

P01-PetsWala

SPROJ Report



Muhammad Tayyab, 22100200

Muhammad Aaish Javed, 22100300

Muhammad Ibrahim Bhalli, 22100098

Syed Raza Abbas, 20100049

Advisor:

Waqar Ahmad,

Basit Shafiq

**School of Science and Engineering
Lahore University of Management Sciences
Submission Date**

Acknowledgement and Dedication

We would like to thank our supervisors for their guidance and assistance every step of the way, without which we would not have been as effective as we were during the course of this project. Every growth-related and learning opportunity we encountered, we owe our esteemed supervisors and mentors.

Special gratitude goes out for Team PO1-PetsWala for the consistent and dedicated efforts of every individual active member without which the timely completion of **PetsWala** would not have been possible.

We wish the team the best for all future endeavors!

Certificate

I certify that the senior project titled “**P01-PetsWala**” was completed under my supervision by the following students:

1. Muhammad Tayyab, 22100200



2. Muhammad Aaish Javed, 22100300



3. Muhammad Ibrahim Bhalli, 22100098



4. Syed Raza Abbas, 20100049,

Did Not Respond

and the project deliverables meet the requirements of the program.

Date:

Advisor (Signature)

Date:

Co-advisor (if any)

Table of Contents

Introduction

- [Objective and Scope](#)
- [Development Methodology](#)
- [Contributions](#)

System Requirements

- [System Actors](#)
- [Functional Requirements](#)
- [Non-functional Requirements](#)

System Architecture

- [Architecture Diagram](#)
- [Architecture Description](#)
- [Justification of the Architecture](#)
 1. [1. Django MVC Justification](#)
 2. [2. Layered Architecture Justification](#)
 3. [3. Client-Server Architecture Justification](#)
- [Tools and Technologies](#)

Requirements Specifications

- [Use Cases](#)
 1. [Sign Up of General Users](#)
 2. [Sign Up of Service Providers](#)
 3. [Sign Up of Veterinary Doctors](#)
 4. [Sign Up for Rescue Services](#)
 5. [Login](#)
 6. [Logout](#)
 7. [Change Password](#)
 8. [Reset Password](#)
 9. [Create Blog Posts](#)
 10. [Edit Blog Post](#)
 11. [Comment on Blog Posts](#)
 12. [Review and Rate Services](#)
 13. [Report User or Service](#)
 14. [Approve Service Provider/Vet/Rescue Service Profile \(Admin\)](#)
 15. [Search on Blog](#)
 16. [Search Marketplace](#)
- [Class Diagram](#)
- [Sequence Diagrams](#)
 1. [Signup of the general user:](#)
 2. [Approve Service Provider/Vet/Rescue Service Profile \(Admin\)](#)

3. [Review and Rate Services](#)
4. [Signup of Service Provider](#)

[Software Development Methodology and Plan](#)

- [Software Process Selection](#)
- [Gantt Chart](#)

[Database Design and Web Services](#)

- [Database Design](#)
- [API Specification](#)

[System User Interface](#)

[Project Security](#)

- [Project Threats](#)
- [Potential Losses](#)
- [Security Controls](#)
- [Static and Dynamic Security Scanning Tools](#)

[Risk Management](#)

- [Potential Risks and Mitigation Strategies](#)

[Testing and Evaluation](#)

[Deployment Guidelines](#)

[Conclusion](#)

- [Summary](#)
- [Challenges](#)
- [Future](#)

[Review checklist](#)

[References](#)

List of Figures

- [Architecture Diagram](#)
- [Use Cases Diagrams](#)
- [Class Diagram](#)
- [Sequence Diagrams](#)
- [Gantt Chart](#)
- [E/R Diagram](#)
- [User Interface Snippets](#)

1. Introduction

In this chapter we will be discussing the objective and scope, developmental methodology and contributions.

a. Introduction

We developed a web application that registers pet owners, vets, rescue, and other pet service and accessories providers and allows them to interact based on their needs. For instance, pet service providers and vets would be able to advertise their services and products and pet owners would be able to contact these service providers and avail their services.

Our web app is business software.

Target users of this application are pet owners, vets, vendors, rescue services, and other pet-related services.

We have developed a web app.

b. Objective and Scope

We chose to develop this web app because currently in the market, there is a gap for such platforms built for pet services that provide all the services under a single platform. Our web app provides all the pet-related services under a single platform. It would attract the customers as it will save them the hassle to keep track of services related to their pets on different platforms.

c. Development Methodology

We used the Scrum development methodology. We believe it is the most productive means of development as the back-and-forth approach to problem-solving takes feedback into account on a regular basis. This leads to fewer gaps in requirement specification and the actual implementation.

d. Contributions

Our platform provides all the pet-related services in a single place; that is something has not been done before. This makes our platform better and more effective than the others as our platform will save the users the hassle of keeping track of services related to their pets on different platforms.

2. System Requirements

In this chapter, we have discussed all the functional and non-functional requirements of our project.

a. System Actors

Actor Name	Description
User	Users can purchase products from the marketplace. They can see the orders that they have placed on their dashboard. Users can find nearby service providers and vets. They can book appointments with these services. They can request a rescue from rescue services and track them on their way to them. They can give services-related feedback, ratings, and reviews. They can create posts for articles on the Blog.
Admin	Admins can Approve/Delete Service Providers, Vets, and rescue services profiles. They can receive reports against all types of users on the platform. They can block the reported users and can also delete the reports.
Vets	Vets can provide their experience, services, and timings so the users can contact them. They can see their booked appointments with the users. They can remove the appointments from their list after they are done with them.
Rescue Services	They can receive the rescue requests from the users and update the rescue request's status appropriately. They can get directions to the user's provided location by redirecting to google maps. They can also post on blogs about the pets that need to be adopted.

Service Providers	Service providers can advertise their services. Users can book appointments with them on the available slots. They can see their booked appointments with the users. They can remove the appointments from their list after they are done with them.
Vendor	Vendors can advertise the products they are selling on the marketplace. Orders from the users will show up on the vendor's dashboard. Vendors can see all the details of the user to deliver them the products. Vendors can update the status of orders accordingly.

b. Functional Requirements

Requirements	
Sr#	Requirement
1	As a general user I want an app that facilitates taking care of my pet. Where I can log in and visit the blog. I can browse the marketplace for products that I might require for my pet. I can add products to the shopping cart and checkout my orders and see the current status of my orders. I need the contact information for rescue services in case of emergencies. I need the location of local vets and service providers. I want to be able to rate and review the aforementioned vets and services. I want to be able to search for accessories and services that I might require. I want to be able to post blogs and post comments on the blogs. I should be able to update my profile information and my login password.
2	As an administrator I need the highest privileges required for managing services, marketplaces, and user interactions. I need the ability to add/edit/delete veterinary profiles. I need the ability to verify and approve service providers in order to filter spam. I need the ability to edit or remove offensive posts from the blog. I need the ability to block the profiles that have been reported.

3	As a service provider I need to get in touch with interested clients conveniently and offer my services through the app. I need access to the marketplace to advertise my services. I want to register with the role of service provider and display my location, contact information, and rates for the users on the marketplace. I want to be able to add/edit my advertised services. I want my reviews and ratings to be displayed alongside my services. I need to see the booked appointments with me and remove the ones that are completed. I should be able to update my profile information and my login password.
4	As a rescue service , I need to deal with user rescue requests, post on the blog, and get in touch with users. I need the ability to register my team. I want to be able to receive requests from the users and resolve them. I want to display my location, and contact information to the users. I want to be able to contact the users in case of an emergency and be able to get directions to the user-specified address. I want a dashboard with the user requests where I can update the requests' status. I should be able to update my profile information and my login password.
5	As a veterinary doctor , I want to display my location, contact information, experience, and the services I am offering to the users. I want to display my timings and also want the users to rate and review me. I also want access to the blog in order to post helpful tips and medical advice. I need to see the booked appointments with me and remove the ones that are completed. I should be able to update my profile information and my login password.
6	As a Vendor , I should be able to advertise the products that I am selling on the marketplace. I should be able to see all the orders that I have received and I should be able to update their status. I should be able to update my profile information and my login password.

c. Non-functional Requirements

Sr#	Requirements
1	The system will process search requests in less than 3 seconds.

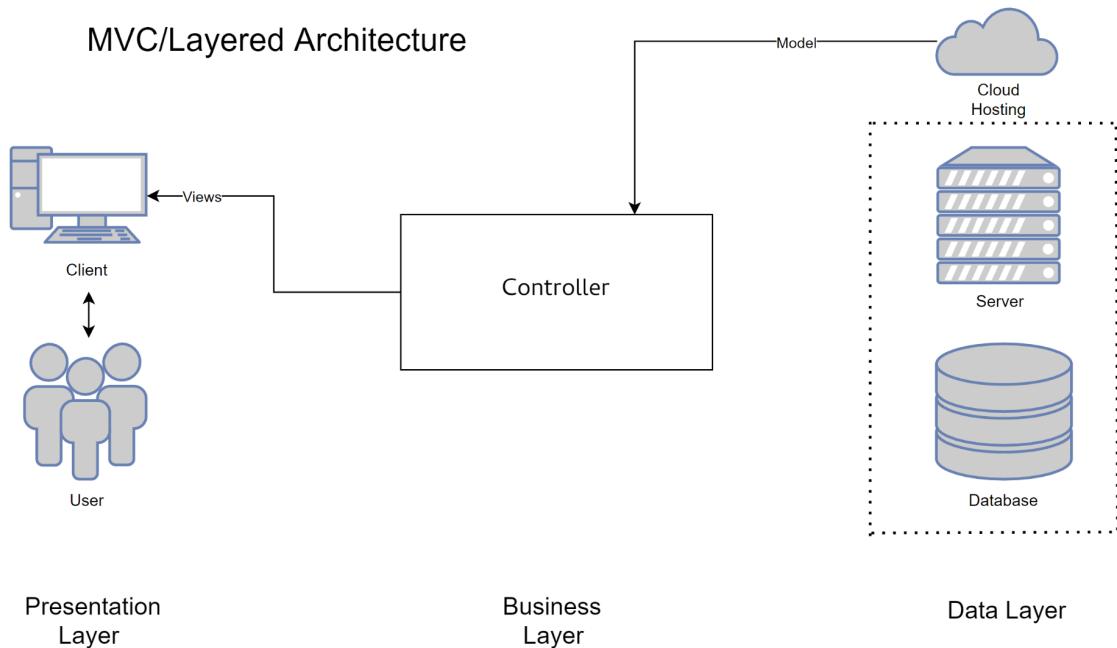
2	The system should have a secure database and should be protected from SQL injection attacks using form data input validation, parameterized queries, etc. This should prevent breaches in sensitive data and will provide a more secure experience to our users.
3	Modified data in a database should be updated for all users accessing it within 3 seconds. If any data variable has been modified, the webpages for all the users should reflect that within 5 seconds.
4	The system should secure user identity and preventative measures should exist against potential security breaches. The passwords will be hashed, and password change functionality will be provided to the users. The user will be able to contact the admin and disable his/her account in case of a breach.
5	Anonymous users are not allowed to access and manipulate data. Only authenticated users are allowed to manipulate and access the database.
6	SSL Certification must be satisfied, which secures an internet connection and safeguards data which is being sent between server and browser.
7	The system must be fully operational at all times unless scheduled maintenance is required for which downtime will be pre-specified.
8	Users must log in and be authorized in order to access their accounts. Admins and every other actor must also follow the authentication procedure in order to access their respective accounts.
9	Admins will have administrator privileges and the most access rights on the system. Every actor will act within its domain and cannot access other rights. The administrator will be responsible for granting or revoking these rights.
10	Users can only place orders once they provide their relevant information.
11	The system shall not be shut down for maintenance more than once in a 24-hour period
12	The system shall provide a uniform look and feel between all the web pages.
13	The system should have an easy-to-use GUI so that all functionality is self-explanatory and intuitive.

3. System Architecture

In this chapter, we discussed the architecture of our project and its justification.

a. Architecture Diagram

Unable to incorporate classes in MVC architecture. The following diagram is simple and explains the high-level architecture of our diagram. **For the sake of simplicity, the modules were not represented in the diagram but have been explained below.**



b. Architecture Description

- **User:**

The user uses a client machine (any device with modern web browser support) with a decent functioning internet connection to interact with our system. Examples of client machines include but are not limited to laptops and desktops (running Windows, Mac OSX, Linux), modern iOS and Android devices, etc.

- **Client (Presentation Layer):**

The client is the machine responsible for communicating with the server over the internet. It must have an active internet connection. This is where all the user interaction and user experience happen (UI/UX). The client fetches data from the server and displays it to the client. This layer is called the presentation layer because the data is rendered from the server and presented to the client in this layer. The client requests the server for files necessary to display the website and data according to the user's needs. The stacks that we use to accomplish this are HTML, CSS, Javascript, and Django. The client is where the front end of the system exists.

- **Server:**

The server is the machine responsible for handling all incoming client requests. This is the physical location where all the application layer and business layer is implemented. We use AWS to rent a cloud-based server responsible for hosting the service (Petswala). There are a variety of AWS packages available, the free ones being the lowest tier obviously. Our server must have a large enough bandwidth to handle incoming traffic without a bottleneck. Preferably there should be flash storage (SSD) to make processing faster and avoid fail-stop failures. The server must have at least 8-16GB of high-speed RAM. The server will be running Ubuntu, and we'll be using services like *Nginx* and *Gunicorn* for the deployment of our website. The server is where the business layer is also implemented. The server is where the backend of the system exists.

- **Business Layer:**

The business layer is the part within the server where the business logic is implemented. Since the business layer is intricately connected with the application layer, we have represented them together in the diagram. The application layer uses Python on Django. This layer is made up of many subsystems, like a model view controller, a CRUD, business logic, etc. For the sake of simplicity, we have not represented every subsystem in the diagram.

1) MVC:

The model view controller is a popular schema/pattern for implementing web applications. The model manages system data and primarily interacts with the database. The view is responsible for managing the data presented to the user. It renders the pages for the users. The controller manages user interactions and passes them to the model and view components. It interacts with the file system in order to fetch and save data.

2) CRUD:

The crud will primarily be used by the administrator and will be part of the admin panel. It has administrator privileges and can edit the database and the file systems. This will be used for registering services, rescue services, and vets. It can also be used to edit or remove blog posts or edit user credentials.

3) Business Logic:

Business logic is the main subsystem and application logic. This will be composed of various modules that are programmed for a specific purpose. These modules will be mostly independent which will ensure efficient maintenance and upgrades. The code will be modular and object-oriented.

It will have a signup and login module, which will be called the authentication module. It will be responsible for storing the user credentials in the database and verifying them upon login.

There will be an admin module that will be responsible for the admin panel. It will have administrative functions and privileges.

There will be a fetch/search module responsible for fetching data from the database. This is necessary for implementing search functionality throughout the website.

Blog management will be a module that implements the functionality of creating blog posts, editing blog posts, etc. It also implements the functionality of commenting and rating/reviewing.

A comms module will be responsible for implementing the chat functionality in the application.

Similarly, there will be a services module responsible for implementing the functionality of services on the website.

A module of the marketplace will deal with the specifics of how the marketplace is implemented, and how the prices, images, and additional information are fetched from the database and file system.

- **Data Layer:**

Data layer provides access to the database and file system. The system will be using the Postgres database. The database will store the encrypted information of the clients. Since the data layer is the last layer in the layered architecture hierarchy, by using this architecture, we are mitigating the danger of byzantine failures.

The data layer also provides access to the file system which holds all the files used by the backend and the server; from HTML, CSS, JS, and Python files to images and other resources that the system may require.

The data layer provides access to everything the system needs to be functional. It's important that the data layer is secure and impervious to failures.

