Dogtor Web Application: Veterinary Scheduling and GateKeeping Application

Requirements Analysis Document

ENGG4000 FR03X
Tash Zaman, Andre Aucoin and Jordan Mitchell

Table of Contents

1. Introduction

1.1 Purpose of the System

This tool will allow users and office administrators at a veterinarian clinic to better book appointments for their pets. The biggest problem when booking a veterinary appointment for your pet is the long wait times and finding someone with authority and qualifications to perform the task. The system will allow a vet client to quickly book appointments for their pet and help the office administrator organize vet appointments more efficiently. The bookings will be customized to the type of appointments the clinic can perform.

1.2 Scope of the System

We will develop a web-based service to get the current system working. This web-based service will need to handle three types of users: staff members performing the appointments (veterinarians and veterinary technicians), vet clients & and office administrators. To accomplish this, we will need an efficient login and credential system to deal with various users. The next step is to allow the office administrators to perform multiple maintenance tasks such as editing types of appointments that can be booked, changing appointments & making changes to an existing account if necessary. The most significant component of the system will be the scheduler. The design needs to help the users identify the type of appointment that they are currently attempting to book. Appointment type will vary between emergencies; the doctor required meetings and the technician appointments. The system will identify to the user what kind of appointment will best suit their needs.

1.3 Objectives of the Project

This project aims to develop a prototype that meets the following goals.

- The system will help direct the user to the best type of appointment that meets the needs of their pet.
- Office administrators can edit and modify what type of appointment their clinics can perform at any given time.
- Office administrators can edit any type of appointment. They can schedule emergency appointments and reschedule affected appointments if an emergency were to occur.
- Veterinarians and technicians can see their current schedule and leave notes on user's pets' profiles regarding their profile and essential information that might be needed for an appointment.

1.4 Definitions, Acronyms, and Abbreviations

Word, acronym or abbreviation	Definition
Dogtor	The working title of the project.
Technician Data Repository	This data store holds all the information regarding the technician and what type of specialties they have.
User Data Repository	A data store that holds all user's information and contains a list of appointments that relates to them.
Veterinarian Data Repository	This data store holds all the information regarding doctors and identifies their current specialties in terms of what they can perform.
Pet Data Repository	A data store that holds the content of the animals, such as medical history, surgeries performed and any other information that might be relevant to the doctor.
Appointment Data Repository	This data store holds all the appointments that have been booked.
Appointment-Type Data Repository	This data store holds all the information in terms of the type of appointment that the clinic can perform and who they can perform.

Table 1: Definitions, acronyms and abbreviations used in the Dogtor project.

2. Current System

By reaching out to local veterinary clinic staff members, our team found out that their current system is a mix of clients calling in and the desk worker using various systems to book appointments. The staff member that we talked to explained that booking an appointment could be a headache since they must use several different software tools to book one appointment. While she did not give us the name of all the other software used, she explained that the system is very complex, tends to break, and requires maintenance often.

3. Proposed System

3.1 Overview

The Dogtor web application is designed to allow vet clients to book their own appointments online to ease the stress of booking appointments for vet clinics. The Dogtor web application will be used by clients to book appointments online, by office administrators to book appointments for clients and to book emergency appointments, as well as by veterinarians and veterinary technicians to check their schedules and to be able to request time off. The Dogtor team is focused on making the web application highly customizable for admins; as this is what will help set the Dogtor application apart from its competitors. For example, one of the customization options requested is to allow admin users to select what kind of appointments veterinary technicians can perform. The application will also allow veterinarians to specify their specialties. This will be shown to clients while booking an appointment to help optimize veterinary care delivered. The Dogtor application will also remind clients of their scheduled appointments, and will remind them to schedule basic appointments such as yearly checkups, vaccinations, and appointments of such nature.

