|  |
| --- |
| Image result for cct college |
| Applied Technology Group Project Pet’s Health Web Application |
| B.S. in Information Technology (NFQ Level 7) |
|  |
| Final Year Project 2018 |
|  |

**CCT College Dublin**

30 - 34 Westmoreland St., Dublin 2 – Ireland

**Supervisor:**  Michael Weiss

**Group Members:**

2015107 - Caroline Santana

2015196 - Danielle Santos

2015325 - Luis Nani Castro

2015187 - Mara Dias

2015254 - Marcionedes de Abreu

2015209 - Michael Faria

**Abstract**

The advent of new technologies such as the internet and other technological tools has changed the world and consequently human behaviour. The intent of this project is to apply these new technologies in order to simplify some tasks for people who own pets. Pets have become part of many human’s life, and inevitably the concern of their health and well-being is important. This project is based on the development of a Web Application to store pets’ data and find the nearest veterinarians and pet sitters. The application aims to be an efficient tool which assists with pet owners many needs. In result, saving time and minimizing concerns.

**Acknowledgments**

We wish to express our sincere gratitude to our supervisor Michael Weiss, to the lecturers Dermot Glanville, Rory O’ Connor, Graham Glanville, Greg South and also Dr. Kyle Goslin who had a great impact in our research project and for their guidance and encouragement for helping us carry out this project.

We also thank all the 2015- 3rd QQI classes members and CCT staff for providing support and suggestions for developing this project, also thank all family and friends for loyalty and love.

Table of Contents

[Introduction 6](#_Toc513395685)

[Proposal 7](#_Toc513395686)

[Project Aim 7](#_Toc513395687)

[Problem Area 7](#_Toc513395688)

[Proposed Solution 8](#_Toc513395689)

[Features and Overview 9](#_Toc513395690)

[Project Planning Schedule 9](#_Toc513395691)

[System Analysis and Design 10](#_Toc513395692)

[Functional Analysis 10](#_Toc513395693)

[Pet’s Health Workflow 13](#_Toc513395694)

[Use Case Diagram 13](#_Toc513395695)

[Sequence Diagram 15](#_Toc513395696)

[Database Structure 18](#_Toc513395697)

[Database Design 18](#_Toc513395698)

[Entity Relation Diagram 18](#_Toc513395699)

[Normalization 18](#_Toc513395700)

[Normalization Rules 19](#_Toc513395701)

[Wireframe Design 20](#_Toc513395702)

[Login page 21](#_Toc513395703)

[Dashboard 21](#_Toc513395704)

[Add Pet 22](#_Toc513395705)

[Medication Page 22](#_Toc513395706)

[Vaccination 23](#_Toc513395707)

[My Pets 23](#_Toc513395708)

[Pet information Page 24](#_Toc513395709)

[Implementation of the System 25](#_Toc513395710)

[Architectural Pattern (MVC) 25](#_Toc513395711)

[Testing Integration 26](#_Toc513395712)

[Methodologies applied 27](#_Toc513395713)

[Agile Methodology 27](#_Toc513395714)

[Type of Agile Methodologies 28](#_Toc513395715)

[Scrum Methodology 28](#_Toc513395716)

[Security Risks 30](#_Toc513395717)

[Project Progression and Further work 32](#_Toc513395718)

[Technology Research and Survey 34](#_Toc513395719)

[GitHub 34](#_Toc513395720)

[Cascading Style Sheets (CSS) 34](#_Toc513395721)

[Web Server 34](#_Toc513395722)

[Bootstrap 35](#_Toc513395723)

[JavaScript (JS) 35](#_Toc513395724)

[XAMMP 36](#_Toc513395725)

[PHP Hypertext Pre-processor (PHP) 37](#_Toc513395727)

[MySQL 38](#_Toc513395728)

[AJAX 38](#_Toc513395729)

[Application Program Interface (API) 40](#_Toc513395730)

[Facebook Login API 40](#_Toc513395731)

[Google Maps API 41](#_Toc513395732)

[Framework 7 41](#_Toc513395733)

[Market Research 41](#_Toc513395734)

[Pet’s Health Survey 42](#_Toc513395735)

[Local Veterinarian Survey 46](#_Toc513395736)

[Appendices 50](#_Toc513395737)

[Appendix A 50](#_Toc513395738)

[Appendix B 51](#_Toc513395739)

[Appendix C 54](#_Toc513395740)

[Reference List 56](#_Toc513395741)

# Introduction

In the present age of information technology, the use of the Internet has became unquestionably popular, it is assumed that most people who have computers use the Internet to exchange messages, access information, purchase products, interact with applications, and so forth.

The project is based on developing a web application focused on a user friendly interface providing simplicity and useful functionalities for those who own pets, allowing users with varying degrees of knowledge in the area of computing technology to use this system.

In order to provide credibility and state facts on the project, surveys were conducted and structured in different forms. The group members interviewed a professional veterinarian and potential users were asked to complete an online questionnaire. Also, a market research was done to ascertain similar applications and test their features.

Wireframes were created for guidance and planning the web application. The application had different versions and the final version used Bootstrap which includes HTML, CSS and Java Script library for the interface, PHP for back end, MySQL for the database, Google Maps API and XAMPP as web server.

The project attempted to apply the best practice for coding and methodologies, the technologies used for building the application are acclaimed and well established on the market for many years and were suitable for the application needs.

The limitations and security risks of the application were listed and taken into consideration. User task analysis such as intuitiveness, fault tolerance, preciseness, and efficiency were performed.

Above all, each group member participated in all phases of the project, which included, research, coding and documentation, improving the development process. This approach was applied in order to make the project understandable in all spheres for each member, resulting in better results and enhanced personal knowledge.

# Proposal

## Project Aim

The main goal of the project is to build a Web Application which can be useful for those who own pets. The intention of the application is to deliver an online tool that pet owners can rely on.  It is practical technology solution to keep their pet’s medical records safe and organised, while promoting the welfare of the animal with suggestions where to find veterinarians and pet sitters available nearby, thereby making life easier for pet owners.

## Problem Area

Pets become part of many Irish families, according to a study published by the Independent (2016). The study was done by Pedigree Ireland and Whiskas Ireland identified that 61 per cent of all Irish households own either a dog or a cat and owners truly cherish their pets with 91 per cent recognising both dogs and cats as members of the family.

According to the interviewed veterinarian *Katy Fortune*, one of the problems Ireland faces in relation to their pets is the process of passing information along from one veterinarian to another. Kate Fortune (transcript of full interview on page 46) says the system that is currently used by the veterinarians is called Teleos (an electronic record management system). Whenever one veterinarian wants to share information with another veterinarian, they export the data from the system and save it as a PDF file and then send it by email (or print and mail it to the new veterinarian).

It is assumed that having a pet can lead to some issues, such as:

* It is hard to keep track of all pet’s information in busy life. For medicines and vaccinations, most people use a journal to keep track of it, but it is not the most secure nor is it the best way to keep this information, because the journal can be lost with all the data;
* The wrong medication or possibly repeated vaccine can be applied to the pet in case the information in the journal is unclear or damaged.
* Pets are exposed to different kinds of danger such as diseases and accidents and depending on the severity of such an event quick action needs to be taken.
* It is often hard to find a reliable or available pet sitter whenever an owner needs to leave their pets alone.

## Proposed Solution

Based on the issues listed above, a Web application would be developed in order to minimize those problems. The application would be based on the following features:

* Consent form will be required to cover any legal implication related to medical records of the pet stored on the database;
* Pet’s personal data - it keeps all the data such as name, weight, breed, date of birth, it also provides a feature where the owner of the pet can upload a picture of the animal;
* Vaccinations - the application has a vaccination tab which keeps track of all information updated, as the pet vaccinations get administrated and how many there are left;
* Medications – similar to the way vaccination works, the Medication tab will keep all the information updated so the user can easily see which medication has been taken and which will be taken;
* Veterinary Clinic - it is possible to quickly find the nearest veterinary clinic through the integrated Google maps API;
* Pet Sitter - The same map technology applied to Veterinary Clinic will be used to find nearby pet sitters willing to take care of pets.

## Features and Overview

The Web Application will be hosted initially on a local host. After it is tested and approved, it will be launched on the cloud and provide the users with the following features:

The ability to:

* Register any pet;
* Insert medication and vaccination the pet has received;
* Register as pet sitter;
* Keep records of veterinarians the pet has visited previously;
* Find nearby veterinary clinic using Google maps API;
* Find nearby pet sitters using Google maps API;
* Add a picture to the pets’ profile.

## Project Planning Schedule

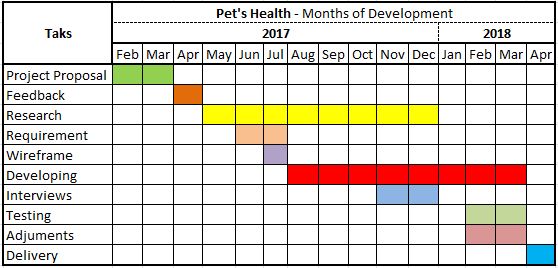


Figure 1 Pet´s Health schedule

# System Analysis and Design

## Functional Analysis

The functional analysis of the Pet´s Health application provides an overview of the program. Pet’s Health allows pets owners to add pets and store medical records in the application, check for nearby veterinary clinics and pet sitters and also register a pet sitter.

The following steps provide a walkthrough of the Pet’s Health Web Application:

Step 1:

The user can access the Application index and select from two options

A - Log in

*Log into the system*

B - Sign up

*Create a new account for a new user. Filling a form with user’s basic information:*

* *First Name*
* *Last Name*
* *Email*
* *Password*

Step 2:

When a user selects from log in or sign up, they will be prompted to the Application home page and select from four options on the dashboard

1. Add Pet

*The user can add one pet at the time into the system. Filling a form with the pets’ basic information:*

* *Pet Type*
* *Name*
* *Breed*
* *Weight*
* *Gender*
* *DOB*

1. My Pets

*The user will see all their registered pets and prompted with three options:*

* Medication

*The user will be able to see medication the pets’ are taking or add new medication to the system filling the form:*

* *Medication*
* *Dose*
* *First Administration*
* *Last Administration*
* *Frequency*
* Vaccination

*The user will be able to see vaccinations the pets’ have taken or add new vaccinations to the system filling the form:*

* *Vaccination Type*
* *Issue Date*
* *Vet Name*
* Extra Notes

*The user can add extra notes for their pets*

1. Vets

*User can check for nearest veterinary clinics using Google Maps API*

1. Pet Sitters

*User can check for pet sitters using Google Maps API*

* Register as Pet Sitter

*User can register as pet sitter filling the form:*

* *First Name*
* *Last Name*
* *Phone Number*
* *Email*
* *Address*
* *Specialization*

1. Log out

*User will log out the system and go back to index page*

## Pet’s Health Workflow

Pet’s Health workflow is a graphical representation of the Web Application activity and execution of the program. It provides extended capabilities in a simplified way.

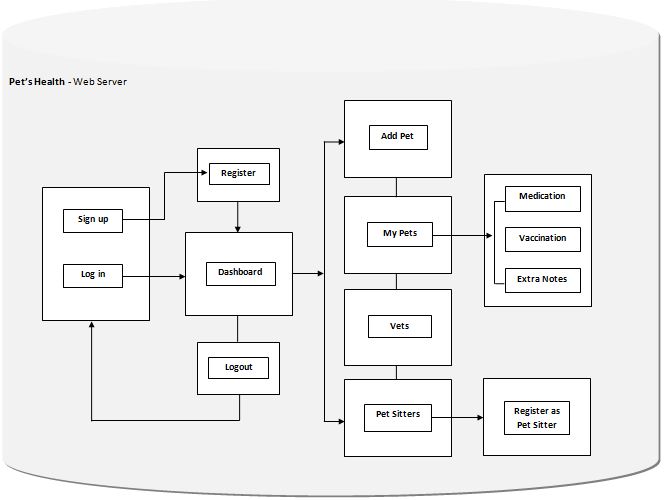


Figure 2 Pet's Health workflow

## Use Case Diagram

Use cases model presents the user’s view of the functionality of a system, in this case the Pet’s Health Application. Use cases are usually presented in a graphical form, the use case diagram, supported by textual descriptions, use case actor, descriptions and scenarios (Briton & Doake, 2004).