- **Interaction of subsystems:**

The front end interacts with the server (backend) in order to render the pages and provide optimal user interaction and experience. The model view controller interacts with the file system, database, and front end. The CRUD and business layer also interact directly with the data layer. The details of how every submodule interacts with every other submodule are beyond the scope of this document.

c. Justification of the Architecture

We're using a combination of three system architectures:

- Client-Server
- Layered
- MVC (Model View Controller)

1. Django MVC Justification

We're developing a complex system with several components, which will require a lot of programming. The Django MVC architectural framework will allow us to use pluggable modules for basic functionality such as user input validation, secure user profiles, information, etc. Since we are already familiar and affluent with Python, the learning curve for the framework will not pose any challenges. Moreover, the framework provides protection against common attacks such as SQL Injection.

Pros	Cons
Instead of writing the code, developers can use readymade packages for adding functionalities. This saves valuable time.	It proves challenging for developers who have worked on frameworks with Convention over Configuration.
Since it is written in Python, a versatile programming language, Django for web development provides more flexibility and dynamism to the developers.	Django calls for a lot of coding, which takes the server's processing time and bandwidth while developing. It is generally used for projects which need scaling or are going to be launched on a large scale.
Faster development. Startups and enterprises can utilize it to develop rapid MVPs and get more time to market the product.	The Steep learning curve of this web framework is challenging for developers who are switching to Python.

The Django REST framework has a modular and configurable architecture which makes API development easier.	Unlike most web development frameworks, Django can't handle multiple requests simultaneously as it encourages developers to explore individual processes and make decisions.
Django is more advanced and is compatible with some of the powerful machine learning libraries like PyTorch, NumPy, etc.	Although it is easy to learn, its mandatory nature means that developers cannot use their own file structure. One needs to follow those rules to deploy anything using Django.
It also includes the prevention of common attacks like SQL injections and cross-site request forgery.	

2. Layered Architecture Justification

Most web applications require a layered architecture. The front end is just static web pages without the backend to link them and implement the functionality. Similarly, the backend depends on the data layer to fetch the data that it requires to implement the business logic and application code.

Pros	Cons
Redundant features (authentication) in each layer can enhance security. Requests for each page will include a token that will recheck authentication and authorization for the user.	Management cost increases if there are too many layers.
Allows replacement of entire layers as long as the interface of the layers does not change.	The performance is affected as more and more layers are added.

When changes occur, only the adjacent layer is affected.	
--	--

3. Client-Server Architecture Justification

No web application can exist without a client-server architecture. We want a web application for people with pets. This non-functional requirement can only be satisfied with a client-server architecture. Furthermore, there are going to be a number of clients connecting to the server and interacting through it. The only practical way of meeting the non-functional requirements of our intended system is that we use the client-server architecture.

d. Tools and Technologies

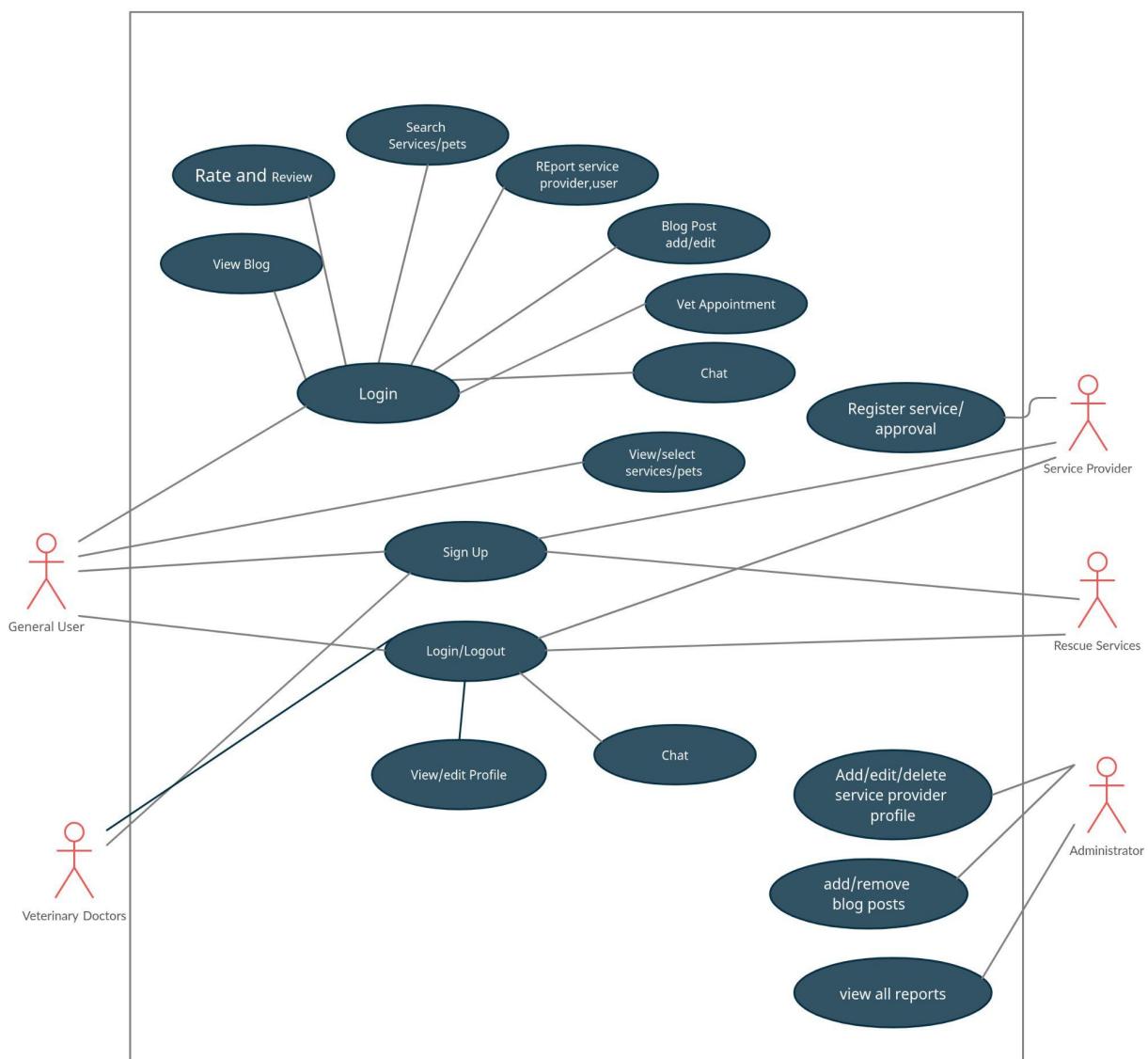
- Django(v3.0 or above)
- Python(v3.7 or above)
- AWS CLI (v2)
- HTML5
- CSS (v2.1)
- Github Project Planner
- Javascript (ES2015)
- Github
- Git

4. Requirements Specifications

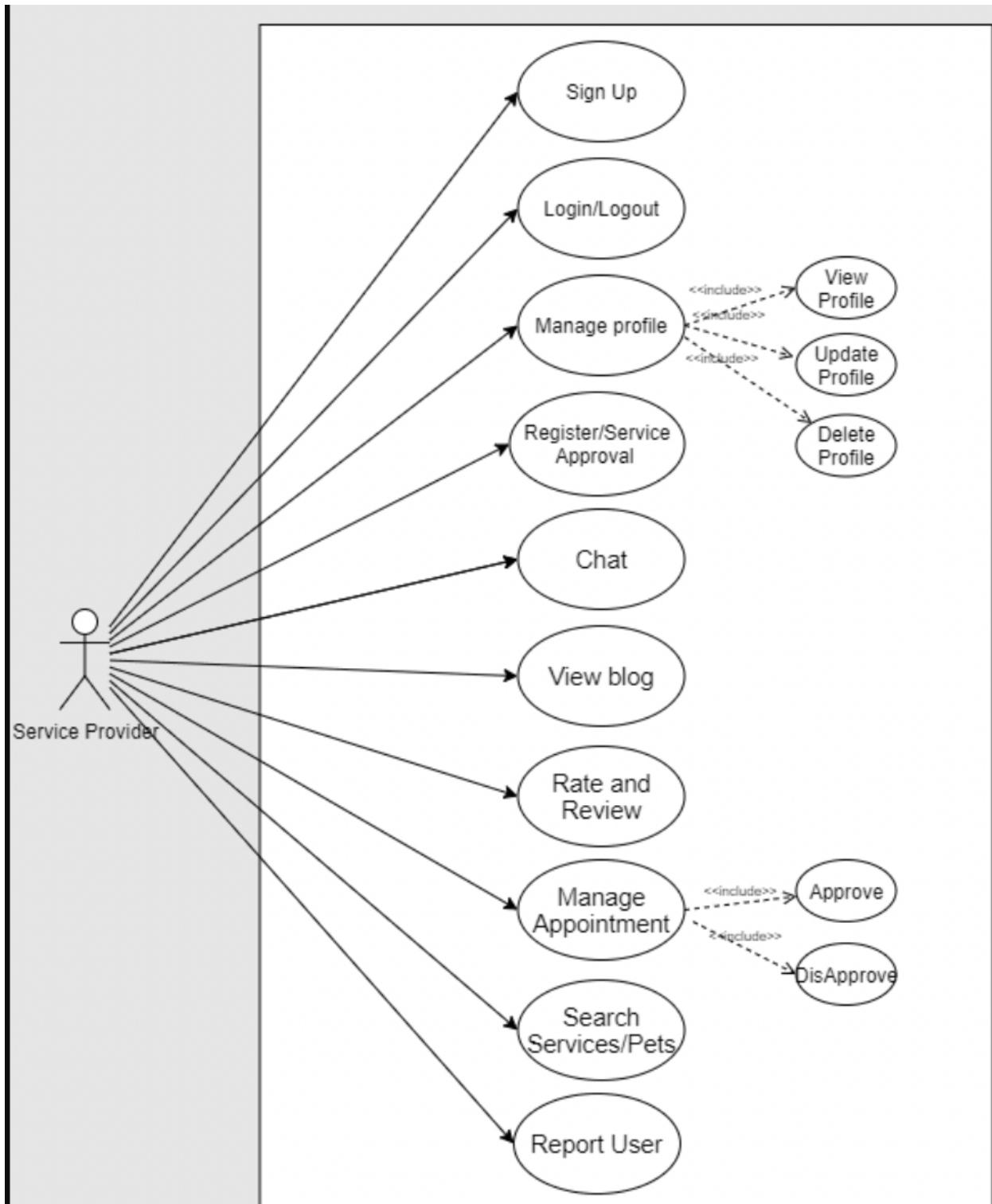
In this chapter, we have discussed the core use cases of our system, their Class diagrams, and their Sequence Diagrams.

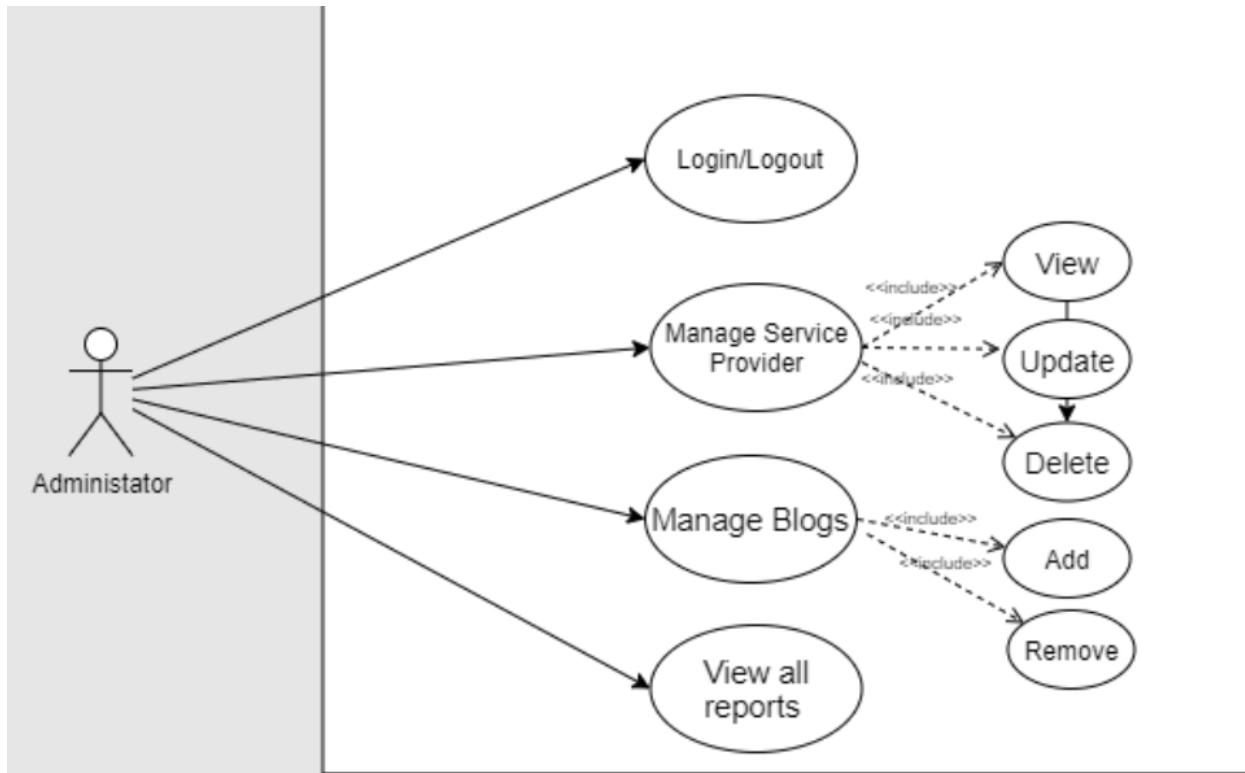
a. Use Cases

- Use Case diagrams:









- **Use Case Description:**

1. **Sign Up of General Users**

Identifier	UC-001
Purpose	The Users register their account on the website.
Pre-conditions	<p>The <i>User</i> should have a valid email address.</p> <p>The <i>User</i> must not already have an account.</p>

Post-conditions		The Sign Up is successful and the account is created. If the User already has an account, they will be directed to the login prompt.
	Typical Course of Action	
1.	The user clicks on the sign-up button for pet owners.	
2.	The system redirects the user to the sign up page.	
3.	The system prompts the user with a form to enter his personal information including name, contact, date of birth, and email address, and asks the user to upload a profile picture .	
4.	The user enters his personal information including name, contact, and date of birth.	
4.	The user enters his email address.	
5.	The user uploads his profile picture.	
6.	The user submits the form.	
7.	The system validates the form and if the user has entered the information correctly then the account is created.	

8.	The use case ends.
	Alternate Courses of Action
	After step 1 the user can choose the signup with Google option and go directly to step 7.
	Exception Paths
	In step 7, if the user enters invalid information or if the email address is already registered an error message is shown.

2. Sign Up of Service Providers

Identifier	UC-002
Purpose	The Service Providers register their account on the website.
Pre-conditions	<i>The Service Providers should have a valid company letterhead and email address.</i>
Post-conditions	The Sign Up is successful and the account is created.

Typical Course of Action	
1.	The Service Provider clicks on the sign-up button for Services. and is redirected to the sign up page.
2.	The system redirects the service provider to its sign up page.
3.	The system prompts the service provider with a form to enter the relevant information, upload a profile picture, and enter the email address, username, and password.
4.	The Service Provider enters their company information.
5.	The Service Provider uploads their profile picture.
6.	The Service Provider enters their email address.
7.	The Service Provider chooses their username and password.
8.	The Service Provider submits the form.
9.	The system validates the form and displays a pending verification message on the user's screen.
10.	The system redirects the user to the home screen.

11.	The system forwards the sign-up request to the Admin team for verification.
12.	After verification, the system sends the Service Provider a confirmation email.
10.	The use case ends.
	Alternative Courses of Action
	The admin team rejects the signup request, and the system sends a rejection email to the service provider.
	Exception Paths
	In step 5, if the Service provider enters invalid information or if the email address is already registered an error message is shown.

