Software Requirements Specification

for

PnP
(Pets and People)

Version 2.1.6

Prepared by:

Arish Qureshi student id: 301176321
Betty Xiaolei Han student id: 301202325
Jocelyne Ojeda Centeno student id: 301197862
Marco Favaretto student id: 301186334
Shibo Wang student id: 301200419

Group 7

2022/01/20

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Arish Qureshi Betty Xiaolei Han Jocelyne Ojeda Centeno Marco Favaretto Shibo Wang	2022/01/20	Discussed and drafted a first draft of the project, added the introduction with the relative points.	1.0
Jocelyne Ojeda Centeno	2022/01/26	Improved introduction and corrections.	1.1
Betty Xiaolei Han	2022/01/27	Added new content: 2. Overall Description.	1.2
Marco Favaretto	2022/02/01	Added new content: 3. External Interface Requirements, Appendix C: Stakeholder List	1.3
Betty Xiaolei Han	2022/02/02	Added new content: 4. Functional Requirements List	1.3.1
Marco Favaretto	2022/02/02	Added new content: Appendix D: Interview question, to be completed.	1.3.2
Marco Favaretto	2022/02/03	Added new content: 5. Non-Functional requirement List.	1.3.3
Shibo Wang	2022/02/03	Improved product features (functions) and functional requirements.	1.3.4
Betty Xiaolei Han	2022/02/04	Added new functional requirement list.	1.3.5
Arish Qureshi	2022/02/04	Added new content: Appendix C- Become a stakeholder as user	Appendix D- Added questions and answers as a user/ answer on marketing of the platform.	1.3.6
Jocelyne Ojeda Centeno	2022/02/04	Formatted and added requirements.	1.3.7
Betty Xiaolei Han	2022/02/06	Added 5 interview questions and answers	1.3.8
Jocelyne Ojeda Centeno	2022/02/08	Added product perspective short description.	1.3.9
Marco Favaretto	2022/02/09	Formatted and answered to some questions.	1.4.0
Betty Xiaolei Han	2022/02/09	Formatted and added reference websites.	1.4.1
Shibo Wang	2022/02/09	Added 5 interview questions and answers. Also defined the name for the project.	1.4.2

Arish Qureshi	2022/02/10	Added interview questions for customer and marketing officer, created logo and tagline for the product, and formatted the document as per APA style.	1.4.3
Arish Qureshi Betty Xiaolei Han Marco Favaretto Shibo Wang Jocelyne Ojeda Centeno	2022/02/11	Final adjustment and revisions.	1.5
Marco Favaretto	2022/14/02	Added Use cases table	1.5.1
Arish Qureshi Betty Xiaolei Han Marco Favaretto Shibo Wang Jocelyne Ojeda Centeno	2022/15/02	Completed Use Case section during Teams meeting.	2.0
Betty Han	2022/02/19	Added comment for changing Use Case Name waiting for team's approval.	2.1
Betty Han	2022/03/03	Revised the Use Case table	2.1.1
Betty Han Marco Favaretto	2022/03/03	Added Use Case for registration	2.1.2
Shibo Wang	2022/03/03	Added Swim-lane diagram	2.1.3
Marco Favaretto	2022/03/03	Added Use Case diagram	2.1.4
Betty Han Marco Favaretto Shibo Wang	2022/03/03	Last review before deploying	2.1.5
Betty Han Marco Favaretto Shibo Wang Jocelyne Ojeda Centeno	2022/03/11	Team meeting review feedback, revising last draft, and discussing classes	2.1.5
Betty Han Marco Favaretto Shibo Wang Jocelyne Ojeda Centeno	2022/03/15	Team meeting discussing class responsibilities	2.1.6
Marco Favaretto	2022/03/17	Added class domain diagram	2.1.6
Betty Han Shibo Wang	2022/03/17	Added CRC index cards	2.1.6

Jocelyne Ojeda Centeno			
Betty Han Marco Favaretto Shibo Wang Jocelyne Ojeda Centeno	2022/03/25	Final review	2.1.6
Betty Han Marco Favaretto Shibo Wang Jocelyne Ojeda Centeno	2022/03/29	Teams Meeting to discuss Part C and divide the work for the first two deliverables.	3.0
Betty Han Jocelyne Ojeda Centeno	2022/03/30	State Diagrams (SearchBar, Registration)	3.0
Marco Favaretto Shibo Wang	2022/03/30	Sequence Diagrams	3.0
Betty Han	2022/04/04	Update SearchBar state diagram with class name	3.1
Betty Han Betty Han Marco Favaretto Shibo Wang Jocelyne Ojeda Centeno	2022/04/06	Completed Deliverable #3	3.1
Jocelyne Ojeda Centeno	2022/04/07	Updated Registration State Diagram as per teacher's feedback	3.1
Betty Han Betty Han Marco Favaretto Shibo Wang Jocelyne Ojeda Centeno	2022/04/20	Discuss and learn "Party" class diagram. Modify our own class.	3.1

1 Introduction

1.1 Purpose:

1. This platform/social media will help pet lovers meet each other, share their pet's photos and stories, and organize virtual and physical meetings with other pet owners.
2. In this platform/social media people will be able to ask for help and advice about their animal's care.
3. Lastly, the platform will evolve to welcome veterinarians, dog trainers, dog walkers and sitters, pet groomers, as well as pet shop owners so they can share their services and connect with potential clients. The platform will allow virtual consultations.

1.2 Document Conventions:

Acronyms	Description
SQL	Structured Query Language
DB	Database
MSVS	Microsoft Visual Studio
C#	C Sharp
ASP.NET	Active Service Pages. Framework
JS	Java Script
UML	Unified Modeling Language
UI	User Interface
OS	Operation System

1.3 Intended Audience and Reading Suggestions:

This document is addressed to all sectors involved in the project such as:

- Investors.
- Project managers.
- Marketing team.
- Developers, such as:
  - Web designer,
  - UX designer,
  - Front-end/Back-end developers,
  - DB developers.
- IT team.

- Some end users such as:
  - Veterinarians,
  - Dog trainers,
  - Pet day walkers and sitters,
  - Pet groomers,
  - Pet shop owners,
- Pet shelters.
- Group of specific users for beta test evaluation purpose.

1.4 Project Scope:

Nowadays, with many people spending time online, we have noticed and increasing in how many pet owners use social media to share pictures of their pets or use social groups or web search engines to find general and/or specific information about their pets. Our project will be to create a platform where pet owners will be able to discuss and share information about their own pet and meet other pet owners. Moreover, this platform will be able to help pet owners discover local businesses specifically services offered by third parties such as veterinarian virtual consultation, dog training lectures, pet care experts. The platform will also welcome pet shop owners, pet groomers and pet walkers and sitters to offer their services and schedule appointments. Pet owners or pet lovers will be able to subscribe to the platform providing their emails, and some general information about their pets if they have one. As for professionals offering their services, an additional verification step will be required to validate their identity and credentials. Other types of services not related to pets/animals will not be allowed.

1.5 References:

- Google API:
  - https://developers.google.com/apis-explorer
- iOS Map kit JS:
  - https://developer.apple.com/documentation/mapkitjs
- Google meet:
 - https://meet.google.com/?pli=1
- Zoom:
  - https://zoom.us/

- Web site of pet product/service seller:
    - https://www.bestinshowconsulting.com/?gclid=EAIaIQobChMIuv2Asrbz9QIVtyCtBh2OgQuBE AAYAiAAEgK3MfD_BwE
- Web site of veterinarian:
    - https://vetster.com/en-ca?utm_source=google&utm_medium=cpc&utm_campaign=vet_clinic_ca&utm_id=12828645592&utm_term=vet_clinic&utm_content=122294164558&gclid=CjwKCAiA6Y2QBhAtEiwAGHybPRwBOSGSre3iD9wl-Zyr9smeKEOoBYB8n5BkzPFjhXIzUtYUNokhkRoCmysQAvD_BwE

2. Overall Description

2.1 Product Perspective

This is a new software. The software will provide an online platform for users to connect with other users, book appointments and be directed to partners website through affiliate links.

