Spring CS 4624 Capstone Project

Final Report

NeuroVeTele

Miguel Lopez, Danny Yang, and Puchuan Song

Client: Richard Shinn

Instructor: Edward A. Fox

Virginia Tech

Blacksburg, VA 24061

May 6, 2024

Abstract

Considered one of anatomy's most complex structures, the nervous system manages the senses and bodily functions. For most animals, the nervous system allows them to see, hear, smell, taste, touch, and feel pain. The interplay of certain areas within the nervous system affects which parts of the animal are affected. Such is the case in veterinarian medicine, where the subject of neurolocalization is an important feature in maintaining the stable lives for pets. Neurolocalizations allow veterinarians to pinpoint any disorders regarding a pet's nervous system. Even for a relatively small animal hospital, the need for a neurolocalization is frequent enough that an efficient system is desirable for streamlining this process. Thus, this report presents an innovative mobile and desktop application named NeuroVeTele that can assist veterinarians to accurately diagnose disorders, given user inputs. The application has two main portions, which are the front-end and back-end. The front-end displays options for the user to choose, with the overall design being both interactive and easy to use. The options that the user chooses from the front-end will be the input needed for the back-end function. The back-end utilizes a weight system for the options the user chose to formulate the best possible neurolocalization given the inputs. This connection between the front-end and back-end is bridged with the usage of the Model-View-Controller format allowing for dynamic manipulations and updates. Ultimately, this application allows veterinarians to go paperless in their neurolocalization endeavors and gives them a helpful aid in finalizing a plan for the prognosis. The latest version includes the point system in the back-end for the behavior part, based on the data sheet sent by our client Dr. Richard Shinn. There is a full-featured basic user interface in the front-end that allow the users, who are veterinarians or veterinary students, to enter and adjust inputs while doing the test. There is also a beta version of a customizable point system that allows the user to adjust the point system by themselves.

Contents

Abstract **1**

List of Figures **5**

List of Tables **6**

1 Introduction **7**

1.1 Problem 7

1.2 Motivation 7

1.3 Objective 8

1.4 Team Roles 8

1.4.1 Danny Yang 8

1.4.2 Miguel Lopez 8

1.4.3 Puchuan Song 9

2 Design **10**

2.1 Front-end 10

2.1.1 Launch Page 10

2.1.2 Dashboard Page 10

2.1.3 General Categories Page 10

2.1.4 Cranial Nerves Page 11

2.1.5 Peripheral Nervous System (PNS) Page 11

2.1.6 Results Page 11

2.2 Back-end 12

2.2.1 Global Variables 12

2.2.2 MVC Architecture for Application Pages 12

2.2.3 Neurolocalization (nl) 13

2.2.4 Option 14

2.2.5 Calculation 14

3 Implementation **15**

3.1 Overview 15

3.2 Model-View-Controller (MVC) Architecture 16

4 Evaluation **17**

4.1 Experimental Setup 17

5 User's Manual **18**

5.1 Launch Page 18

5.2 Input Pages 18

5.3 Neurolocalization/Results Page 25

5.4 Video Demo 28

6 Developer's Manual **31**

6.1 Prerequisites 31

6.2 Applications 31

6.3 GitHub 32

6.4 Running in VSCode 33

6.5 Dashboard 33

7 Lessons Learned **35**

7.1 Encountered Problems 36

7.2 Timeline 38

7.3 Future Plans 38

8 Acknowledgements **41**

9 References **42**

10 Appendix **43**

List of Figures

1 Simplified Framework of NeuroVeTele 15

2 Launch Page 18

3 Proposed Pet Selection Main Page (left) and Current (right) 19

4 Sample Input Page Showing Progress Bar 20

5 Proposed Qualitative Sample Input Page (left) and Current (right) 21

6 Proposed Showcase of the Dropdown Bar (top) and Current (bottom) 22

7 Instance of Multiple Options 23

8 Instance of Conflicting Options 23

9 Sample Completed Input Page 24

10 Neurolocalization/Results Page 25

11 Simplified Results Page 26

12 Wireframe of the Demo Sequence 29

13 Options of Devices 33

14 Sample Dashboard 34

15 Figma Implementation 43

16 Simplified UML of NeuroVeTele. On the left are functions. On the right are details for the 6 pages. 44

List of Tables

1 Sample PDF Table of a Normal Neurolocalization 27

2 Timetable of the Video Demo 28

1 Introduction

The NeuroVeTele app is designed to aid veterinarians in diagnosing and treating pets with neurological disorders, which may affect the brain, spinal cord, muscles, and nerves (Rylander, 2022). Traditionally, neurolocalizations are conducted manually, involving comprehensive assessments of various bodily regions such as the head (cranial nerves), gait, neck, front legs, torso, hind legs, anus, and tail (Schubert, 2018). NeuroVeTele facilitates a transition to a paperless process, aiming to streamline these extensive evaluations.