3. Sign Up of Veterinary Doctors

Identifier	UC-003
Purpose	The Vets register their accounts on the website.

Pre-conditions	The Vets should have a valid medical license and email address.
Post-conditions	The Sign Up is successful and the account is created.
Typical Course of Action	
1.	The Vet clicks on the signup button for Vets.
2.	The system redirects the vet to its sign up page.
3.	The system prompts the vet with a form to enter medical information, email address, username, password, and upload a profile picture.
4.	The Vet enters their medical information.
5.	The Vet uploads their profile picture.
6.	The Vet enters their email address.
7.	The Vet chooses their username and password.
8.	The Vet submits the form.
9.	The system validates the form and displays a pending verification message on the vet's screen.

10.	The system redirects the vet to the home screen.
11.	The system forwards the Sign-up request to the Admin team for verification.
12.	After verification, the system sends the Vet a confirmation email.
10.	The use case ends.

	Alternative Courses of Action
	The admin team rejects the sign-up request, and the system sends a rejection email to the vet
	Exception Paths
	In step 5, if the Vet enters invalid information or if the email address is already registered an error message is shown.

4. Sign Up for Rescue Services

Identifier	UC-004
------------	--------

Purpose	The Rescue Services register their account on the website.
Pre-conditions	<i>The Rescue Services should have a valid organization and email address.</i>
Post-conditions	The Sign Up is successful and the account is created.
	Typical Course of Action
1.	The Rescue Service clicks on the signup button for Vets. and is redirected to the sign-up page.
2.	The system redirects the rescue service to the sign up page.
3.	The system prompts the rescue service with a form to enter company information, email address, username, password and upload a profile picture.
3.	The Rescue Service enters their company information.
3.	The Rescue Services upload their profile picture.
4.	The Rescue Service enters their email address.
5.	The Rescue Service chooses their username and password.

6.	The Rescue Service submits the form.
7.	The system validates the form and displays a pending verification message on the rescue service's screen.
8.	The system redirects the rescue service to the home screen.
9.	The system forwards the Sign-up request to the Admin team for verification.
10.	After verification, the system sends the Rescue Service a confirmation email.
11.	The use case ends.

Alternative Courses of Action	
	The admin team rejects the signup request, and the system sends a rejection email to the rescue service.
Exception Paths	
	In step 6, if the Rescue Service enters invalid information or if the email address is already registered an error message is shown.

5. Login

Identifier	UC-005
Purpose	The Users can log in to their account.
Pre-conditions	<p><i>The Users should have already registered their account.</i></p> <p>The Users should be logged out at the time they wish to log in.</p>
Post-conditions	The Users log in and are redirected to their dashboard.
Typical Course of Action	
1.	The user clicks on the login button
2.	The system redirects the user to the login page
3.	The system prompts the user with a form to enter their username and password.
4.	The User enters their Username.
5.	The User enters their password.
6.	The user clicks on login

7.	The system validates the form and if the credentials match it redirects the user to the dashboard
10.	The use case ends.
Alternate Courses of Action	
	In case of a failed validation, the user can click on the forgot password button.
	Exception Paths
	In step 3, if the Username/Password combination is incorrect an error message is shown.

6. Logout

Identifier	UC-006
Purpose	The Users can Log Out from the system.
Pre-conditions	The <i>Users should already be logged in.</i>
Post-conditions	The Users are logged out and redirected to the home page.

	Typical Course of Action
1.	The User clicks on the Logout Button
2.	The system logs the user out.
2.	The system redirects the user to the home page.
3.	The use case ends.
	Alternate Courses of Action
	Exception Paths

7. Change Password

Identifier	UC-007
Purpose	The Users can change their password.
Pre-conditions	<i>The Users should know their old password.</i>
Post-conditions	The password is changed
Typical Course of Action	
1.	The User navigates to the Edit profile page
2.	The User clicks on the change password button.
3.	The system redirects the user to the password change page.
4.	The system prompts the user with a form asking for their old password and new password.
5.	The User enters their old password.
6.	The User enters a new password.
7.	The system validates the form, updates the password in the database, and redirects the user to the password change successful page.
8.	The use case ends.

	Alternate Courses of Action
	If the form is not valid and the credentials are incorrect, the system displays an error message to the user.
	Exception Paths
	In step 5, if the old Password is incorrect an error message is shown.

8. Reset Password

Identifier	UC-008
Purpose	The Users can reset their password in case they forget.
Pre-conditions	The <i>Users should know their registered email address.</i>
Post-conditions	The password is changed. The previous password should not work.
Typical Course of Action	

1.	The User clicks on the forgot password button on the home page.
2.	The system redirects the user to the password reset page.
3.	The User enters his email address.
4.	The User gets an email link to reset their password.
5.	The User clicks on the link and is redirected to the change password page.
6.	The User enters a new password.
7.	The password is validated and reset.
8.	The User is redirected to the homepage.
7.	The use case ends.
	Alternate Courses of Action
	In step 7, if the user enters an invalid password or the old password, an error message will be shown and the User will be prompted to re-enter the password

	Exception Paths
	In step 2 if the email address is incorrect an error message is shown.

9. Create Blog Posts

Identifier	UC-009
Purpose	The Users can create a new blog post.
Pre-conditions	The <i>User</i> should be present on the Blog Page.
Post-conditions	A new Blog post is created.
Typical Course of Action	
1.	The user clicks on the Add Post option.
2.	The system redirects the user to an Add New Post page.
2.	The user enters text and/or images.
3.	The user clicks on the submit button.

4.	The System shows a “Post Successfully added” message.
5.	The post is published.
6.	The user is redirected to the Blog Page.
7.	The use case ends.

	Alternate Courses of Action
	Exception Paths

10. Edit Blog Post

Identifier	UC-010
------------	--------

Purpose	The Users can edit their Blog posts.
Pre-conditions	The Post should already be published.
Post-conditions	The Post is updated.
	Typical Course of Action
1.	The user clicks on the edit post button.
2.	The system redirects the user to an edit blog post on page
2.	The user makes changes to their Blog Post.
4.	After making the changes the user clicks on save changes
5.	The system redirects the user to the home page with a success message that the edit has been successful.
6.	The system updates the database
7.	The use case ends.
	Alternate Courses of Action

	Exception Paths
	In step 4, if the post is not approved the user receives a message in their inbox.

11. Comment on Blog Posts

Identifier	UC-012
Purpose	The Users can comment on Blog Posts.
Pre-conditions	The <i>User</i> should be present on the Blog Post they want to comment on.
Post-conditions	The User must be able to see their comment posted under the post.
Typical Course of Action	
1.	The user clicks on the Add comment button.
2.	The system opens a text box for the user to type in.

3.	The user types in their comment.
4.	The user clicks on the post comment button.
5.	The system shows a “comment successfully added” message.
6.	The comment is posted
7.	The use case ends.

	Alternate Courses of Action
	None
	Exception Paths
	None

12. Review and Rate Services

Identifier	UC-013
------------	--------

Purpose	The Users can review and rate services
Pre-conditions	<p>The <i>user</i> should have availed the service.</p> <p>The <i>user</i> must be present on the service profile.</p>
Post-conditions	The User's review and rating is updated on the service profile.
	Typical Course of Action
1.	The user clicks on the review button/field.
2.	The system opens a text box for the user to type in.
3.	<p>The user types in their review.</p> <p>The user rates the service out of 5 using stars.</p>
4.	The user clicks on the post review button.
5.	The system shows a “successfully posted review” message to the user.
6.	The user review and rating are updated on the service profile.
7.	Use case ends.

	Alternate Courses of Action
	If the user has not availed of the service, they will not be able to click on the review and rating button.
	Exception Paths
	None

13. Report User or Service

Identifier	UC-014
Purpose	The User can report inappropriate content posted by other users or service providers on the blog.
Pre-conditions	<ul style="list-style-type: none"> • The user should be logged in and viewing the post. • Precondition removed as per instructions
Post-conditions	The Users review and rating is updated on the service profile.

	Typical Course of Action
1.	The user clicks on the report button/field on a post.
2.	The system redirects the user to the report page.
3.	The system prompts the user with a form to enter the reason for reporting
4.	The user types in the reason for reporting.
5.	The user clicks on the submit report button.
6.	The system validates the form.
7.	The system updates the database with report
8.	The system forwards the report to the administrator
9.	The system redirects the user after a success message that the report has been submitted
10.	Use Case ends
	Alternate Courses of Action

	Exception Paths

14. Approve Service Provider/Vet/Rescue Service Profile (Admin)

Identifier	UC-015
Purpose	The Admin can Approve or Reject account registration requests
Pre-conditions	<ul style="list-style-type: none"> • The admin should be logged in to their dashboard
Post-conditions	The New account is registered.
Typical Course of Action	
1.	The admin clicks on view registration request

2.	The system displays the registration request to the admin (redirects the admin to the request page).
3.	The admin clicks on approve request
4.	The system displays the message to the admin that the request is approved and redirects the admin to the registration requests page.
5.	The account is registered.
6.	The system sends an email confirmation to the user.
7.	Use Case ends
	Alternate Courses of Action
	<p>Continuing from step 3:</p> <p>The admin clicks on reject request</p> <p>The system displays the message to the admin that the request is rejected and redirects the admin to the registration requests page.</p> <p>The account is not registered</p> <p>Use case end</p>
	Exception Paths

--	--

15. Search on Blog

Identifier	UC-016
Purpose	The user can search for relevant stuff on the blog
Pre-conditions	None
Post-conditions	None
Typical Course of Action	
1.	The user clicks on the search field.
2.	The user types in queries or keywords
3.	The system uses the typed-in query to search and filter the database
4.	The system presents the fetched data to the user

5..	Use Case ends
	Alternate Courses of Action
	Removed as instructed
	Exception Paths
	If the system does not find any matches, it displays a message to the user that no results were found

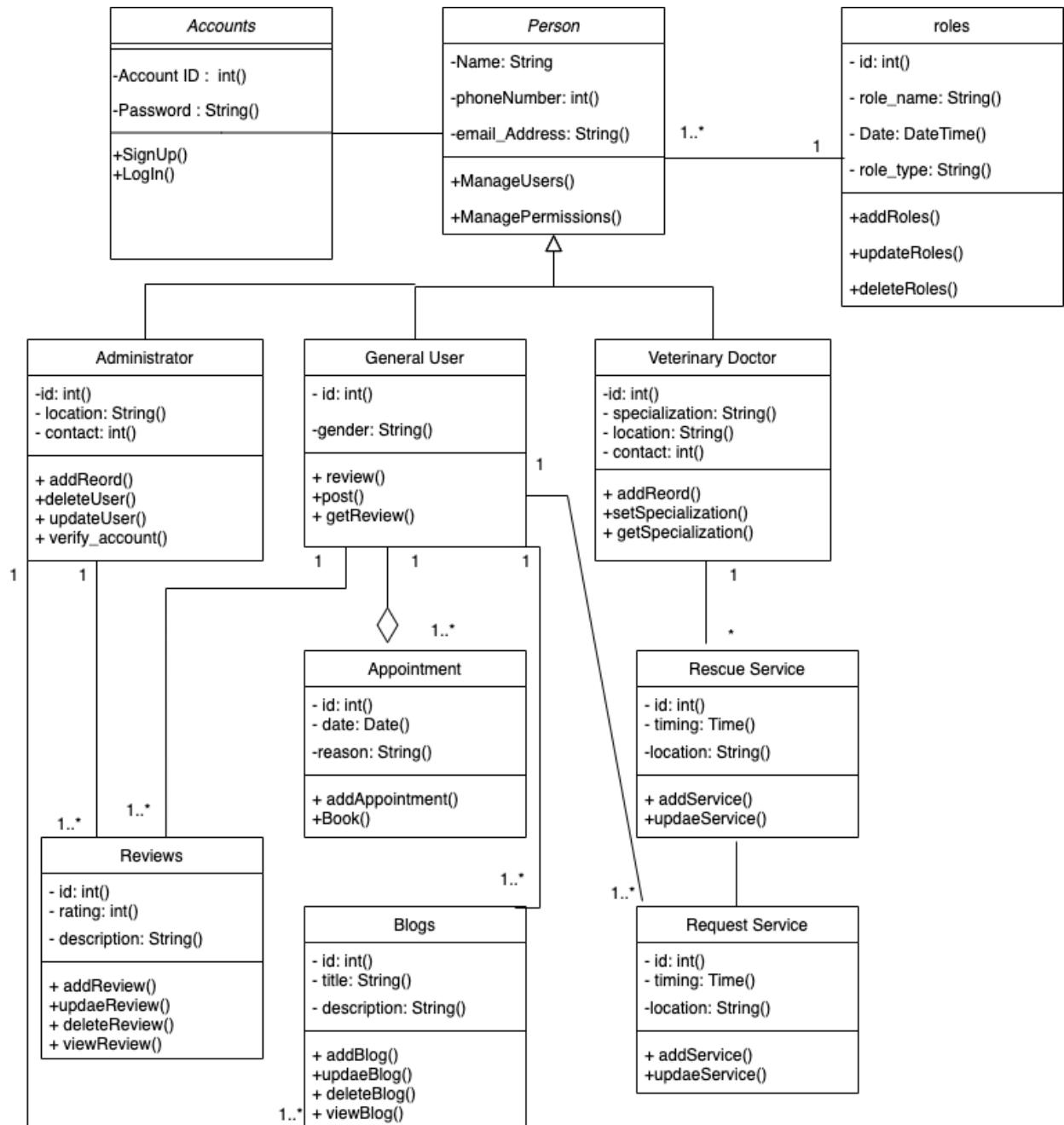
16. Search Marketplace

Identifier	UC-017
Purpose	The user searches for a specific service or vet in the marketplace
Pre-conditions	The user should be logged into the account The user should be in the marketplace
Post-conditions	Search results are filtered according to the user's query and displayed

	Typical Course of Action
1.	The user clicks on the search field
2.	The user types in queries or keywords
3.	The system uses the typed-in query to search and filter the database
4.	The system presents the fetched data to the user
5..	Use case ended
	Alternate Courses of Action
	<p>Continued from step 4:</p> <p>The user has the option to filter based on location</p> <p>The user clicks on the filter by location</p> <p>The system filters the results by location</p> <p>The system displays the results to the user</p> <p>Use case ended</p>