Diagram:
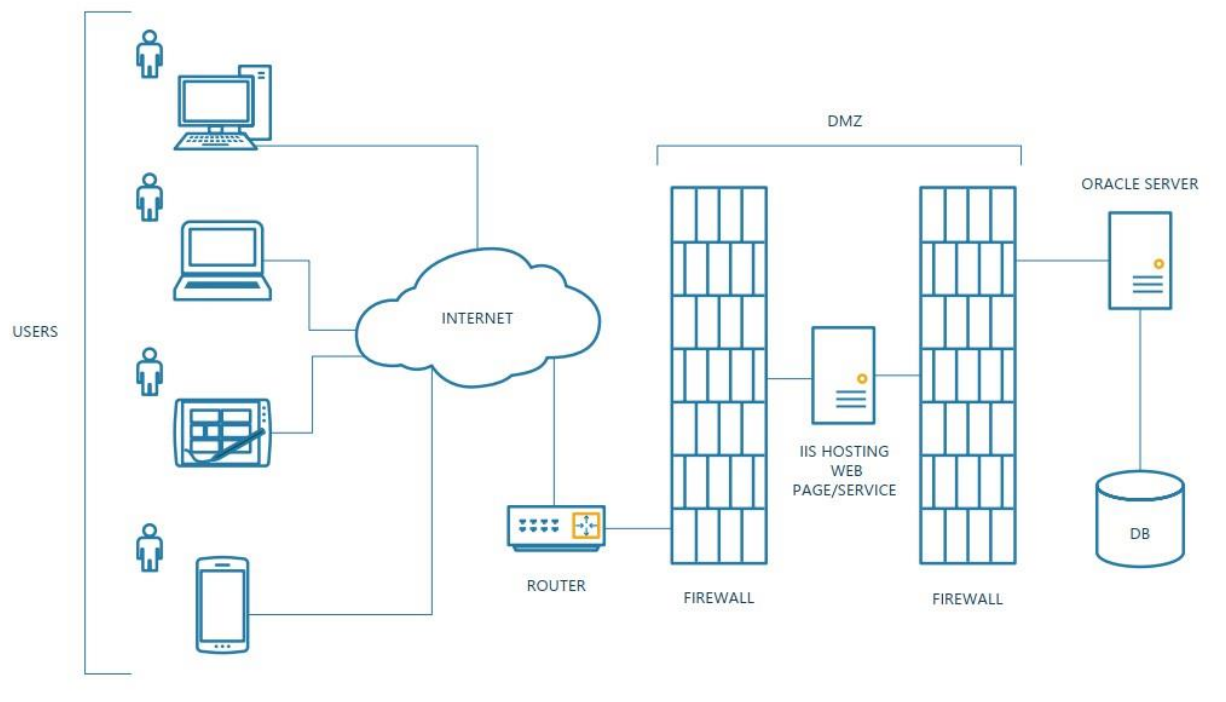
2.2 Product Features (Functions):

The new software will provide the following functionalities:

1. Users can register, login, and create their pets' profile on the software
2. Users can share and edit pets' stories, photos, and videos
3. Users can search and connect with each other
4. Users can comment and message each other
5. Users can use the search bar to search for pet food and vet locations. This will be made possible through a Python searching function and database we created.
6. Users can link to local PetSmart shops; users will know shops' locations by google map
7. Users can link to local animal hospital/clinic; users will know hospitals' locations by google map
8. Users can post their new baby pets on the pet market page; they can trade and sell pets.
9. Services providers can register here to be able to advertise their product and services, and to provide online consultations.

2.3 User classes and Characteristics:

1. pet owners: will use the new software to connect with other pet owners or service providers

2. pet service providers: will use the new software to connect with potential clients, answer   inquiries and advertise their services

3. system administrators: will use the new software to manage users' profile and provide technical support when needed

2.4 Operating Environment:

- will run on a Unix server
- will use an Oracle DB to store users' information
- supports the following browsers: IE, Safari and all browser based on Chromium (Chrome, MS Edge, Opera, etc.…)
- will be developed using Html, C# and JavaScript for UI; PHP and Python for back-end
- supports iOS, android for the mobile version

2.7 Assumption and Dependencies:

- Marketing database creating
- Search bar creating
- Some clinics may not like to share information with the app
- Some users may post inappropriate content or information on the chatting and pet market interface
- Google map function may not find the right information on the map interface

3.0 Section "External Interface Requirements"

3.1 User Interfaces:

Users will be able to access our system through a web page mostly developed in C# and JavaScript. The web page should be accessible from any devices such as desktop/laptop running Linux/Windows/Mac OS, tablets, and smartphones. Our website will follow the W3C rules to be

more accessible with the highest quality for user interface. Our back end and search engine will be developed in PHP and python, respectively.

3.2 Hardware Interfaces:

- Camera,
- GPS,
- Geo location,
- Wi-Fi and Network signal.

3.3 Software Interfaces:

- Calendar,
- Photos Album,
- Maps:
  - Google API,
  - iOS Map kit JS.
- Google meet,
- Zoom.

Use cases			
Use Case name	List of related Requirements ID	Actor (s)	Brief Description
Register account	FR00 FR02	Pet Owner & Service Provider	New user clicks on register button on the homepage, then the website will respond with a registration form for

			new user to fill up. After finish filling up the form, the user will go to his/her email to activate the account and the website will redirect the user to a setting page to complement his/her info.
Post stories	FR03 FR10	Pet Owner & Service Provider	The pet owner/service provider clicks on the post button, in response the website let the actor to post stories, photos, and videos of their pets /services.
Search pet owners/comments/pet services	FR04	Pet Owner	The pet owner will click on the search bar, type in @username or #keyword of a tag or comment, in response the website will present the pet owner a list of information that match the user's input keywords. The user will select the member/tag/comment he/she likes to connect or view.
	FR05 FR06 FR09	Pet Owner	The pet owner will click on search bar, input one genetic keyword (such as vet, grooming, training, food) of the pet product & services he/she wants to search. In response, the website will present a list of providers that match the keyword. The pet owner clicks on the service provider, in response the website will present the provider's

			contact information, details of services, distance from the pet owner and an option to show on the map.
Manage user data/questions	FR08 FR07	Administrator	The system administrator clicks on manage and monitor users' data button, and the system should generate the users' data according to the admin's input date. The administrator clicks on frequently asked questions button, in response the website will present a list of unsolved questions from users.
Send notification	FR01	Pet Owner & Service Provider	If the pet owner/service provider turns on notification in settings, then he/she will get a mobile notification when he/she gets a new message/comment on the website.
ABSTRACT Log into website	FR11		The pet owner/service provider clicks on the sign in button, and the website will let the member to put in their username and password. The member puts in the correct credentials to log into their account.