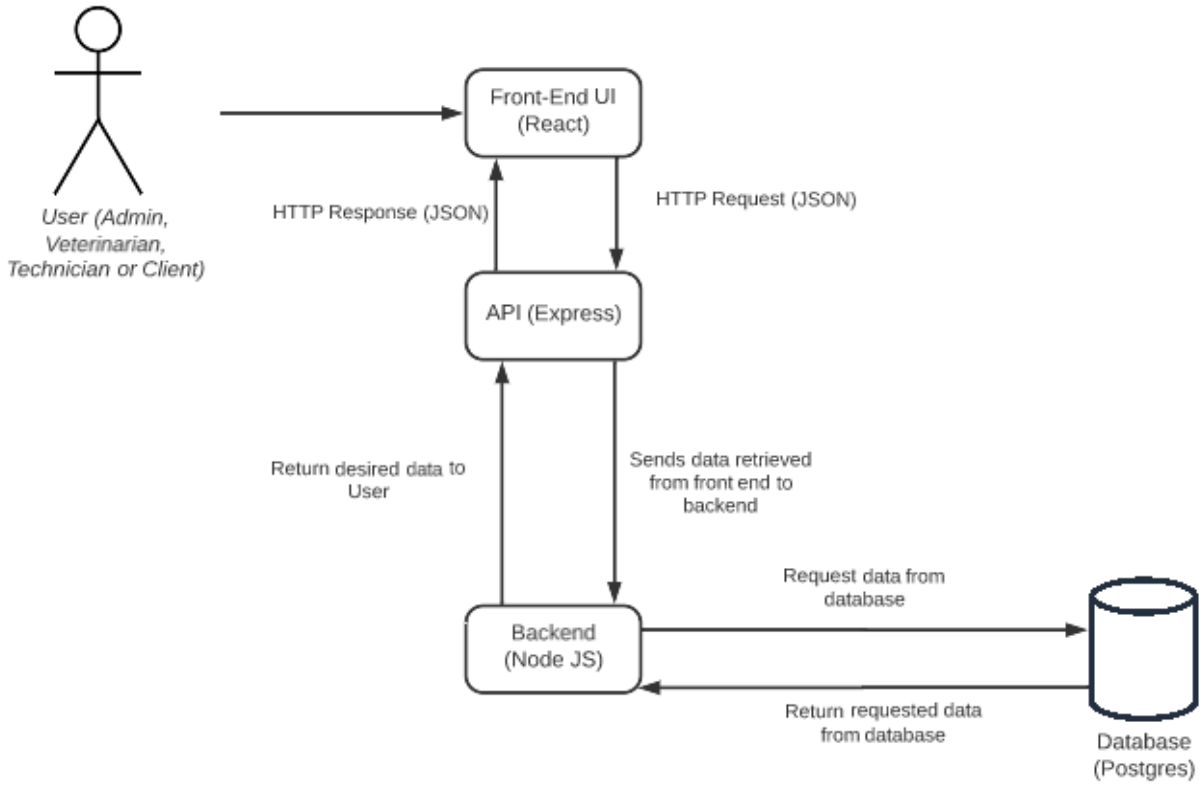

Figure 1: Dogtor application conceptual model.

3.2 Functional Requirements

The Dogtor web application is a tool that will allow veterinary clinics to book appointments more efficiently for their clients. The application will allow clients, veterinarians/technicians, and admins to login and create a profile. The profile consists of a set of attributes with values. The user can access their profile through the system at any time to make and save changes.

If a client has created a profile, they are able to access the scheduling page which would be precluded by a home page. The scheduling page would allow the client to select a date and time they wish to have the appointment. Once a time has been chosen, the clients are then taken to a questionnaire page where they can fill out set parameters; these parameters will allow the application to determine which veterinarian/technician would be most appropriate for their services. After a matching veterinarian/technician is found, the client will then be able to view their appointment on the scheduling page with some additional information.

If a veterinarian/technician has created a profile, they are able to view their client's profiles as well as scheduled appointments.

There is also some functionality that is limited to users with administrative access. An administrator can cancel and reschedule appointments made by clients. Administrators can also assign vet specializations to veterinarian/technicians which will help with client scheduling.

3.3 Non-Functional Requirements

3.3.1 Usability

Generally, the operations required from the users are limited, they will just need to be experienced in navigating through the different pages which are not very complex. Therefore, the usability of the system will be high. One of the key reasons why this system will be developed is to reduce appointment scheduling wait time and increase client satisfaction. The application must be designed to be user friendly to achieve this goal. The application will handle much of the backend logic to provide users with a pleasant experience. The application must also be accessible to a large demographic due to vet clients having a large group of clients.

3.3.2 Reliability

The reliability of the application is moderately important because it will improve the chances of successfully booking an appointment. Veterinarians are busy so by the time the client arrives at the vet clinic to manually book an appointment, their desired time slot may not be available. With all that said, this software is not meant to replace the manual booking process so clients will still be able to book appointments without it.

3.3.3 Implementation

The application will be web-based so users will be able to interact with it via web browser. User information will be stored in relational databases in the backend. A matchmaking algorithm will need to be developed for connecting clients to veterinarians/technicians.

3.3.4 Maintainability

Maintainability will be accounted for by prioritizing efficient coding style and providing extensive documentation throughout the development process. This will greatly aid in evolving the application in the future. The application will have a precise architecture that will minimize encountered bugs and defects throughout the lifecycle of the software.

3.3.5 Legal

The application will be licensed and distributed under whichever licenses best enable the success of the system's requirements. This may include licensing to host and secure the web application before deployment.