	Exception Paths

b. Class Diagram



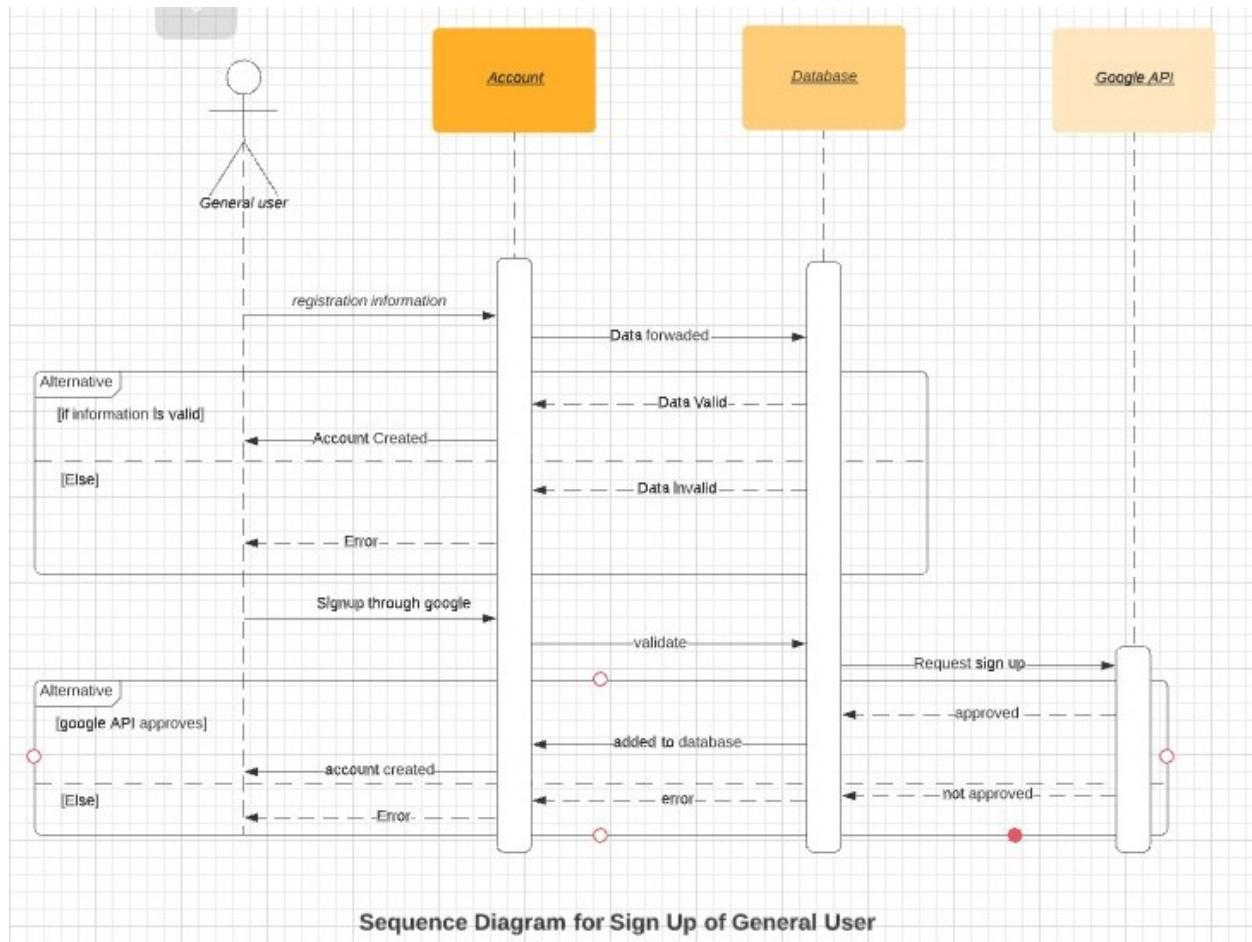
Description:

This Class Diagram contains most of the classes that comprise the system. The Doctor, User, Service Provider, and Admin all will inherit the attributes of their parent class, Person since they have the common attributes and methods that are defined in the Person class. The other

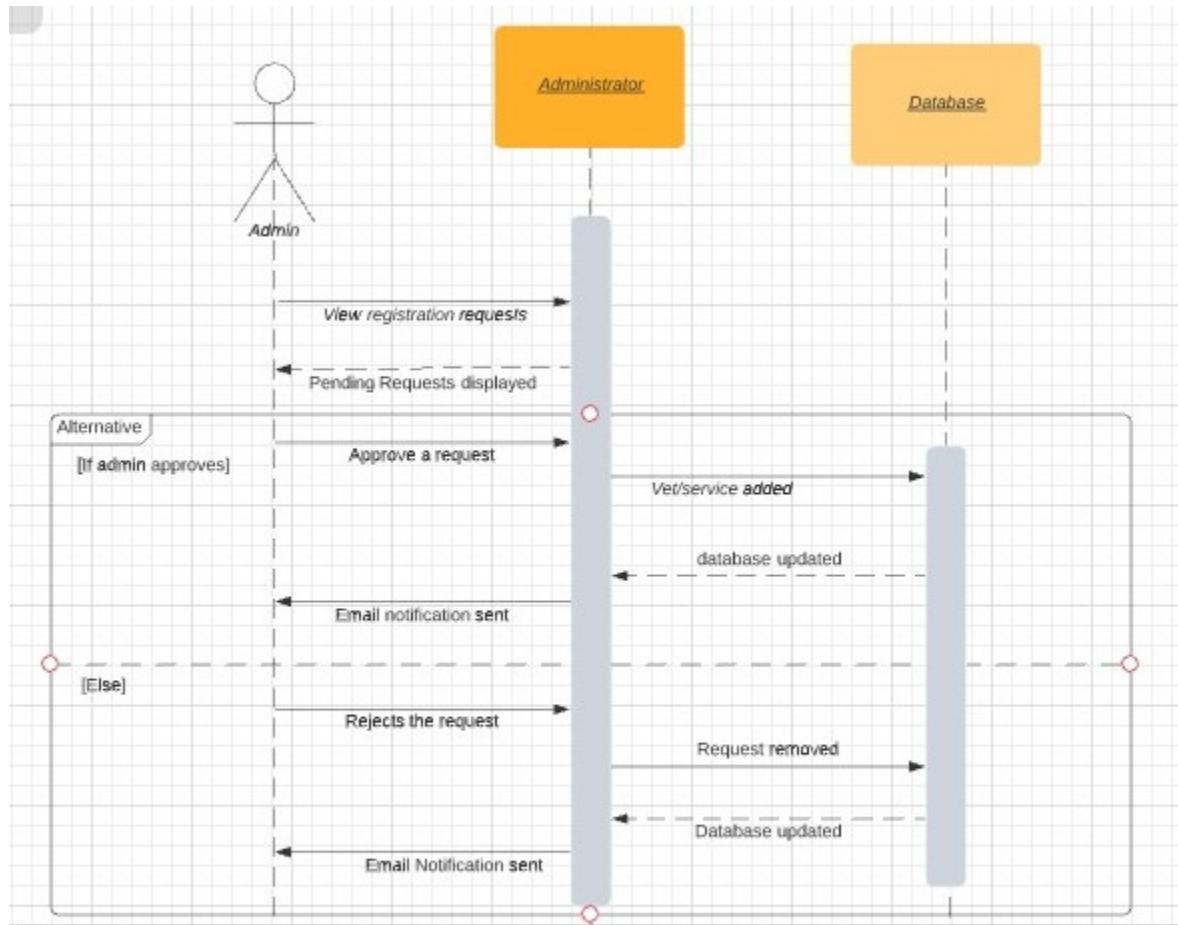
classes like the appointment class will have methods related to the appointment of the doctor and the Pet owner, like the detail of the issue for which they need an appointment. And last but not least the Accounts class will deal with the security and the privacy and have attributes such as account IDs and passwords. This class will also take care of signup/login and service registration approvals. The class diagram is providing a high-level idea and abstraction of how the many different classes interact and function. Moreover, the sequence diagrams in the following section provide details about the flow of data in specific use case scenarios.

c. Sequence Diagrams

c.1. Signup of general user:

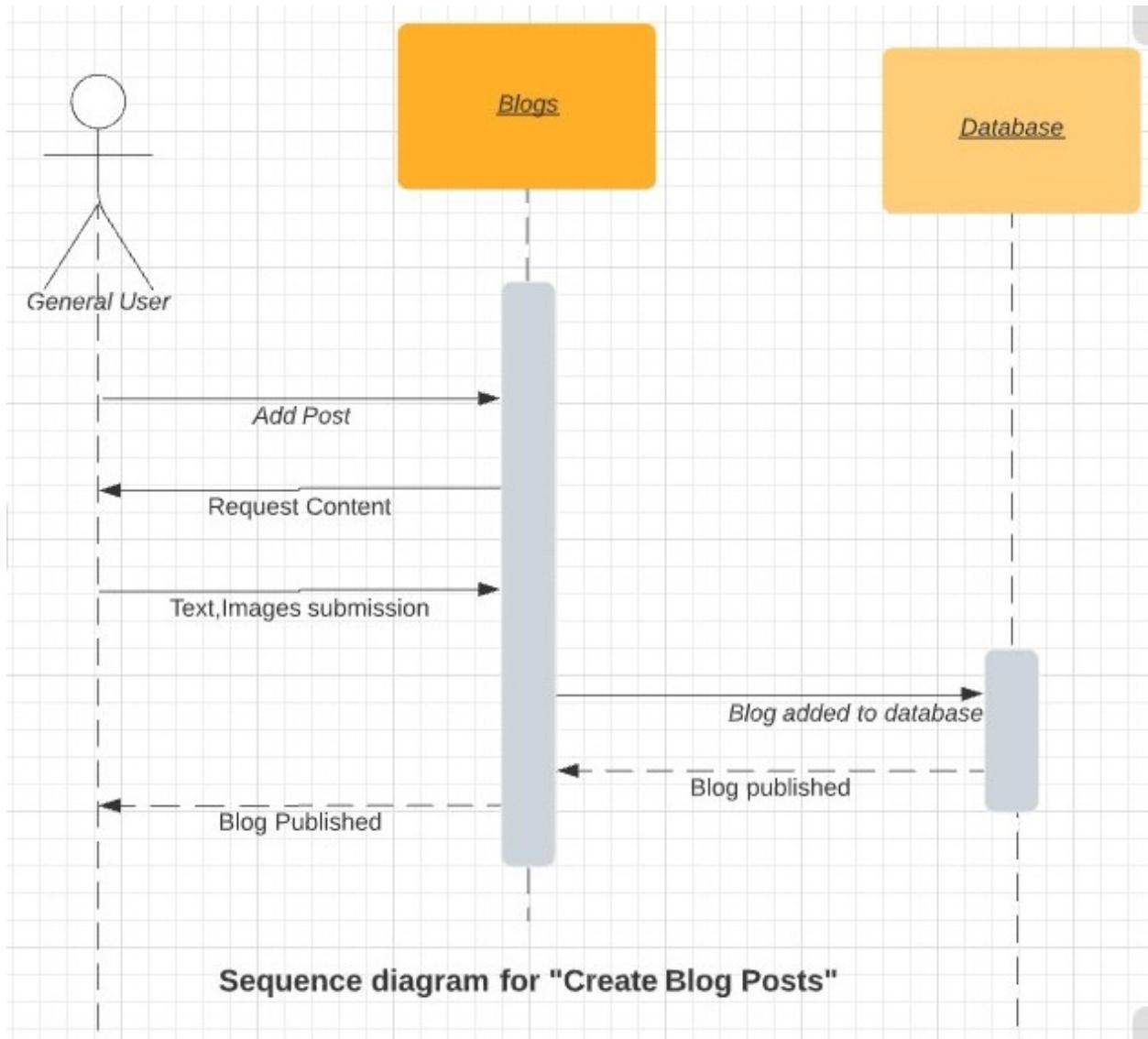


c.2. Approve Service Provider/Vet/Rescue Service Profile (Admin)

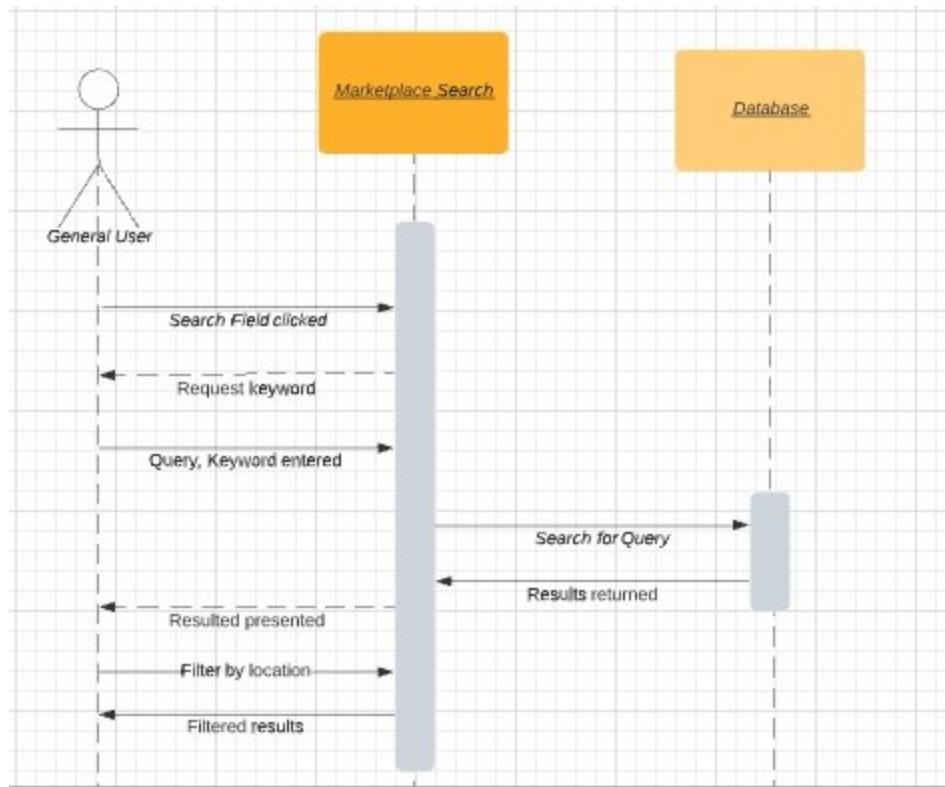


Sequence diagram for "Approve/reject Service Provider/Vet/Rescue Service Profile (Admin)"

