

The VetClinic

Product documentation

- Leo Carvalho Ramos Antunes
- Xialong Li
- Arshad Najeeb Shaik
- Hanan Younes

Table of Contents

Objective of the VetClinic Software	2
Detailed Requirements	2
User Interface Design.....	2
Input	2
List	2
Save and Retrieve Data	2
Search	2
Code Design Pattern.....	3
View	3
ViewModel	3
Model	4
Core	5
Screenshots	5
References	8

1. Objective of the VetClinic Software:

The Vet Clinic software is a Windows Presentation Foundation product that aims to help veterinarian clinics with their daily appointment bookings and patients' registration. The product allows any clinic to register different kinds of animals based on their own preferences. The application offers book an appointment functionality; appointments can be either at the clinic or at the client's house depending on the animal needing help. The product offers a selection of services that can be either therapeutic, surgery, or vaccination. In addition, the software provides search feature that allows intended users to search for appointments for a specific day or any appointment for a specific animal.

2. Detailed Requirements:

- **User Interface Design (Main Window):**

The user interface of the VetClinic application has a main window that is divided into four sections. These sections provide different functionalities as per the project requirements.

- Input:** the VetClinic application offers data input at the top left-side of the main window with two options, **Patient Menu** and **Appointment**. The **Patient Menu** option allows end users to enter new pet-owner information by selecting the **Add** button or edit/delete existing patient from the patient list by selecting the **Manage** button. Meanwhile, the **Appointment** option creates a new appointment for any of the existing patients with their intended service and at their convenient place. All data input options open as a new window or a dialogue box on top of the main window.
- List:** this section shows a list of available appointments on the right-hand side of the main window. A data grid is used to display information that has been given through the different input options.
- Save and Retrieve Data:** users can load, save, and delete data from patient and appointment list by selecting **Load**, **Save** or **Clear** buttons from the **File** dropdown menu. Data is saved and retrieved from an XML file.
- Search:** The application enables users to search through the data currently displayed and saved in lists using a set of filters, including pet name, date range, type of service,

and type of animal. The search results represent only a subset of the full saved patients/appointments' lists.

- [Code Design Pattern \[1\]](#):

The VetClinic application is designed to take full advantage of the Model-View-ViewModel (MVVM) design pattern, which would result in more readable and clear code. The MVVM pattern is divided into three sections.

- i. **View:** This section includes all of the user interface elements of the application that the intended users would mainly with. A wide selection of input controls has been used to help users perform a variety of tasks related to patients and appointments, including but not limited to combo boxes, buttons, checkboxes, textboxes etc. All of these UI elements are placed in four windows that are all XAML files:
 1. **MainWindowView:** this is the application's home screen and main point of interaction for users.
 2. **AppointmentDialogView:** this view is a pop-up dialog that allows users to create a new appointment to one of the registered patients. Users can create new appointment by clicking on **Add** button under **Appointment** tab.
 3. **PatientDialogView:** this view is a pop-up dialog that offers adding a new pet/owner information to the patient list. Users can add a new patient by clicking on **Add** button under **Patient** tab.
 4. **PatientListDialogView:** this view is a pop-up dialog that offers modifying or deleting an existing patient information from the patient list. Users can edit or delete a patient by clicking on **Manage** button under **Patient** tab.
- ii. **ViewModel:** This section includes all of the objects that provide data, business logic, and functionality for each view in the view section listed above. The ViewModel is acting as a connection between the View, which faces users, and the Model, which models the business rules. Similarly, there are four view model classes that have a direct mapping with their prospective view through binding.

1. **MainWindowViewModel:** this is the main ViewModel that provides most of the data and functionality to the Vet Clinic's MainWindowView including saving, loading and clearing data from all lists.
 2. **AppointmentDialogViewModel:** this ViewModel is mapped to the AppointmentDialogView to provide appointment related data, and help users to create a new appointment to one of the registered patients based on available time slots, length of one time slot and place of service. The number of time slots available, the start of the first appointment, and the length of one slot are customizable.
 3. **PatientDialogViewModel:** this ViewModel is mapped to the PatientDialogView to facilitate the functionality of adding new patient/owner information to the patient list.
 4. **PatientListDialogViewModel:** this ViewModel is binded to PatientListDialogView that is mainly concerned about modifying or deleting an existing patient information from the patient list.
- iii. **Model:** This section includes all of the application's custom classes, enums, and all necessary generic collections. It is data centric Classes used are as follows:

1. **Person Class:** includes all pet owner properties.
2. **Pet Class:** includes all patient/pet related properties and to connect a pet to a specific person object (owner).
3. **Appointment Class:** includes all appointment properties.
4. **Schedule Class:** to create a schedule based on available appointments in the list.
5. **Clinic Class:** this class handles daily schedule based on the appointment list and registered patients based on the patients' list.

In addition to these main classes, the application makes use of other classes/enumerations that facilitate certain functionalities, which are customizable, as follows:

1. **AppointmentFilter Class:** to set and reset search filters.
2. **AppointmentPlace Enum:** to specify a list of possible appointment places.
3. **Gender Enum:** to specify a list of pets' genders.

4. **PetType Enum:** to specify a list of pet types accepted at the clinic.
5. **ServiceType Enum:** to specify a list of available services.

iv. **Core:** In addition to the MVVM pattern and sections explained above, the VetClinic application is offering a notification system (property changed notifications) that is achieved by implementing the **INotifyPropertyChanged Interface**. By doing so, binding clients are notified when a property value has changed.