Successful Use Case

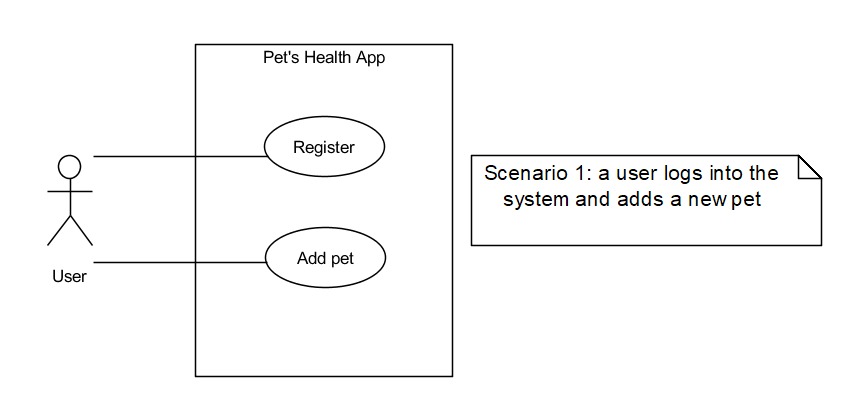


Figure 3 Pet’s Health use case diagram successful scenario

Not Successful User Case

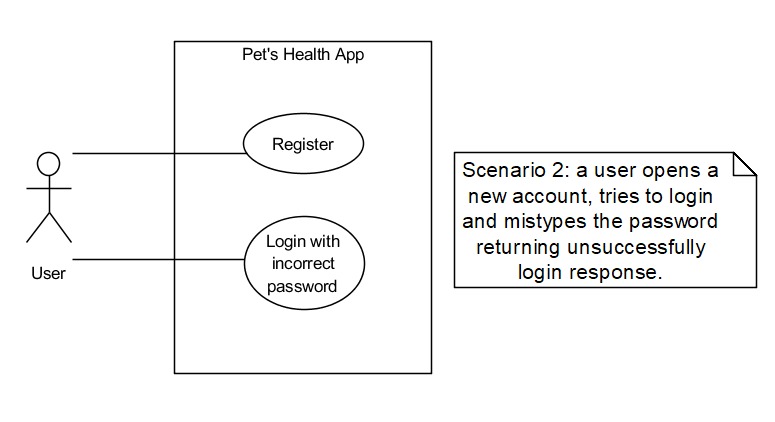


Figure 4 Pet's Health use case diagram unsuccessful scenario

## Sequence Diagram

This Sequence diagram shows the flow of control between the objects required to execute a scenario. A scenario outlines the order of steps in one occasion of a use case, from the user’s side. The Sequence diagram shows how those steps happen on the computer’s side.

For each the following Sequence diagrams displayed, the user must attempt to login into the system before using a feature.

Add Pet

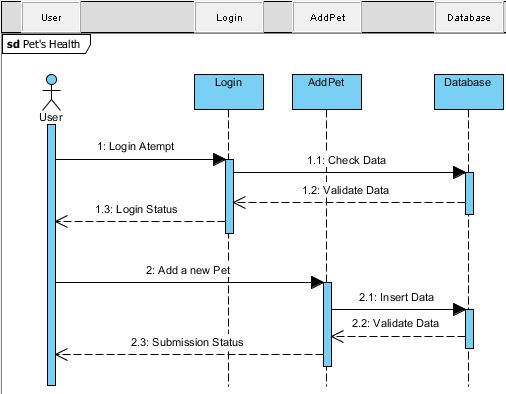


Figure 5 Pet's Health sequence diagram where the user adds a pet into the database

My Pets

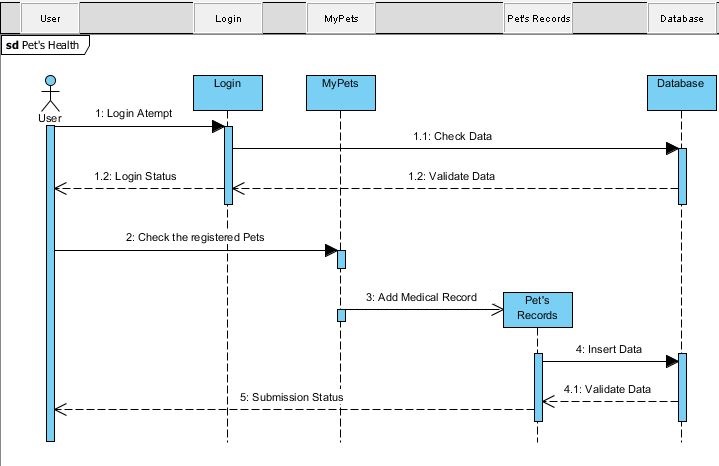


Figure 6 Pet's Health sequence diagram to check added pet

Veterinary Clinics

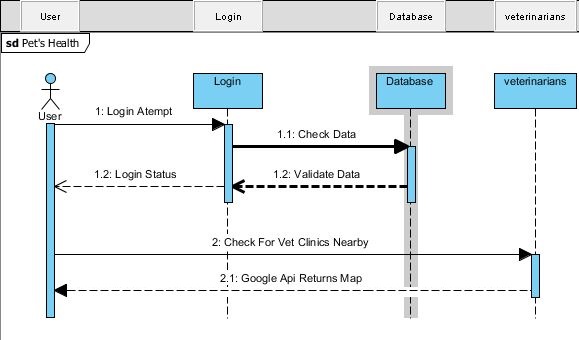


Figure 7 Pet's Health sequence diagram to check veterinary clinics

Pet Sitters

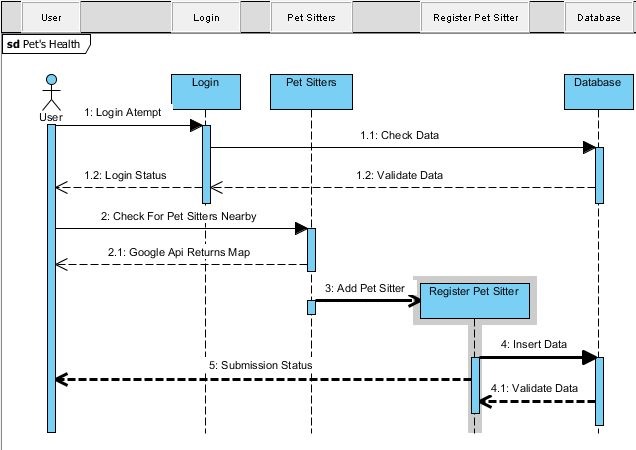


Figure 8 Pet's Health sequence diagram to check pet sitters

**Database Structure**

## Database Design

A database as a collection of data, schemas, tables, queries and reports (Elmasri & Navathe, 2016). To design Pet’s Health’s database was a crucial component to implement the ideal elements and model according to the needs of the system proposed, which is to store the web application data.

## Entity Relation Diagram

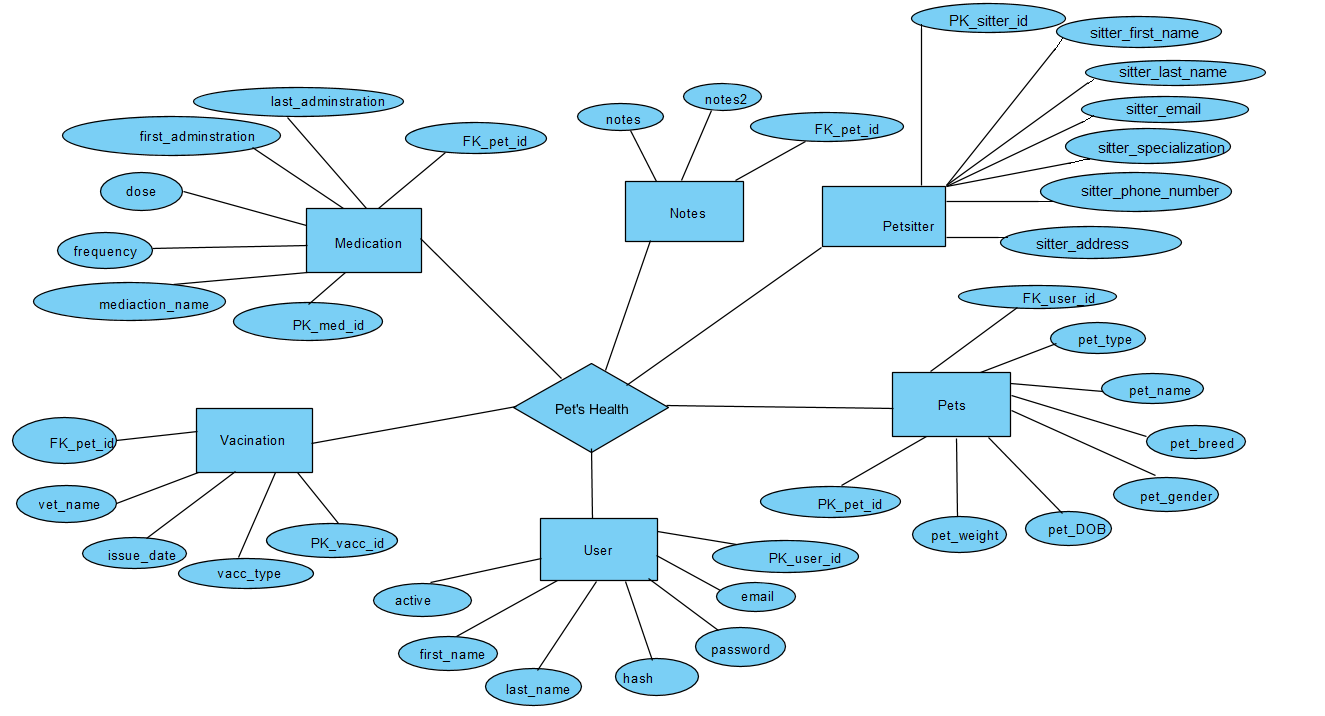


Figure 9 Entity relation diagram

## Normalization

Normalization is the process of organizing data in the database to avoid redundancy and ensure the data is logically stored and independent, but related.

Data redundancy is the presence of unnecessary data repetition which makes difficult to retrieve information and increases the size of the database and leads to more issues.

Databases which do not have normalization are more likely to lose data and provide anomalies in the process of insertion, update and deletion.

*Insertion anomaly* occurs when data cannot be inserted all at once, because this data may rely on later information, so this field must be set as NULL.

*Updating anomaly* occurs when all field needs to be updated, and for a mistake some data it is lost, leading to data inconsistency.

*Deletion anomaly* occurs when a field is related to another, and also in the process of deletion, if one of the data item is deleted, the information will be lost from the other field (i.e. if the Primary key is deleted from the “user” table the foreign key will be deleted on the “pets” table).

## Normalization Rules

First Normal Form (1NF)

The process of designing the database in such a way that is easily extended and easy to retrieve the data when requested. It should follow some rules:

* Single valued attributes, should not contain multiple values;
* Attribute domain should not change, values should be the same type;
* Unique name for attributes and columns;
* Order does not matter.

Second Normal Form (2NF)

For a table to in Second Normal Form it should follow only two rules:

* The table must be in 1NF;
* It should not have *partial dependency*, when an attribute depends on only a part of the primary key and not the whole key.

Third Normal Form (3NF)

The Third Normal Form has only two rules:

* The table should be in 2NF;
* It should not have *transitive dependency*, when a non-prime attribute depends on other non-prime attributes instead of depending upon the prime attributer or primary key.

For the project needs, the Pet’s Health database provides the Third Normal Form (3NF) normalisation.

# Wireframe Design

The Pet’s Health wireframes were created for the purpose of arranging information and functionalities. The structures and architecture of the web application were used as a guide for developing the application and identifying priorities.

The layout was designed to have an intuitive user experience. The backend was developed in PHP and the frontend was developed in JavaScript and Bootstrap.

The wireframes below show the Web Application overall:

## 

## Login page

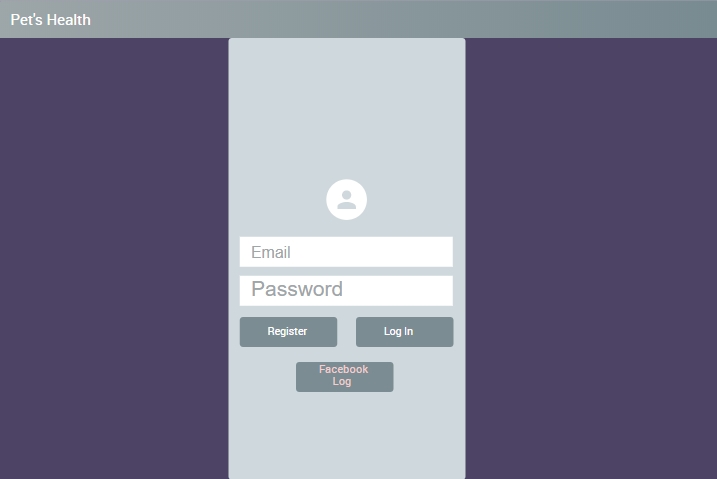


Figure 10 Login page

## Dashboard

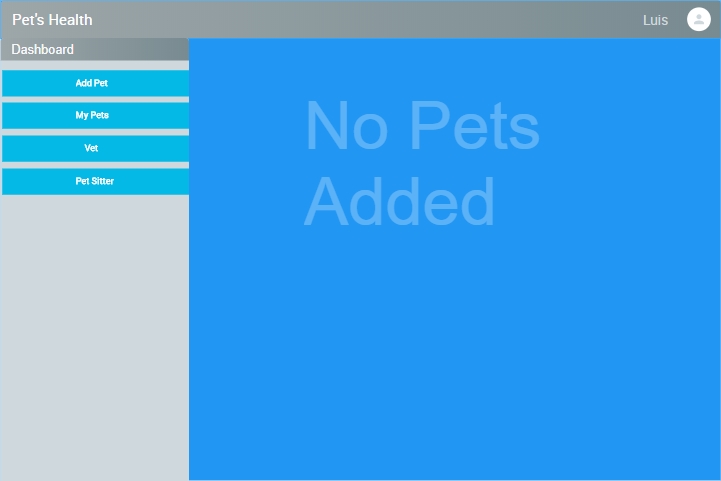


Figure 11 Dashboard

## Add Pet

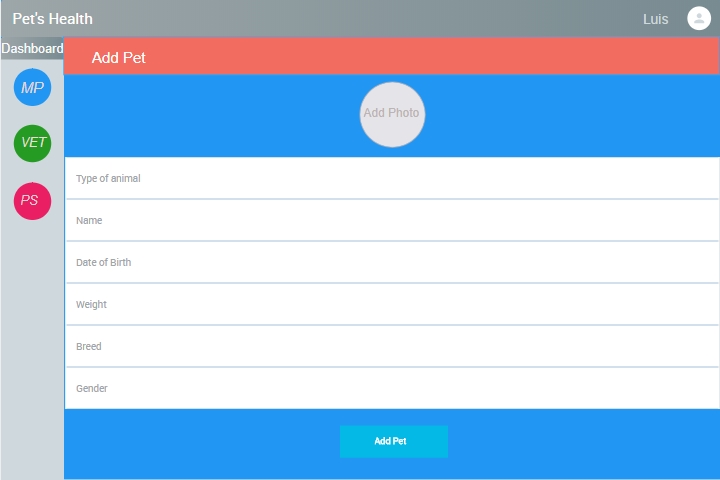


Figure 12 Add Pet

## Medication Page

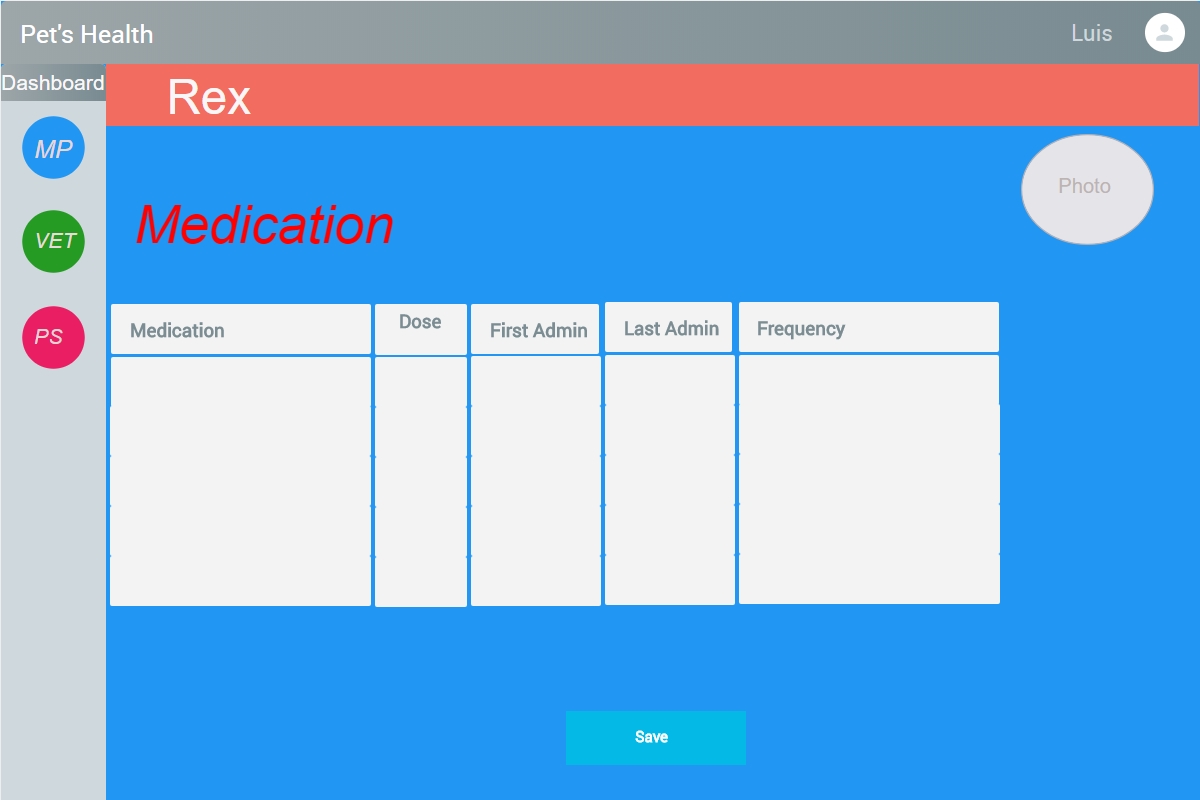


Figure 13 Medication page

## Vaccination

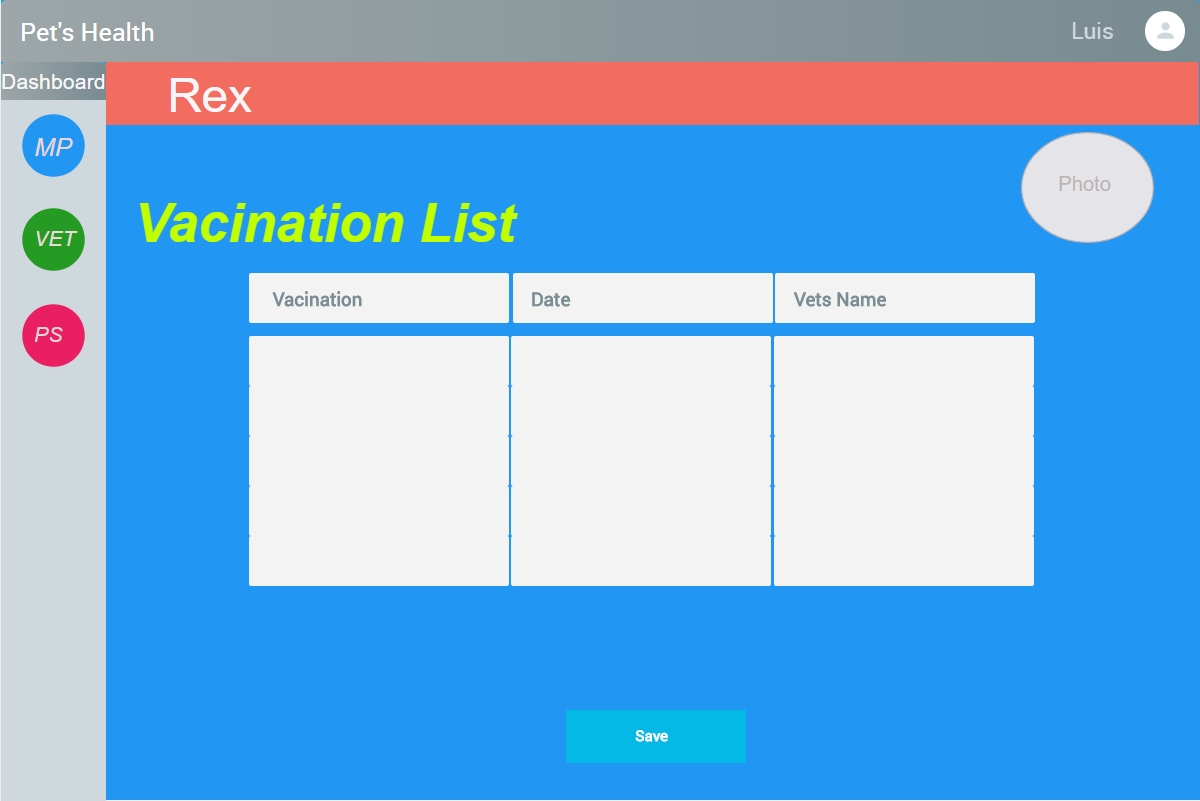


Figure 14 Vaccination Page

## My Pets

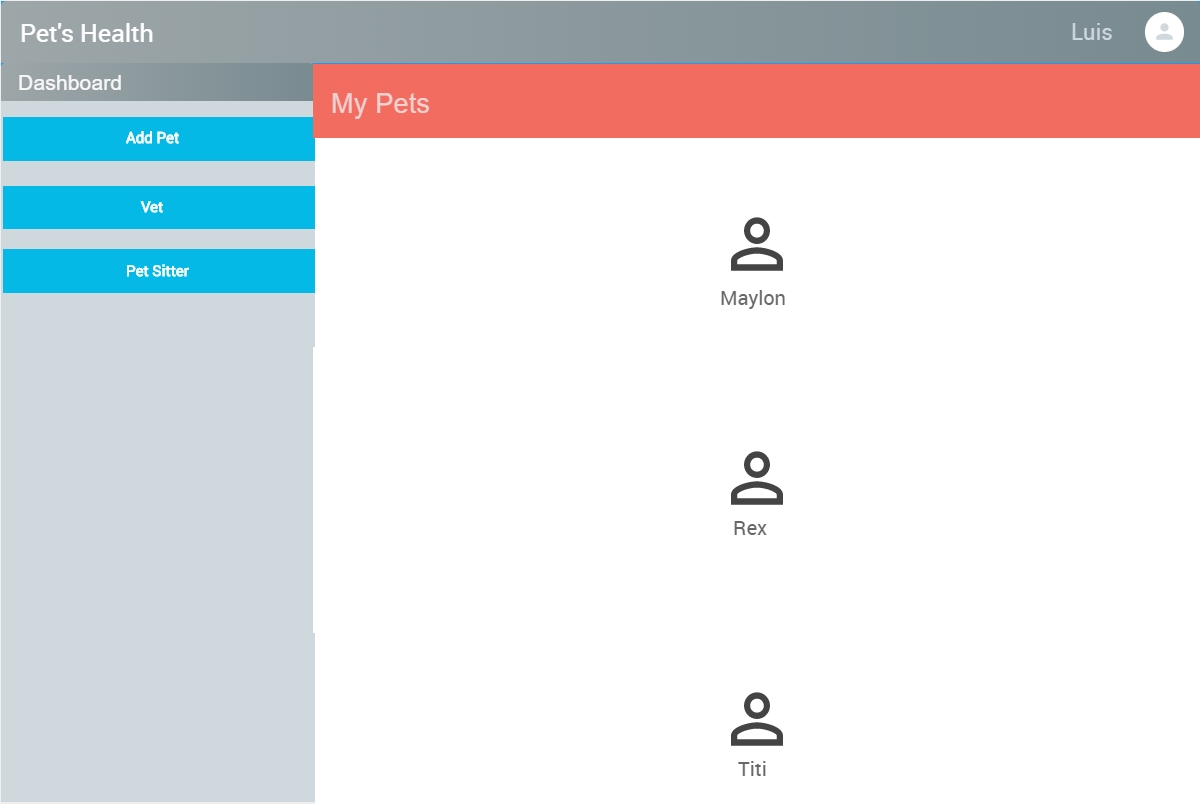


Figure 15 My Pets

## Pet information Page

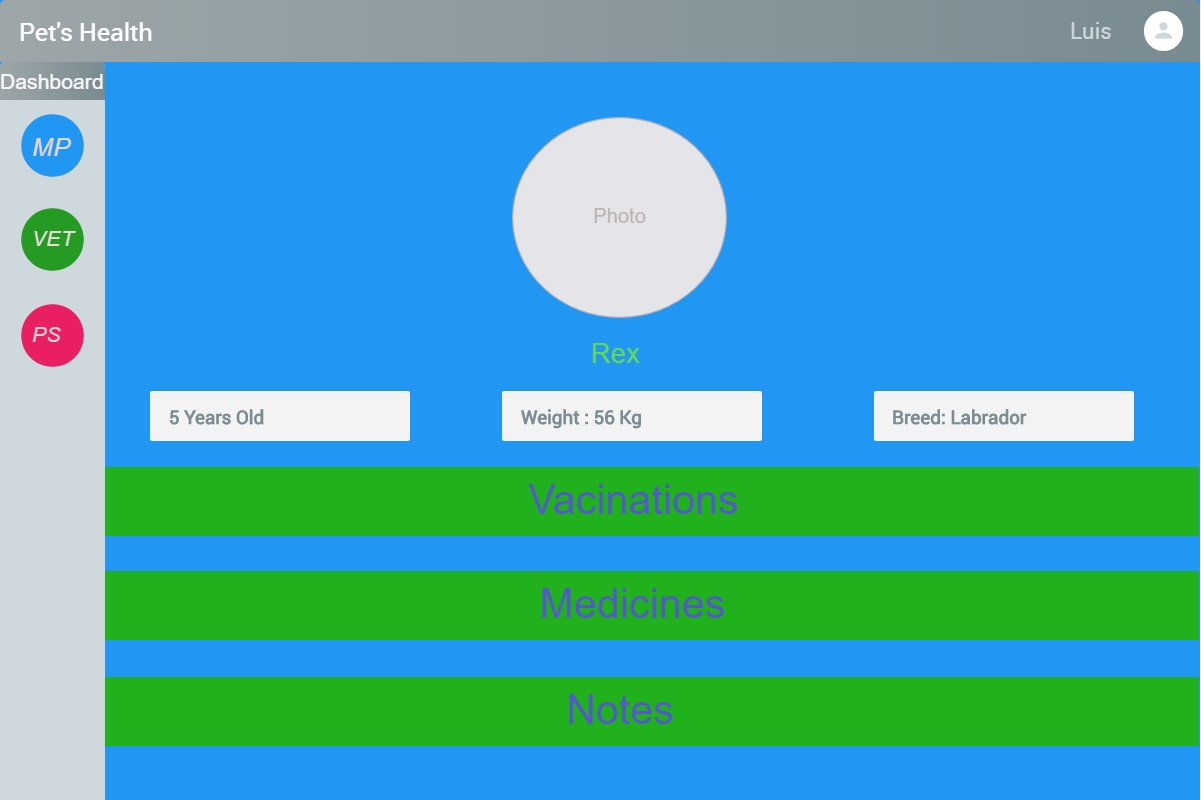


Figure 16 Pet Information

# Implementation of the System

## Architectural Pattern (MVC)

According to Microsoft MVC (Model, View Controller) is an architectural pattern for developing applications, where each part has a responsibility that is different from another.

* Model: The data of the application
* Views: The template files that the application will use to dynamically generate HTML responses.
* Controllers: Classes that handle incoming URL requests to the application, retrieve model data, and then specify view templates that render a response back to the client.