c.3. Create Blog Posts

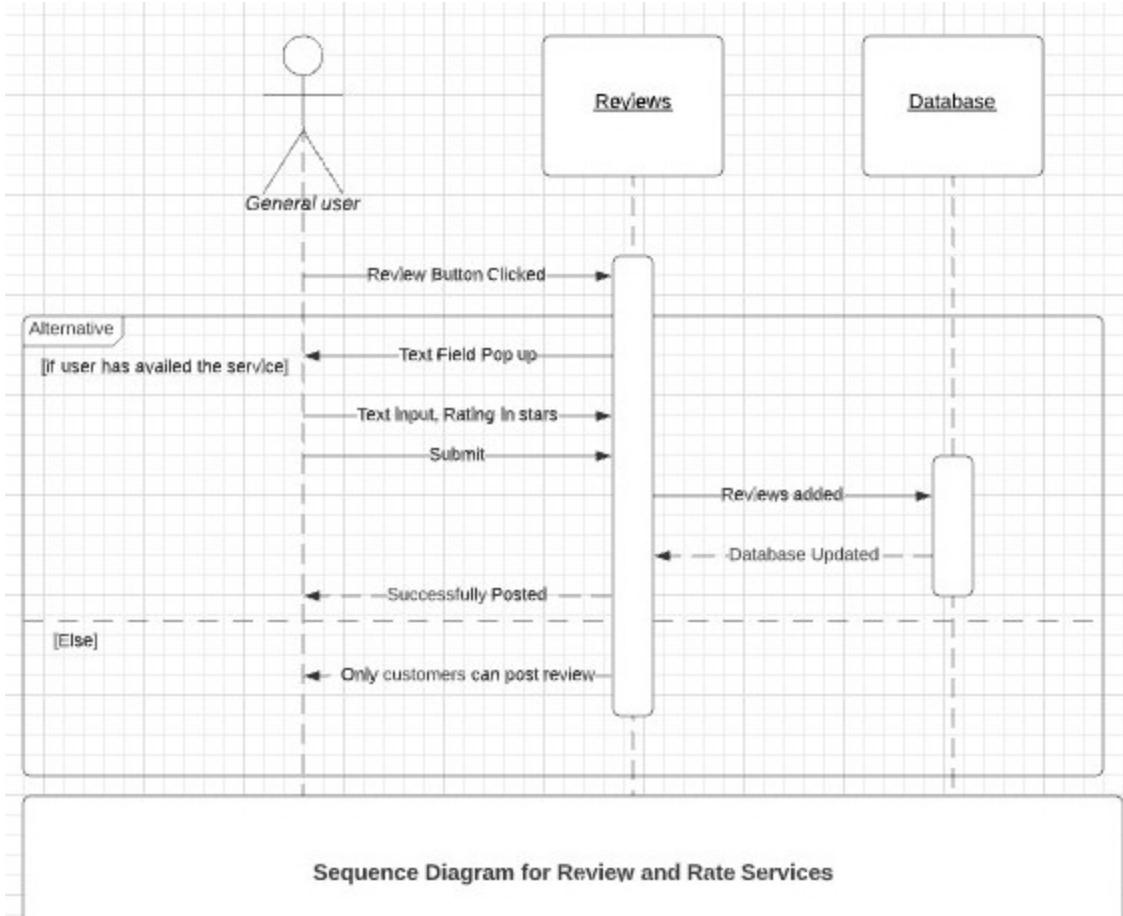


c.4. Search Marketplace

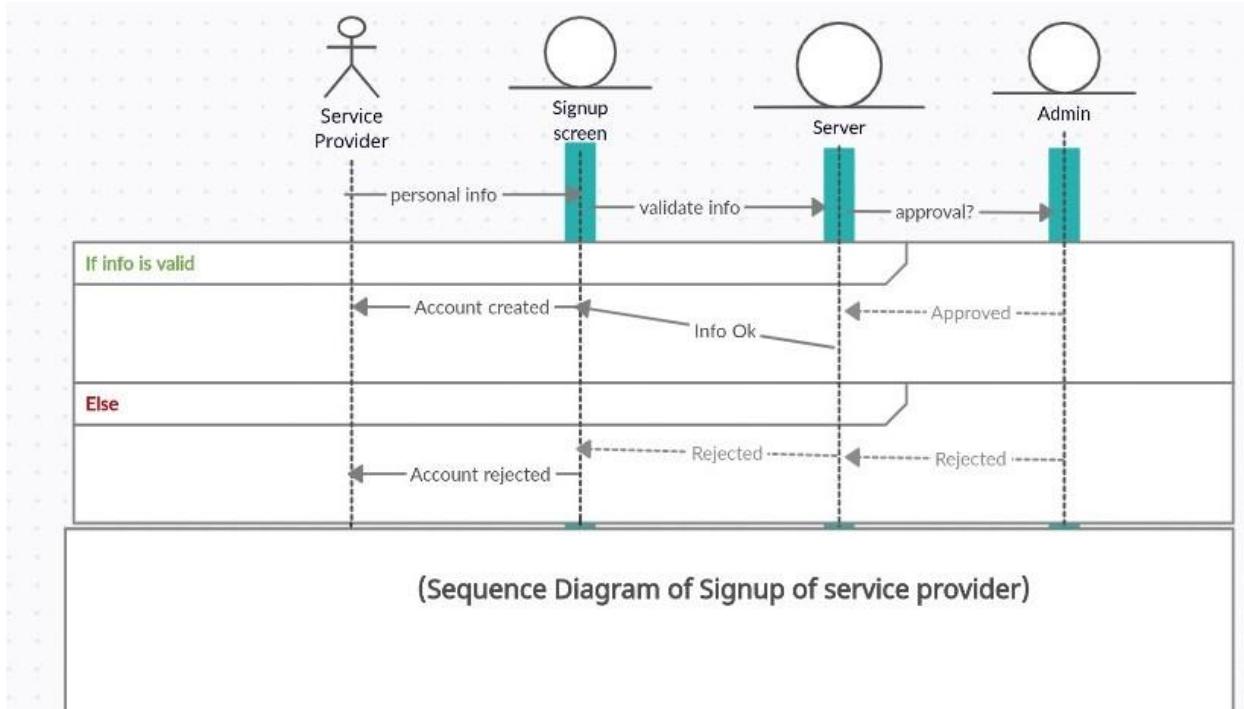


Sequence diagram for "Search Marketplace"

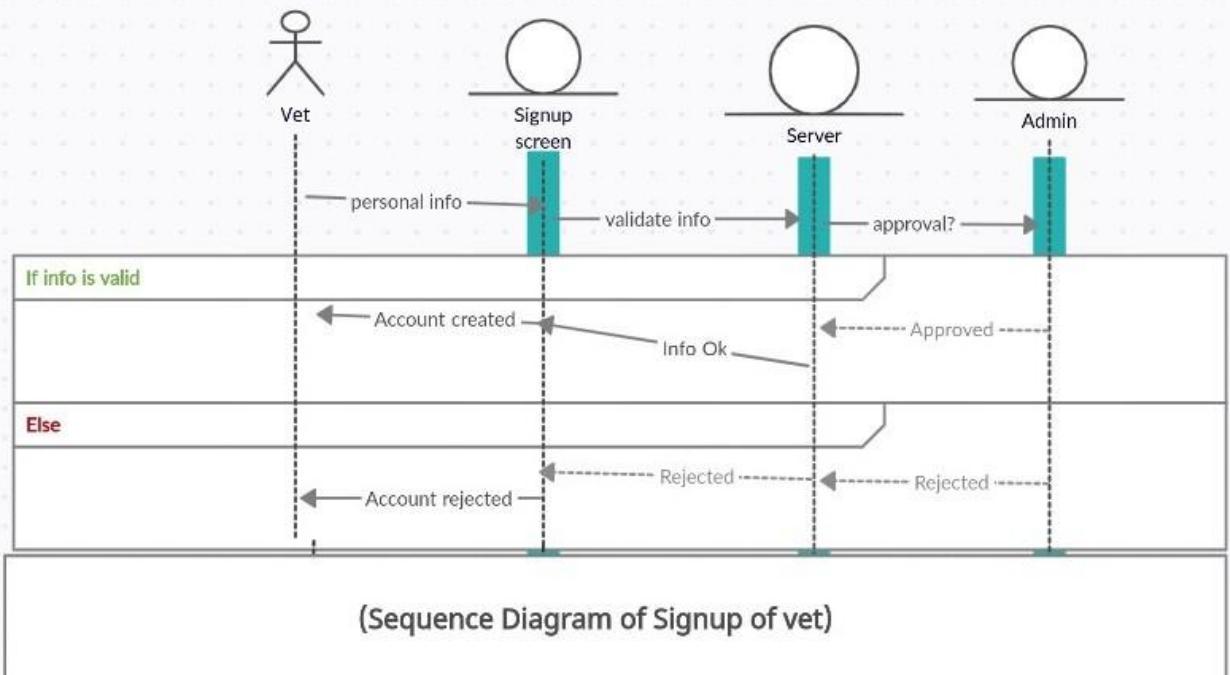
c.5. Review and Rate Services



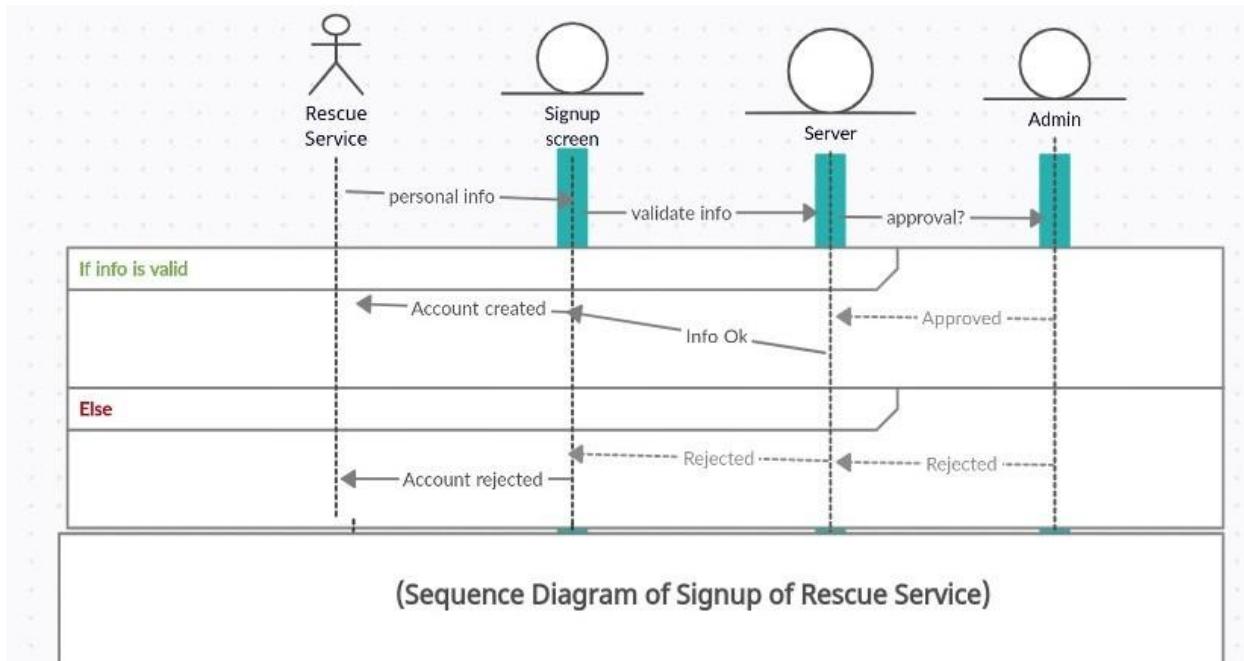
c.6. Signup of Service Provider



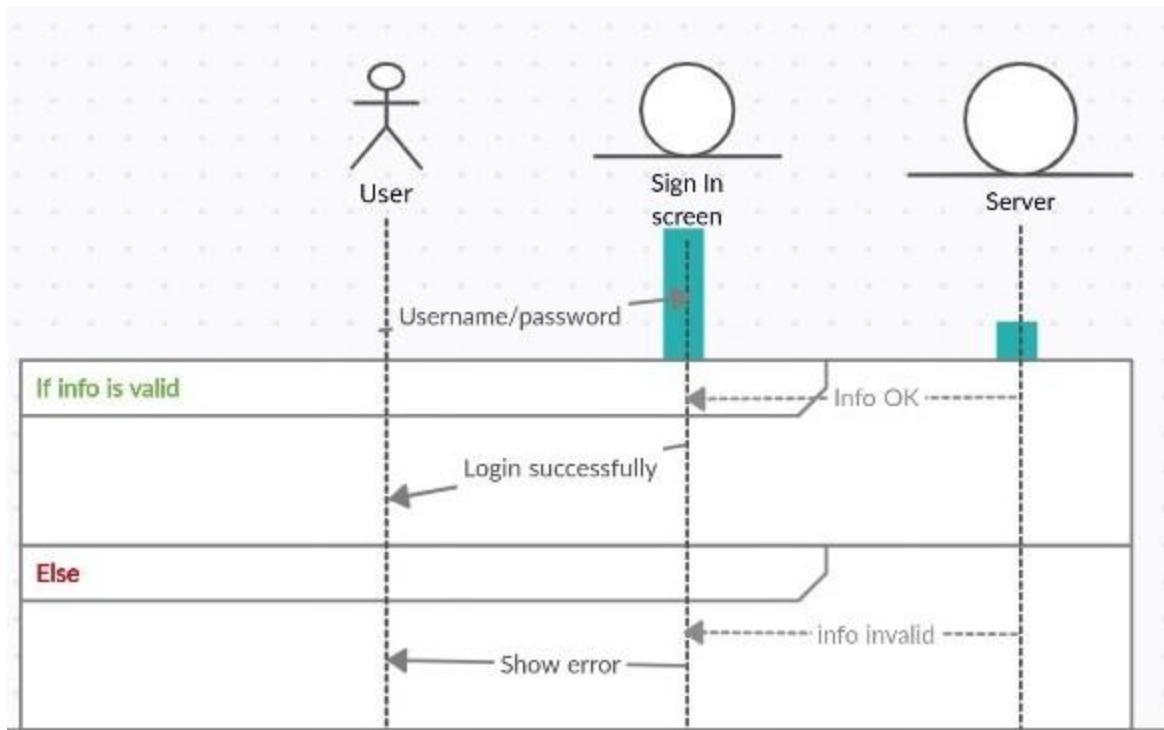
c.7. Signup of Vet



c.8. Signup of Rescue Services

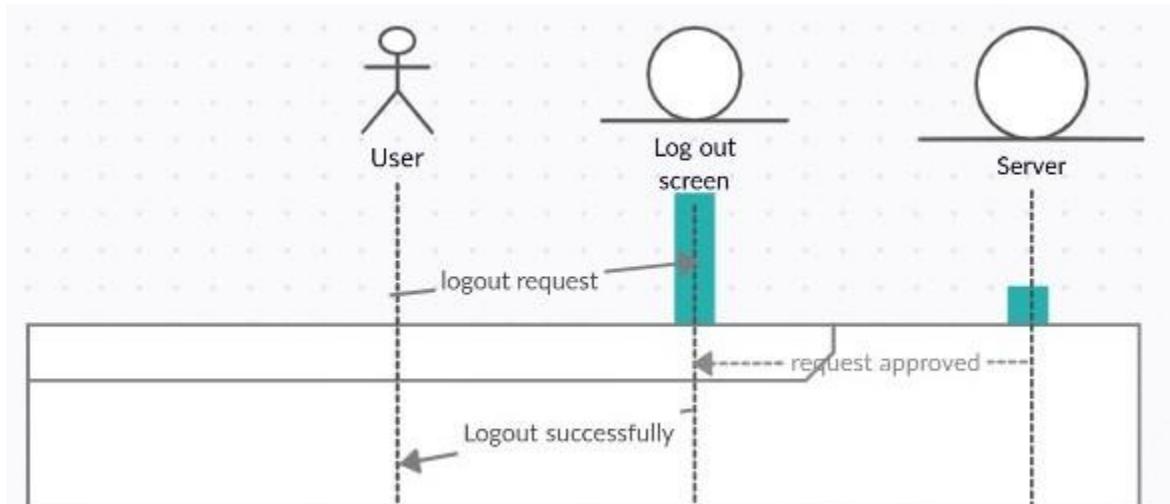


c.9. Login

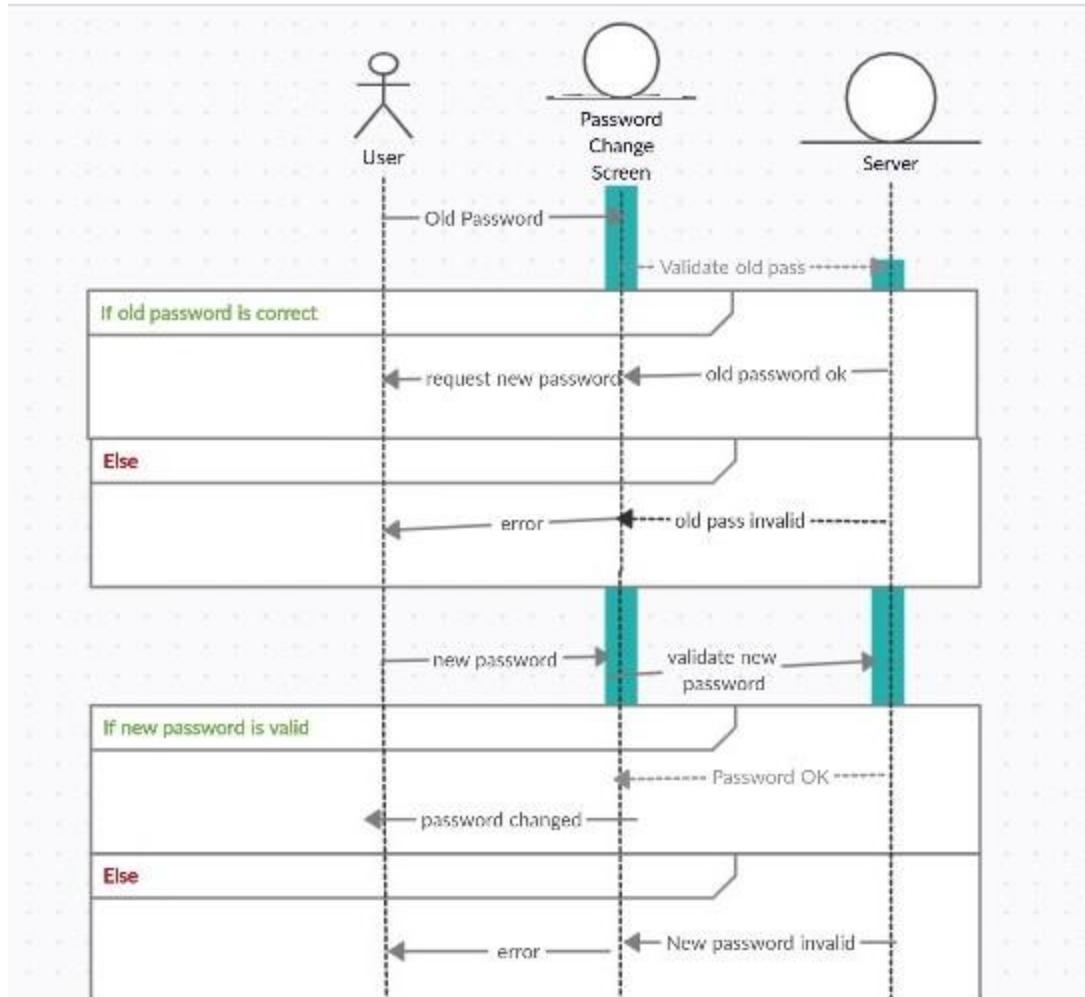


(Sequence Diagram of Login)

c.10. Logout



c.11. Password Change



(Sequence Diagram of Password change)