1.1 Problem

Diagnosing neurological conditions presents significant challenges due to the complexity of the disorders and the subtlety of symptoms (Bush, 2019). For instance, minor deviations in an animal's behavior, such as a slight head tilt, could indicate severe underlying conditions like seizures or forebrain issues (Nagendran et al., 2024). Inaccuracies during the diagnostic phase can lead to misdiagnoses, potentially resulting in inappropriate treatments that could harm the animal (Berg, 1989). In extreme cases, such errors might even lead to the unnecessary euthanasia of pets.

1.2 Motivation

Dr. Shinn, a veterinary neurologist at Virginia Tech, recognized the need for a tool that mitigates these diagnostic challenges. The NeuroVeTele app is developed to support general veterinarians by providing access to a comprehensive knowledge base and algorithmic guidance for neurolocalization, aiming to bridge existing knowledge gaps and enhance diagnostic accuracy.

1.3 Objective

The primary objective of the NeuroVeTele app is to simplify and enhance the accuracy of neurolocalization in veterinary neurology. The app provides immediate, reliable guidance to non-specialist veterinarians, enabling more precise neurolocalizations and facilitating appropriate treatment plans. This tool is intended to augment the quality of veterinary care and contribute to the professional development of veterinarians. However, it is crucial to note that the app is designed to assist rather than replace veterinarians. We recommend that an experienced veterinary neurologist review any results generated by NeuroVeTele to confirm their validity.

1.4 Team Roles

1.4.1 Danny Yang

Danny Yang plays a key role in the NeuroVeTele app development team, focusing on back-end development. He is responsible for creating the algorithm that can process inputs of options and come up with an accurate output. As the secondary contact, Danny provides essential support in communication and coordination tasks. Additionally, he is tasked with submitting the app's documentation and updates to VTechWorks, ensuring that all relevant information is accessible and archived properly.

1.4.2 Miguel Lopez

Miguel Lopez is the main front-end developer and designer for the NeuroVeTele app. His responsibilities include developing the user interface, ensuring that it is intuitive, approachable, and aesthetically pleasing. As the main designer, Miguel is in charge of the app's visual aspects, focusing on creating an engaging and efficient user experience. He is also responsible for app compatibility, ensuring that the app functions seamlessly across different devices and platforms, providing a consistent user experience regardless of the user's hardware.

1.4.3 Puchuan Song

Puchuan Song is responsible for the interaction between back-end and front-end; he also develops the APIs for the application. He is responsible for scheduling meetings and keeping the project on schedule, as well as full-stack development. As the primary contact with the veterinarian and instructor, he is responsible for analyzing the client's needs and reporting the team's progress to the client and professor.

2 Design

2.1 Front-end

The front-end design of the NeuroVeTele app is structured to provide a straightforward interface that is easy to navigate across six distinct pages, each dedicated to gathering specific types of neurological examination data.

2.1.1 Launch Page

The first page that users are presented with when they open the NeuroVeTele app, has two buttons “DOG” and “Go to Dashboard”. The dog button navigates users to the neurolocalization test for a dog. The dashboard button navigates users to the dashboard page.

2.1.2 Dashboard Page

This page is exclusively for administrative purposes. It features a simple interface where admins can adjust the weights of any of the options through editable text boxes. Initially, these boxes are populated with the default weights, which are the latest values provided by Dr. Shinn. To modify a weight, an admin needs to clear the existing number and enter the new value. After making the desired changes, admins can save all updates by clicking the save icon located in the top bar.

2.1.3 General Categories Page

This initial page captures broad neurological categories such as behavior, mentation, posture, gait, and involuntary movements. The interface includes multiple-select drop-down menus with search functionalities, allowing quick and efficient selection of relevant options.

2.1.4 Cranial Nerves Page

This page is dedicated to assessing the function of cranial nerves. It uses interactive elements like sliders and increment/decrement buttons to adjust values for each cranial nerve assessment. A swiping feature enables users to easily switch between assessing left-side and right-side cranial nerves, such as comparing the functionality of the left and right eyes.

2.1.5 Peripheral Nervous System (PNS) Page

This page collects detailed information on the limbs, spine, and muscles. It employs a similar interface to the General Categories Page, featuring drop-down menus and increment/decrement buttons with swiping functionality. The drop-down menus have search capabilities and include error alerts for incompatible selections to ensure precise data entry.

2.1.6 Results Page

After all necessary data has been provided by the user, this final page will display the top three potential neurolocalizations, labeled as "Most Likely," "Could Also Be," and "Or Even." This page serves as the culmination of the app's diagnostic assistance, providing a clear and concise summary of the findings and suggesting next steps based on the entered examination data.

2.2 Back-end

The back-end of the NeuroVeTele app is developed using Dart, utilizing object-oriented programming (OOP) principles to effectively manage and process the neurological localization (nl) data. At the heart of the architecture is a point-based system, where each option is assigned a unique weight. A Model-View-Controller (MVC) architecture is integral to this back-end, ensuring that data entered by users or administrators is consistently saved and communicated with the point-based data structure. This report provides an overview of the fundamental aspects of the back-end rather than a file-by-file explanation. Below is a detailed description of the back-end architecture, emphasizing data handling and class structures.