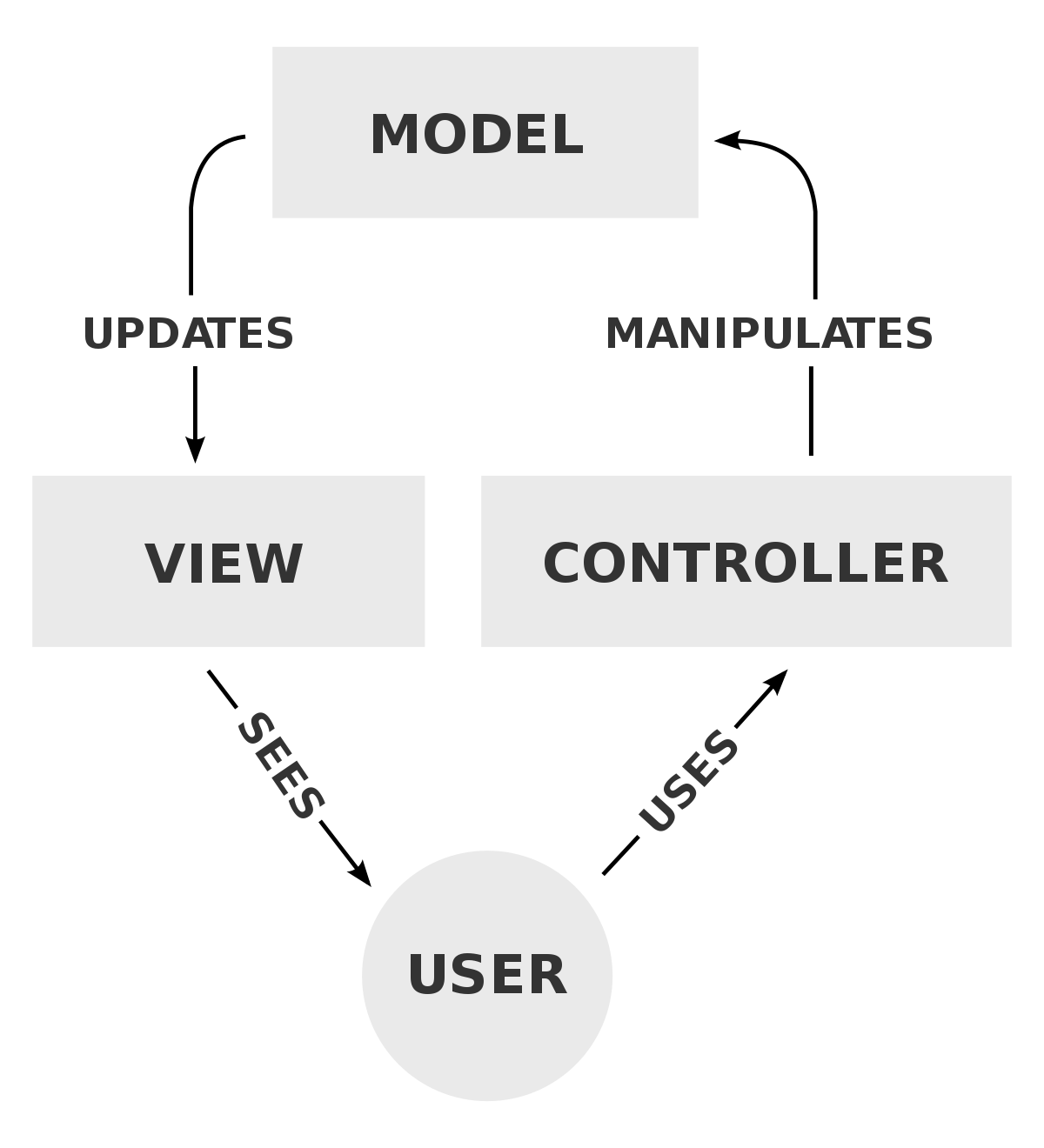


Figure 17 MVC diagram

Follow down a MVC interaction for Pet’s Health Web Application exemplified by steps:

When a user access the login page, the first interaction of MVC occurs.

Step1:

* The user supplies an email and a password to the Controller via the View.

Step 2:

* The Controller checks if the email and password supplied by the user matches to the data stored on the Model.

Step 3:

* If the user had entered the right email and password the Controller pull the data from the Model and redirect the user to the profile page. If the user had provided an email or a password that is not stored on the Model an error message is displayed.

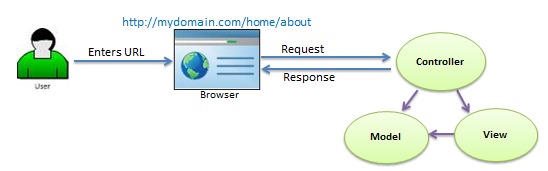


Figure 18 MVC request

## Testing Integration

From the Pet´s Health perspective of testing, User Acceptance Testing (UAT) was applied on this project, is the last phase of the program testing process. UAT was done by group of members as end user testing, validating and verifying Pet’s Health Application.

The Pet’s Health Application was tested on Localhost server and its Database has been hosted using XAMPP control panel. The tables were queried using HeidiSQL and MySQL Workbench.

## Methodologies applied

In order to maintain productivity and set goals, a research of various project methodologies were carried out and Agile Methodology along with Scrum were implemented on Pet’s Health project.

## Agile Methodology

Agile Development focuses on achieving personal, technical, and organization successes delivering and decreasing costs. Agile methods set expectations early in the project. The teams communicate quickly and accurately reviewing the process and continually improving the code making software easier to maintain and enhance over time (Shore & Warden, 2007).

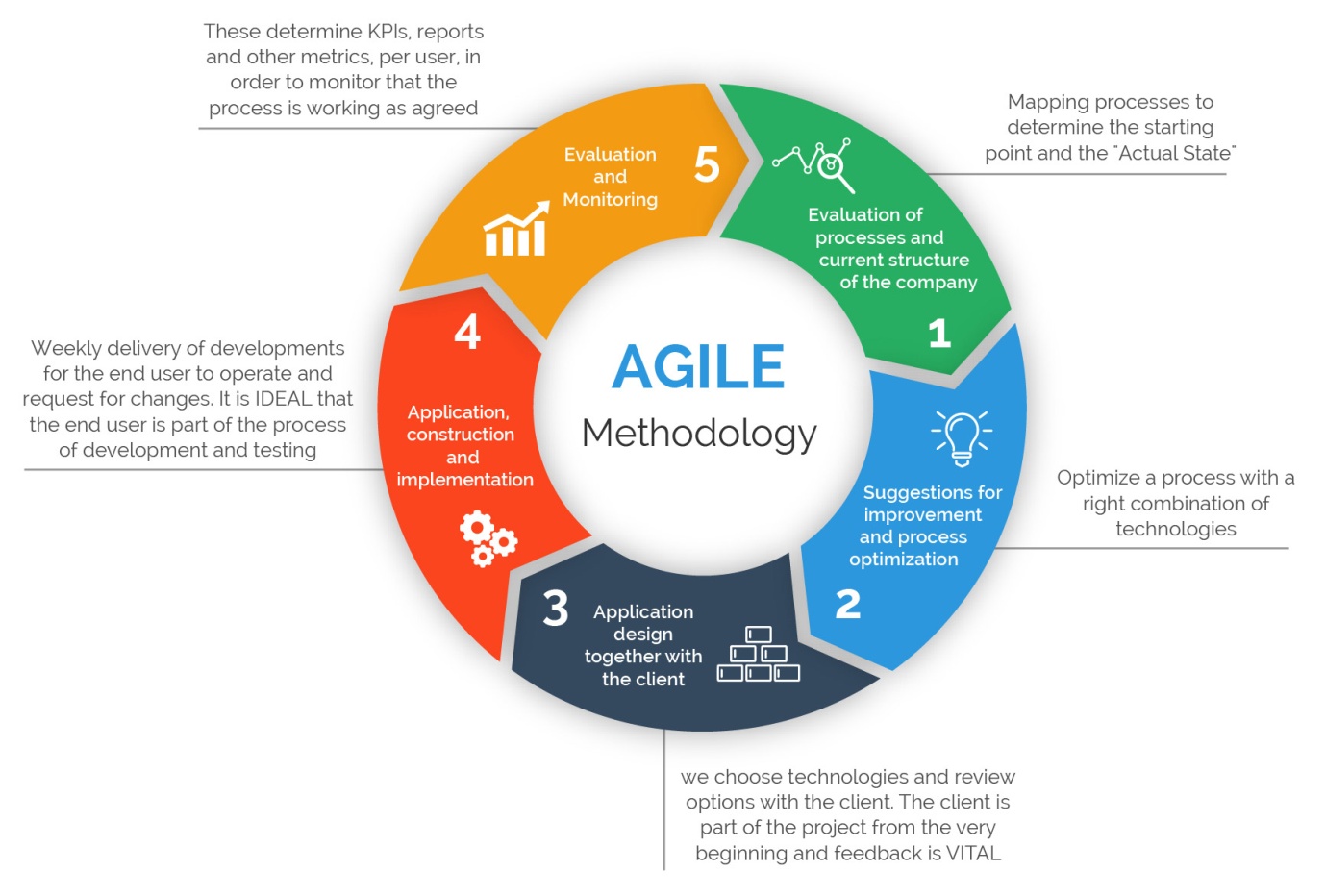


Figure 19 Agile Methodology

## Type of Agile Methodologies

Agile methodologies are distinct and have their own particular approach. However, the methodologies have identical outcome, they incorporate iterative submissions of projects by one way or the other, which helps them get frequent feedbacks, increasing the rate of successful software submission. Mubasher (2017). Follow down the Agile Methodologies:

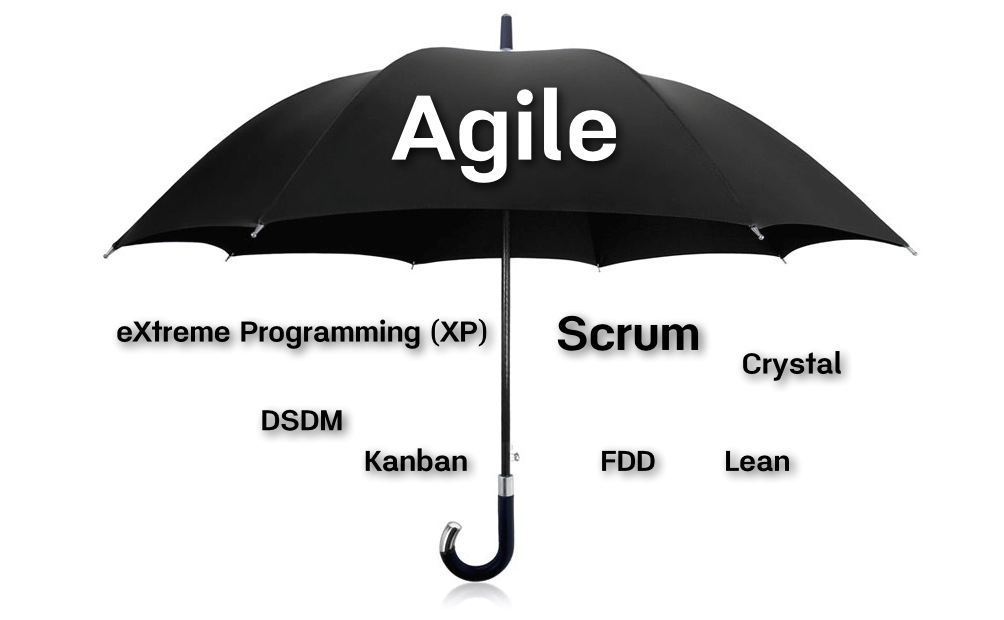


Figure 20 Types of Agile Methodology

## Scrum Methodology

Scrum is a framework used to defining roles in a project, leading to effectiveness in prioritizes and complete the task. It uses sprints (time-boxed monthly increments), each sprints focus on a particular list of requirement called backlogs. Islam (2013) and Layton (2015).

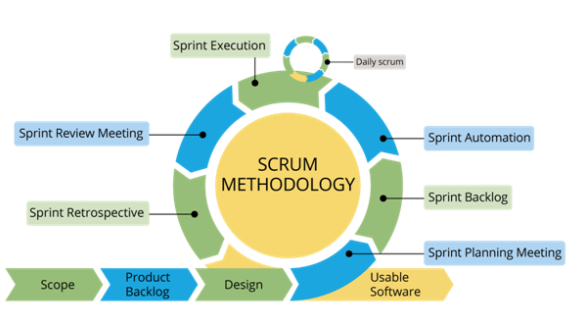


Figure 21 Scrum Methodology

This methodology was partially implemented on Pet’s Health project. Scrum meeting was held every Friday at CCT College, where our Scrum Master delegates the tasks that had to be done by the end of every week. During the meetings were discussed what everyone of the team had done, and what could have been improved for following meetings to achieve best results.