3.4 System Models

3.4.1 Use Case Model

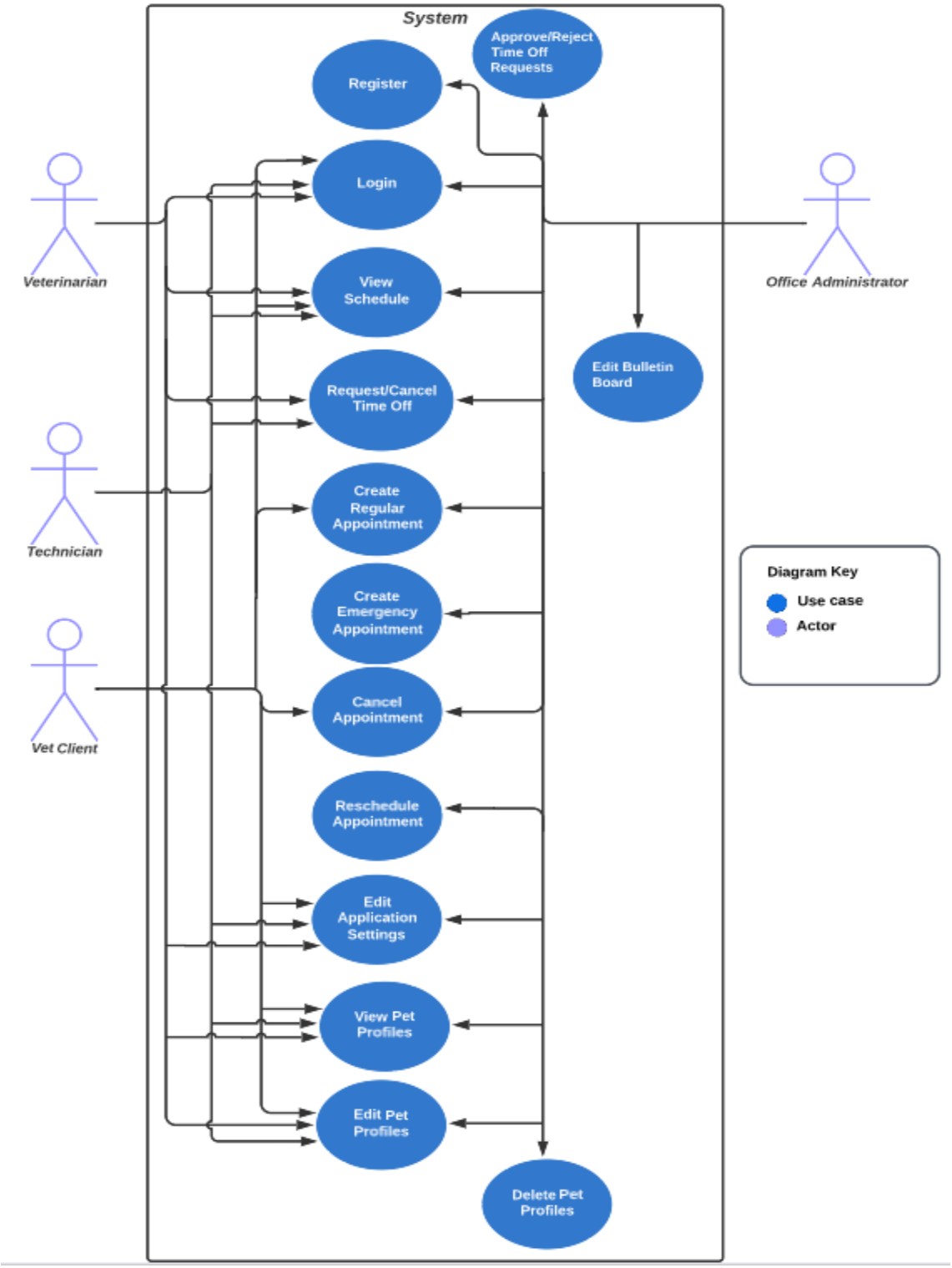

Figure 2: Use case model diagram.

Use case name	Register
Participating actors	Office Administrator
Flow of events	Main Flow 1. The Office Administrator visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome message and the Office Administrator clicks on the *Continue* button. 3. The Office Administrator inputs their username (email) and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen depending on the role. 5. The Office Administrator clicks on the *Register* tab. 6. The Office Administrator clicks on the *Register User* button. [A1: The Office administrator clicks on the *Register Pet* button] 7. The Office Administrator enters user information: first name, last name, email and role (Veterinarian, Technician, Vet Client or Admin). 8. The Office Administrator clicks the *Next* button. [A2: The Office administrator clicks on the *Back* button] 9. The Office Administrator has registered a new Admin role and is sent, by the Dogtor web application, to the home page. [A3: The Office Administrator registers a Vet Client] [A4: The Office Administrator registers a Veterinarian or a Technician] [E1: User role creation fails] 10. The Dogtor web application displays a notification that the new Admin has been created and can find their current password in an email sent to the email that was registered. Alternate Flows A1: The Office Administrator clicks on the *Register Pet* button. 1. The Office Administrator enters pet information: vet client email, pet name, pet type (dropdown), pet details (including weight, physical description, etc.). 2. The Office Administrator clicks the *Register* button. [A2: The Office Administrator clicks on the *Back* button] A2: The Office Administrator clicks on the *Back* button. 1. If the *Back* button has been clicked by the Office Administrator the Dogtor web application displays the home screen.

	A3: The Office Administrator registers a Vet Client. 1. The Dogtor web application displays a notification that the new Vet Client has been created and can find their current password in an email sent to the email that was registered. [E1: User role creation fails] A4: The Office Administrator registers a Veterinarian or a Technician. 1. The Dogtor web application displays a notification that the new Veterinarian or Technician has been created and can find their current password in an email sent to the email that was registered. [E1: User role creation fails] Exceptions E1: User role creation fails 1. If the email already exists in the system for a particular role, a notification at the top of the screen will indicate that there is an error. 2. The Dogtor web application will remain on the current registration page until the Office Administrator either clicks the *Back* button (returning to the home page) or enters a unique email address for a role that does not exist in the system.
Entry condition	The Office Administrator visits the Dogtor application where it is hosted on the web. The Office Administrator is logged into the system.
Exit condition	The newly created user has been sent an email containing their current password, the Dogtor web application returns to the home page and a notification is present at the top of the screen indicating that the new user has been created. If a pet is registered then the Dogtor web application returns to the home page and a notification is present at the top of the screen indicating that the new pet profile has been created. On failure, a notification at the top of the screen will indicate that there is an error and the Dogtor web application will remain on the current registration page until the Office Administrator either clicks the *Back* button (returning to the home page) or enters a unique email address for a role that does not exist in the system.

Quality requirements	The welcome and login pages each load within 10 seconds once the website has been visited and logged into. The registration page loads within 10 seconds of pressing the Register tab. Once the *Register* button has been pressed, the form is processed within 15 seconds and is responsive after processing has occurred.

Use case name	Login
Participating actors	User (Office Administrator, Veterinarian, Technician, Vet Client)
Flow of events	Main Flow 1. The User visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome message and the User clicks on the *Continue* button. 3. The User inputs their username (email) and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen depending on the role.
Entry condition	The User visits the Dogtor application where it is hosted on the web.
Exit condition	The User is provided certain access (based on their role) to the Dogtor web application's home page if the User's credentials exist in the system. If the User's credentials don't exist within the system, the User is asked to re-enter their credentials.
Quality requirements	The welcome and login pages each load within 10 seconds once the website has been visited and logged into.