2.2.1 Global Variables

The back-end architecture of the NeuroVeTele app incorporates crucial global variables that store default values for each possible option and their respective neurolocalizations, as provided by Dr. Shinn. These variables are fundamental for the initial setup of the neurolocalization process, ensuring consistency and accessibility of data throughout the application. Initialized at launch, these variables are central to the application's functionality, linked directly to each user interface option. This integration ensures that any adjustments in the Dashboard or assessments in the General, Cranial Nerve, or PNS Pages are based on accurate and consistent baseline values. The robust and reliable neurolocalization calculations provided by these global variables enhance the application's diagnostic capabilities, leveraging the foundational data to maintain the integrity and effectiveness of all related computations.

2.2.2 MVC Architecture for Application Pages

The application utilizes a Model-View-Controller (MVC) architecture to structure the interaction between data processing and user interface across different pages: General Categories, Cranial Nerves, Peripheral Nervous System (PNS), and Dashboard.

The General Categories and PNS Pages are designed to capture and process user selections from dropdown menus in their respective views. The selection process is managed through an update method defined in the controller, which then saves the selections into the model as a list, making them available for further processing by other class files. The General Categories Page is fully implemented. The PNS page is mostly implemented, as it contains the main categories and their suboptions related to the Peripheral Nervous System.

The Cranial Nerves Page, designed to facilitate detailed assessments of cranial nerve functions. Like the other pages, it will utilize MVC principles to manage interactions.

The Dashboard Page uses text boxes instead of dropdown menus in its view component, where each option has its defined text box. When an admin modifies the weights of options, the changes are temporarily stored. The official update to the global variables, which affect the neurolocalizations across the application, occurs only when the admin presses the save button, triggering the 'save all' method. This page is fully implemented.

This section of the application leverages the MVC architecture to ensure that changes in the user interface trigger updates in the model efficiently, maintaining the integrity and responsiveness of the system.

2.2.3 Neurolocalization (nl)

Neurolocalizations, such as L4-S3, T3-L3, and Normal, are fundamental data elements in the back-end. Each neurolocalization is associated with an integer value, enabling quantitative analysis and comparison. These values are crucial for determining the three most likely neurolocalizations based on the input data.

2.2.4 Option

Within the app, options refer to selectable choices derived from the MVCs of the General Categories Page, Cranial Nerves Page, PNS Page, and Dashboard Page. An abstract class, *Option*, includes operations such as `add`, `sub`, `isGuaranteed`, and `isNever`, all of which modify the neurolocalization (nl) weights. By default, the values for each option are set to the global variables. Accompanying this abstract class is an abstract factory responsible for creating each option. For every option, there are initializer methods. When the neurolocalization is computed, every method is initialized alongside all operations (add, sub, etc.) within each method, influencing the weights of potential nls based on the selected options.

2.2.5 Calculation

All the neurolocalizations are stored in a list, where their values are sorted using the built-in Dart sort method. After sorting, the top three neurolocalizations are selected and returned for further processing. These top neurolocalizations are then displayed on the Results Page, providing a clear and concise summary for diagnostic support.