4.0 Section Functional Requirements:

	Functional Requirements list			
Requirement ID	Requirement title	Short Description	Priority	Requester
FR00	Creating profiles	Users and service providers will be able to create an account.	Expected	All users
FR01	Alerts	The system should give the users a mobile notification/alert when they get new messages or requests	Expected	all users
FR02	Profiles	The system should allow the users to edit/update their profiles	Expected	all users
FR03	Uploading and posting	The system should allow pet owners to post stories, photos, and videos of their pets	Expected	all users
FR04	Search for other users	The system should allow users to search for other pet owners, service providers and send connect/message requests	Expected	all users
FR05	Hyperlinks	The system should provide accurate and functional third-party links of service providers for pet owners	Expected	service providers
FR06	Locate service providers	The system should allow pet owners to search and locate service providers on Google Map	Exciting	pet owners
FR07	Manage data and postings	The system should allow administrators to manage and monitor users' postings data and comments	Expected	Administrator
FR08	Frequently asked questions FAQ	The system should give the administrator the ability to manage frequently asked questions from users.	Exciting	Administrator

FR09	Search for pet foods and vet location	The system should give pet owners the ability to search pet foods and vet locations. The information is from our database, which is provided by the market officers.	Expected	pet owners
FR10	Store Service and products data into database	The system should allow our "partner service providers" to input their services and products into our database for pet owners to search	Expected	Service providers
FR11	Log into website	The system should allow all users to log into our website after registration with the correct username and password	Expected	All users

Use case:	**Register new user to the platform**
Iteration:	
Primary actor:	**New user.**
Goal in the context:	**Create an account for new users.**
Preconditions:	**New users must have an email and be legally of age.**
Trigger:	**New users want to create an account for the platform to be able to post their pets photos, share opinions and ideas about pets, find other pet lovers, and to search for pet services.**

Scenarios:

1. New user access to the home page and click on create new account.
2. New users must fulfill the form providing the following mandatories data:
    a. First Name,
    b. Last Name,
    c. Birth date,
    d. Username,
    e. Password,
    f. Email (the email must be verified email),
    g. Security question for profile recovery purpose.
    h. Security answer for profile recovery purpose.
3. The new user will have to click on the "Create Now" button to receive a confirmation email.
4. To finalize the process, new users must open the email and click on "confirm your account here", to confirm the account.
5. The new user will be redirected to his/her profile page, where will be possible to complement some information with the following data:
    a. Privacy,
    b. Who can find me,
    c. Who can see my homepage and post,
    d. Location,
    e. Favourite language,
    f. Mobile number (for notification purpose),
    g. Notification alert of new posts (on/off),
    h. Notification alert of new services (on/off),
    i. Interests on (to filter the advertise that will receive on user home page).

Exceptions:

1. The date of birth does not correspond to the age of majority.
2. Username already in use or does not meet the system requirements such as the use of special characters or inappropriate/offensive words).

3. The password does not respect the minimum safety criteria (password must contain at least 8 alphanumeric characters, must contain at least one uppercase character, must contain a number, and cannot contain the user's birth date).
4. The email must be a valid and active email.
5. All fields marked as mandatory must be filled in.
6. The new user has not received or confirmed the "confirmation" email.

Priority: **Hight priority, this is fundamental for users to be able to access to the portal.**

When available: **Second increment.**

Frequency of use: **Frequent.**

Channel to actors: **From any device with an internet connection and a web browser.**

Secondary actors: **System administrator**

Channels to secondary actors:

1. System administrator: PC-based system, email.

Open issues:

1. What mechanism will prevent fictious account to be created?
2. What mechanism will prevent use and registration by minors?
3. User data are sufficiently safe and protected? Risk of hacker attacks to steal personal data.

Swim lane diagram for register user:
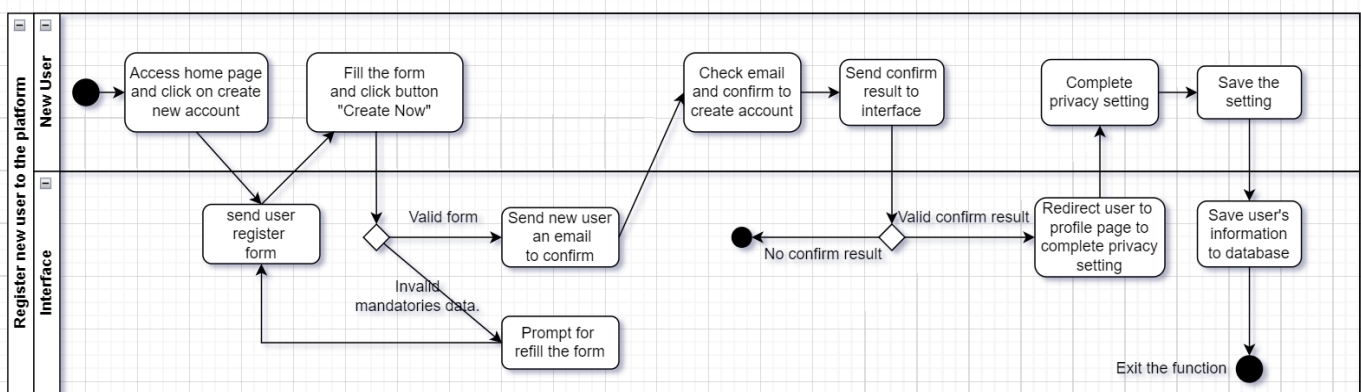
Use case diagram:
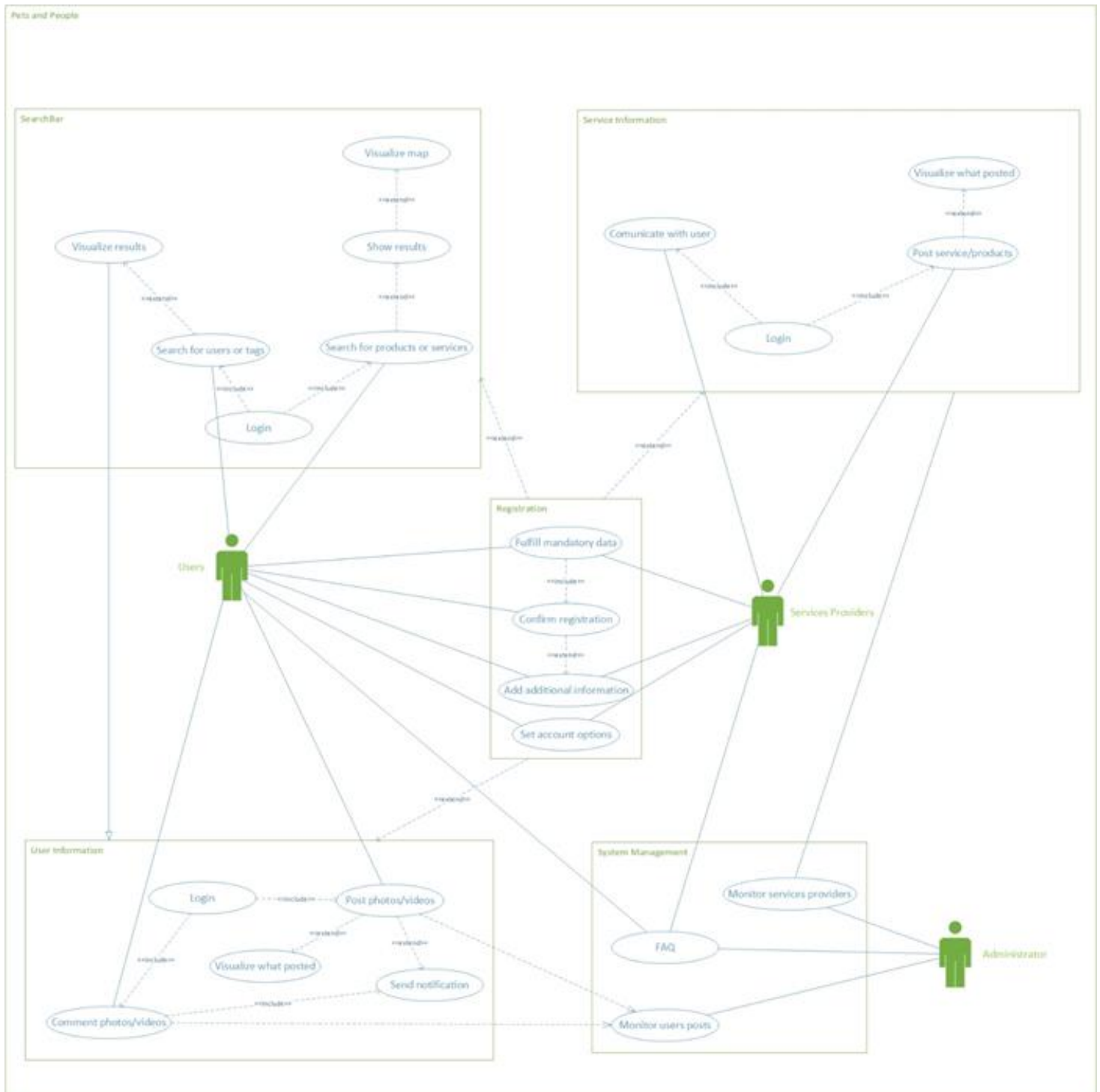
5.0 Non-Functional Requirements:

Non-Functional Requirements List				
Requirement ID	Requirement title	Short Description	Priority	Requester
NFR01	DB Performance	The database must handle the spike in user requests from the application.	High	DB developer
NFR02	Redundant Capacity	A redundant system will provide failover or load balancing support to protect an active system in the event of an unexpected failure.	High	CIO
NFR03	Security - Cyber	Safety tests must be performed periodically such as: vulnerability scanning, penetration testing, API security testing, Web application security testing.	High	IT security administrator
NFR04	Device Compatibility	The platform must be viewable and adaptable to any type of device with different display sizes, and constantly updated with new entry models.	Medium	UI developer
NFR05	Software Reliability	The development languages used, and related Frameworks, must always be updated to the latest stable version.	Low	CTO
NFR06	FAQ, Help Desk	The system should be able on its own to identify inappropriate content or user behavior. In addition, the system must manage the FAQs, and upon request, a	Low	CIO/CTO

			help desk should intervene to help users.		

Appendix C:

Stakeholder Register					
Stakeholder Name	Stakeholder Position	External / Internal	Stakeholder contact details	Operational / Executive	Interest (high, medium, low)
Stewart Paterson	Customer	External	stewart.paterson@petsmart.ca	Operational	High
Kylie Murray	Customer	External	Kylie.murray@vcahospital.ca	Operational	High
Ella Gibson	Customer	External	ella.gibson@dogtopia.ca	Operational	High
Arish Qureshi	User	External	qureshi.arish@gmail.com	Operational	Low
Victoria Mitchell	User	External	victoria.mitchell@gmail.com	Operational	Low
Nicholas Jackson	User	External	nicholas.jackson@gmail.com	Operational	Low
David Hemmings	Marketing Officer	Internal	david.hemmings@projcorp.com	Operational	Medium
Sally Mackay	CIO	Internal	sally.mackay@projcorp.com	Executive	High
Angela James	CTO	Internal	angela.james@projcorp.com	Executive	High
Sonia Peters	DB Administrator	Internal	sonia.peters@projcorp.com	Operational	Medium

Dylan Russell	IT security administrator	Internal	dylan.russle@projcorp.com	Operational	Medium
Joe Butler	Investor	Internal	joe.butler@projcorp.com	Executive	High
Stephen Marshall	Investor	External	stephen.marshall@gmail.com	Executive	High

Appendix D:

Interview Questions		
Question	Stakeholder position	Answer
What kind of visibility are you expecting on our platform?	Customer	We would like to expand our visibility to social media lovers and those who for some reason cannot physically reach us.
What kind of services would you like to advertise on our platform?	Customer	Pets grooming, daycare services, online veterinary consultations, and some pet products
What made you consider our product?	Customer	The unique idea of connecting people virtually with same interests in pets is something which attracted us towards your product.
In what direction is this platform heading in your opinion?	Customer	We believe your product is heading into a category of social networking website where the user will share photos of their pets and connect virtually.
Why do you think there is a need of this product?	Customer	The lack of a platform where specifically people with same pet interests can connect virtually is why this product is required
What kind of services would you like to find on our platform?	User	I would like the platform to connect me with other pet owners. Moreover, I would like the platform to provide me with pet category and breed filter.

What do you like most about social networks?	User	I love how social networks bring me closer to like-minded people around the globe. The experience of being able to communicate with people who share same interests as I fascinate me the most.
What do you like least about social networks?	User	I do not like a social media platform which shows unnecessary advertisement pops unless I get a premium version for unrealistic prices.
What additional feature do you want our platform to have?	User	I would love if the platform provided me with an option to set the level of importance of the messages, I send to other pet owners. So that if an urgent respond is needed from the other end, it become easier.
What feature of our software are you going to use the most?	User	Connecting with other pet owners in my community and share our pets' life with each other
How to advertise our platform and make it as visible as possible?	Marketing Officer	The best way to advertise our platform is to start small. Basically, starting with free social media platforms such as Twitter, YouTube etc. Moreover, we can use google ad campaign, promoting attractive videos, and graphic ads related to what our platform will do.
What are the short- and long-term business goals?	Marketing Officer	For short goal, we are going to create a demo prototype in 3 months. This prototype can work basic functions, to attract 10 merchants' recruitment. For long goal, IPO.
How would you measure whether this project is a success?	Marketing Officer	When we will have one million users, or we have one million users, and we have over two hundred featured brands supports and join (include vet, food, and toys supply).
Who are your main competitors?	Marketing Officer	Facebook and Instagram. Some users create social media pages for their pets.

		Amazon Pet Supplies. It has many featured brands support. PetSmart and PETCO account for nearly half of all online pet product revenue.
What should be the tagline of our platform?	Marketing Officer	"Your pet's best friend is one post away", as our platform connects people and pets.
How will our platform be designed from an IT point of view? Will it be necessary to update and / or purchase new hardware or software?	CIO	Please refer to 2.1 Product perspective diagram. It depends on how our server room is structured today. A capacity analysis and stress test of the current server park will be done. Based on the results, it will be determined whether new hardware and / or software will be required.
Are there any user limitations?	CIO	There are so many old pet owners who cannot use social media, they do not use smartphones, laptops. Our software cannot impact their life.
Which technologies and languages will be most appropriate for the development of our platform?	CTO	Html, C#, JS, PHP, Python.
Will bots be used for certain types of activity in response to user demand?	CTO	It will be created a bot in Python to guide users to predefined FAQs, in specific cases, a channel will be available, via email, to answer more specific questions.
Will the bots be used to verify the good conduct and ethics of all users?	CTO	It will be developed a bot in Python to filter the content of discussion, and images.
How do you test the software to ensure the software quality?	CTO	- We apply agile methodology to deliver functions in each sprint. - We follow industry best practice like continuous integration.

		• We follow-up with users' experience after each delivery.
What will be the aesthetic aspect of our site? Will it contain features for people with deficiencies? and for who needs it?	UI designer	The aesthetic of the platform will be clean and modern. It will not allow ad banners from outside companies, so we will be able to support our website with auto-reading features. Finally, our website will follow the W3C rules to be more accessible.
What type of Database is used in your company?	DB Administrator	Oracle because oracle has better performance in dealing with large database and our team has valuable experience with Oracle.
How do you make sure our clients' data is secure?	DB Administrator	• apply network security system rules, • restrict users' access permissions, • apply data encryption and masking against, • sensitive customer information, • scheduled operations of data backup.
What will be the available budget?	Internal investor	The starting budget is 35,000 CAD, which can be expanded up to 75,000 CAD in case of need to purchase hardware and software.
What annual earnings expectation? What ROI do you expect and in how long?	External investor	Expect 2.5% of ROI in the first 6 months, to close at 8% in the first year. After that we plan to progressively increase our volume from 25% up to 50% per year.
What are the interface languages that may be provided?	UI designer	Right now, we are going to test this app in Canada, so the interface languages would be English and French. Future version may include more languages such as Spanish, Mandarin, Cantonese etc.