Use case name	View Schedule
Participating actors	User (Office Administrator, Veterinarian, Technician, Vet Client)
Flow of events	Main Flow 1. The User visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome

	message and the User clicks on the *Continue* button. 3. The User inputs their username (email) and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen depending on the role. 5. The User clicks on the *Schedule* tab. 6. The Dogtor web application displays the User's weekly schedule. 7. The User can navigate through the weeks by clicking either the < button (to navigate to the previous week) or the > button (to navigate to the next week).
Entry condition	The User is logged into the system.
Exit condition	The User can view their schedule (or more in the case of the Office Administrator) when selecting the *Schedule* tab in the navigation bar.
Quality requirements	Once the *Schedule* button has been pressed, the schedule is displayed on the page within 15 seconds and is responsive after it has been displayed. The schedule should automatically display the current week within this time period.

Use case name	Create/Cancel Time Off
Participating actors	User (Office Administrator, Veterinarian, Technician)
Flow of events	Main Flow 1. The User visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome message and the User clicks on the *Continue* button. 3. The User inputs their username (email) and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen depending on the role. 5. The User clicks on the *Schedule* tab. 6. The Dogtor web application displays the current User's schedule for the current week on the Schedule page. 7. The User clicks on the *Edit Time Off Requests* button. 8. The Dogtor web application displays the current User's a page allowing the User to either create a time off request or cancel an already existing time off request.

	9. The User clicks on the *Create Time Off Request* tab. [A1: The User clicks on the *Cancel Time Off Request* tab] 10. The User enters how much time they'd like to take off from time and date to time and date. 11. The User clicks on the *Submit* button. [A2: If the Veterinarian or Technician has appointments scheduled during their time off] 12. The Dogtor web application displays the current User's schedule and displays a notification at the top of the page stating that the time off request has been sent. Alternate Flows A1: The User clicks on the *Cancel Time Off Request* tab. 1. The User selects (from a dropdown) the time off request that they would like to cancel. 2. The User clicks on the *Submit* button. 3. The Dogtor web application displays the current User's schedule and displays a notification at the top of the page stating that the time off request has been canceled. A2: If the Veterinarian or Technician has appointments scheduled during their time off. 1. If the User has appointments during the selected time off period a pop up will warn them with the message "You have appointments scheduled during this time. Are you sure that you want to proceed?" with two buttons at the bottom - *Continue* and *Cancel*. 2a. The User clicks the *Continue* button, the Dogtor web application displays the current User's schedule and displays a notification at the top of the page stating that the time off request has been sent. 3a. If the time off request is approved by the Admin, the Admin will receive a notification with the name of the User and the day of the appointments that must be rescheduled by the Admin. 2b. If the User clicks the *Cancel* button, the Dogtor web application displays the current User's time off request so that it can be edited by the User.
Entry condition	The User is logged into the system. The User desires to take time off.
Exit condition	The Dogtor web application displays the current User's schedule and displays a notification at the top of the page stating that the time off request has been sent or the User remains on the Create/Cancel Time Off page to edit their

	request. The time off request is added to the list of time off requests on the Review Time Off Requests page (which can only be accessed by an Admin User).
Quality requirements	The creation/deletion of a time off request must be done in a reasonable amount of time depending on the size of the request, how many appointments are to be rescheduled by the admin upon approval of the request and the connection of the client.

Use case name	Create Regular Appointment
Participating actors	User (Office Administrator, Vet Client)
Flow of events	Main Flow 1. The User visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome message and the User clicks on the *Continue* button. 3. The User inputs their username (email) and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen depending on the role. 5. The User clicks on the *Schedule* tab. 6. The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page. 7. The User clicks on the *Edit* button. [A1: The User clicks on the *Book Appointment* button] 8. The Dogtor web application displays the Edit Schedule page. 9. The User clicks on the *Create Appointment* tab. 10. The User enters appointment information: client first name, client last name, pet name (dropdown corresponding to the client), appointment type (dropdown), number of Technicians required and a Veterinarian (if required). Veterinarians who are specialized in the appointment type or the pet type will be starred (recommended). The approximate appointment time is displayed. 11. The User clicks on the *Submit* button. [A2: The User clicks on the *Cancel* button] 12. The Dogtor web application displays the Schedule of

	the Veterinarian selected. [A3: A vet is not required to conduct the appointment] 13. The User enters their desired start time by entering a time and date. 14. The User clicks on the *Submit* button. 15. The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page and a success notification is shown at the top of the page. [E1: Appointment cannot be booked - not enough staff] Alternate Flows A1: The User clicks on the *Book Appointment* button. 1. The Dogtor web application displays a warning message notifying the User that emergency appointments should be booked over the phone by an Admin. 2. The User clicks the *Proceed* button. [A2: The User clicks on the *Cancel* button] 3. The User enters appointment information: pet name (dropdown corresponding to the client), appointment type (dropdown) and a Veterinarian (if required). Veterinarians who are specialized in the appointment type or the pet type will be starred (recommended). The approximate appointment time is displayed. 4. The User clicks on the *Submit* button. [A2: The User clicks on the *Cancel* button] 5. The Dogtor web application displays the Schedule of the Veterinarian selected. [A3: A vet is not required to conduct the appointment] 6. The User enters their desired start time by entering a time and date. 7. The User clicks on the *Submit* button. 8. The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page and a success notification is shown at the top of the page. [E1: Appointment cannot be booked - not enough staff] A2: The User clicks on the *Cancel* button. 1. The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page and a success notification is shown at the top of the page. A3: A vet is not required to conduct the appointment 1. The Dogtor web application displays the Schedule of

	the User. 2. The User enters their desired start time by entering a time and date. 3. The User clicks on the *Submit* button. 4. The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page and a success notification is shown at the top of the page. [E1: Appointment cannot be booked - not enough staff] Exceptions E1: Appointment cannot be booked - not enough staff. 1. On failure, there will be a notification letting the User know that there aren't enough Technicians available at the date/time specified. 2. The User will stay on the page and can submit an appointment at a new time/date 3. Once the appointment has been submitted successfully, the client will be redirected to their schedule and will receive a success notification.
Entry condition	The User is logged into the system. The User desires to schedule a regular appointment (not an emergency appointment).
Exit condition	The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page and a success notification is shown at the top of the page or the User stays on the page that allows them to book an appointment and an error notification is shown at the top of the page asking the User to choose a different time and date. The schedule has been updated to reflect the changes made.
Quality requirements	The creation of appointments must be done in a reasonable amount of time. Clients must be warned not to book emergency appointments as quickly as possible. The loading time for all pages should not pass 10 seconds unless heavy computation is involved (then 20 seconds should not be passed).