3 Implementation

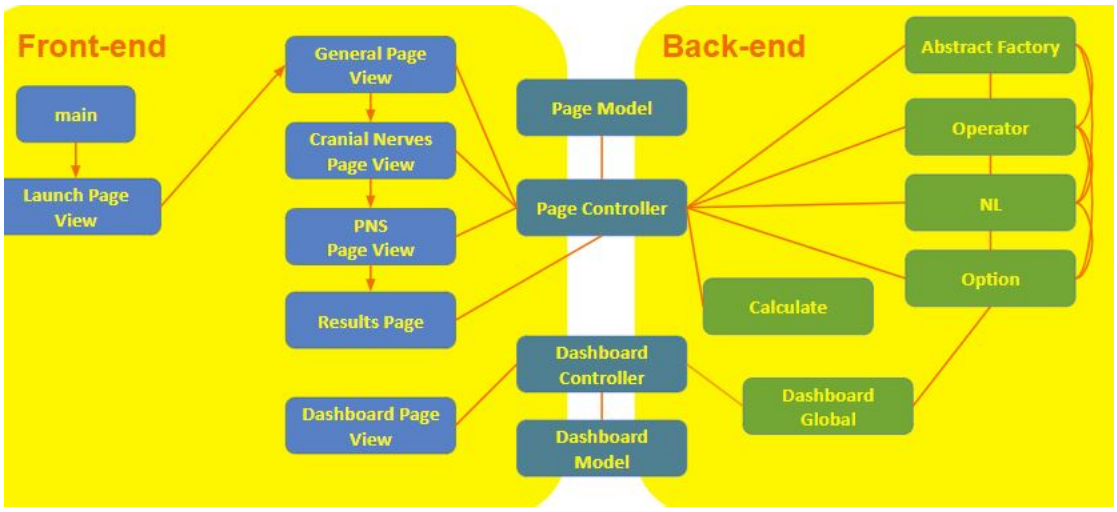


Figure 1: Simplified Framework of NeuroVeTele

3.1 Overview

The entire structure behind NeuroVeTele revolves around the Model-View-Controller (MVC) architecture in connecting the front-end and back-end. Both the front-end and back-end were developed using Flutter, which is an open-source UI software development kit created by Google (Lougheed, 2024). One of the main advantages of Flutter is that it allows for code to be used for multiple platforms based on one central language. In using Flutter, the entirety of NeuroVeTele was written in Dart, a language with similarities to Java (Lougheed, 2024). As mentioned in the Front-end subsection in the Design section, the front-end contains all the pages needed for the general sequence of a neurolocalization. The front-end utilizes the material library found in the Flutter API (flutter/material.dart) for the creation of widgets such as buttons, dropdown menus for the General and PNS Pages, number pickers for the Cranial Nerves Page, and textboxes in the Dashboard. The Dashboard specifically sets up the global variables used as the initial default values for the points system. These variables can be accessed by the back-end, where they are implemented into the multitude of classes used to establish the points for each neurological option. When the program is run, Flutter

is able to create different formats for iOS, Android, and web applications by translating Dart code into the respective codes of each platform.

3.2 Model-View-Controller (MVC) Architecture

As seen in Figure 1, the front-end and back-end is bridged by a Model-View-Controller (MVC) architecture, specifically by the Model and Controller. The main role of the View is to create the visual components seen by the user and to act as the conduit for displaying input and output. When the user operates the NeuroVeTele, they activate the Controller, which utilizes the input that was submitted through the View to make specific commands in code either for the View or for the Model. The Model stores the central component of NeuroVeTele's front-end, the inputs the user selects for a particular neurological category. This is done so by having multiple arrays that store these options. When all the categories are filled with options, the MVC system is able to notice this and send the information to the back-end through the Controller and Model. Once the back-end is able to calculate the top few neurolocalizations, this information is integrated into the MVC system through the Controller, which assists in giving the results to the View for showcase. This type of architecture was selected to provide the user with a more immediate interaction, as the options can be dynamically showcased to the user when they perform changes to their selections.

4 Evaluation

4.1 Experimental Setup

To prove the validity of NeuroVeTele's accuracy for neurolocalization, NeuroVeTele is set to undergo at least 100 trials where edge cases are involved and emphasized. Especially with the extreme number of the combinations involved in neurolocalizations, the points system needs to be tested more to see if accuracy is sustained with this complexity. No major testing has been done, but the experimental setup has been prepared. User experiments are to be done under our client Dr. Richard Shinn's supervision, with his veterinary students acting as the experimental groups in this study. Dr. Shinn plans to involve two experimental groups (Group 1 and 2). Groups are randomly assigned, and have more than 10 students in each group. In Dr. Shinn's experimentation setup, a session of testing takes two weeks. In the first week, Group 1 uses NeuroVeTele and Group 2 does not, and instead utilizes traditional neurolocalization techniques. After the first week, the two groups switch their usage of NeuroVeTele. After these two weeks, the results are collected from students regarding NeuroVeTele's accuracy. Besides testing NeuroVeTele's accuracy, its front-end design is also commented on by the same experimental groups for its usability. These sessions repeat until NeuroVeTele yields high accuracy in neurolocalizations (above 95%) and receives satisfactory comments from veterinary students.

5 User's Manual

5.1 Launch Page

When the user first enters the application, they are greeted by the launch page (Figure 2). The launch page showcases the NeuroVeTele logo along with its name. Currently, this launch page transitions to the pet selection page (Figure 2) which begins the input pages.

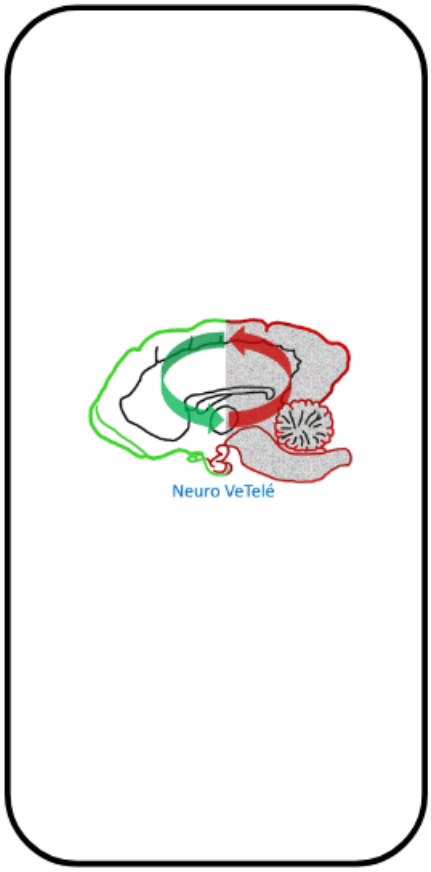


Figure 2: Launch Page

5.2 Input Pages

The pet selection main page (Figure 3) serves as the first input page. This page simply asks the user to select between dog or cat as the patient. Currently, as the application only has data regarding dogs, the app only showcases the dog option.