5. Software Development Methodology and Plan

In this chapter, we will analyze the pros and cons of different developmental models and present the gantt chart.

a. Software Process Selection

Waterfall Model: Pros and Cons

Pros	Cons
Spend more time and effort on the preceding phases to mitigate errors in the succeeding phases	Responding to unstable customer requirements can be hard due to the nature of the development lifecycle in waterfall model
Enforces discipline and time scales are adhered to	More suitable for large systems engineering projects
Since the end result is planned and document, the outcome of the project is explicitly clear	Functional system only results late into the development lifecycle
Financial requirements can accurately be estimated due to predefined outcomes	Longer delivery time
Emphasis on documentation which helps plan better	Client/end user involvement in the project is minimal

Agile(SCRUM) Model: Pros and Cons

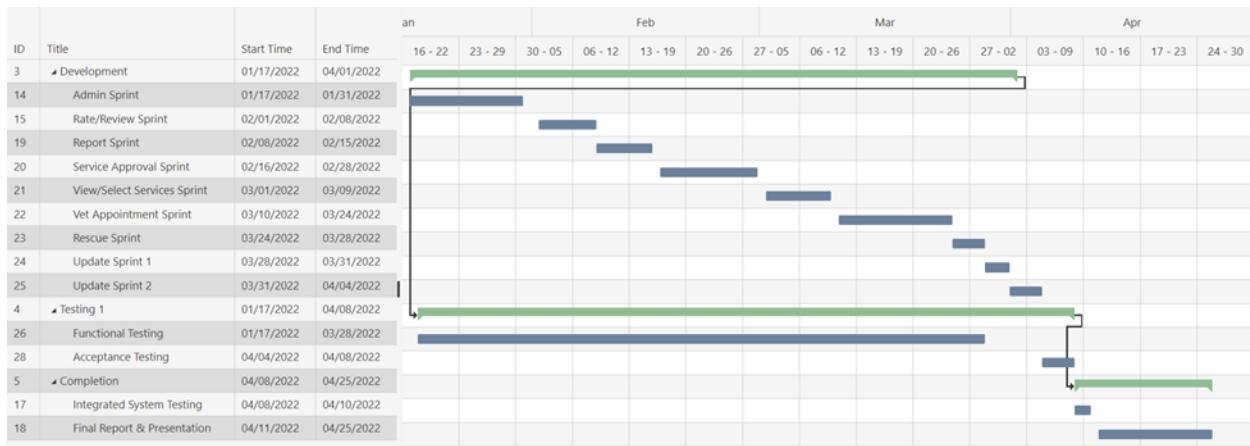
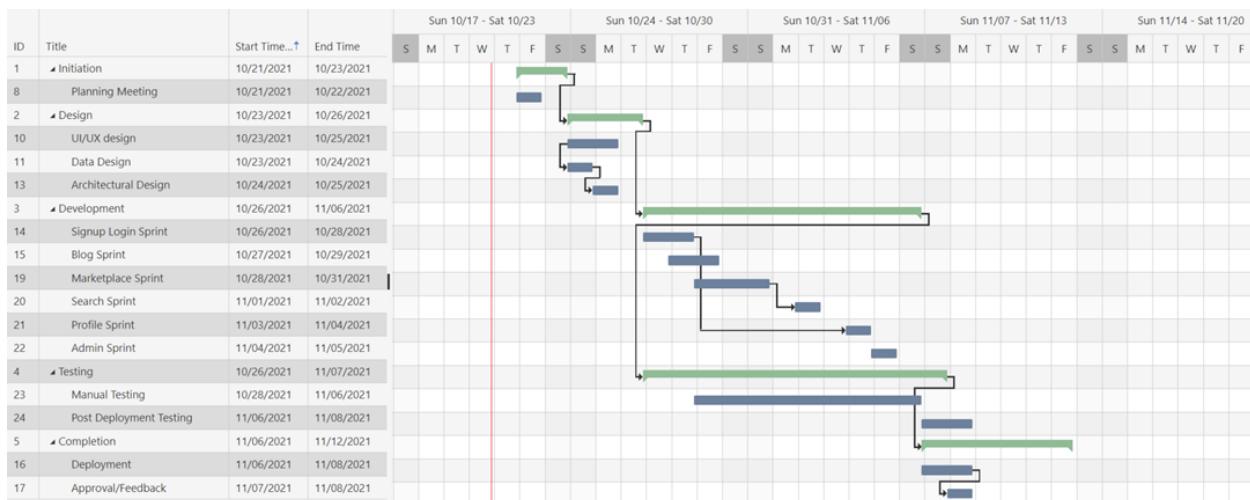
Pros	Cons
Project deliverables can be completed quickly and efficiently.	Prioritizing changes in case of multiple stakeholders can be challenging
Very responsive to changing user requirements.	Scrum master must be keeping meeting productive and everyone in check, and team members must communicate properly
Team members have more freedom with their developmental responsibilities as long as they meet team goals	Usually results in a lack of documentation
Large projects are divided into easily manageable sprints	Adopting SCRUM for larger teams is challenging
Team gets clear visibility through scrum meetings	May result in scope creep due to lack of a definite end-date

We have used an agile scrum model for our developmental process. We have a small team of 4 and we believe that scrum helped us collaborate better and become more productive. As developers, it was easier for us to break down the problem into small manageable chunks that we were able to deliver on time.

Scrum proves very suitable for this as daily scrum meetings will minimize procrastination and ensure the team stays on track. Scrum meetings enabled us to have defined goals for every day and since a progress report was due on the next day, it kept us accountable. The team did not have the same developmental experience, and scrum helped us learn and work together better since there is relatively more freedom with the developmental responsibilities in the scrum.

We believe it is better to work in sprints and see results sooner, instead of trying to develop the entire project in one go. Scrum helped us implement different subsystems and modules as sprints. We could work on adjusting our project requirements and scope as our abilities and skills improved. Scrum helped us develop a system that is receptive to stakeholder feedback. Sprints regularly enforced deadlines on us which urged us to work harder and be productive instead of slacking off.

b. Gantt Chart



- Muhammad Ibrahim Bhalli - Backend + Frontend Development, SCRUM Master, UI-IX
- Muhammad Aaish Javed - Backend + Frontend Development, UI-UX

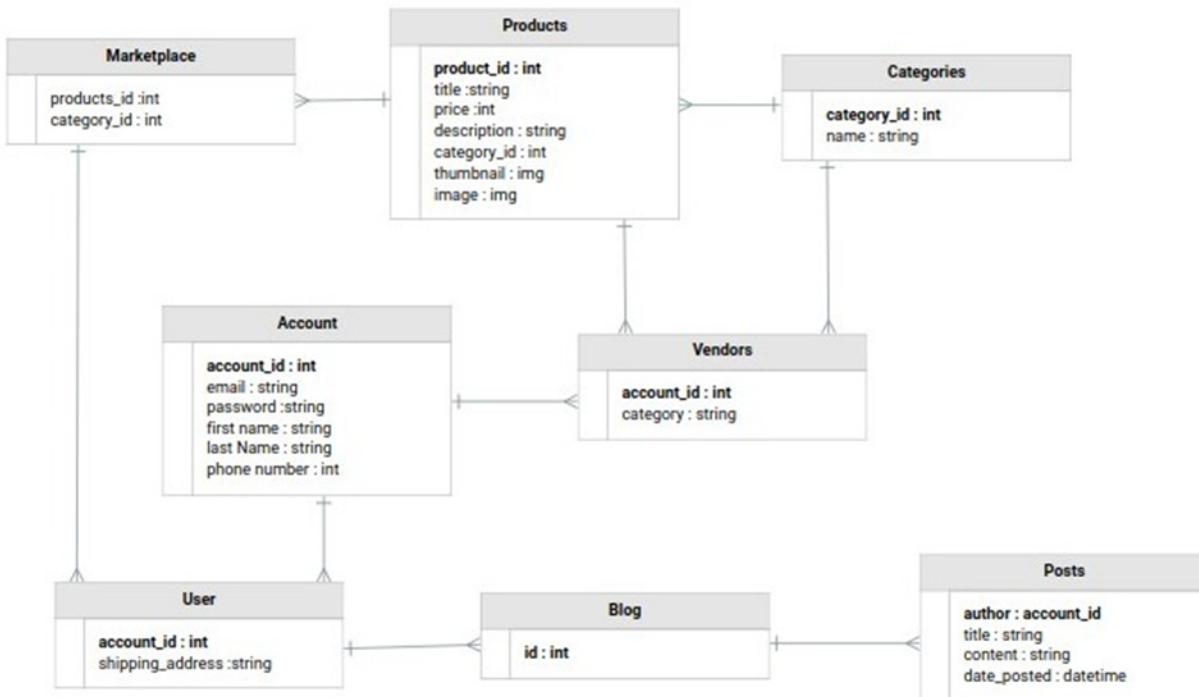
- Muhammad Tayyab
 - Backend + Frontend Development, UI-UX
- Syed Raza Abbas
 - Frontend Development, UI-UX

6. Database Design and Web Services

In this chapter, we have discussed the database design and description of our project.

a. Database Design

E/R Diagram



Brief Description:

- **Products:** This entity has data on all the products that the vendor will publish on the marketplace.
- **Categories:** This entity has all the categories for products that vendors will add.

- **Account:** This entity has data of all the user accounts made on the website.
- **Vendors:** This entity is inherited from the accounts user model and will have the additional data of vendors.
- **User:** This entity keeps data of general users who benefit from services provided on the website.
- **Blog:** This entity has a record of all the posts inherited from the entity named “Posts”.
- **Marketplace:** This entity inherits data from entities “**Products**” and “**Categories**” and offers it to “**User**”.

b. API Specification

We have used the following external APIs:

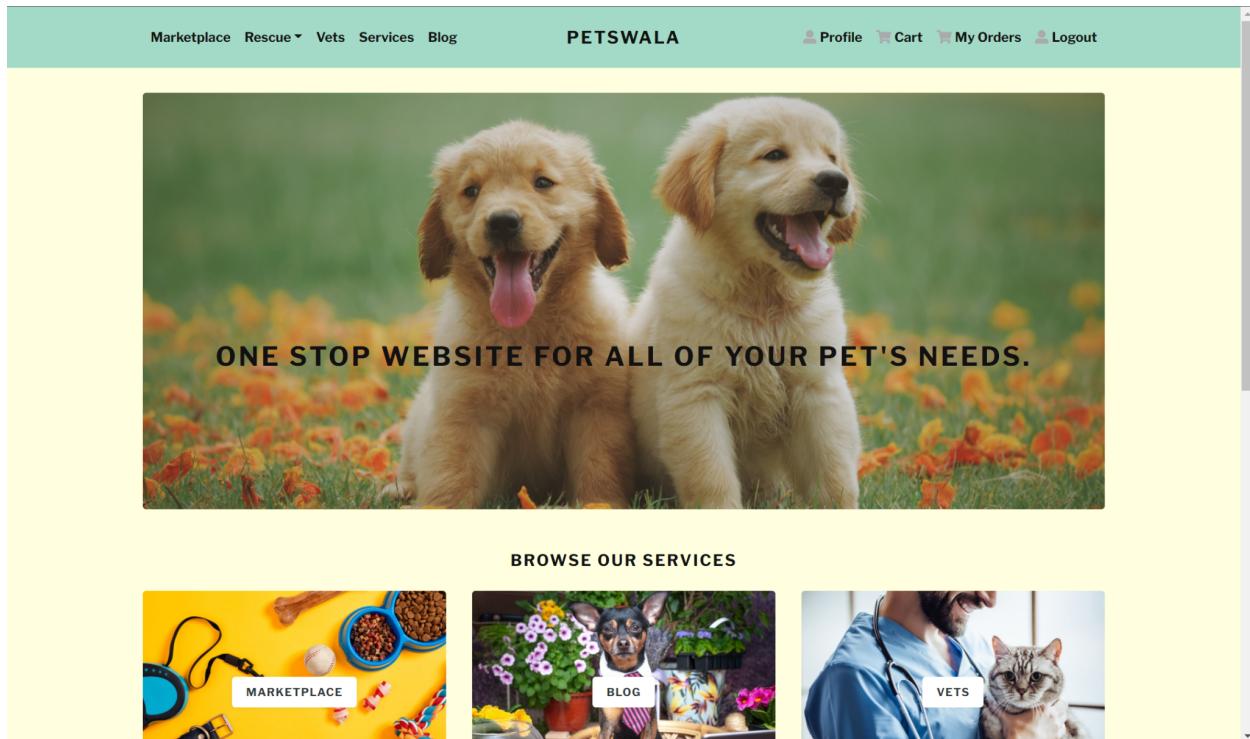
- Google Maps JavaScript API
- Google Places API
- Google Geocoding API

We used these APIs to embed auto-completion functionality in the address field, to get the pin location of users for rescue purposes, etc. We incorporated these APIs into our system using vanilla Javascript.

7. System User Interface

In this chapter we will be going in an in depth analysis of the user's interaction and experience with our project.

Our User Interface focused on ease of access of information as well as the general aesthetics of a modern web application. In this chapter, we display sample user interface screenshots from the deployed website, the explanation and the rationale behind certain design choices.



ABOUT

A SINGLE STOP ONLINE PLATFORM FOR PET OWNERS TO SHOP PET PRODUCTS AND TO ENGAGE WITH VETERINARY SERVICES.

CUSTOMER SERVICES

- Help & Contact Us
- Returns & Refunds
- Online Stores
- Terms & Conditions

COMPANY

- What We Do
- Available Services
- Latest Posts
- FAQs

SOCIAL MEDIA

- Twitter
- Instagram
- Facebook

© 2021 All rights reserved.

We designed a simple and intuitive, easy-to-use user interface specifically with the end-user in mind. There are multiple types of users that will be using our site, therefore we made sure that the unique core functionalities for every user type are easily accessible. The site has a similar home page to the various other eCommerce websites and pet websites. We chose a color scheme of a lighter shade that feels welcoming. On the top of the page, we have the navbar, where all the functionalities that the user may need from the site can be conveniently accessed. Sticking to the widespread login and sign-up conventions, the option to do so appears in the top left corner of the navbar. These nuances make sure that the site feels familiar to the user and that there is no learning curve associated with the site. Furthermore, the navbar has been intuitively designed so the more personalized functionalities are on the right side whereas the site functionality options are on the left. We used javascript to make sure that the buttons and icons are responsive. There are tiles on the home screen as well with icons that serve as an alternate way to explore site functionality. They are more visual as compared to the navbar links. At the end of the page, we have our miscellaneous information and further information.