Appendix E:

Class diagrams

LoginWebPage
Class
→ Form
- Fields
  - btnLogin : Button
  - btnSignUp : Button
  - components : IContainer
  - txtPassword : TextBox
  - txtUserName : TextBox
- Properties
  - Password { get; } : string
  - Username { get; } : string
- Methods
  - Dispose(bool disposing) : void
  - InitializeComponent() : void
  - LogIn(string Username, string Password) : bool
  - LoginWebPage()
  - SignUp() : void

Registration
Class
- Fields
  - confirmed : bool
- Properties
  - Confirmed { get; set; } : bool
- Methods
  - ConfirmEmail(string email, bool Confirmed) : bool
  - CreateAccount(Account newAccount) : Account

RegistrationWebPage
Class
→ Account
- Fields
  - btnCreateNow : Button
  - components : IContainer
  - dateTimePickercBirthdate : DateTimePicker
  - txtEmail : TextBox
  - txtFirstName : TextBox
  - txtLastName : TextBox
  - txtPasword : TextBox
  - txtSafetyAnswer : TextBox
  - txtSafetyQuestion : TextBox
  - txtUsername : TextBox
- Methods
  - CreateNow() : void
  - Dispose(bool disposing) : void
  - InitializeComponent() : void
  - RegistrationWebPage()

HomePage
Class
→ Account
- Fields
  - btnComment : Button
  - btnCreatePost : Button
  - btnDeletePost : Button
  - btnFAQ : Button
  - btnLogOut : Button
  - btnProfile : Button
  - btnSearchBar : Button
  - btnUpload : Button
  - components : IContainer
  - lstBoxPosts : ListBox
  - txtComment : TextBox
- Properties
  - Comment { get; set; } : string
  - PostID { get; set; } : int
- Methods
  - Dispose(bool disposing) : void
  - HomePage()
  - InitializeComponent() : void

ProfileWebPage
Class
→ Account
- Fields
  - btnHomePage : Button
  - btnSave : Button
  - chkBoxLocation : CheckBox
  - chkBoxNofitfyNewPosts : CheckBox
  - chkBoxNotifyNewServices : CheckBox
  - chkBoxPrivacy : CheckBox
  - cmbBoxFavoriteLanguage : ComboBox
  - cmbBoxWhoCanFindMe : ComboBox
  - cmbBoxWhoCanSeeMyHomepageAndPost : ComboBox
  - components : IContainer
  - dateTimePickerBirthdate : DateTimePicker
  - lstBoxInterestOn : ListBox
  - txtEmail : TextBox
  - txtFirstName : TextBox
  - txtLastName : TextBox
  - txtMobileNumber : TextBox
  - txtPassword : TextBox
  - txtSafetyAnswer : TextBox
  - txtSafetyQuestion : TextBox
  - txtUsername : TextBox
- Methods
  - Dispose(bool disposing) : void
  - HomePage() : void
  - InitializeComponent() : void
  - ProfileWebPage()
  - SaveProfile() : void

Account
Abstract Class
- Fields
  - answer : string
  - email : string
  - favLanguage : string
  - firstName : string
  - interestOn : List<string>
  - lastName : string
  - location : int
  - mobileNumber : int
  - newPostNotification : bool
  - newServicesNotification : bool
  - password : string
  - privacy : bool
  - question : string
  - username : string
  - whoCanFindMe : list<string>
  - whoCanSeeMe : list<string>
- Properties
  - AccountID { get; set; } : int
  - Comment { get; set; } : string
  - Confirmed { get; set; } : bool
  - Email { get; set; } : string
  - Password { get; set; } : string
  - PostID { get; set; } : int
  - SearchResult { get; set; } : List<string>
  - UserName { get; set; } : string
- Methods
  - ChangeOptions(List<options> options) : void
  - CommentPost(int PostID, string Comment) : void
  - DeleteAccount(string username, string password) : void
  - DeletePost(int PostID) : void
  - FAQ() : void
  - RecoverPassword() : void
  - SaveOptions() : void
  - Upload(object objToUpload) : void

ServiceProvider
Class
→ Account
- Properties
  - AccountID { get; set; } : int
- Methods
  - DeletePost() : void
  - Messaging() : void
  - Streaming() : void
  - Uploading() : void

PetOwner
Class
→ Account
- Fields
  - postType : object
- Properties
  - AccountID { get; set; } : int
  - Comment { get; set; } : string
  - PostID { get; set; } : int
  - SearchResult { get; } : List<string>

FAQWebPage
Class
→ Account
- Fields
  - btnAskToUs : Button
  - btnHomepage : int
  - components : IContainer
  - lstFAQ : ListBox
- Properties
  - FAQ { get; set; } : List<string>
- Methods
  - Dispose(bool disposing) : void
  - FAQWebPage()
  - InitializeComponent() : void

FAQ
Enum
- Question1
- Answer1
- Question2
- Answer2
- Question3
- Answer3

Administrator
Class
- Properties
  - FAQ { get; set; } : int
- Methods
  - DeleteServiceProvider() : void
  - DeleteUser() : void
  - MonitorPetOwners() : void
  - MonitorServicesProvider() : void

Hashtag
Enum
- Users
- Services
- Products

SearchBar
Class
- Fields
  - location : int[]
  - petOwner : int
  - serviceProvider : int
  - txtSearch : string
- Properties
  - Hashtag { get; set; } : int
  - SearchResult { get; set; } : int
- Methods
  - Search() : void

SearchBarWebpage
Class
→ SearchBar
- Fields
  - btnHomePage : Button
  - btnSearch : Button
  - components : IContainer
  - lstItemResult : object
  - lstResults : ListBox
  - txtSearch : TextBox
- Methods
  - InitializeComponent() : void
  - OpenMapResult(object lstItemResult, Array location) : void
  - SearchBarWebpage()
  - SelectResult(object lstItemResult) : void

CRC index cards

Class: Registration	
Description:	
Private field include bool trigger to check email confirmed or not, then create an account for user.	
Responsibility:	**Collaborator:**
Create Account for user	Account
Send User confirm email	

Class: Account	
Description:	
Abstract class for PetOwner and ServiceProvider, connecting with HomePage, ProfileWebPage, FAQWebPage, and RegistrationWebPage interface	
Responsibility:	**Collaborator:**
User can set and change all mandatory data	
Create AccountID	
Confirm registration requirement	
User can delete Account	
User can ask for FAQ	
User can ask for recovering password	
Save user's options	
PetOwner can comment under posting	PetOwner
PetOwner can delete posting	PetOwner
ServiceProvider can upload object	ServiceProvider

Class: SearchBar	
Description: SearchBar involves functions like searching for other users, hashtags, pet product and services as well as displaying locations of users and services.	
Responsibility:	**Collaborator:**
shows search results for hashtag	Hashtag
shows search results for users/products/services	PetOwner, PetProvider
shows locations of users/serviceProvider	Account

Class: Administrator	
Description: Administrator includes functions like accessing FAQ, monitoring all users' posting content, account security, and disable accounts when necessary.	
Responsibility:	**Collaborator:**
Monitor users and users' postings	
Disable user accounts when impropriate information is posted by the account	
Update FAQ webpage	

Class: PetOwner

Description: The PetOwner can navigate the website to connect with other pet owners, search for service providers by name or location, browse the FAQ section, and upload and post content to their page.

Responsibility:	Collaborator:
Pet Owner can create and modify an account	Account
Pet Owner can get their AccountID	Account
Pet Owner can upload pictures	
Pet Owner can upload daily stories	
Pet Owner can write posts	
Pet Owner can delete/modify posts	
Pet Owner can comment below other pet owner's post	
Pet Owner can use the SearchBar function	SearchBar

Class: ServiceProvider

Description: The Service Provider will be able to create an account which they will use to advertise their services to pet owners and build a clientele. They will be able to upload content to share relevant information, reply to pet owners' inquiries through instant messaging and stream for virtual consultations.