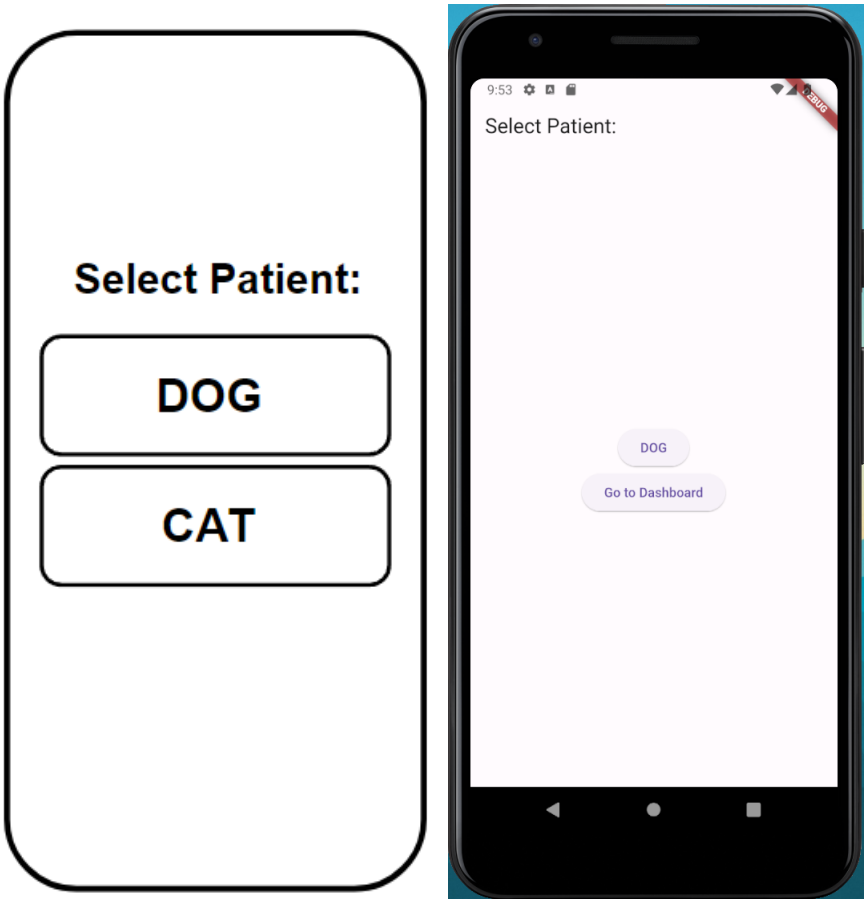


Figure 3: Proposed Pet Selection Main Page (left) and Current (right)

The input page is mainly divided into three parts, as seen in the progress bar (Figure 4). The first part, indicated by a cat or dog face, addresses the behavior and other qualitative parts of the patients. The second part, indicated by the brain-stem icon, is mainly about the cranial nerves and contains mostly quantitative options. The third part, indicted by the paw, pertains to options concerning the peripheral nervous system, and is both quantitative and qualitative. When the user finishes a part, it fills up the progress bar. Additionally, the user can click on the icons after they have completed the current input page if they need to go back and make alterations to their options. The sample page of Figure 4 also shows the setup for the user to select quantitative options for cases where the left and right can have separate inputs. The arrows allow the user to pick the options for that specific side. The letters "L" and "R" under it refer to the other page that the user can get to for inputs of

a different side. When the user does not swipe to the right, the application will vibrate to notify of this decision.