Use case name	Create Emergency Appointment

Participating actors	Office Administrator
Flow of events	Main Flow 1. The Office Administrator visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome message and the Office Administrator clicks on the *Continue* button. 3. The Office Administrator inputs their username (email) and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen depending on the role. 5. The Office Administrator clicks on the *Schedule* tab. 6. The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page. 7. The Office Administrator clicks on the *Edit* button. 8. The Dogtor web application displays the Edit Schedule page. 9. The Office Administrator clicks on the *Create Appointment* tab. 10. The Office Administrator enters appointment information: client first name, client last name, pet name (dropdown corresponding to the client), appointment type is set to emergency, number of Technicians required and a Veterinarian. Veterinarians who are specialized in the appointment type or the pet type will be starred (recommended). 11. The Office Administrator clicks on the *Submit* button. [A2: The Office Administrator clicks on the *Cancel* button] 12. The Dogtor web application displays the Schedule of the Veterinarian selected. 13. The Office Administrator enters the start time by entering a time and date. The Office Administrator also enters the estimated end time by entering a time and date. 14. The Office Administrator clicks on the *Submit* button. 15. The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page and a success notification is shown at the top of the page. The schedule has been updated to reflect the changes made. Alternate Flows A1: The Office Administrator clicks on the *Cancel* button. 1. The Dogtor web application displays the weekly

	schedule(s) that can be accessed by that role on the Schedule page and a success notification is shown at the top of the page.
Entry condition	The Office Administrator is logged into the system. The Office Administrator must schedule an emergency appointment which was either brought into the clinic or called in.
Exit condition	The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page and a success notification is shown at the top of the page. The schedule has been updated to reflect the changes made.
Quality requirements	The creation of appointments must be done in a reasonable amount of time (especially since emergencies are time sensitive). The loading time for all pages should not pass 10 seconds per page unless heavy computation is involved (then 20 seconds should not be passed).

Use case name	Cancel Appointment
Participating actors	User (Office Administrator, Vet Client)
Flow of events	Main Flow 1. The User visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome message and the User clicks on the *Continue* button. 3. The User inputs their username (email) and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen depending on the role. 5. The User clicks on the *Schedule* tab. 6. The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page. 7. The User clicks on the *Edit* button. [A1: The User clicks on the *Cancel Appointment* button] 8. The Dogtor web application displays the Edit Schedule page. 9. The User clicks on the *Cancel Appointment* tab. 10. The User enters appointment information: client first

	name, client last name, pet name (dropdown corresponding to the client), appointment date (time/date dropdown of existing appointments) and appointment type (dropdown). 11. The User clicks on the *Submit* button. [A1: The User clicks on the *Cancel* button] 12. The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page and a success notification is shown at the top of the page. Alternate Flows A1: The User clicks on the *Cancel* button. 1. The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page and a success notification is shown at the top of the page.
Entry condition	The User is logged into the system. The User desires to cancel an appointment. The User has appointments in the system.
Exit condition	The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page and a success notification is shown at the top of the page.
Quality requirements	All pages must load within 10 seconds once the website has been visited and logged into. The system must process cancel requests within 15 seconds. After the appointment(s) has been canceled it must be reflected within the affected schedules.

Use case name	Reschedule Appointment
Participating actors	Office Administrator
Flow of events	Main Flow 1. The Office Administrator visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome message and the Office Administrator clicks on the *Continue* button. 3. The Office Administrator inputs their username (email)

	and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen depending on the role. 5. The Office Administrator clicks on the *Schedule* tab. 6. The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page. 7. The Office Administrator clicks on the *Edit* button. 8. The Dogtor web application displays the Edit Schedule page. 9. The Office Administrator clicks on the *Reschedule Appointment* tab. 10. The Office Administrator enters appointment information: client first name, client last name, pet name (dropdown corresponding to the client), current appointment date and time (time/date dropdown of existing appointments) and appointment type (dropdown). 11. The Office Administrator clicks on the *Submit* button. [A1: The Office Administrator clicks on the *Cancel* button] This will then follow the main flow of the "Create Regular Appointment" use case mentioned above. Alternate Flows A1: The Office Administrator clicks on the *Cancel* button. 1. The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page and a success notification is shown at the top of the page.
Entry condition	The Office Administrator is logged into the system. The Office Administrator desires to reschedule an existing appointment. The appointment must exist for a pet under the client's profile.
Exit condition	The Dogtor web application displays the weekly schedule(s) that can be accessed by that role on the Schedule page and a success notification is shown at the top of the page. If there is not enough staff, the Office Administrator will remain on the Schedule Edit page and an error notification is shown at the top of the page.
Quality requirements	The creation of appointments must be done in a

	reasonable amount of time. The loading time for all pages should not pass 10 seconds per page unless heavy computation is involved (then 20 seconds should not be passed).