Responsibility:	Collaborator:
Service Provider can create/modify an account	Account
Service Provider can get their AccountID	Account
Service Provider can write posts	
Service Provider can delete/modify posts	
Service Provider can reply to PetOwners' inquiries	

Class Diagram Images:

Class Details - Account

Name	Type	Modifier	Summary	Hide
Methods				☐
▲ ChangeOptions	void	public		☐
(options	List<options>	None		
) <add parameter>				
▲ CommentPost	void	public		☐
(PostID	int	None		
, Comment	string	None		
) <add parameter>				
▲ DeleteAccount	void	public		☐
(username	string	None		
, password	string	None		
) <add parameter>				
▲ DeletePost	void	public		☐
(PostID	int	None		
) <add parameter>				
▷ FAQ	void	public		☐
▷ RecoverPassword	void	public		☐
▷ SaveOptions	void	public		☐
▲ Upload	void	public		☐
(objToUpload	object	None		
) <add parameter>				
<add method>				
Properties				☐
AccountID	int	public		☐
Comment	string	public		☐
Confirmed	bool	public		☐
Email	string	public		☐
Password	string	public		☐
PostID	int	public		☐
SearchResult	List<string>	public		☐
UserName	string	public		☐
<add property>				
Fields				☐

Class Details - Account

Name	Type	Modifier	Summary	Hide
) <add parameter>				
<add method>				
Properties				☐
AccountID	int	public		☐
Comment	string	public		☐
Confirmed	bool	public		☐
Email	string	public		☐
Password	string	public		☐
PostID	int	public		☐
SearchResult	List<string>	public		☐
UserName	string	public		☐
<add property>				
Fields				☐
answer	string	private		☐
email	string	private		☐
favLanguage	string	private		☐
firstName	string	private		☐
interestOn	List<string>	private		☐
lastName	string	private		☐
location	int	private		☐
mobileNumber	int	private		☐
newPostNotification	bool	private		☐
newServicesNotification	bool	private		☐
password	string	private		☐
privacy	bool	private		☐
question	string	private		☐
username	string	private		☐
whoCanFindMe	list<string>	private		☐
whoCanSeeMe	list<string>	private		☐
<add field>				
Events				
<add event>				

Class Details - Administrator

Name	Type	Modifier	Summary	Hide
▲ **Methods**				☐
▲ ⚙ DeleteServiceProvider	void	public	Delete service provider account	☐
(accountID	int	None		
) <add parameter>				
▲ ⚙ DeleteUser	void	public	Delete pet owner accout	☐
(accountID	int	None		
) <add parameter>				
▲ ⚙ MonitorPetOwners	void	public	Monitor posts of pet owner	☐
(postID	string	None		
) <add parameter>				
▲ ⚙ MonitorServicesProvider	void	public	Monitor posts of service provider	☐
(postID	string	None		
) <add parameter>				
⚙ <add method>				
▲ **Properties**				☐
🔧 FAQ	int	public		☐
🔧 AccountID	int	public		☐
🔧 PostID	int	public		☐
🔧 <add property>				
▲ **Fields**				
🌐 <add field>				
▲ **Events**				
⚡				

Class Details - FAQWebPage

Name	Type	Modifier	Summary	Hide
▲ **Methods**				☐
▷ ⚙ Dispose	void	protected	Clean up any resources being used.	☐
▷ ⚙ FAQWebPage		public		☐
▷ ⚙ InitializeComponent	void	private	Required method for Designer support - do not modify	☐
⚙ <add method>				
▲ **Properties**				☐
🔧 FAQ	List<string>	public		☐
🔧 <add property>				
▲ **Fields**				☐
🔧 btnAskToUs	Button	private	Open the user's e-mail program to send a request	☐
🔧 btnHomepage	int	private	Redirect the user to the Homepage	☐
🔧 components	IContainer	private	Required designer variable.	☐
🔧 lstFAQ	ListBox	private		☐
🌐 <add field>				
▲ **Events**				
⚡				

Class Details - HomePage

	Name	Type	Modifier	Summary	Hide
	⊿ **Methods**				
	▷ Dispose	void	protected	Clean up any resources being used.	☐
	▷ HomePage		public		☐
	▷ InitializeComponent	void	private	Required method for Designer support - do not modify	☐
	<add method>				
	⊿ **Properties**				☐
	Comment	string	public	Get the string from txtComment	☐
	PostID	int	public	Get the PostId from the lstBoxPost	☐
	<add property>				
	⊿ **Fields**				☐
	btnComment	Button	private		☐
	btnCreatePost	Button	private		☐
	btnDeletePost	Button	private		☐
	btnFAQ	Button	private	Redirect the user to FAQWebage	☐
	btnLogOut	Button	private	LogOut the user	☐
	btnProfile	Button	private	Redirect the user to the ProfileWebPage	☐
	btnSearchBar	Button	private	Redirect the user to SearchBarWebpage	☐
	btnUpload	Button	private		☐
	components	IContainer	private	Required designer variable.	☐
	lstBoxPosts	ListBox	private		☐
	txtComment	TextBox	private		☐
	<add field>				
	⊿ **Events**				

Class Details LoginWebPage

	Name	Type	Modifier	Summary	Hide
	⊿ **Methods**				☐
	▷ Dispose	void	protected	Clean up any resources being used.	☐
	▷ InitializeComponent	void	private	Required method for Designer support - do not modify	☐
	⊿ LogIn	bool	public	This method allow user to Login, if the credential are confirmed, the user will be redirected to his/her homepage	☐
	(Username	string	None		
	Password	string	None		
	) <add parameter>				
	▷ LoginWebPage		public		☐
	▷ SignUp	void	public	The user will be redirected to registration webpage	☐
	<add method>				
	⊿ **Properties**				☐
	HomePage	HomePage	public		☐
	Password	string	public		☐
	Username	string	public		☐
	<add property>				
	⊿ **Fields**				☐
	btnLogin	Button	private		☐
	btnSignUp	Button	private		☐
	components	IContainer	private	Required designer variable.	☐
	txtPassword	TextBox	private		☐
	txtUserName	TextBox	private		☐
	<add field>				
	⊿ **Events**				

Class Details - PetOwner

	Name	Type	Modifier	Summary	Hide
	⊿ **Methods**				
	<add method>				
	⊿ **Properties**				
	AccountID	int	public		☐
	Comment	string	public		☐
	PostID	int	public		☐
	SearchResult	List<string>	public		☐
	<add property>				
	⊿ **Fields**				
	postType	object	private	postType is an object: Image, Post	☐
	<add field>				
	⊿ **Events**				

Class Details - ProfileWebPage

Name	Type	Modifier	Summary	Hide
Methods				☐
▷ Dispose	void	protected	Clean up any resources being used.	☐
▷ HomePage	void	public	Redirect the user to the homepage	☐
▷ InitializeComponent	void	private	Required method for Designer support - do not modify	☐
▷ ProfileWebPage		public		☐
▷ SaveProfile	void	public	Save any profile options changed	☐
<add method>				
Properties				
<add property>				
Fields				☐
btnHomePage	Button	private		☐
btnSave	Button	private		☐
chkBoxLocation	CheckBox	private		☐
chkBoxNofitfyNewPosts	CheckBox	private		☐
chkBoxNotifyNewServices	CheckBox	private		☐
chkBoxPrivacy	CheckBox	private		☐
cmbBoxFavoriteLanguage	ComboBox	private		☐
cmbBoxWhoCanFindMe	ComboBox	private		☐
cmbBoxWhoCanSeeMyHomepageAndPost	ComboBox	private		☐
components	IContainer	private	Required designer variable.	☐
dateTimePickerBirthdate	DateTimePicker	private		☐
lstBoxInterestOn	ListBox	private		☐
txtEmail	TextBox	private		☐
txtFirstName	TextBox	private		☐
txtLastName	TextBox	private		☐
txtMobileNumber	TextBox	private		☐
txtPassword	TextBox	private		☐
txtSafetyAnswer	TextBox	private		☐
txtSafetyQuestion	TextBox	private		☐
txtUsername	TextBox	private		☐
<add field>				
Events				
<add event>				