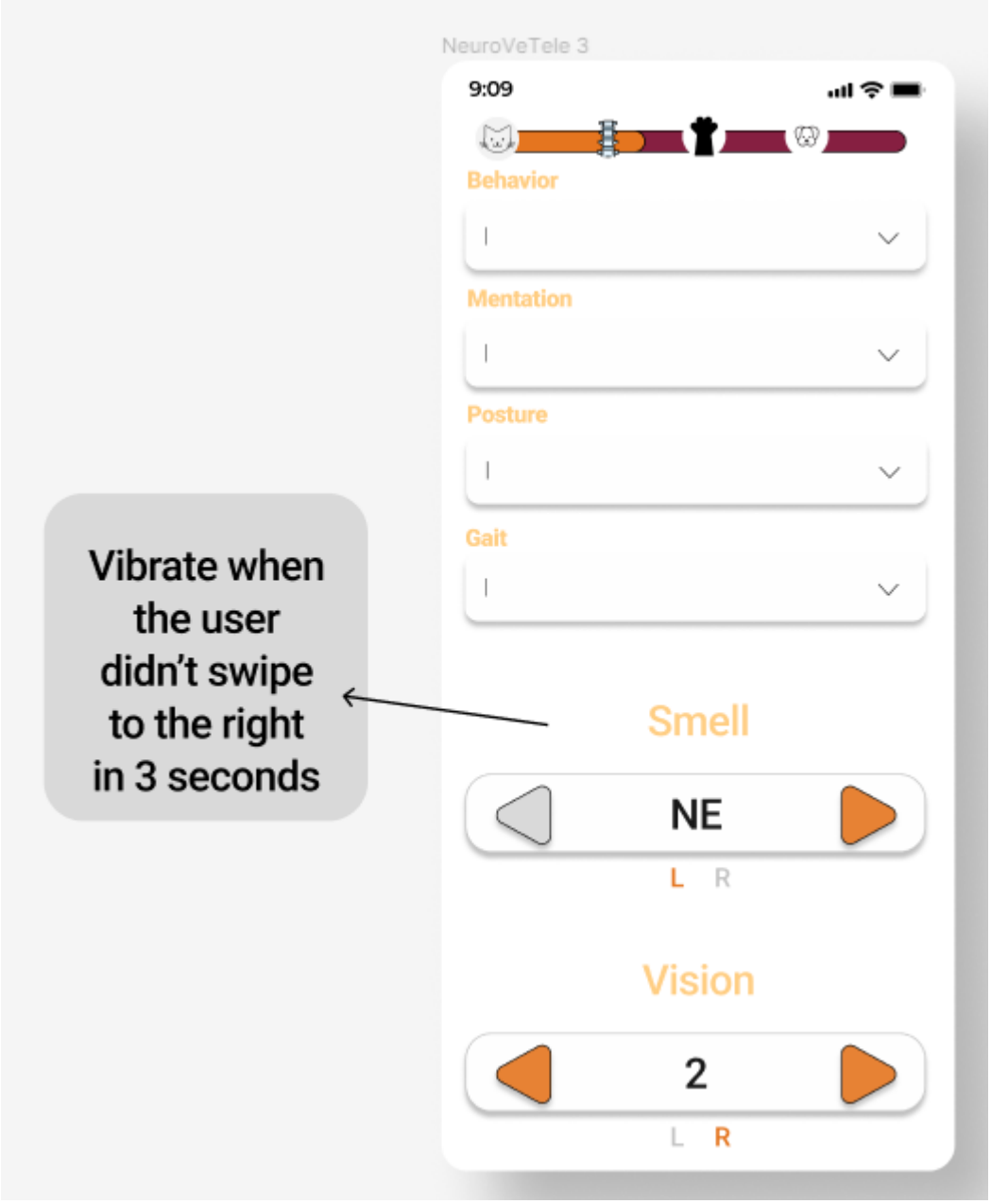


Figure 4: Sample Input Page Showing Progress Bar

As seen in Figure 5, most of the options that are qualitative will utilize the drop down bar. When the drop down arrow is clicked it will showcase all the available options for that option (Figure 6). As the user can pick multiple options, the drop down options will

dynamically move the best next options to the top, and conflicting options to the bottom (Figure 7). When the user selects two options that contradict each other, the application is aware of this conflict and will prompt a pop-up telling the user of this conflict (Figure 8).