- Screenshots:

- i. Main application window:

The screenshot shows the main application window of VetClinic. It features a sidebar on the left with filters and a main area with a table of appointments.

Filters:

- ☐ Patient name: [Text input]
- ☐ Date from: [Date picker: 2019-11-21]
- ☐ Date to: [Date picker: 2020-01-05]
- ☐ Type of animal: [Dropdown menu]
- ☐ Type of service: [Dropdown menu]
- [Reset] [Apply]

Appointments Table:

Patient	Type	Breed	Owner	Date	Time	Service	Place			
Flicker	Dog	Husky	Leo Antunes	2019-12-04	12:00:00	Therapeutic	Clinic	Edit	Delete	
Flicker	Dog	Husky	Leo Antunes	2019-12-05	10:00:00	Vaccination	Clinic	Edit	Delete	
Kitty	Cat	Siamese	John Smith	2019-12-20	16:00:00	Vaccination	ClientHouse	Edit	Delete	
Little	Cat	Unknown	Mark Harman	2019-12-28	09:00:00	Therapeutic	Clinic	Edit	Delete	
kat	Cat	cat	Hanan Younes	2019-12-07	13:00:00	Vaccination	Clinic	Edit	Delete	

ii. Add a new patient:

VetClinic

File Patient Appointment

Filters

☐ Patient name:

☐ Date from:

☐ Date to:

☐ Type of animal:

☐ Type of service:

Reset **Apply**

Appointments

Patient	Type	Breed	Owner	Date	Time	Service	Place		
Flicker	Dog	Husky	Leo Antunes	2019-12-04	12:00:00	Therapeutic	Clinic	Edit	Delete
Flicker	Dog	Husky	Leo Antunes	2019-12-05	10:00:00	Vaccination	Clinic	Edit	Delete
Kitty	Cat	Siamese	John Smith	2019-12-20	16:00:00	Vaccination	ClientHouse	Edit	Delete

Register Patient

Owner information

First name*:

Last name*:

Address:

City:

Province:

Postal code:

Phone*:

E-mail*:

Pet information

Type of pet*:

Name*:

Breed*:

Age*:

Gender*:

Weight(kg):

Length(m):

Height(m):

Cancel **OK**

iii. Manage an existing patient:

VetClinic

File Patient Appointment

Filters

☐ Patient name:

☐ Date from:

☐ Date to:

☐ Type of animal:

☐ Type of service:

Reset **Apply**

Appointments

Patient	Type	Breed	Owner	Date	Time	Service	Place		
Flicker	Dog	Husky	Leo Antunes	2019-12-04	12:00:00	Therapeutic	Clinic	Edit	Delete
Flicker	Dog	Husky	Leo Antunes	2019-12-05	10:00:00	Vaccination	Clinic	Edit	Delete
Kitty	Cat	Siamese	John Smith	2019-12-20	16:00:00	Vaccination	ClientHouse	Edit	Delete

Patient List

Patient	Breed	Age	Type	Gender	Owner		
Flicker	Husky	10	Dog	Male	Leo Antunes	Edit	Delete
Kitty	Siamese	12	Cat	Female	John Smith	Edit	Delete
Little	Unknown	5	Cat	Female	Mark Harman	Edit	Delete
kat	cat	2	Cat	Female	Hanan Younes	Edit	Delete

OK

iv. Create a new appointment:

The screenshot shows the 'VetClinic' application window with the 'Appointment' tab selected. On the left, there are filters for Patient name, Date from, Date to, Type of animal, and Type of service. The main area displays a table of appointments. A 'Create an Appointment' dialog box is open, showing fields for Patient*, Date*, Time*, Type of Service*, and Place*, with 'OK' and 'Cancel' buttons.

Patient	Type	Breed	Owner	Date	Time	Service	Place			
Flicker	Dog	Husky	Leo Antunes	2019-12-04	12:00:00	Therapeutic	Clinic	Edit	Delete	
Flicker	Dog	Husky	Leo Antunes	2019-12-05	10:00:00	Vaccination	Clinic	Edit	Delete	
						Vaccination	ClientHouse	Edit	Delete	
						Therapeutic	Clinic	Edit	Delete	
						Vaccination	Clinic	Edit	Delete	

Pet information

Patient*:

Date*:

Time*:

Type of Service*:

Place*:

v. Search the appointment list using the type of animal filter:

The screenshot shows the 'VetClinic' application window with the 'Appointment' tab selected. The 'Type of animal' filter is checked and set to 'Cat'. The main area displays a table of appointments filtered by cat.

Patient	Type	Breed	Owner	Date	Time	Service	Place			
Kitty	Cat	Siamese	John Smith	2019-12-20	16:00:00	Vaccination	ClientHouse	Edit	Delete	
Little	Cat	Unknown	Mark Harman	2019-12-28	09:00:00	Therapeutic	Clinic	Edit	Delete	
kat	Cat	cat	Hanan Younes	2019-12-07	13:00:00	Vaccination	Clinic	Edit	Delete	

References

1. Getting started with Model-View-ViewModel (MVVM) pattern using Windows Presentation Framework (WPF). Retrieved from <https://intellitect.com/getting-started-model-view-viewmodel-mvvm-pattern-using-windows-presentation-framework-wpf/> on Dec 4th, 2019.