Class Details - Registration

Name	Type	Modifier	Summary	Hide
Methods				☐
▲ ConfirmEmail	bool	public	If the email is confirmed, this method set the parameter confirmed to true, and redirect the user to his/her profile page	☐
(email	string	None		
, Confirmed	bool	None		
) <add parameter>				
▲ CreateAccount	Account	public		☐
(newAccount	Account	None		
) <add parameter>				
<add method>				
Properties				☐
Account	Account	public		☐
Confirmed	bool	public		☐
<add property>				
Fields				☐
confirmed	bool	private		☐
<add field>				
Events				
<add event>				

Class Details - RegistrationWebPage

Name	Type	Modifier	Summary	Hide
▲ **Methods**				☐
▷ ⚙ CreateNow	void	public	If the mandatory fields are fullfilled, the system will send an email to confirm the registration	☐
▷ ⚙ Dispose	void	protected	Clean up any resources being used.	☐
▷ ⚙ InitializeComponent	void	private	Required method for Designer support - do not modify	☐
▷ ⚙ RegistrationWebPage		public		☐
⚙ <add method>				
▲ **Properties**				
🔧 <add property>				
▲ **Fields**				☐
🔩 btnCreateNow	Button	private		☐
🔩 components	IContainer	private	Required designer variable.	☐
🔩 dateTimePickercBirthdate	DateTimePicker	private		☐
🔩 txtEmail	TextBox	private		☐
🔩 txtFirstName	TextBox	private		☐
🔩 txtLastName	TextBox	private		☐
🔩 txtPasword	TextBox	private		☐
🔩 txtSafetyAnswer	TextBox	private		☐
🔩 txtSafetyQuestion	TextBox	private		☐
🔩 txtUsername	TextBox	private		☐
🔩 <add field>				
▲ **Events**				
⚡				

Class Details - SearchBar

Name	Type	Modifier	Summary	Hide
▲ **Methods**				☐
▷ ⚙ Search	void	public		☐
⚙ <add method>				
▲ **Properties**				☐
🔧 Hashtag	int	public		☐
🔧 SearchResult	int	public		☐
🔧 <add property>				
▲ **Fields**				☐
🔩 location	int[]	private		☐
🔩 petOwner	int	private		☐
🔩 serviceProvider	int	private		☐
🔩 txtSearch	string	private		☐
🔩 <add field>				
▲ **Events**				
⚡ <add event>				

Class Details - SearchBarWebpage

Name	Type	Modifier	Summary	Hide
▲ **Methods**				☐
▷ InitializeComponent	void	private	Required method for Designer support - do not modify	☐
▲ OpenMapResult	void	public		☐
(lstItemResult	object	None		
, location	Array	None	This array containing longitude and latitude	
) <add parameter>				
▷ SearchBarWebpage		public		☐
▲ SelectResult	void	public		☐
(lstItemResult	object	None		
) <add parameter>				
⚙ <add method>				
▲ **Properties**				☐
SearchResult	PetOwner	public		☐
<add property>				
▲ **Fields**				☐
btnHomePage	Button	private		☐
btnSearch	Button	private		☐
components	IContainer	private	Required designer variable.	☐
lstItemResult	object	private		☐
lstResults	ListBox	private		☐
txtSearch	TextBox	private		☐
<add field>				
▲ **Events**				
⚡				

Class Details - ServiceProvider

Name	Type	Modifier	Summary	Hide
▲ **Methods**				☐
▲ Messaging	void	public	Send invitation, appointment to the pet owner	☐
(accountID	int	None		
) <add parameter>				
▲ Streaming	void	public	Open 3rd part software (Zoom, Skype, GoogleMeet)	☐
() <add parameter>				
⚙ <add method>				
▲ **Properties**				☐
AccountID	int	public		☐
Commest	string	public		☐
PostID	int	public		☐
SearchResult	List<string>	public		☐
<add property>				
▲ **Fields**				
<add field>				
▲ **Events**				
⚡ <add event>				

State Diagrams

SearchBar

New Search

Click on SearchBar → Awaiting text input — Type in search keywords → Get input for searching

Search in Database

Start New Search

Loading — Search failure → No match result

Search Sucess

Display Search Result

End Search

End search

Registration

Click on "Create Account" button → Opening New Form → Loading User Information Page

Click on 'Cancel' Button → Loading Home Page

entry /
Email
Date time
First Name
Last Name
Password
Choose/
Safety Question
entry/
Safety Answer
Username

Click on "Create Now" button

Data is Not Valid

Confirming Data Is Valid

Loading User Information Page

do/display invalid information

Data is Valid

Click on "Create Now" button

Page Loading

Creating Account

do/
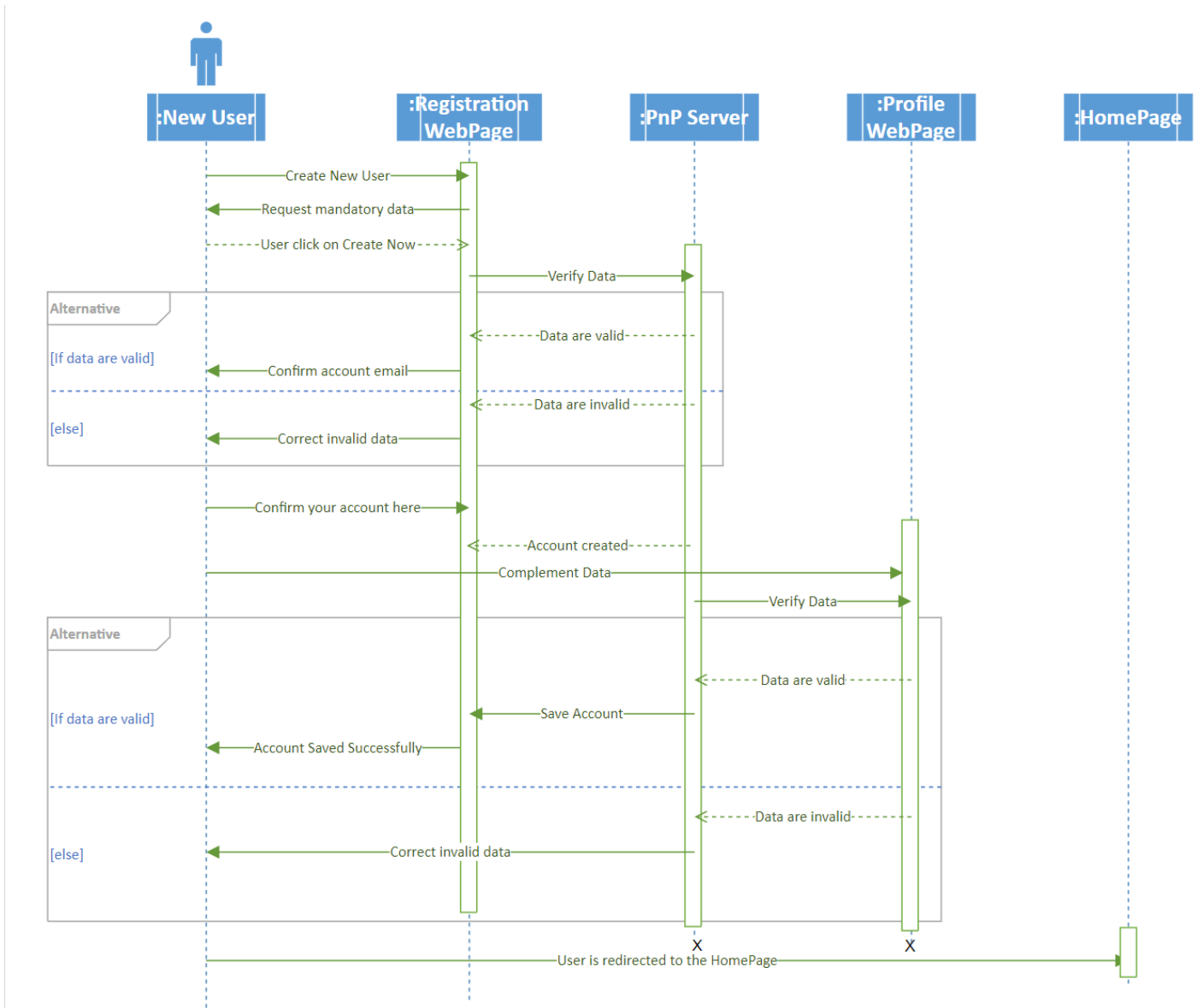Display page to Confirm Account has been created and confirmation email has been sent.