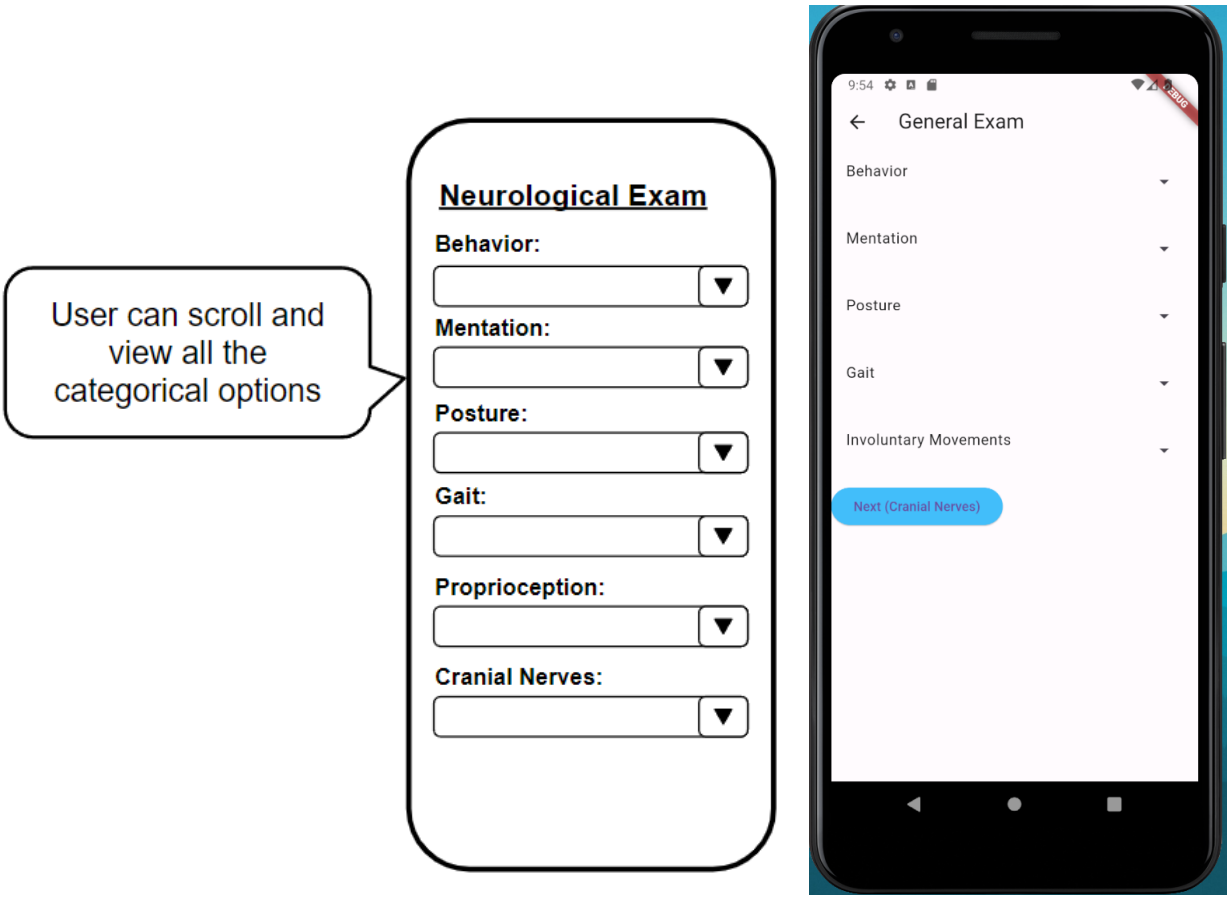


Figure 5: Proposed Qualitative Sample Input Page (left) and Current (right)

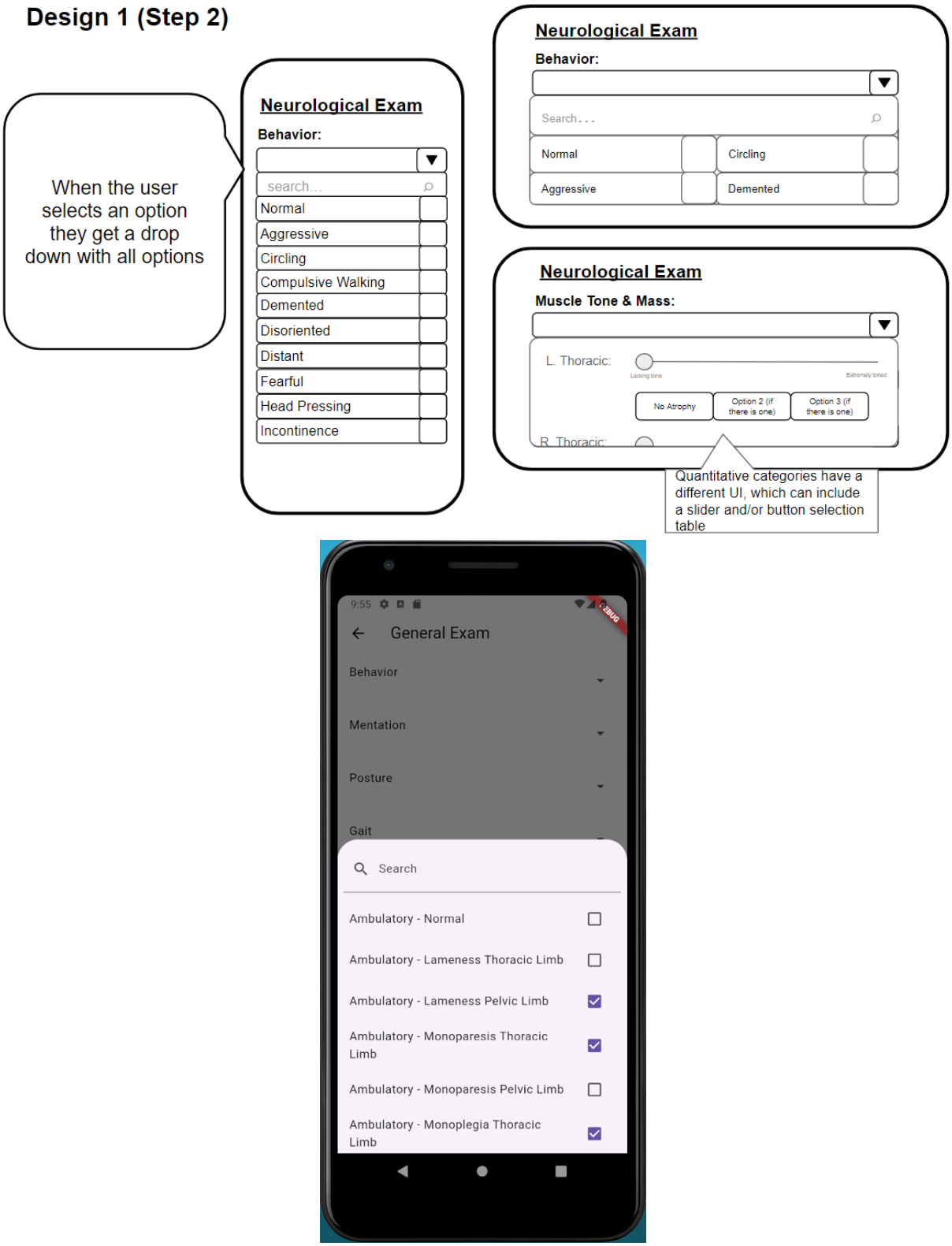


Figure 6: Proposed Showcase of the Dropdown Bar (top) and Current (bottom)