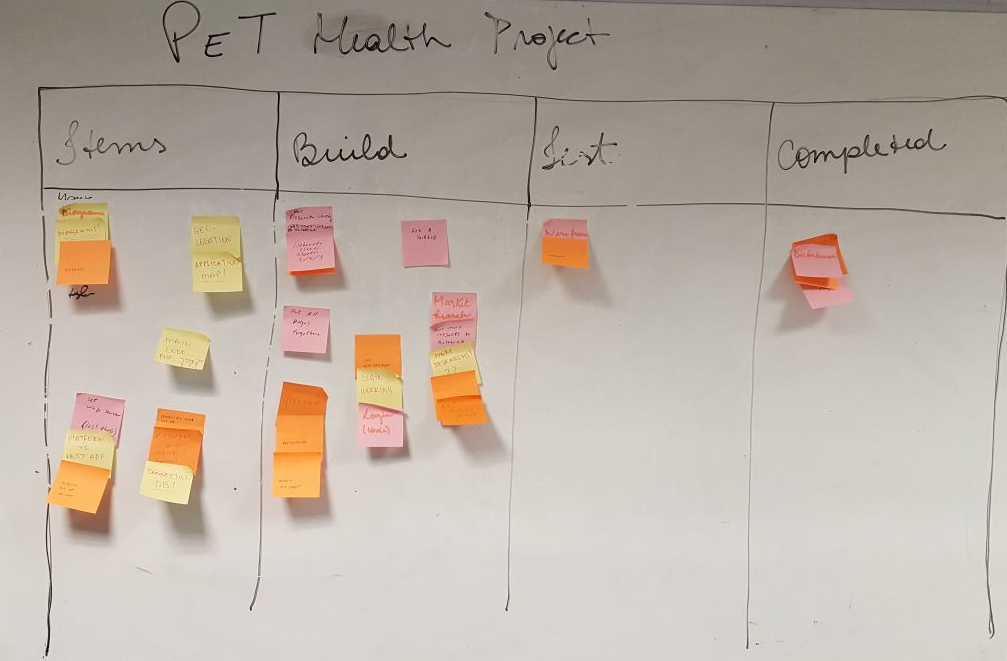


Figure 22 Pet's Heath roadmap broken down into quarters

## Security Risks

Web Application security risks are effective and it is a serious threat that affect its availability.  There are practices to reduce security risks around web applications; Prescott (2018) lists three front-line approaches:

Better training - developers must be trained in and employ secure coding practices,

Rigorous testing - professional security testers must test the applications before deployment.

Stringent policies and procedures - policies and procedures must be in place to prohibit the deployment of applications with vulnerabilities.

With the expansion of the internet and the proliferation of Web-based apps, security vulnerabilities are an open door for web attacks.  An attack of a Web-based application may yield information that should not be available, browser spying, identify theft, theft of service or content, damage the application, etc.

Many aspects make web applications vulnerable, for instance the HTTP protocol, which is clear text, attackers find it very easy to modify the parameters and execute functionality that was not intended to be executed as a function of the application.

There are known vulnerabilities that simple programming practices can reduce. However, some professionals may not be aware of them. The organizations must provide the necessary information and guidance for awareness.  Unfortunately, even some organizations are not fully aware of its vulnerabilities.

The Open Web Application Security Project (OWASP) lists the Top 10 2017 security risks:

[Injection](https://www.owasp.org/index.php/Top_10_2010-A1) - involves the exploiter breaking out of a data context and switching into a code context by using special coding characters.

[Broken Authentication](https://www.owasp.org/index.php/Top_10_2010-A3) - involves the attacker stealing or assuming the identity of the unprotected authentication credentials of a user.

Sensitive Data Exposure - rather than directly attacking crypto, attackers steal keys, execute man-in-the-middle attacks, or steal clear text data off the server, while in transit, or from the user’s client.

Sensitive Data Exposure - attackers can exploit vulnerable XML processors if they can upload XML or include hostile content in an XML document, exploiting vulnerable code, dependencies or integrations.

Broken Access Control- the exploitation of access control is a core skill of attackers. There are tools can detect the absence of access control but they cannot verify if it is functional when it is present.

Security Misconfiguration - attackers often attempt to exploit unpatched flaws or access default accounts, unused pages, unprotected files and directories, etc to gain unauthorized access or knowledge of the system.

Cross-Site Scripting (XSS) - automated tools can detect and exploit all three forms of XSS, and there are freely available exploitation frameworks.

Insecure Deserialization - exploitation of deserialization is somewhat difficult, as off the shelf exploits rarely work without changes or tweaks to the underlying exploit code.

Using Components with Known Vulnerabilities - while it is easy to find already-written exploits for many known vulnerabilities, other vulnerabilities require concentrated effort to develop a custom exploit.

Insufficient Logging & Monitoring - exploitation of insufficient logging and monitoring is the bedrock of nearly every major incident. Attackers rely on the lack of monitoring and timely response to achieve their goals without being detected.

OWASP provides great detailed information describing each of these risks, and how to avoid, review the code and make test for them.

## Project Progression and Further work

For this project become achievable, an important goal was to work efficiently as a team, each member had a valuable role on the development process and commitment was required.

The members of the group have shown dedication and patience to accomplish the final result of the project respecting delivery dates for each task and working together as a team. Follow the collaboration roles for the Pets’ Health project:

***Caroline Santana*** worked on the code developing the first template proposed (framework 7), helped with ideas on the layout of the program also had a significant cooperation on the research section, reference list format and designed the use case diagram. Caroline had shown to be a strong team player bringing all of her ideas and thoughts to the brainstorm sections.

***Danielle Santos*** as part of Pet’s Health group had a considerable contribution with management of ideas, project planning, suggesting new methodologies as Agile and applying Scrum as a part of the meetings, encouraging team members to do frequent deliveries. Related to documentation Danielle had a significant role organizing topics, uploading new versions and keeping the files tidy on basecamp. Also had shown availability to help all group members and always treat them with respect.

***Luis Castro*** introduced the application idea to the group, designed the wireframes and mock-ups. Luis structured an online survey to gather information refining features of the application. Worked on the code developing the first template proposed (framework 7), played an important role providing the design and functionality for the presentations, and also assisted on the documentation structure. Luis had cooperated on brainstorm meetings bringing new ideas on technologies and features.

***Mara Dias*** worked on the code developing the first template proposed (framework 7), helped with ideas on the layout of the program. Mara had a significant cooperation on the research section of Databases structures, market research, reference list format and table of figures format, designed use case diagrams. Mara had shown to be a strong team player bringing all of her ideas, and analytical skills to the thoughts to the brainstorm sections.

***Marcionedes de Abreu*** was focused and responsible for developing the design and back end code. Worked on the code developing the first template proposed (framework 7) and brought new technology concept to achieve the desired result. Marcionedes had assisted on the development of the database structure, providing minor changes, in order to promote a better user experience. Had the role as a counsellor trying to motivate and promote encouragement among group members, assuming the leader role in Michael’s absence occasions. Contributed on the research technologies, documentation, and live presentations.

***Michael Faria*** had an important role in the group which was being a group leader. Michael had assisted on the development of the database structure, worked on the code developing the first template proposed (framework 7). Gathered constructive criticism from lectures and classmates and organized it for future references, mediated the communication between lectures and group members, had organized files and researches on folders to keep easy access for all the members and guided reading sections, promoting direction and instructions on the document structure. Reviewed, monitored and managed every task during the stages of the project.

# Technology Research and Survey

## GitHub

GitHub is a code hosting platform for version control and collaboration. It offers all of the distributed version control and source code management. It provides access control and several collaboration features such as bug tracking, features request, task management.

GitHub gives a free and private version both is a good way to share and publish service or it is a social networking site for programmers. GitHub was used for manage and review changes on the source code.

## Cascading Style Sheets (CSS)

CSS (Cascading Style Sheets) it is a simple design language intended to simplify the process of making web pages presentable, and you can control colours, layout, background images and fonts and also tailor pages to different devices different types of devices. Once that CSS is written it can be reuse in multiple HTML pages. CSS can store web application locally with the help of an offline cache which ensures faster loading and better overall performance of the website. CSS was implemented on the pages to provide a better system design.

## Web Server

Also known as, an Internet Server, Web Server is a computer or systems that deliver web pages, contents or service, to end users.

It delivers content throw Hypertext Transfer Protocol, HTTP, pages may content HTML documents, images, style sheets and text content.

This process is made by the browser communicating with the server sending a GET request, asking for a specific file, then the server send a HTML text for the web page to the browser, where its read and formatted onto the screen. Apache web server was used to host the application on local host.

## Bootstrap

Bootstrap is a [free and open source](http://whatis.techtarget.com/definition/Free-and-open-source-software-FOSS-or-free-libre-open-source-software-FLOSS) [front-end](http://whatis.techtarget.com/definition/front-end) development framework for the creation of websites and [web apps](http://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app). The Bootstrap framework is built on [HTML](http://searchmicroservices.techtarget.com/definition/HTML-Hypertext-Markup-Language), [CSS](http://searchmicroservices.techtarget.com/definition/cascading-style-sheet-CSS), and JavaScript ([JS](http://searchmicroservices.techtarget.com/definition/JavaScript)) to facilitate the development of [responsive](http://whatis.techtarget.com/definition/responsive-design), [mobile-first](http://searchmobilecomputing.techtarget.com/definition/mobile-first) sites and apps.

Bootstrap includes user interface components, layouts and JS tools along with the framework for implementation. The software is available precompiled or as [source code](http://searchmicroservices.techtarget.com/definition/source-code).

Mark Otto and Jacob Thornton developed Bootstrap at Twitter as a means of improving the consistency of tools used on the site and reducing maintenance. The software was formerly known as Twitter Blueprint and is sometimes referred to as Twitter Bootstrap. Bootstrap was implemented on the final version of the web application as front-end framework.

## JavaScript (JS)

JavaScript (often shortened to JS) is a lightweight, interpreted, object-oriented language, and is best known as the scripting language for Web pages, but it's [used in many non-browser environments](https://en.wikipedia.org/wiki/JavaScript#Uses_outside_web_pages) as well.

JavaScript runs on the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event.

JavaScript can function as both a [procedural](https://en.wikipedia.org/wiki/Procedural_programming) and an [object oriented language](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Introduction_to_Object-Oriented_JavaScript). Objects are created programmatically in JavaScript, by attaching methods and properties to otherwise empty objects at run time, as opposed to the syntactic class definitions common in compiled languages like C++ and Java. Once an object has been constructed it can be used as a blueprint (or prototype) for creating similar objects.

The development of dynamic Web pages has been strongly influenced by the rise of “new” architectures for developing web-based information system, Sun’s Java 2 Enterprise Edition (J2EE) architecture, which allows web pages to be generated from data drawn from relational databases, components, message, queues and legacy systems. ( Stanek and Watters, 2006).

JavaScript and Java are completely different languages, both in concept and design.

JavaScript was invented by Brendan Eich in 1995, and became an ECMA standard in1997.ECMA-262 is the official name of the standard. ECMA Script is the official name of the language. JavaScript was used as a front-end language on the application.

## XAMMP

XAMPP is a free open source web server package which works in different platforms. It stands for Cross - Platform as Apache, MySQL, PHP and Perl. Also as cross-platform, it works equally well on Linux, Mac and Windows. It makes transitioning from a local test server to a live server is very easy to manage. These are four primary components:

* Apache: Apache is the actual web server application that processes and delivers web content to a computer. Apache is the most popular web server online, powering nearly 54% of all websites.
* MySQL: Every web application, howsoever simple or complicated, requires a database for storing collected data. MySQL, which is open source, is the world’s most popular database management system. It powers everything from hobbyist websites to professional platforms as WordPress.
* PHP: It is a server-side scripting language that powers some of the most popular websites in the world, including WordPress and Facebook. It is open source, relatively easy to learn, and works perfectly with MySQL, making it a popular choice for web developers.
* Perl: Perl is a high-level, dynamic programming language used extensively in network programming, system admin, etc. Although less popular for web development purposes, Perl has a lot of niche applications.

XAAMP provides the components needed for the project and hosted the web application.

**PHP Hypertext Pre-processor (PHP)**

PHP Hypertext Pre-processor (PHP) is open source language, which means anyone can access the source code and is free of charge.

PHP is an HTML-embedded Web scripting language. This means PHP code can be inserted into the HTML of a Web page. When a PHP page is accessed, the PHP code is read or “parsed” by the server the page resides on. The outputs from the PHP functions on the page are typically returned as HTML code, which can be read by the browser.

A lot of the syntax of PHP is borrowed from other languages such as C, Java and Perl. , PHP has a number of unique features and specific functions as well.

PHP allows:

* To generate dynamic page content;
* To create, open read, write, delete and close files on a server;
* To send and receive cookies;
* To add, delete, or update data databases;
* To control user-access;
* To encrypt data.

PHP was implemented as back-end language, server side for the web application.

## MySQL

It is an open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL) is developed and supported by Oracle Corporation. MySQL databases server is fast, reliable and scalable. MySQL is a client/server system that consists of a multi-threaded SQL server that support different back ends and application programming interface (APIs). Its name is a combination of the name of co-founder Michael Widenius’ daughter, the word “My” and the acronym “SQL”, stands for “My Structured Query Language”.

MySQL is written in C and C++ and its SQL parser is written in YAAC. It runs as a server and allows different users to create and manage various databases and was developed to handle extensive databases.

MySQL is a key element of the LAMP (Linux, Apache, MySQL and Perl/PHP/Python) web application software stack and is used in many high profile websites such as Google, Facebook, Twitter and YouTube. The web application uses MySQL database to store the data.

## AJAX

Ajax is a client-side script that communicates to and from a server without the need for a post-back or compete page refresh. In another words, Ajax can update part of the webpage without reloading the entire page. Ajax is not a programming language or tool, but a concept.

Ajax acronym for Asynchronous JavaScript and XML where asynchronous means: multiple events happening independently of one another.

Benefits of Ajax:

* Call-backs: Makes a quick trip to and from the server to retrieve and/or save data without posting the page back to the server.
* Making asynchronous calls: This allows the client browser to avoid waiting for all data to arrive before allowing the user to act once more.
* User-friendly: because the page post-back is being eliminated, Ajax enabled applications will always be more responsive, faster and more user-friendly.
* Increased speed: The main purpose of Ajax is to improve the speed, performance and usability of a web application. A known example of Ajax being used is Netflix film rating feature. The user rates a film and their personal rating for that will be saved to their database without waiting for the page to refresh or reload, so the user can keep doing whatever its doing at the moment.