Marketplace Rescue ▾ Vets Services Blog

PETSWALA

Profile Cart My Orders Logout

Search Search

Search Search

Dog Food
PKR 80.00

PAI PAI Puppy Dog Food
PKR 796.00

PAI PAI Adult Dog Food
PKR 969.00

PAI PAI Cat Food
PKR 699.00

CUSTOMER SERVICES

- [Help & Contact Us](#)
- [Returns & Refunds](#)
- [Online Stores](#)
- [Terms & Conditions](#)

COMPANY

- [What We Do](#)
- [Available Services](#)
- [Latest Posts](#)
- [FAQs](#)

SOCIAL MEDIA

- [Twitter](#)
- [Instagram](#)
- [Facebook](#)

© 2021 All rights reserved.

The marketplace was designed in such a way that the most important information is displayed on the icons and tiles, and the user has the option to choose and learn more about the desired product before buying it. There is a search bar at the top of the products that facilitates the user in

finding whatever they need. This helps it make it easier for the user to navigate and find their desired product without getting annoyed or feeling the hassle. This makes the user experience much more straightforward and rewarding.

The screenshot shows the PETSWALA website interface. At the top, there is a navigation bar with links for Marketplace, Rescue (with a dropdown arrow), Vets, Services, and Blog. To the right of the navigation is the PETSWALA logo. Further to the right are links for Profile, Cart, My Orders, and Logout. Below the navigation bar, the main content area has a yellow header titled "Approved Rescue Services". The content is displayed in three columns, each containing a placeholder profile picture of a person, the provider's name, and two buttons: "Request Rescue" and "View Profile". Each entry also includes a "Description:" section with a truncated text snippet.

Provider Name	Description	Action Buttons
Bilal Asif	I will give first aid to your pet and help you tak...	[Request Rescue] [View Profile]
Daniyal Faheem	I will give first aid to your pet and help you tak...	[Request Rescue] [View Profile]
Rescue Brothers company	I will give first aid to your pet....	[Request Rescue] [View Profile]

The rescue providers page is designed to make the rescue pipeline as streamlined as possible. All the approved rescue providers are displayed to the user with their short descriptions and profile. The user also has the option to request the service directly from here in case of an emergency they can go through the process flow as soon as possible.

The screenshot shows a web page from the PETSWALA website. At the top, there is a navigation bar with links for Marketplace, Rescue (with a dropdown arrow), Vets, Services, and Blog. In the center of the header is the PETSWALA logo. On the right side of the header are links for Profile, Cart (with a shopping bag icon), My Orders, and Logout. Below the header, a large yellow rectangular area contains a white rounded-corner form. The form has a title "Requesting Rescue from Bilal Asif". It includes fields for "Title*" (a text input box), "Description*" (a large text area), "Image" (a section with a "Choose File" button and a message "No file chosen"), and location options ("Enter Address" and "Select Pin Location" buttons). There is also a text input field for "Enter a location" and a "Submit" button at the bottom.

Marketplace Rescue ▾ Vets Services Blog

PETSWALA

Profile Cart My Orders Logout

Requesting Rescue from **Bilal Asif**

Title*

Description*

Image

Choose File No file chosen

Enter Address Select Pin Location

Enter a location

Submit

Upon requesting the rescue service, the user is provided a simple form that asks only the necessary information so that in case of an emergency, no time is wasted. The user makes their request and has the option to attach an image and enter the location of the area where assistance is required.

Marketplace Rescue ▾ Vets Services Blog

PETSWALA

Profile Cart My Orders Logout



Rescue Service
Bilal Asif
Pet Shelter

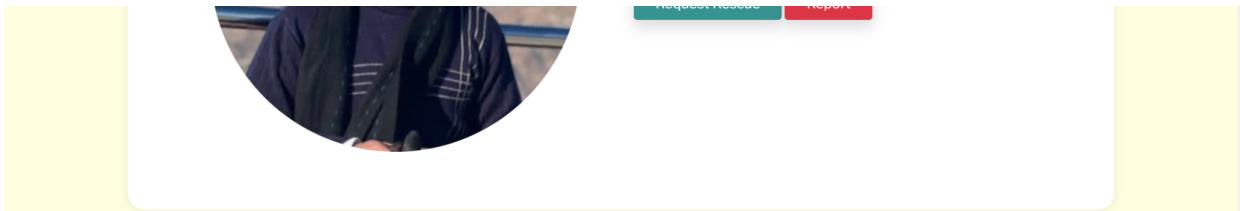
Contact Information

- ✉ putman@outlook.com
- 📞 0332649732

[Request Rescue](#) [Report](#)

REVIEWS

If the user wants to view the rescue provider information, then we have a detailed page that has all the necessary information about the rescue provider. The buttons use the universal color schemes of red and green to make navigation feel as natural as possible. The user has the option to use the contact information to contact the rescue service directly.



REVIEWS



Billu

A Quick and Responsive Rescue Service. Saved My Cat From A Tree!



Posted at: 22 Apr 2022 07:06

WRITE YOUR REVIEW

Your Review

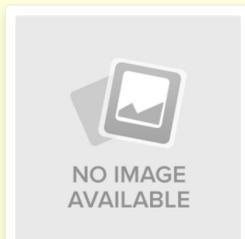
YOUR RATING

★ ★ ★ ★ ★

Post

Furthermore, we have provided the opportunity for user feedback. The user has the option to write a review about the rescue service and rate them too which helps us ensure customer satisfaction.

Requested Rescues

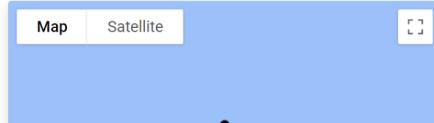
**Cat Emergency**[View details](#)**Status:** TO DO**Note:**

Updated: 1 week, 1 day ago

If the user clicks on the requested rescues, they will have the option to check the status of all their rescue service requests. The tile view shows the basic information of the status of their request, and the last time that they were updated.

Title: Cat Emergency**Description:** My cat fell, please hurry up**Address:** Manawala cantt, FCC9+P2J, Bedian Rd, near sector D, Manawala Sector D Phase 5 D.H.A, Lahore, Punjab, Pakistan**Phone#:** 0332649732**From Rescuer:****Status:** TO DO**Note:**

Updated: 1 week, 1 day ago

[Get Direction To Rescuer's Latest Location](#)**Rescuer's Last Updated Location:**

Upon further details, the user is shown the description of the request alongside extra details that they may have specified. And the google maps integration makes sure that the user can track and get direction to the rescuer's latest location on the same page instead of trying to find the address and navigating themselves.

The screenshot shows a web page titled "Approved Veterinary Doctors". At the top, there is a navigation bar with links for Marketplace, Rescue, Vets, Services, and Blog on the left, the PETSWALA logo in the center, and Profile, Cart, My Orders, and Logout on the right. Below the navigation bar, the main content area has a yellow background. It displays a single veterinary doctor profile card. The card features a placeholder user icon (a gray person silhouette), the name "Waseem Sabir", a green "View Profile" button, and a "Description" section stating "I have 5 years of experience in treating pets....".

As you can see, the theme of the site remains constant throughout different pages and functionalities. The list of all the approved veterinary doctors is displayed to the user, with their profile pictures and experience. This makes sure that the users don't have to dig through extra information to choose the veterinary doctor. If their decision depends on the experience of the vet, then they can easily filter out the one for their needs.

The screenshot shows a profile page for a Veterinary Doctor named Waseem Sabir. At the top, there is a navigation bar with links for Marketplace, Rescue, Vets, Services, and Blog. The central logo is PETSWALA. On the right side of the header are links for Profile, Cart, My Orders, and Logout. The main content area features a placeholder image for the doctor's profile picture. To the right of the image, the title "Veterinary Doctor" and the name "Waseem Sabir" are displayed in bold. Below the name, a short bio states, "I have 5 years of experience in treating pets." A section titled "Contact Information" lists two items: an email address (waseemsabir200@gmail.com) and a phone number (0315156065). A red "Report" button is located at the bottom right of this section.

The detail page is the same as the other detail pages, and this is by design as consistency throughout will help keep the user experience stable.

The screenshot shows a booking form titled "Book Appointment Now!". The form includes fields for "Your Name", "Your Phone", "Your Email", and "Your Address". Below these fields is a dropdown menu listing various time slots and days of the week. The dropdown is currently open, showing options like "9AM to 10AM, Monday" through "1PM to 2PM, Friday". To the right of the dropdown is a green "Book Appointment Now" button. On the left side of the form, there are initials "R" and "W" followed by lists of available times for those initials. At the bottom of the form, there is a section titled "YOUR RATING" with a five-star rating scale and a "Post" button.

Our user has the option to book an appointment with the vet through our app instead of trying to reach out to them on their own. The site makes sure that it is an all in one solution, and therefore the appointment system is integrated into it. The available time slots are displayed to the user, and to avoid clashes, time slots that have already been booked are removed from the selection list. This makes sure that there are no conflicts and the user makes the most of his time.

The same goes for the service providers, as the user has the ability to schedule appointments with them, review and rate them etc.

The screenshot shows the PETSWALA platform's blog section. At the top, there is a navigation bar with links for Marketplace, Rescue, Vets, Services, Blog, Profile, Cart, My Orders, and Logout. Below the navigation bar, there is a search bar with a placeholder 'Search Posts' and a 'Search' button. A teal button labeled 'Add New Post' is also visible. The main content area displays three blog posts in a grid:

- Ahmed Tariq** (Posted on: April 22, 2022)
To care for a cat you will need to:
Be a decent human being. That's the basic requirement to have any pet or live with any other living thing for that matter. Also, congratulations, now another thing depends on you for survival, no pressure.
- Cat Food Vendor** (Posted on: April 22, 2022)
Hey, best cat food vendor is back!
Hello world, I am pleased to inform you that we have restocked on our cat food. Hurry up before it runs out again.
- Root Admin** (Posted on: April 22, 2022)
First Post
A Very Warm Welcome To Our Pet Enthusiasts and Animal Lovers. Happy Trading!

Our site features a blog section where users, vendors, vets, service providers etc can post content. This personalizes the experience for the users as they can use the platform to voice their opinions, spread knowledge, obsess over pets, or just help other people out with their queries. The search bar at the top serves the purpose that if the user is looking for something specific, the searching process should be as efficient as possible. That's why the search bar is at the top of the page. Then the option to add new posts. This is a place where the users can skim through the posts with the short descriptions, and click on them to find the detailed view where they can post comments on the blog posts and have discussions.

Update Info

Username*

Required: 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name*

Last name*

Phone number*

Email address

Address*

Image*

Currently: default.jpg

Change:

 No file chosen

Using the profile page, the user can view and edit the information on their own profile. The page is designed in such a way that the user does not have to dig through the settings to modify personal information and reduces hassle.

Marketplace		Rescue	Vets	Services	Blog	PETSWALA			Profile	Cart	My Orders	Logout
Product		Quantity		Price		Item Total						
	Dog Food	x1		80.00		80.0		<input type="button" value="Remove"/>				
	Status: In Stock											
								Total	80.00			
								<input type="button" value="Continue Shopping"/>	<input type="button" value="Checkout"/>			

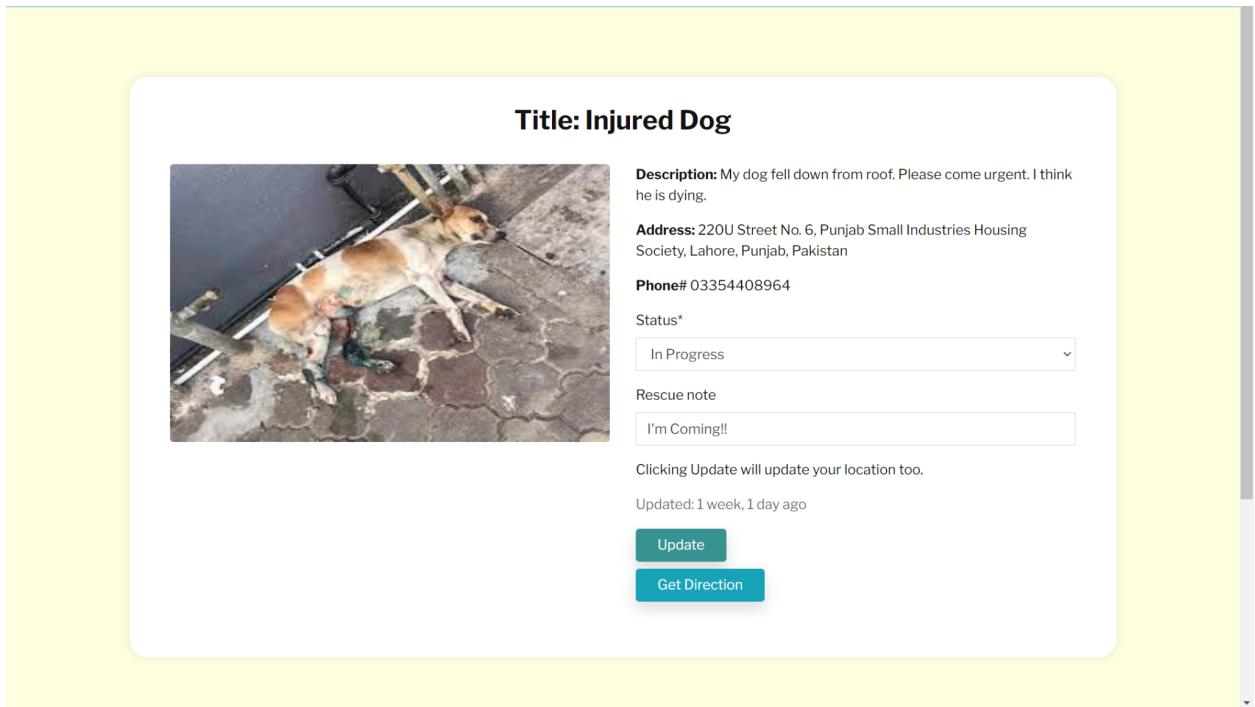
The shopping cart was designed to be simple and functional, with only the essentials being displayed. The buttons are color coded similar to their functionality. It is easy to use and to the point, showing the necessary totals and the options to modify the cart before placing orders.

The screenshot shows the PETSWALA sign-up page. At the top, there is a navigation bar with links for Marketplace, Rescue, Vets, Services, and Blog. The PETSWALA logo is in the center. On the right side of the header are Login and Sign Up buttons. The main content area has a yellow background and features the heading "Register as a". Below this, there is a list of roles: User, Vendor, Vet, Service Provider, and Rescue Service. The "Service Provider" option is highlighted in blue.

The register page is straight forward for the user. Depending on the needs of the end user, they have the option to sign up as their desired role. This simplifies things for the user, as the fields and information that might be necessary for one role are not required for the other one. These different sign up pages make sure that only the necessary information is asked from the user and that they are not annoyed by irrelevant information.

The screenshot shows the PETSWALA vendor add product page. The title is "Add Product". The form includes fields for Title*, Slug, Category*, Description, Price*, Image, and Thumbnail. Each field has a placeholder text. To the right of the form is a background illustration of a person sitting in a chair and another person standing next to a computer monitor, both looking at it. A vertical scroll bar is visible on the right side of the page.

The vendor has the option to easily add products to the marketplace thanks to the personalized add product functionality. It makes editing the database without any knowledge of programming a breeze.



Rescue services have the option to choose from the list of rescue requests. They can respond to the request and update status. It ensures that the communication between the service and the user is as efficient as possible. The rescue service also has the option to get directions to the specified location. We basically try to remove as many hindrances as possible.

Marketplace Rescue ▾ Vets Services Blog **PETSWALA** Profile Appointments Logout

Appointment Requests

Appointments for vet2:

Appointment on Monday, 9AM to 10AM
Appointment on Thursday, 9AM to 10AM

Select Completed Appointments:

Completed

The appointment requests page for the vets is simple and convenient. It lists the appointments for the vet and once they have completed the appointment, they can mark it as completed and simultaneously, the time slot will be updated.

8. Project Security

In this chapter, we have discussed the possible security threats to our project, the losses due to these threats, and the security controls. Then we discussed which tools we have chosen for static and dynamic security scanning.

a. Project Threats

- Broken Access Control:

Users acting out of their intended permissions.