Use case name	Edit Application Settings
Participating actors	User (Office Administrator, Veterinarian, Technician, Vet Client)
Flow of events	Main Flow 1. The User visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome message and the User clicks on the *Continue* button. 3. The User inputs their username (email) and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen depending on the role. 5. The User clicks on the *Settings* tab. 6. The Dogtor web application displays the appropriate settings page depending on the role. 7. The User clicks on the *Edit* button to edit the settings that they desire to change. 8. The *Edit* button is replaced with two buttons in edit mode: the *Save* button and the *Cancel* button. 9. The User changes the settings then clicks on the *Save* button to save their changes. [A1: The User clicks on the *Cancel* button.] 10. The Dogtor web application remains on the Settings page and a success notification is shown at the top of the page. Alternate Flows A1: The User clicks on the *Cancel* button. 1. The Dogtor web application remains on the Settings page and the changes made to the settings are not saved.
Entry condition	The User is logged into the system. The User desires to change their existing settings.
Exit condition	The Dogtor web application remains on the Settings page and a success notification is shown at the top of the page (changes have been saved). If the changes have been

	canceled the settings that were changed will revert to their last saved values and there will not be a notification at the top of the page.
Quality requirements	The loading time for the Settings pages should not pass 10 seconds per page. The processing time for the Settings page, once new changes are saved, should not pass 10 seconds once the *Save* button has been pressed.

Use case name	View Pet Profiles
Participating actors	User (Office Administrator, Veterinarian, Technician, Vet Client)
Flow of events	Main Flow 1. The User visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome message and the User clicks on the *Continue* button. 3. The User inputs their username (email) and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen depending on the role. 5. The User clicks on the *Pet Profiles* tab. 6. The Dogtor web application displays a page requesting the client's first name and last name. [A1: Tab directs User directly to *Pet Profiles*] 7. The User inputs the client's information. 8. The User clicks on the *Next* button. 9. The Dogtor web application displays the Pet Profiles page which can be viewed by the User for the specified client. [E1: Client does not exist] Alternate Flows A1: Tab directs User directly to Pet Profiles. 1. Once the *Pet Profiles* tab is clicked by a Vet Client, the Dogtor web application displays their Pet Profiles page which can be viewed by the User. Exceptions E1: Client does not exist. 1. The Dogtor web application will remain on the page

	requesting the client's information. An error notification will appear at the top of the page letting the User know that the client could not be found within the system.
Entry condition	The User is logged into the system. The User desires to view a client's Pet Profiles. A Pet Profile exists for the client that is being searched for.
Exit condition	On success, the Dogtor web application displays the Pet Profiles page which can be viewed by the User for the specified client. On failure, the Dogtor web application will remain on the page requesting the client's information. An error notification will appear at the top of the page letting the User know that the client could not be found within the system.
Quality requirements	The loading time for the Pet Profiles pages should not pass 10 seconds per page. Searching for a client's Pet Profiles should not take any longer than 20 seconds.

Use case name	Edit Pet Profiles
Participating actors	User (Office Administrator, Veterinarian, Technician, Vet Client)
Flow of events	Main Flow 1. The User visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome message and the User clicks on the *Continue* button. 3. The User inputs their username (email) and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen depending on the role. 5. The User clicks on the *Pet Profiles* tab. 6. The Dogtor web application displays a page requesting the client's first name and last name. [A1: Tab directs User directly to *Pet Profiles*] 7. The User inputs the client's information. 8. The User clicks on the *Next* button. 9. The Dogtor web application displays the Pet Profiles

	page which can be viewed by the User for the specified client. [E1: Client does not exist] 10. The User clicks on the tab with the name of the pet who's profile they would like to view. 11. The Dogtor web application displays the desired pet's Pet Profile. 12. The User clicks the *Edit* button. 13. The *Edit* button is replaced with two buttons: a *Save* button and a *Cancel* button. 14. The User makes the desired changes to the Pet Profile. 15. The User clicks the *Save* button to save their changes. [A2: The User clicks on the *Cancel* button] 16. The Dogtor web application remains on the Pet Profiles page and a success notification is shown at the top of the page. Alternate Flows A1: Tab directs User directly to Pet Profiles. 1. Once the *Pet Profiles* tab is clicked by a Vet Client, the Dogtor web application displays their Pet Profiles page which can be viewed by the User. A2: The User clicks on the *Cancel* button. 1. The Dogtor web application remains on the Pet Profile page and the changes made to the profile are not saved. Exceptions E1: Client does not exist. 1. The Dogtor web application will remain on the page requesting the client's information. An error notification will appear at the top of the page letting the User know that the client could not be found within the system.
Entry condition	The User is logged into the system. The User desires to edit a client's Pet Profiles. A Pet Profile exists for the client that is being searched for.
Exit condition	On success, the Dogtor web application remains on the Pet Profiles page and a success notification is shown at the top of the page. On failure, the Dogtor web application will remain on the page requesting the client's information. An error notification will appear at the top of the page letting the User know that the client could not be found within the

	system.
Quality requirements	The loading time for the Pet Profiles pages should not pass 10 seconds per page. Searching for a client's Pet Profiles should not take any longer than 20 seconds. Saving changes made to a pet profile should take no longer than 20 seconds.

Use case name	Delete Pet Profiles
Participating actors	Office Administrator
Flow of events	Main Flow 1. The Office Administrator visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome message and the Office Administrator clicks on the *Continue* button. 3. The Office Administrator inputs their username (email) and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen depending on the role. 5. The Office Administrator clicks on the *Pet Profiles* tab. 6. The Dogtor web application displays a page requesting the client's first name and last name. [A1: Tab directs Office Administrator directly to *Pet Profiles*] 7. The Office Administrator inputs the client's information. 8. The Office Administrator clicks on the *Next* button. 9. The Dogtor web application displays the Pet Profiles page which can be viewed by the Office Administrator for the specified client. [E1: Client does not exist] 10. The Office Administrator clicks on the tab with the name of the pet who's profile they would like to view. 11. The Dogtor web application displays the desired pet's Pet Profile. 12. The Office Administrator clicks the *Delete* button. 13. The Dogtor web application displays a pop up asking the Office Administrator if they are sure that they'd like to delete the current Pet Profile. 14. The Office Administrator clicks the *Confirm* button. [The Office Administrator clicks on the *Cancel* button]