Ajax should be used anywhere in a web application where small amounts of information could be saved or retrieved from the server without posting back the entire pages. A good example of this is data validation on save actions. Another example would be to change the values in a drop down list-box based on other inputs, such as state and college list boxes. When the user selects a state, the college list box will repopulate with only colleges and universities in that state.

How AJAX Works

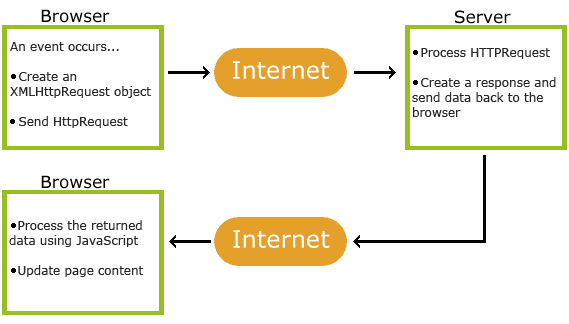


Figure 23 Ajax implementation

AJAX calls were implemented with PHP in order to provide more interactivity on the application.

## Application Program Interface (API)

An application program interface (API) is code that allows two software programs to communicate with each other.

The API defines the correct way for a developer to write a program that requests services from an operating system (OS) or other application. APIs are implemented by function calls composed of verbs and nouns. The required syntax is described in the documentation of the application being called.

APIs are made up of two related elements. The first is a specification that describes how information is exchanged between programs, done in the form of a request for processing and a return of the necessary data. The second is a software interface written to that specification and published in some way for use.

The software that wants to access the features and capabilities of the API is said to call it, and the software that creates the API is said to publish it. The Google Maps API was implemented on the application final version.

## Facebook Login API

Facebook Login with the Facebook SDK for JavaScript enables people to sign into a web page with their Facebook credentials. To implement Facebook Login with the JavaScript SDK, a Facebook App ID is needed. The implementation of Facebook Login API was discussed and discouraged, due our assumption of its low contribution and relevance to the application.

## 

## Google Maps API

The Maps JavaScript API offers the customization of maps with own content and imagery for display on web pages and mobile devices. The Maps JavaScript API features four basic map types (roadmap, satellite, hybrid, and terrain) which allows modify using layers and styles, controls and events, and various services and libraries. Google Maps JavaScript API was implemented on the application to show nearby veterinarian’s clinics and pet sitters.

## Framework 7

Framework7 is a free and open source mobile HTML framework to develop hybrid mobile apps or web apps with iOS & Android native look. It is also used as a prototyping apps tool to show working app prototype.

The main approach of the Framework7 is to give an opportunity to create iOS & Android apps with HTML, CSS and JavaScript. Framework7 was used as font-end on the first version of the application; due its low usage on the market and implementation complexity it was replaced for Bootstrap.

## Market Research

There are similar applications available on the market, specially focused on dogs, our group downloaded and used two popular native applications (Dog Heath, DogBuddy) to test their functionalities.

*Dog Health*

The app has some similarities with Pet’s Health such as adding a new pet and managing the following information: name, DOB, weight, withers (size), breed, chip, sex (gender). Once all the form is filled and submitted, the pet is registered into the database and the user choose between few options:

Memo: check any notification that is coming in the near future.

Medical Records: shows six more tabs where the user can add more info (vaccination, antiparasitics, visits and surgeries, medicine/administrations, notes and veterinarians which opens locations of nearby clinics).

The app was developed in Java for Android, Objective-c for iOs and backend developed in Google App engine (Java). Dog Health was developed by Luca Biasotto.

*DogBuddy*

DogBuddy is an application available for Iphone/Ipad, providing two versions Dog Buddy Free and Dog Buddy Pro (Paid Version). The app offers a diary for the dog, record milestones such as birth, grown teeth, special dates, etc.

Also allows the addition of veterinaries, stores photos, add reminders, keep track of dog’s medication and information such as weight and breed. It provides dogs first aid with an instruction list. The paid version (Dog Buddy Pro) supports Dropbox backup and restore, also allows the export of the all data.

In general DogBuddy is a useful application; its features are practical and relevant. However, the interface is not fully friendly; some of the features provided are difficult to understand its functionalities. DogBuddy was developed by Maxwell Software.

## Pet’s Health Survey

The following survey is a voluntary online questionnaire. The survey was done with Google Forms and posted on Facebook.

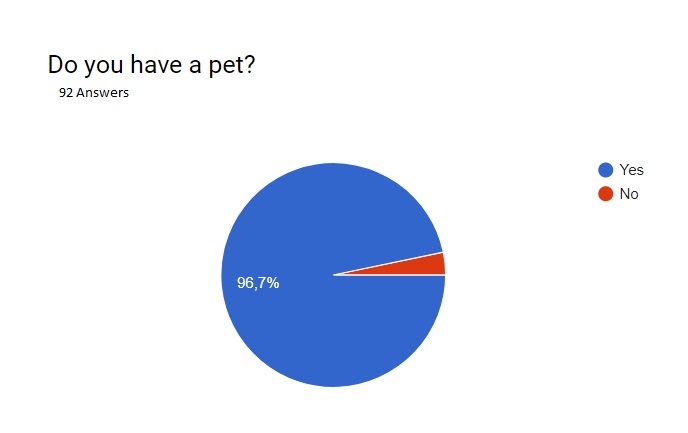


Figure 24 Pet's Health survey Question 1

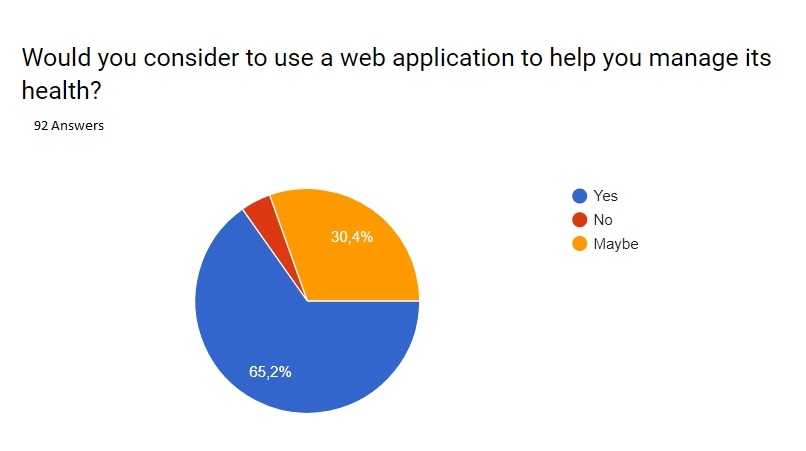


Figure 25 Pet's Health survey Question 2

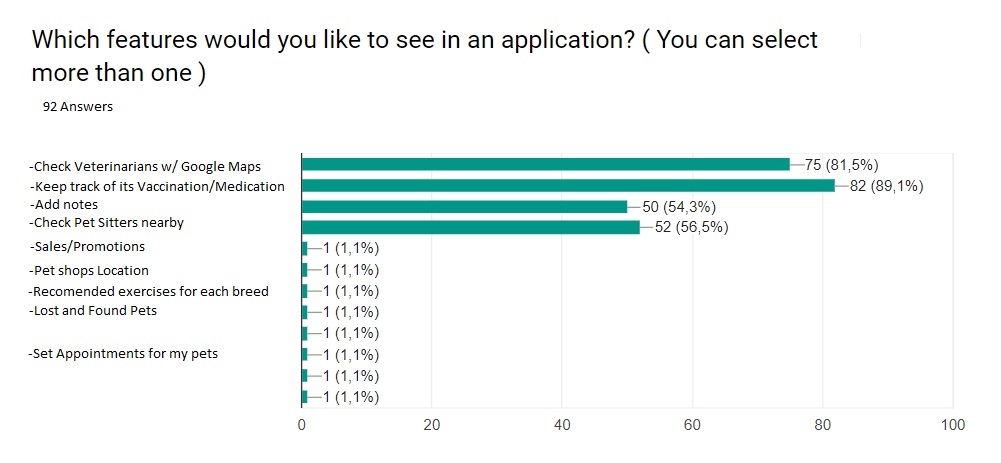


Figure 25 Pet's Health survey Question 3

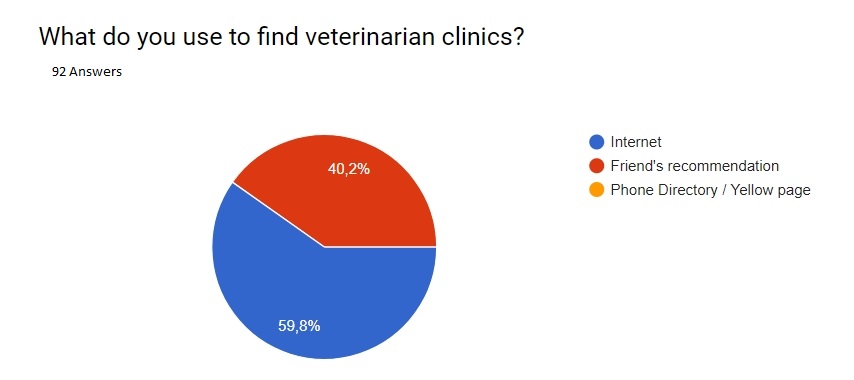


Figure 27 Pet's Health survey Question 4

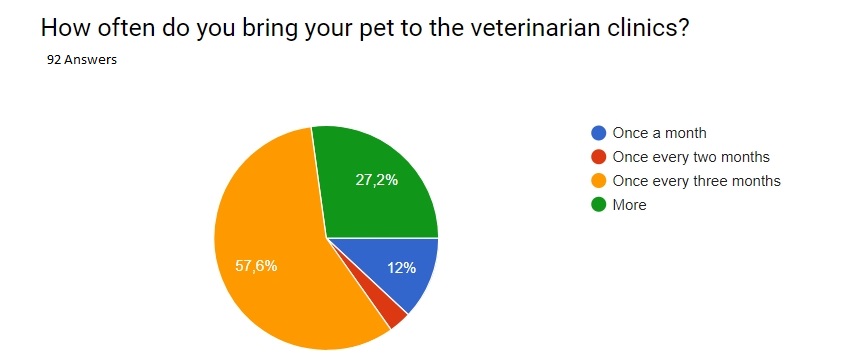


Figure 28 Pet's Health survey Question 5

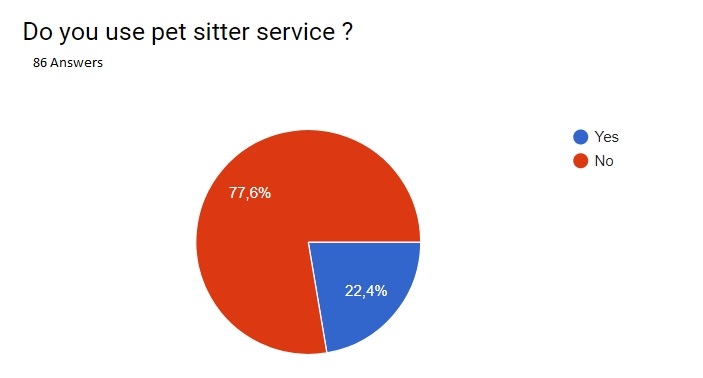


Figure 29 Pet's Health survey Question 6

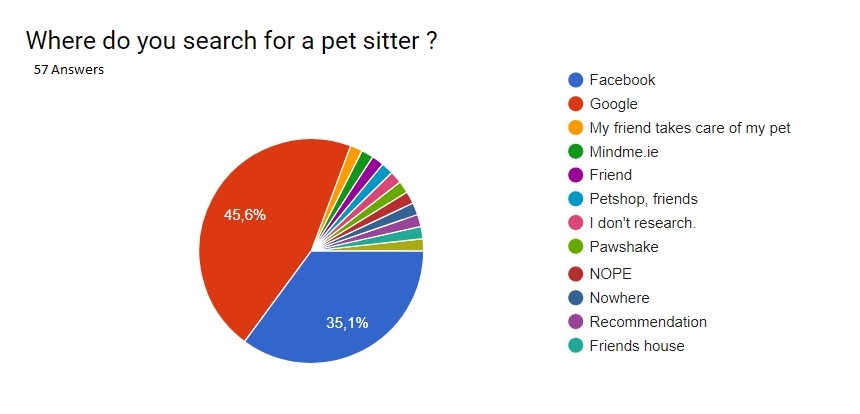


Figure 30 Pet's Health survey Question 7

## Local Veterinarian Survey

Questions answered by *Katy Fortune*, Registered Veterinary nurse, RVN.

Legal questions directed to one professional veterinarian about legal utilization and implementation of our app.

**I. What are the legal implications of having medical records stored in my App?**

The records are treated with confidence. The vet and the pet owners are the only people that have access to the information. If anyone else wants to get permission to see the data, they just must ask the owner of the pet for permission to see the data.

**II. Is it something veterinarians would use to minimize the amount of paperwork and also put together all information about the Pets?**

The system that is currently used by the vets is called Teleos (http://www.teleosvet.com). It is an electronic record management system.

X-ray information is handled by another system and records are kept for 7 years. X-rays are digital so they all appear on the computer, some clinics use paper based systems however.

**III. What kind of information would we need for medical records and vaccination records?**

Date, Medication, Amount, Location, Vet Name

**IV. As a Vet, Would you recommend the idea of keeping all data (information) in one place?**

Yes, keeping the information in one place is one of the most useful elements in the system. However, consent forms etc. are not stored in the system and are kept in paper based format. This does not exist in the system.

