:** User

**Description:** User selects Gender to be filtered.

**Includes:** None

**Extends:** None

**Use-Cases:** User must navigate to post page

**Use Case: Select Location**

**Actors:** User

**Description:** User selects Location to be filtered.

**Includes:** None

**Extends:** None

**Use-Cases:** User must navigate to post page

**4.2 Classes / Objects**

**4.2.1** **<User>**

4.2.1.1 Attributes

* -Name:String
* -Surname:String
* -birthDate:Date
* -Photo:ImgView
* -Pets:List<Pet>
* -favouritePetTypes:List<String>
* -Proclomations:List<ProclomationCard>
* -forumQuestions:List<ForumQuestionCard>

4.2.1.2 Functions

* +createProclamation(Pet pet, String def, String city, String address, String proclamationType):void
* +addPet(String type, String name, Int age, Photo imgView, List diseaseList):void
* +createHelpProclomation(String city, String petType, ImgView photo, String adress, String definition):void
* +addForumQuestion(String question):void
* getters and setters

**4.2.2** **<Pet>**

4.2.2.1 Attributes

* -Type:String
* -Name:String
* -birthDate:Date
* -Photo:ImgView
* -Photo:ImgView
* -Disease:List<String>
* -Owner:User

4.2.2.2 Funcitons

* getters and setters

**4.2.3 <ForumQuestionCard>**

4.2.3.1 Attributes

* -Question:String
* -questionDate:Date
* -askedBy:User

**4.2.4 <ForumQuestionDetail extends ForumQuestionCard>**

4.2.4.1 Attributes

* -questionDetail:String
* -Photo:imgView

**4.2.5 <ProclamationCard>**

4.2.3.1 Attributes

* -Pet:Pet
* -Owner:User
* -City:String
* -proclomationDate:Date
* -proclomationType:String

**4.2.6 <ProclamationDetail extends ProclamationCard>**

4.2.6.1 Attributes

* -Definition:String
* -Adress:String

**4.2.7 <HelpProclamationCard>**

4.2.7.1 Attributes

* -City:String
* -petType:String
* -Photo:ImgView
* -Status:String

**4.2.8 <HelpProclamationDetail extends HelpProclamationCard>**

4.2.8.1 Attributes

* -Definition:String
* -Adress:String

**4.2.9 <ForumPage>**

4.2.9.1 Attributes

* +allQuestions:List<QuestionCard>

**4.2.9 <helpProclomationPage>**

4.2.9.1 Attributes

* +allHelpProclomations:List<helpProclomationCard>

**4.2.10 <matingProclomationPage>**

4.2.10.1 Attributes

* +allMatingProclomations:List<ProclomationCard>

**4.2.11 <adoptionProclomationPage>**

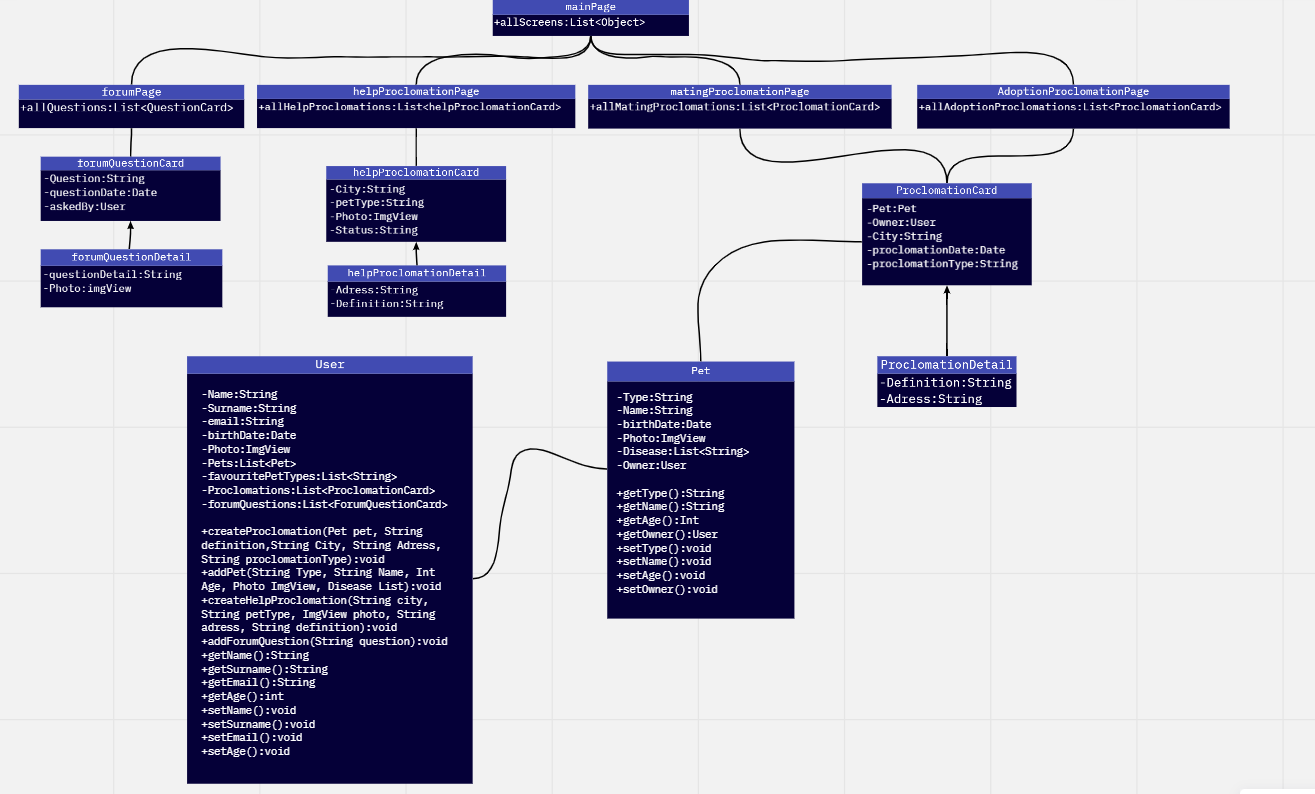
4.2.11.1 Attributes

* +allAdoptionProclomations:List<ProclomationCard>

**4.2.12 <mainPage>**

4.2.12.1 Attributes

* +allScreens:List<Object>



**A. Appendices**

*Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.*

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

**A.1 Appendix 1**

**A.2 Appendix 2**

Acknowledgment: Sections of this document are based upon the IEEE Guide to Software Requirements Specification (ANSI/IEEE Std. 830-1984). The SRS templates of Dr. Orest Pilskalns (WSU, Vancover) and Jack Hagemeister (WSU, Pullman) have also be used as guides in developing this template.

**1.1, 1.2, 1.3, 1.5, 1.2,3.5, 2.5, 3.1.1, 3.1.2, 3.1.3, 3.2.2, 3.2.9, 3.2.5, 3.3, 4.1.3, 4.1.4, 4.1.6, 4.1.7, 4.1.9**

**3.2.4, 3.6, 3.2.8, 3.1.4, 4**