	15. The Dogtor web application remains on the Pet Profiles page (with the profile of the pet deleted) and a success notification is shown at the top of the page. Alternate Flows A1: Tab directs Office Administrator directly to Pet Profiles. 1. Once the *Pet Profiles* tab is clicked by a Vet Client, the Dogtor web application displays their Pet Profiles page which can be viewed by the Office Administrator. A2: The Office Administrator clicks on the *Cancel* button. 1. The Dogtor web application remains on the Pet Profile page and the changes made to the profile are not saved. Exceptions E1: Client does not exist. 1. The Dogtor web application will remain on the page requesting the client's information. An error notification will appear at the top of the page letting the Office Administrator know that the client could not be found within the system.
Entry condition	The Office Administrator is logged into the system. The Office Administrator desires to edit a client's Pet Profiles. A Pet Profile exists for the client that is being searched for.
Exit condition	On success, the Dogtor web application remains on the Pet Profiles page (with the profile of the pet deleted) and a success notification is shown at the top of the page. On failure, the Dogtor web application will remain on the page requesting the client's information. An error notification will appear at the top of the page letting the Office Administrator know that the client could not be found within the system.
Quality requirements	The loading time for the Pet Profiles pages should not pass 10 seconds per page. Searching for a client's Pet Profiles should not take any longer than 20 seconds. Deleting a pet profile should take no longer than 10 seconds.

Use case name	Approve/Reject Time Off Requests
Participating actors	Office Administrator
Flow of events	Main Flow 1. The Office Administrator visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome message and the Office Administrator clicks on the *Continue* button. 3. The Office Administrator inputs their username (email) and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen depending on the role. 5. The Office Administrator clicks on the *Schedule* tab. 6. The Dogtor web application displays the Office Administrator's Schedule page. 7. The Office Administrator clicks on the *Review Time Off Requests* tab. 8. The Dogtor web application displays the Office Administrator's Review Requests page. 9. The Office Administrator reviews the table containing the time off requests. 10. The Office Administrator clicks on the *Approve* button at the end of a table entry to approve a request. [A1: The Office Administrator clicks on the *Reject* button] 11. The *Approve* and *Reject* buttons are replaced with the text "Approved" and a notification is sent to the staff member (that requested the time off) letting them know that their request was rejected. Their schedule is updated accordingly. Alternate Flows A1: The Office Administrator clicks on the *Reject* button 1. The Office Administrator clicks on the *Reject* button at the end of a table entry to reject a request. 2. The *Approve* and *Reject* buttons are replaced with the text "Rejected" and a notification is sent to the staff member (that requested the time off) letting them know that their request was rejected. Their schedule is not updated.
Entry condition	The Office Administrator is logged into the system. The Office Administrator desires to review time off requests made by staff members. Time off requests exist within the system.

Exit condition	On success, the *Approve* and *Reject* buttons are replaced with the text “Approved” or “Rejected” and a notification is sent to the staff member (that requested the time off) letting them know that their request was reviewed.
Quality requirements	The loading time for the Review Requests page should not pass 10 seconds. Processing and sending notification to staff members whose requests have been should take no longer than 20 seconds.

Use case name	Edit Bulletin Board
Participating actors	Office Administrator
Flow of events	Main Flow 1. The Office Administrator visits the URL where the Dogtor web application is being hosted. 2. The Dogtor web application displays a welcome message and the Office Administrator clicks on the *Continue* button. 3. The Office Administrator inputs their username (email) and password and clicks on the *Login* button. 4. The Dogtor web application displays the appropriate home screen for the Admin role. 5. The Office Administrator clicks on the *Edit* button on the home page. 6. The *Edit* button is replaced with two buttons once clicked: the *Save* button to save their changes to the bulletin board and the *Cancel* button to revert the changes made before saving. 7. The Office Administrator makes changes to the bulletin board. 8. The Office Administrator clicks the *Save* button and their changes are saved to the bulletin board. [A1: The Office Administrator clicks the *Cancel* button] Alternative Flows A1: The Office Administrator clicks the *Cancel* button 1. The Office Administrator clicks the *Cancel* button and none of the changes made (once the *Edit* button was pressed) are saved to the bulletin board.
Entry condition	The Office Administrator visits the Dogtor application where it is hosted on the web.

Exit condition	All bulletin boards (each role contains the same welcome page but only the Admin can edit the bulletin board) reflect the changes that were made to it by the Office Administrator once they have clicked either the *Save* button or the *Exit* button.
Quality requirements	The welcome and login pages each load within 10 seconds once the website has been visited and logged into. Once changes have been made to the bulletin board the board is processed within 20 seconds and is responsive after processing has occurred. All changes made by the Admin are reflected in the bulletin boards for all user roles.