When the dog is micro chipped we keep a copy which the owner filled it but it's send off and registered online so we never actually use the forms. The microchip number is on their teleos file, but the registration is paperwork. Consent forms are kept also,

So if the animal is admitted for a procedure the owner signs a consent form and they are kept. Details of procedure are on Teleos but signed form is paperwork stored

**V. Is this all data we need about medication or vaccination?**

Information about times, dates, location and who gave the medications needs to be recorded. Does everything really! Appointments, payments etc. all on system too

**VI. Is there sample paperwork for tracking the vaccination and medication?**

There is no sample paperwork as it is all kept inside of the system.

**VII. Can you provide sample paperwork?**

Medication is all dispensed through Teleos, Labels printed from there and detailed on Teleos file. If it's something we don't stock prescription forms can also be printed from Teleos

**VIII. Do vets swap information between each other? If so how?**

When vets want to share information to another vet, they export the data from the system and export the data as a PDF and send it by email or print and sent it to the new vet.

**Conclusion**

The final year project was carried out over three semesters, with plenty of research on the newest technologies available and programming languages, to achieve the best performance to deliver an innovative, effective and functional project. The group was formed in 2016 when we started with the first steps. To create a project proposal, through lots of meetings and brainstorming sections, the idea to create an application to aid pet owners came to mind.

The surveys applied had a great impact on the development process; they worked as a guide to structure our ideas towards the application end result. The report provided by the veterinarian suggested the need of an efficient and secure system to manage pets’ health records. The online questionnaire offered an important experience taking into consideration the opinion from possible potential users of the system; it offered a substantial support for our ideas on the system features. Also, the research and usage of similar applications was relevant in order to learn how different systems work and extracted their greatest functionalities.

The initial idea was to develop a native application (available for Android and iOS) for Pet owners. However the development took another approach due to the lack of experience on this technology, we then came to the conclusion that we needed to create a web application. Although it was easier to develop a web application, the process of coding also hit some barriers and had some difficulties; it brought us to a final change of paths to achieve the desired result when the group members decided to apply Bootstrap technology which includes HTML, CSS and Java Script library for the interface, PHP for back end, MySQL for the database, Google Maps API and XAMPP as web server.

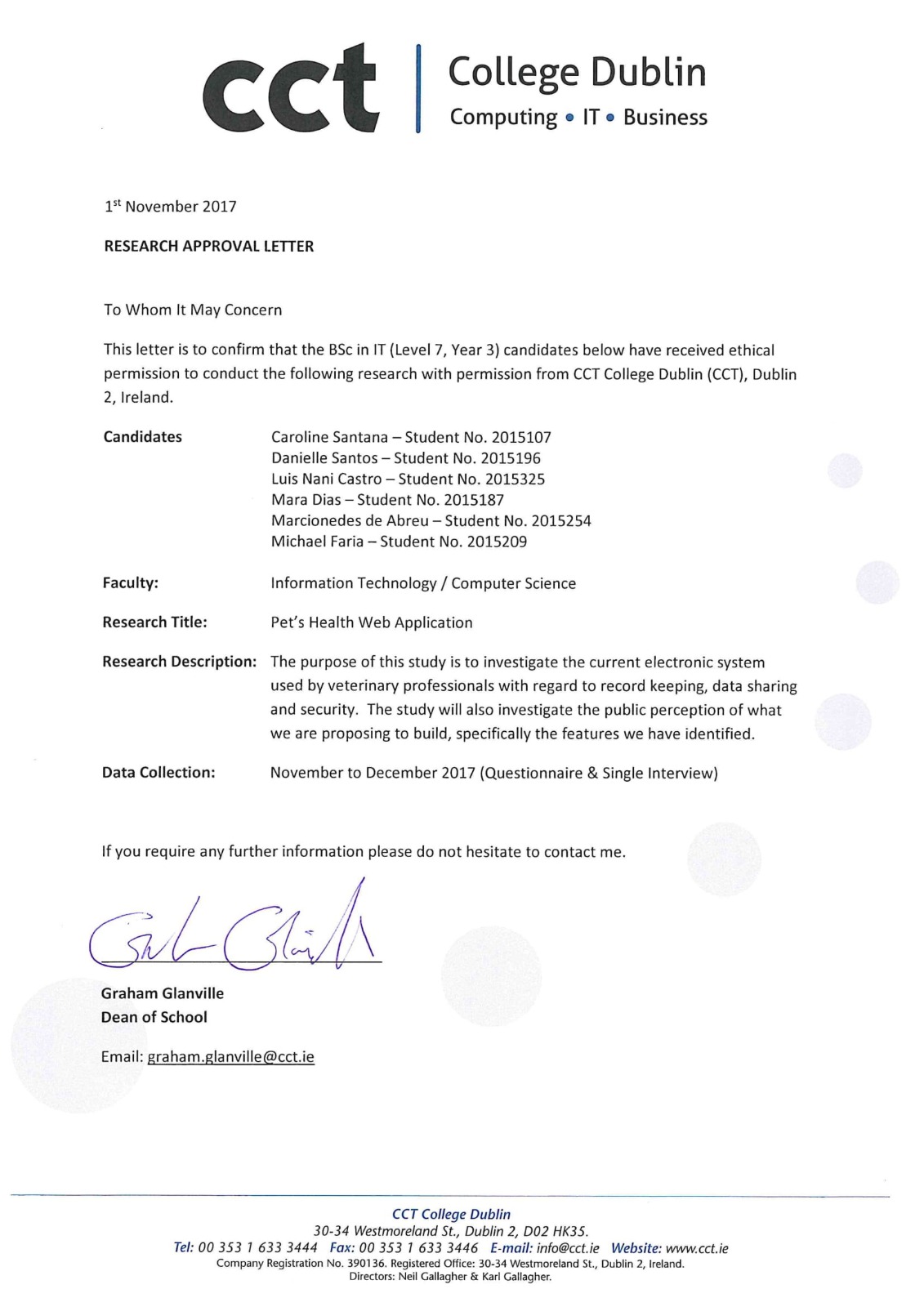
That said, the final version of the application brings users an interactive platform which reduces the amount of paperwork they would carry around, and offers them a centralized source of information when it comes to their pet’s data. Users can see pet’s medical records, veterinarian clinics and pet sitters through the application.

Considering all the challenges that the group had through the academic journey during the past three years, the group has produced and delivered the expected results. A key point was communication, and working effectively as a team. Developing Pet’s Health Application code and documentation brought us both joy and improvement of personal knowledge.