- Cryptographic Failures:

OWASP describes it as a “description of a symptom, not a cause” that leads to exposure of sensitive data.

- Injection:

Unvalidated, unfiltered and unsanitized input of data or code (exploit) by user.

b. Potential Losses

- Broken Access Control:

1. Elevation of Privilege
2. Denial of Access

- Cryptographic Failures:

1. Sensitive Data Exposure
2. Prone to Injection (exploits)

- Injection:

1. Loss of normal flow of control (malicious code)
2. DoS and DDoS attacks

c. Security Controls

- **Broken Access Control:**

1. Implement access control mechanisms and minimize Cross-Origin Resource Sharing usage. (Protective)
2. The cross-origin opener policy (COOP) header allows browsers to isolate a top-level window from other documents by putting them in a different context group so that they cannot directly interact with the top-level window. (Protective)
3. Disable webserver directory listing and ensure file metadata and backup files are not present within web roots. (Protective)
4. Log access control failures and alert admins. (Detective, Responsive, Recovery)
5. Rate limit API and controller access to minimize harm from automated attack tooling. (Detective, Protective)
6. Deny by default, except for public resources. (Protective)

- **Cryptographic Failures:**

1. Make sure to encrypt all sensitive data at rest. (Protective)
2. Store passwords using strong adaptive and salted hashing functions with a work factor. (Protective)
3. Avoid deprecated cryptographic functions and padding schemes, such as MD5, SHA1, etc. (Protective)
4. Disable caching for responses that contain sensitive data. (Protective)
5. Verify independently the effectiveness of configuration and settings. (Detective, Recovery, Responsive)

- **Injection:**

1. Django's query sets are protected from SQL injection since their queries are constructed using query parameterization. (Protective)
2. For any residual dynamic queries, escape special characters using the specific escape syntax for that interpreter. (Protective)
3. Don't use vulnerable functions like extra() and RawSQL in Django. (Protective)
4. Use LIMIT and other SQL controls within queries to prevent mass disclosure of records in case of SQL injection. (Protective)

d. Static and Dynamic Security Scanning Tools

- **Acunetix:**

For Dynamic Security Scanning

- **Raxis:**

For Static Security Scanning

- **SQLMate, <https://linuxsecurity.expert/tools/sqlmate/>:**

To find Injection vulnerabilities.

- **django.contrib.sessions,**

https://docs.djangoproject.com/en/1.8/_modules/django/contrib/sessions/middleware/:

To prevent Broken Access Control.

9. Risk Management

In this chapter, we analyzed potential risks and their mitigation strategies.

Potential Risks and Mitigation Strategies

Sr.	Risk Description	Mitigation Strategy
1.	Risks can take the form of a new cyber security threat, a supplier or service provider that's no longer able to service your company, or an equipment failure.	Identification of vulnerabilities, implementation of firewalls, strict input validations. Investigating for new service providers to be replaced. For Example, we will use AWS for hosting our website, in case they can no longer provide service to us, we will investigate some other service provider.
2.	Breach in user credential security	Make use of secure frameworks that ensure strong encryption and multiple verification techniques and steps.
3.	Risk of losing to competitors	Designing and executing a proper marketing plan, innovating and improving services to customer satisfaction.
4.	Risk of not finishing the project on time	Keeping track of team members' contributions to the project on a daily basis and providing them feedback on how to improve.
5.	Risk of the client not liking the end product and requesting a change in functional requirements	We will be using the scrum method for development which will ensure frequent reviews from the client.

6.	Team members become ill during critical times in the projects.	Reorganize the team so that there is more overlap of work and team members should have knowledge of each other's work.
7.	Security gaps or undiscovered technical bugs in modules that might leave room for vulnerabilities.	Identifying potential risks and replacing buggy components with more reliable and thoroughly tested bought-in components.
8.	Database performance does not keep up with the demand on the system.	Investigate the possibility of investing in a higher performing database.
9.	Drawbacks of an initiative before investing resources, time, or money. For example, choosing services like a hosting platform that may cost more or is unreliable, choosing development software and tools with which the entire team is not comfortable working.	Cost-benefit analysis in light of budget and scope of the initiative and getting the whole team on the same page to make choices about developing software and tools.
10.	Lack of developmental skills	Team members will work together on the developmental process, collaborating and learning together instead of having defined developmental roles.

10. Testing and Evaluation

This chapter discusses the testing strategies and test cases.

Strategy:

Unit Testing

This type of testing entails testing the smallest testable parts of an application. These individual unit tests are crucial for program development and ultimately for integration testing and system testing. This testing usually takes place during the development phase. In the SCRUM approach, this kind of testing is done on a regular basis.

Integration Testing

This type of testing involves testing the interface between two software units or modules. This type of testing takes place after unit testing of multiple software units is successful and their co-functionality needs to be tested. The purpose of this testing is to expose and fix faults between the interaction of two or more units or modules.

Validation Testing

Validation testing focuses on the requirements angle of the developed and integrated modules, ensuring that the program is functioning according to the business requirements specified and agreed upon prior to the development cycle. It basically ensures that the product satisfies the clients' needs.

System Testing

As the name implies, it is the testing of a complete and fully integrated software product. Since this testing focuses on the functionality of the product or application, and not on the way it is implemented, it is also referred to as Black Box Testing and can be performed by a non-technical team.

Our team has made extensive use of all these types of standard testing of program applications. It is important to note that black box testing is not always following completion of the project but also during unit testing and integration testing. Black Box Testing reveals logical errors in the implementation which can be technically fixed after the root cause or error has been identified.

Sample Test Cases

Test Case 1:

Test Case ID	T12	Use Case Name	Edit Blog Post
Test Created by		Test executed by	
Test Case Priority	High/Medium/Low	Test Case Objectives	Check the functionality of edit blog post
Test browser/platform	OS: Windows 10, Browser: Google Chrome etc.		
Pre-conditions	User has signed in. Logged-in user id is U01. User is on blog page.		
Post-conditions	Blog post is edited by the user and updated in the database.		

Step No	User actions	Inputs	System response	Expected Outputs
1	The user chooses to edit a blog post.	Click Edit Blog Post	Website redirects to edit blog post page.	Edit blog post page with form having the content of the previous post
2	The user edits the title of the blog. The user edits the content of the blog.	Title : "Blog Title Edited" Content: "This is the edited content of the blog"	Website is idle and awaiting next step	The entered content is displayed to the user on the form
3	The user is done writing the blog post and confirms.	Click Done	Website updates the database and redirects the user to the blog.	Blog Post edited successfully
Test Case Execution Result		<i>Passed</i>		

	Test Case ID	T13	T14	T15	T16	T17
	Pre-conditions	Logged in user id = U01	Logged in user id = U01	Logged in user id = U01	Logged in user id = U01	Logged in user id = U01
	Post-conditions	Blog post edited successfully	Blog post by different user, cannot edit	Blog title missing in post	Blog content missing in post	Both title and content missing in post
Step No						
1	Inputs	Click Edit Blog Post	Click Edit Blog Post	Click Edit Blog Post	Click Edit Blog Post	Click Edit Blog Post
	Expected Output	Edit blog post page with form having the content of the previous post	Error; User does not have privileges to edit this post	Edit blog post page with form having the content of the previous post	Edit blog post page with form having the content of the previous post	Edit blog post page with form having the content of the previous post
2	Inputs	Title: Welcome again to Petswala Content: This is the edited blog post	...	Title: Content: Edited post without title	Title: Edited post without content Content:	Title: Content:
	Expected Output	Form not yet submitted, system idle	...	Form not yet submitted, system idle	Form not yet submitted, system idle	Form not yet submitted, system idle
3	Inputs	Click Done	...	Click Done	Click Done	Click Done
	Expected Output	Blog post edited successfully	...	Error; Post has no title	Error; Post has no content	Error; Post has no title Error; Post has no content
Test Case Execution Result		Passed	Passed	Passed	Failed	Passed

Test Case 2:

Test Case ID	T18	Use Case Name	Withdraw Cash
Test Created by		Test executed by	
Test Case Priority	High/Medium/Low	Test Case Objectives	Edit Profile
Test browser/platform	OS: Windows 10, Browser: Google Chrome etc.		
Pre-conditions	User is Logged In. Logged-in user id is U01. User is on their profile page.		
Post-conditions	Profile is edited by the User and updated in the database.		

Step No	User actions	Inputs	System response	Expected Outputs
1	The user chooses edit their profile information.	Click Edit Profile	Website redirects to edit profile page.	Edit Profile page with form having the content of the previously saved information
2	The user edits their username The user edits their description The user edits their profile picture	Username : <new_username> Description: "This is the new User Description!" Profile picture: new_picture.jpeg	Website is idle and awaiting next step	The entered content is displayed to the user on the form
3	The user is done editing their profile and confirms.	Click Done	Website updates the database and redirects the user to their profile.	User Profile edited successfully
Test Case Execution Result		Passed		

Test Case Variations (Test cases with different inputs go here)						
	Test Case ID	T19	T20	T21	T22	T23
	Pre-conditions	Logged in user id = U01	Logged in user id = U01	Logged in user id = U01	Logged in user id = U01	Logged in user id = U01
	Post-conditions	Profile Edited successfully	Profile picture missing	Profile username missing	Profile description missing	Both Profile username and description missing
Step No						
1	Inputs	Click Edit Profile	Click Edit Profile	Click Edit Profile	Click Edit Profile	Click Edit Profile
	Expected Output	Edit Profile page with form having previously saved content	Edit Profile page with form having previously saved content	Edit Profile page with form having previously saved content	Edit Profile page with form having previously saved content	Edit Profile page with form having previously saved content
2	Inputs	Username : "New_User" Description: "This is the new User Description!" Profile picture: new_picture.jpeg	Username : "New_User" Description: "This is the new User Description!" Profile picture: new_picture.jpeg	Username : "" Description: "This is the new User Description!" Profile picture: new_picture.jpeg	Username : "New_User" Description: "" Profile picture: new_picture.jpeg	Username : "" Description: "" Profile picture: new_picture.jpeg
	Expected Output	Form not yet submitted, system idle	Form not yet submitted, system idle	Form not yet submitted, system idle	Form not yet submitted, system idle	Form not yet submitted, system idle
3	Inputs	Click Done	Click Done	Click Done	Click Done	Click Done
	Expected Output	Profile edited successfully	Error: Profile picture missing	Error: Profile username missing	Error: Profile description missing	Error: Profile username and description missing
Test Case Execution Result		Passed	Failed	Passed	Passed	Passed

Automation Tools:

- Selenium
- Selenide

11. Deployment Guidelines

In this chapter, we discussed how we deployed our system on AWS.

Project's Code link: <https://github.com/razaabbas12/PO1-Petswala>

Deployment Process:

1. First of all, we bought an ec2 server on AWS.
2. Then, by using the ssh, we connected to the above server.
3. After that, we installed the server dependencies.
4. We set up the configuration file for “Nginx” which can be accessed here:
https://drive.google.com/file/d/1z6fX1yTh2RgPgMXXBnHossSFvA9IkB4S/view?usp=s_haring
5. After setting up the Nginx configuration file, we ran the Nginx reverse proxy.
6. Pulled the latest code from Github for which the link is provided above.
7. After that, we start the Django server with gunicorn.
8. In the end, we set up the certificates with a cert bot.

The project's deployed link:

<https://www.petswala.site/>

For Admin Access:

<https://www.petswala.site/admin>

Username: admin

Password: Petswala@123

12. Conclusion

In this Chapter, we have concluded our report and reported the challenges that we faced and future work that can be done on this project.

a. Summary

We used a well-organized software development plan parallel to a Agile (SCRUM) methodology of implementation in order to develop a fully functional, industry-ready and scalable web project. Our project “Petswala” is designed to provide pet owners, enthusiasts and service providers a single platform to engage in the provision of all sorts of pet-related services and in trade at the marketplace, and as well as interact in the community blog.

We learnt the importance of organizing our sprints and having prior agreement on the specification of use-case development during each sprint of the development process. We learnt the importance of and harnessed the skill of assigning tasks and delegating work in a team. Moreover, we understood how significant the impact of communication amongst a team is, and learned to maintain a healthy team environment which takes into account each individual, their strengths and weaknesses, and their work ethic.

In conclusion, the whole process of software planning, product designing and development has allowed each and every team member to grow at their pace whilst maintaining a healthy and safe work and collaboration environment.

b. Challenges

Mostly during the planning and developing cycle, we functioned as a team in harmony and remained on the same page regarding decisions and workflow relevant to the project. However, there were some challenges we encountered.

- Division of Skill**

The skill and experience level, regarding the web framework in use, was not equal among the team. This led to a rocky start when the development process began. But timely communication, guidance from our supervisors, and the determination to learn eventually landed the team on the common ground.

- Division of Labor**

Initially, there were significant challenges revolving around contribution to the documentation phase and the prototype phase mostly due to the gap in level of skill and experience. However, the team pulled through, consistently putting in

the efforts and hours and by the time the sprint cycle phase initiated, the team was putting in equal effort and contribution.

- **Smaller Team**

A significant disadvantage the team experienced was due to the fewer members (4 members) in the team than there were in other teams (5 members). This led to an increase in workload on the already busier-than-usual schedules of the individual team members. No team member was enrolled in fewer than 21 CH of courses during this final year. **Even more challenging was the inactivity of one of the team members (Syed Raza Abbas) who played quite a passive role in the entire software development process.** But the consistent efforts of the remaining (3 members) eventually led to the completion of the project in time and up to the standard that was promised in the documentation and requirements specification phase.

c. Future

The implementation of the web project (codebase) is very modular, which means that code is well organized and the workflow is documented with comments in the code. This ensures that any and all future work related to the application will be feasible and extendible by this team or any other team that may be interested in further development.

The application currently covers all general base use cases and leaves plenty of room for upscaling and further addition of features. The extensive project documentation also helps ensure that future teams will understand the rationale behind certain choices exercised during implementation such as the Data Model, the ER Diagram, the Class diagram and use-case flows, all of which reflect very well the way the application is implemented.

Further features could potentially include a subdivision of the blog application, where users can create sub-community forums or threads and interact in very specific discussions of common interest. Moreover, there is significant room for improvement in the UI-UX of the application. The web project would potentially be extended to an Android/iOS mobile application which would increase the accessibility of the application according to generational trends in technology and population.

13. Review checklist

Chapter/Section Name	Reviewer Name(s)
Sec 2,3,7,8,9,11	Muhammad Tayyab
Sec 1,3,5,6,8,10,12	Muhammad Aaish Javed
Sec 4,8,10,3	Muhammad Ibrahim Bhalli
Worked on None, Reviewed None	Syed Raza Abbas

14. References

- “Owasp Top Ten.” *OWASP*, <https://owasp.org/www-project-top-ten/>.
- “Google Cloud Services.” *Google Cloud Platform*, Google, <https://console.cloud.google.com/>.
- *Cloud Computing Services - Amazon Web Services (AWS)*. <https://aws.amazon.com/>.
- “Build Software Better, Together.” *GitHub*, <https://github.com/users/razaabbas12/projects/2/views/1>.
- “Sqlmate Review (a Friend of Sqlmap with Additional Features).” *Linux Security Expert*, <https://linuxsecurity.expert/tools/sqlmate/>.
- “Django Contrib Sessions Documentation.” *Django*, https://docs.djangoproject.com/en/1.8/_modules/django/contrib/sessions/middleware/.
- “Free Online Architecture Diagram Maker: EdrawMax Online.” *Edrawsoft*, <https://www.edrawmax.com/architecture-diagram/create-architecture-diagram-online/>.
- “Free Online Diagrams Maker.” *Creately*, 10 July 2019, <https://creately.com/>.
- “Free Online Gantt Chart Software.” *Free Online Gantt Chart Software*, <https://www.onlinegantt.com/>.
- Course Material, for example: Slides, Templates, Documentations etc.