Sequence Diagram

"Party Analysis pattern"

For deliverable 5, we carefully analyzed and compared the benefits of the domain class diagram and the analysis pattern model. Although we concluded that having a supertype can be helpful in showing common attributes of the two users as they relate to the "party" class, we felt it was necessary to provide a more in-depth diagram due to the complexities of our application. To help us better understand the analysis pattern

model we put together the diagram which was helpful in demonstrating the different relationships that can exist between supertypes and the user but also highlighting their commonalities.

In the updated class diagram using the analysis pattern, we define a party as the supertype of an account of either a pet owner or a pet service provider. It shows the sharing attributes like addresses, phone numbers, or emails for both account types, which was already reflected in the original domain class diagram. The unique part about the updated diagram is the reflection of the relationship between the party and the common attributes, such as one to one or one to many. Overall, the original domain class diagram has more thorough information and details of each class, its attributes, methods and the relationships between different classes.
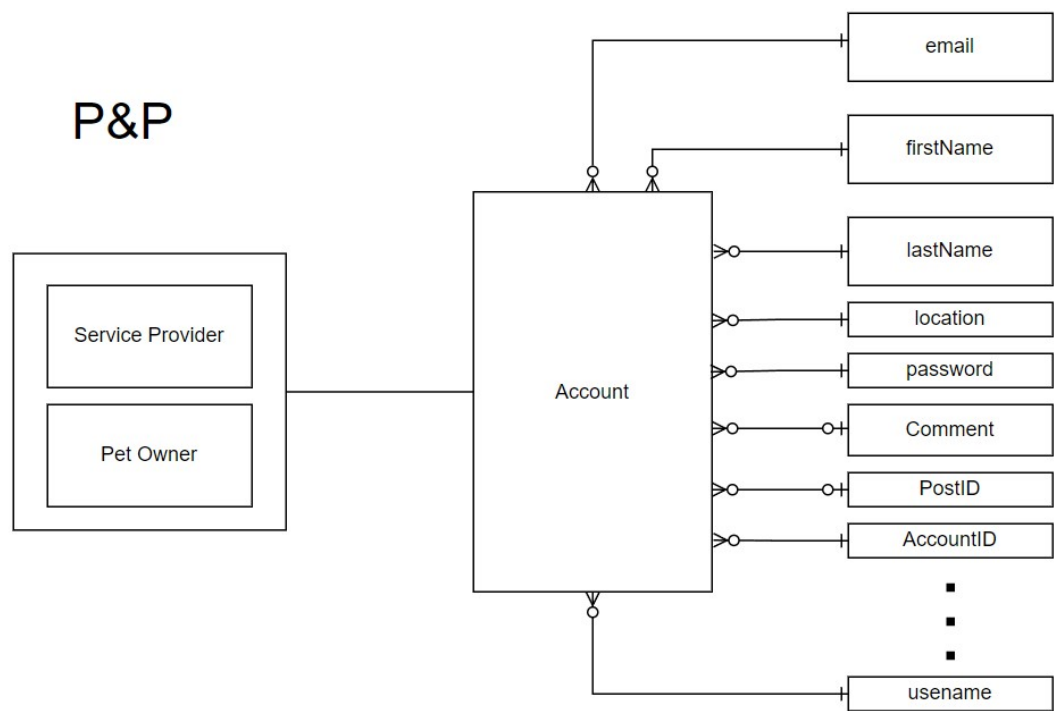
Quality Report

Attribute	Metric

1. Ambiguity	Number of ambiguous modifiers:
2. Completeness	Number of TBA, TBD: 3
3. Understandability	Number of sections/subsections:
4. Volatility	Number of changes per requirement Time (by activity) when change is requested: 3
5. Traceability	Number of requirements not traceable to design/code: 1
6. Model clarity	Number of UML models: 4 Number of descriptive pages per model: 3 Number of UML errors: 13

1. Ambiguity

The SRS document presented a minimal number of ambiguous modifiers. In total, we counted the word "many" two times and the word "people" 1 time. This demonstrates the authors had a great understanding of their project and in turn, makes it easier for the reader to have clear expectations of the software.

2. Completeness:

The report is overall complete and follows the instructions specified in each part. Each part of the report is thoroughly developed with essential details.

3 TBD and 0 TBA were found: Other Requirements, Appendix A Glossary, and Appendix B Analysis Models, which do not affect the report completeness at this stage.

3. Understandability

The paper's structure presented both sections and subsections which were outlined in the table of content, making the document easy to read, follow and reference. The document had a total of 12 sections and 41 subsections. We did note a comment was left under section 3.2, which contains information that could be kept for the authors only.

4. Volatility

Number of Volatility: 3

The use case is perfectly adapted to requirements. 13 functional requirements, 13 use cases, no requirements are modified, added, deleted, or changed.

Among the 4 user documentation requirements, one was indicated to be delivered in a later version, and the other three were not set to the required time or reflected on the non-functional requirements. It was considered 3 changes.

5. Traceability

Number of Traceability: 1

Display results with links to each respected website (Req-5).

Since not every book has an electronic version and not every book is authorized to the website database, there will be a challenge to have a database that can meet every student's search needs. Also, the required database is too large, so it is considered that this requirement is not traceable to design/code.

6. Model clarity

Number of UML models: 4

1 Use Case Diagram (4.16, page 13),

1 Swimlane Diagram (4.17, page 14),

2 Domain Class Diagram (ii–iii, pages 24-25)

Number of descriptive pages per model: 3

1 Formal Use Case (4.15, page 11)

      useful for Use Case Diagram and Swimlane Diagram

1 Functional Requirements (pages 16-17)

      useful for Domain Class Diagram

1 CRC cards (Appendix E, pages 20-23)

      useful for Domain Class Diagram

Number of UML errors: 13

Use Case Diagram:

    1. The Use Case Diagram Each use case does not show if "include" or "extend" from/to another use case.
    2. In the Account subsystem the Admin, or Third Party Database actors should validate the "Access to the LiBook", and "Log in through post-secondary institution" use cases.

3. There are no relations between the use cases "Search for a study material" and "Display and sort results" since the "Display and sort results" is a result of "Search for a study material".

4. The use case "Scan book cover and convert into searchable parameters" does not show any results.

5. If the use case "Search image parameters" is intended as an option to have a better result from the search, it should be related to "Search for a study material" use case.

6. The use case "Place order", "Check Out", and "Payment Method" should be under another subsystem rather than Search subsystem.

7. "Track order" use case should be accessible after an order has been placed.

8. Looks like only for the subsystem Account the Student needs to be logged in.

Swimlane Diagram:

1. The Swimlane Diagram shows the option "Add to cart" after clicking on a result, that is not mentioned in the Formal Use Case. If "Add to cart" is part of the "Search for a study material" Formal Use Case, it should be added to the document, but if not, it should not be shown in the Swim Lane Diagram.

Domain Class Diagram:

1. The class "Student" doesn't own any kind of class methods, I'm not sure how "Student" can perform any actions.

2. The class "Student" should use the "Search Method" or at least visualize the result of his/her search. The "Student" class has a list of objects "Study Materials", which is also used by the "Third Party Database" class, but there is not any relation between "Third Party Database" class and "Study Materials" class.

3. The class "Student" does not interact with "Cart" class, and neither uses any class method to perform payment or download the course text after payment.

4. There are no classes to log in or register a user in the software.