# Appendices

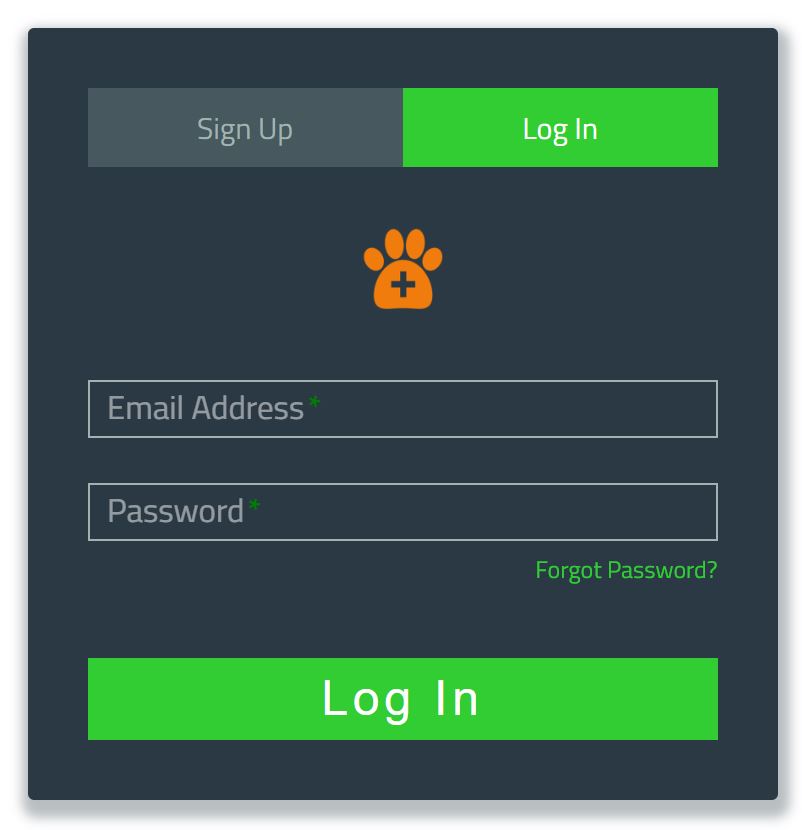
## Appendix A

Research Approval Letter

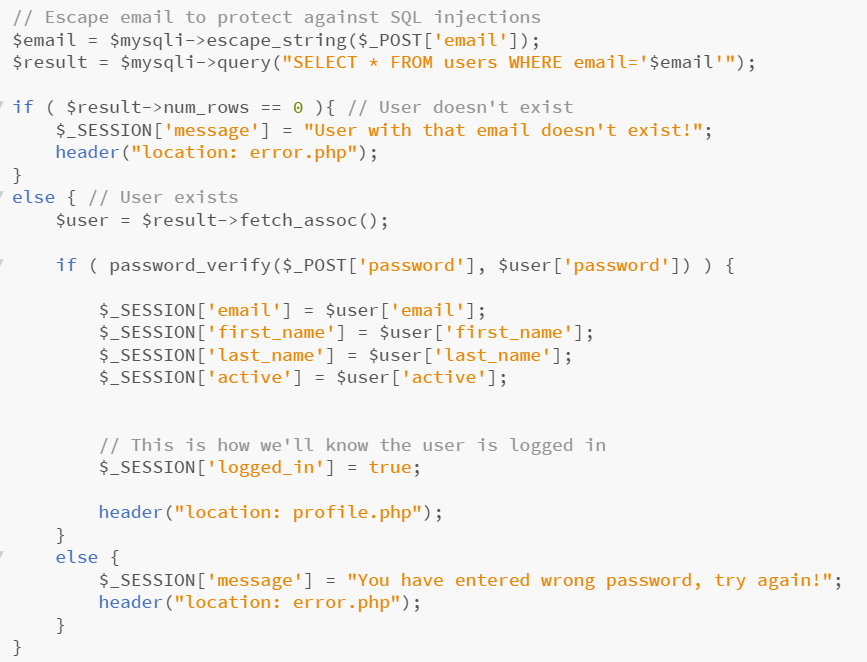


## Appendix B

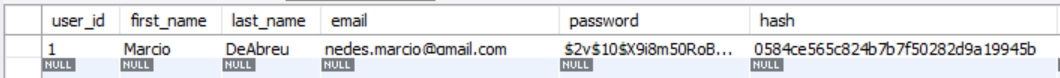
Login Page



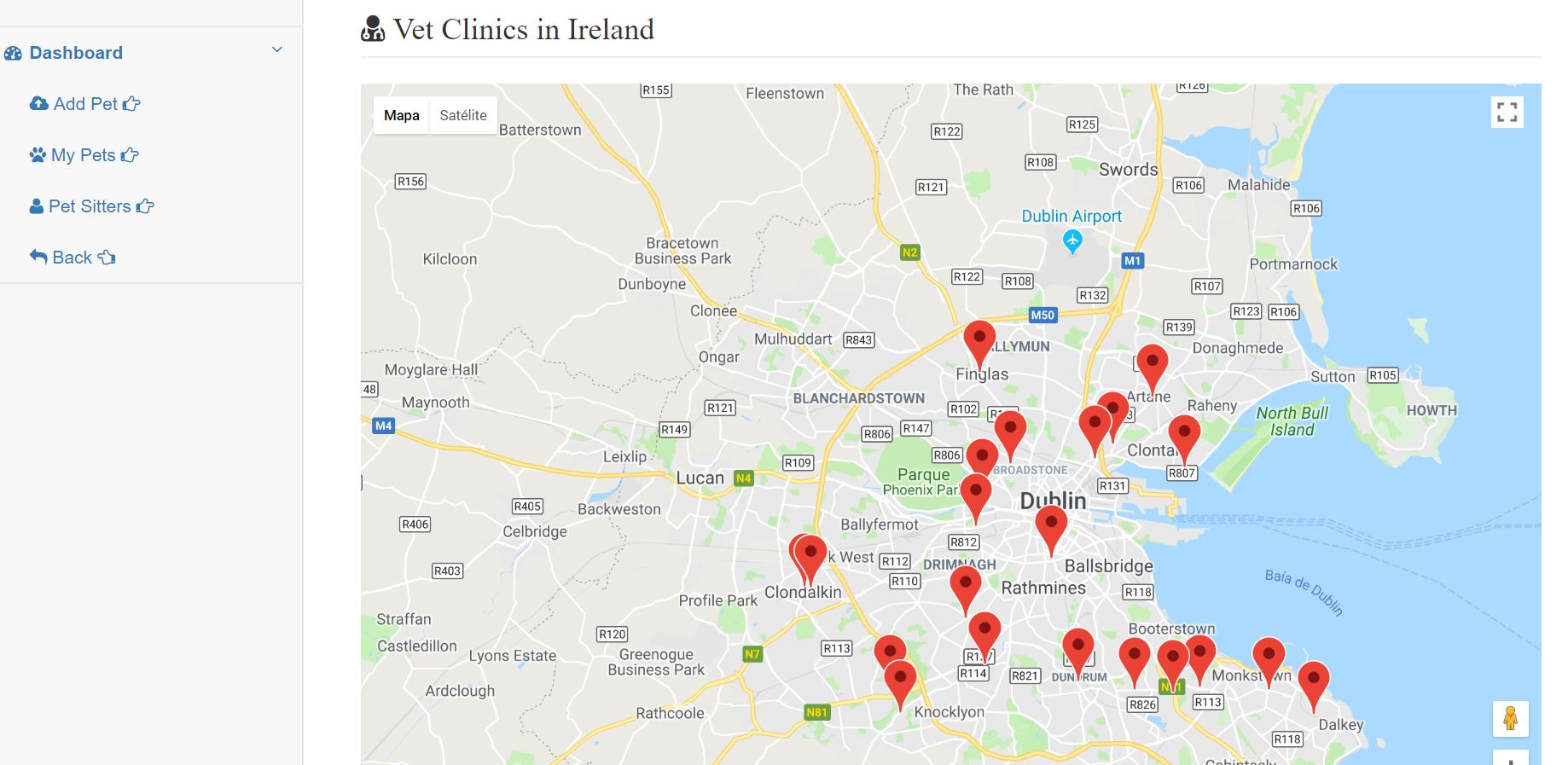
PHP login validation code



User registered on the database



My Vets Page



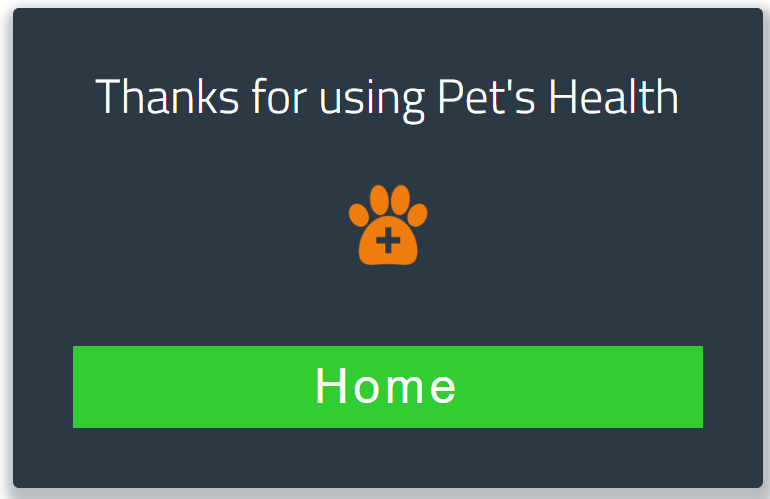
Google Maps JavaScript API code



Add New Pet Page

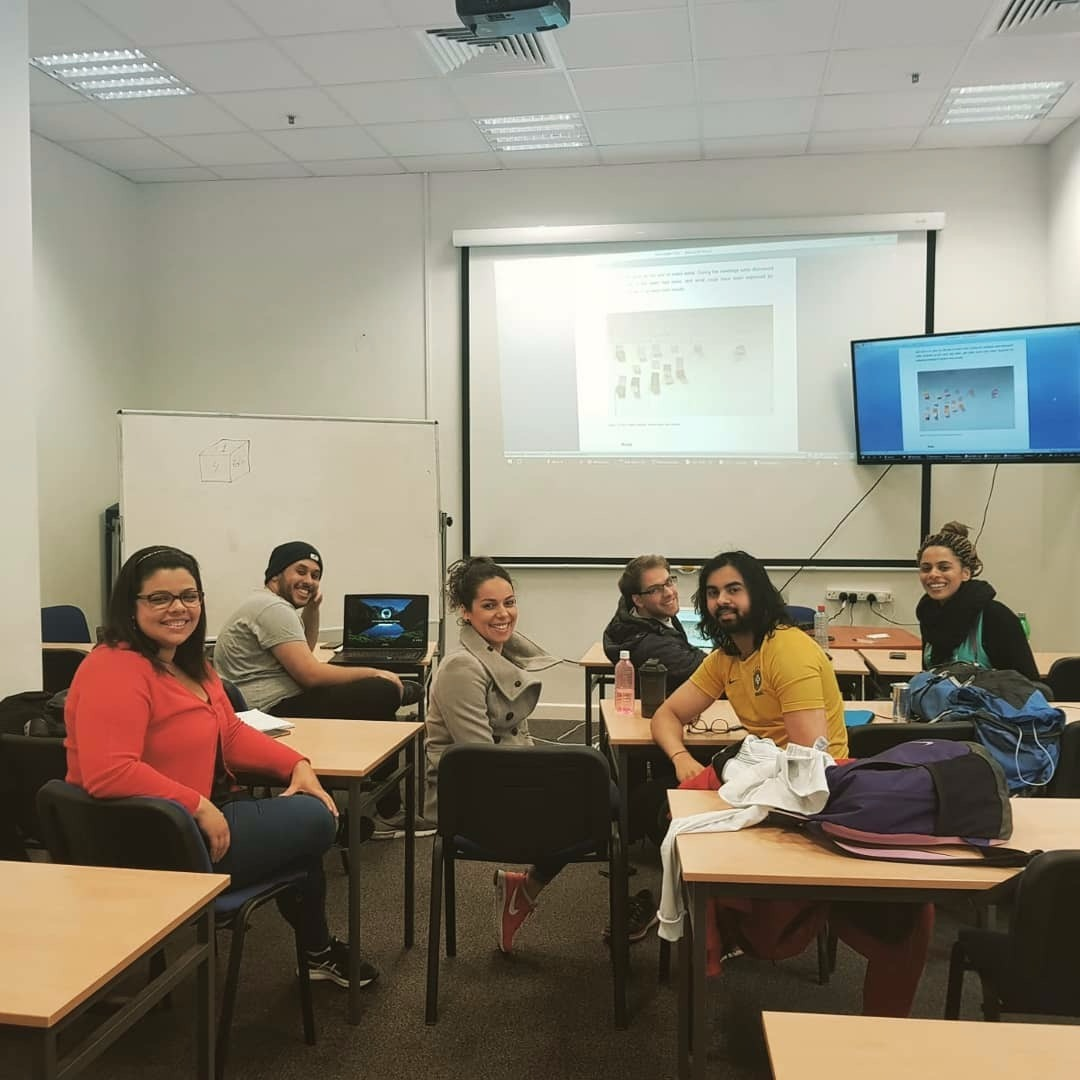


Logout Page



## Appendix C

Final documentation meeting



**Table of Figures**

[Figure 1 Pet´s Health schedule 9](#_Toc511986182)

[Figure 2 Pet's Health workflow 13](#_Toc511986183)

[Figure 3 Pet’s Health use case diagram successful scenario 14](#_Toc511986184)

[Figure 4 Pet's Health use case diagram unsuccessful scenario 14](#_Toc511986185)

[Figure 5 Pet's Health sequence diagram where the user adds a pet 15](#_Toc511986186)

[Figure 6 Pet's Health sequence diagram to check added pet 16](#_Toc511986187)

[Figure 7 Pet's Health sequence diagram to check veterinary clinics 16](#_Toc511986188)

[Figure 8 Pet's Health sequence diagram to check pet sitters 17](#_Toc511986189)

[Figure 9 Entity relation diagram 18](#_Toc511986190)

[Figure 10 Login page 21](#_Toc511986191)

[Figure 11 Dashboard 21](#_Toc511986192)

[Figure 12 Add Pet 22](#_Toc511986192)

[Figure 13 Medication page 22](#_Toc511986193)

[Figure 14 Vaccination 23](#_Toc511986192)

[Figure 15 My Pets 23](#_Toc511986192)

[Figure 16 Pet Information 24](#_Toc511986192)

[Figure 17 MVC diagram 25](#_Toc511986194)

[Figure 18 MVC request 26](#_Toc511986195)

[Figure 19 Agile Methodology 27](#_Toc511986196)

[Figure 20 Types of Agile Methodology 28](#_Toc511986197)

[Figure 21 Scrum Methodology 29](#_Toc511986198)

[Figure 22 Pet's Heath roadmap broken down into quarters 29](#_Toc511986199)

[Figure 23 Ajax implementation 40](#_Toc511986200)

[Figure 24 Pet's Health survey Question 1 43](#_Toc511986201)

[Figure 25 Pet's Health survey Question 2 44](#_Toc511986202)

[Figure 26 Pet's Health survey Question 3 44](#_Toc511986203)

[Figure 27 Pet's Health survey Question 4 45](#_Toc511986204)

[Figure 28 Pet's Health survey Question 5 45](#_Toc511986205)

[Figure 29 Pet's Health survey Question 6 46](#_Toc511986206)

[Figure 30 Pet's Health survey Question 7 46](#_Toc511986207)

# Reference List

Whitaker, K. (2010) *Principles of Software Development Leadership.* 4th ed. Boston: Cengage Learning, p.12.

Stanek, W. & Watters, P. (2006) *Web Guru Guide to JavaScript*. 1st ed. New Jersey: Pearson Education, pp. 7-8.

Briton, C. & Doake, J. (2004) *A Student Guide to Object-Oriented Development*. 1st ed. Butterworth-Heinemann: Elsevier, pp.10-12.

Elmasri, R. & Navathe, B. (2016) *Fundamentals of Database Systems.* 7th ed. England: Pearson Education Limited, pp. 50- 51.

*Pilone, D. (2005) Uml 2.0 in a Nutshell.* 1st ed. Farnham*: O'Reilly* Media Inc.p. 48.

Layton, M. (2015) *Scrum for Dummies*. 1st ed.  New York: John Wiley & Sons, p. 17.

Islam, K. (2013) *Agile Methodology for Developing & Measuring Learning*: Training *Development for Today’S World.* 1st ed. Bloomington: AuthorHouse, p.28.

Shore J., Wardern S. (2007) The Art of Agile Development.1st ed. Sebastopol: O’Reilly Media, Inc. pp.5-7.

[Mubasher](https://medium.com/@mubasherkhawaja?source=post_header_lockup), K. (2017) *Manifesto to Agile Methodology*. Retrieved from: https://medium.com/@mubasherkhawaja/manifesto-to-agile-methodology-c9e9b157eaf6 on 7 of April 2018.

Peham, T. (2013) *5 Types Of User Acceptance Testing.* Retrieved from: <https://usersnap.com/blog/types-user-acceptance-tests-frameworks/> on 13 of April 2018.

Tutorials Teacher (2018) *MVC Architecture*. Retrieved from: http://www.tutorialsteacher.com/mvc/mvc-architecture on 13 of April 2018.

Brain, M. (2017) *How Web Servers Work*. Retrieved from: https://computer.howstuffworks.com/web-server.htm on 30 of November 2018.

[Rouse](http://www.techtarget.com/contributor/Margaret-Rouse) M. (2017) *Web server*. Retrieved from: http://whatis.techtarget.com/definition/Web-server on 30 of November 2017.

Apache Friends (2007) *About.* Retrieved from: <https://www.apachefriends.org/about.html> on 30 of November 2017.

Microsoft (2017) *MVC Overview*. Retrieved from: https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview on 13 of April 2018.

*Techterms (2017) PHP*. Retrieved from: <https://techterms.com/definition/php> on 29 of November 2017.

*W3Schools (2017) PHP 5 Introduction*. Retrieved from: <https://www.w3schools.com/php/php_intro.asp> on 23 of November 2017.

Segue Technology (2013) *What is Ajax and Where is it Used in Technology*. Retrieved from: <https://www.seguetech.com/ajax-technology/> on 20 of November 2017.

*W3Schools (2017) Ajax Introduction. Retrieved from:* https://www.w3schools.com/xml/ajax\_intro.asp on 23 of November 2017.

Chraiet (2018) *About Ajax*. Retrieved from: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About\_JavaScript on 15 of November 2017.

*W3Schools (2017) JavaScript Tutorial*. Retrieved from: https://www.w3schools.com/js/ on 12 of November 2017.

Rouse, M. (2017) *Application Program Interface*. Retrieved from: http://searchmicroservices.techtarget.com/definition/application-program-interface-API on 20 of November 2017.

*Google* (2017) Maps JavaScript API. Retrieved from: <https://developers.google.com/maps/documentation/javascript/tutorial> on 18 of November 2017.

*Facebook (2018) Login for the Web with the JavaScript SDK*. Retrieved from: https://developers.facebook.com/docs/facebook-login/web on 15 of February 2018.

*MySQL(2017) History of MySQL*. Retrieved from: https://dev.mysql.com/doc/refman/5.7/en/history.html on 26 of October 2017.

*MySQL (2017) What is MySQL*. Retrieved from: https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html on 26 of October 2017.

*Framework7 (2017) About Framework 7*. Retrieved from: https://www.tutorialspoint.com/framework7/framework7\_overview.htm on 26 of October 2017.

Prescott, S. (2018) *The top 10 web application security risks*. Retrieved from: https://www.business.att.com/learn/operational-effectiveness/the-top-10-web-application-security-risks.html on 15 of April 2018.

OWASP (2017) *OWASP Top 10 – The Ten Most Critical Web Application Security Risks*. Retrieved from https://www.owasp.org/index.php/Category:OWASP\_Top\_Ten\_Project on 15 of April 2018.

Clever Techie (2017) *Create a Login System using PHP and MYSQL Database*. Retrieved from: <https://clevertechie.com/php/20/login-system-php-mysql-database> on 15 of January 2018.

Bootstrap (2018) *SB Admin 2*. Retrieved from: <https://startbootstrap.com/template-overviews/sb-admin-2/> on the 14 of February 2018.

Independent (2016) Ireland's largest pet ownership study. Retrieved from: <https://www.independent.ie/life/family/irelands-largest-pet-ownership-study-heres-all-the-facts-you-need-to-know-34505427.html> on the 12 of January 2018.