3.4.2 Object Model

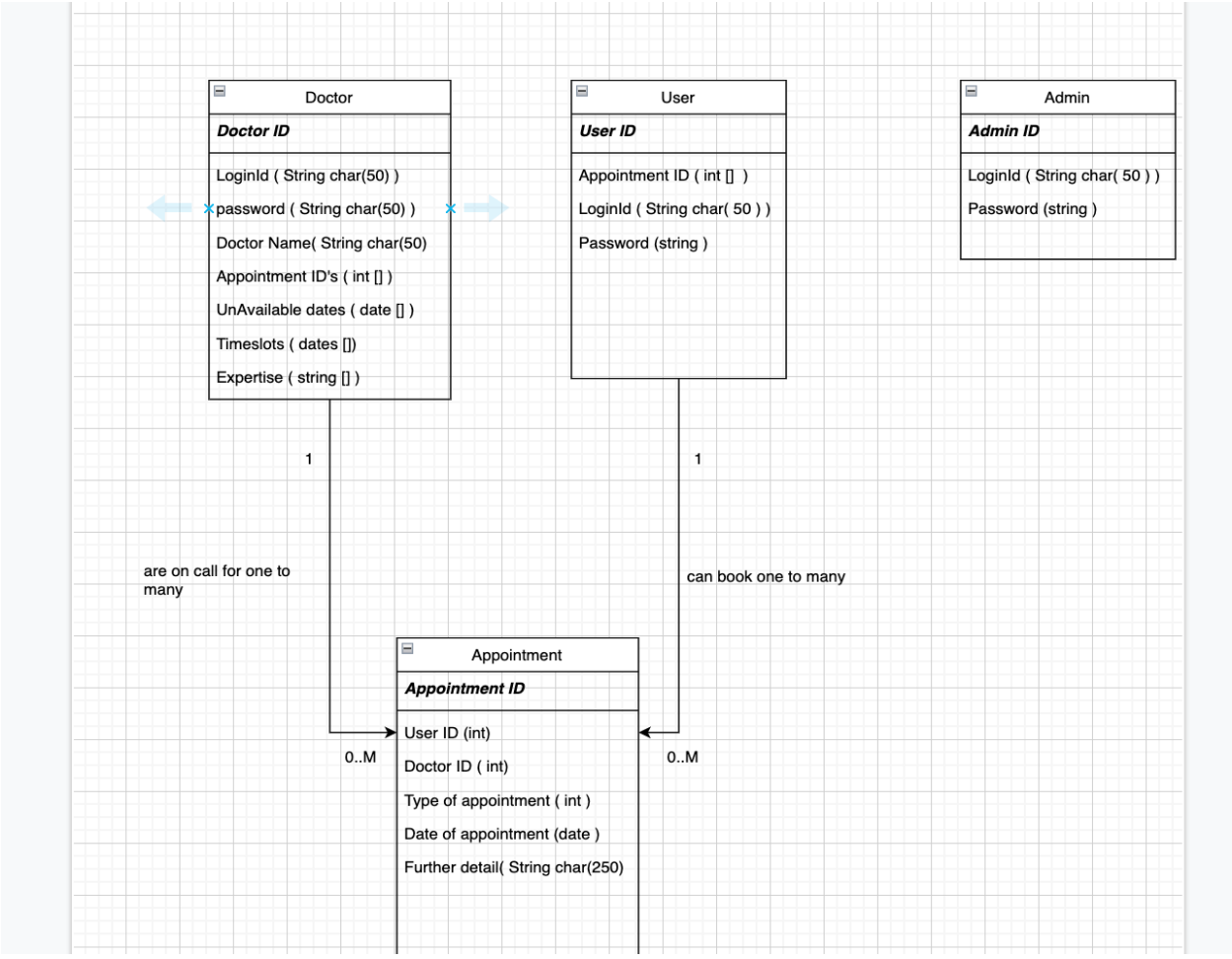

Figure 3: Class diagram.

3.4.3 Dynamic Model

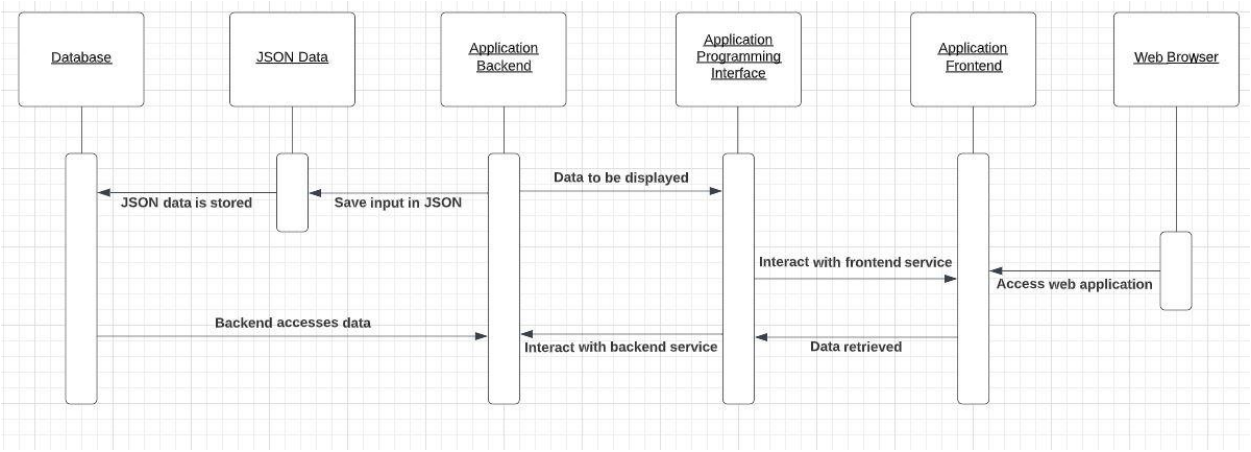

Figure 4: General sequence diagram.

This general sequence occurs whenever a user interacts with the system. The user interacts with the web browser running the Dogtor web application (specifically the Application Frontend). If information from the database is required for an interaction, an HTTP request (JSON) is sent from the Application Frontend to the Application Programming Interface (API). The work required for the request is then completed within the Application Backend. The Application Backend interacts with the database to send and receive values being stored that are required to complete the request. The Application Backend then returns the requested information to the API, which then returns it to the Application Frontend in the form of an HTTP response (JSON). The Dogtor web application's Frontend reflects the changes that were made by the user, and is ready to complete the next request in the same fashion.

3.4.4 User Interface

The User Interface can be viewed at the following link:
https://www.figma.com/file/ghE9zeqHKIiFqUY4FxWYfp/Dogtor-UI?node-id=0%3A1&t=HD7zC5APJ9G1xN4n-1