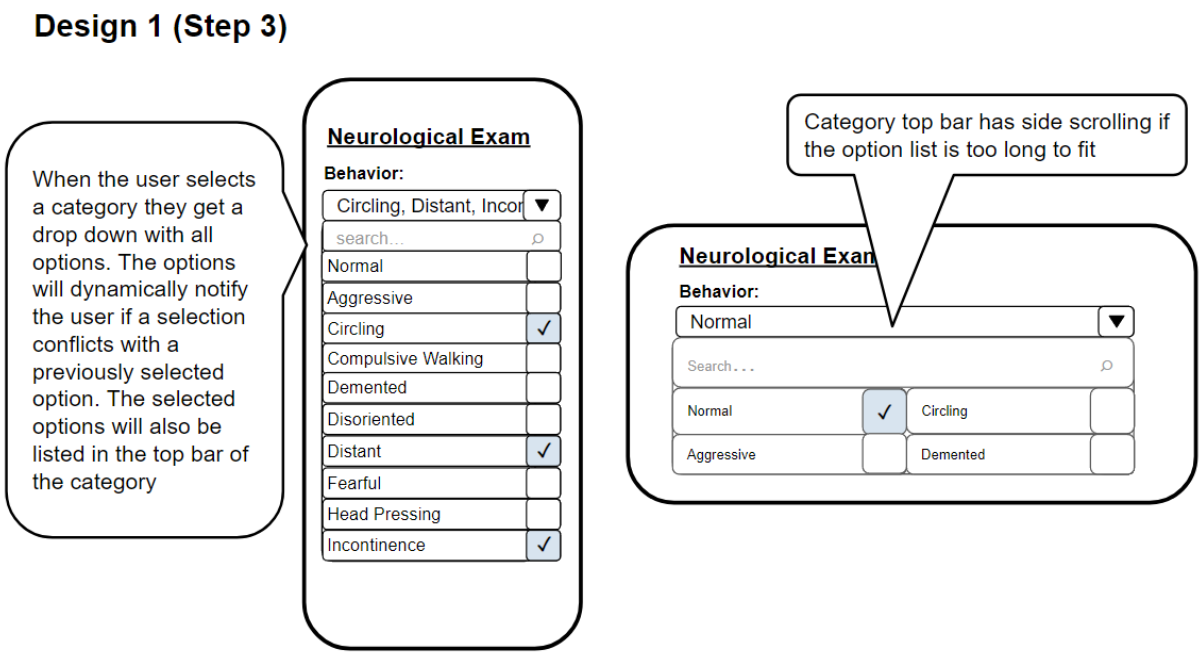


Figure 7: Instance of Multiple Options

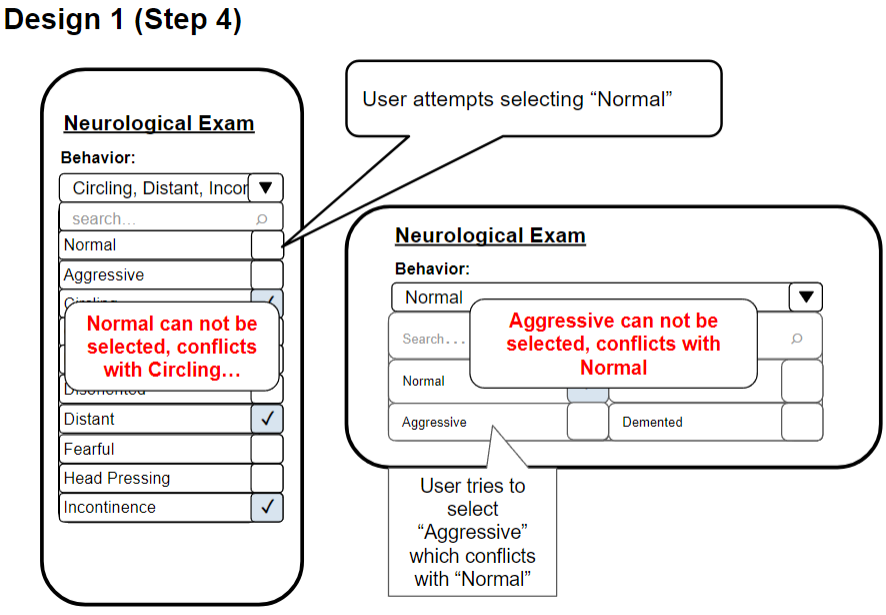


Figure 8: Instance of Conflicting Options

As the user finishes up and populates the inputs, the "Neurolocalize!" button will activate at the end (Figure 9). By clicking this button, it will bring the user to the neurolocalization page.

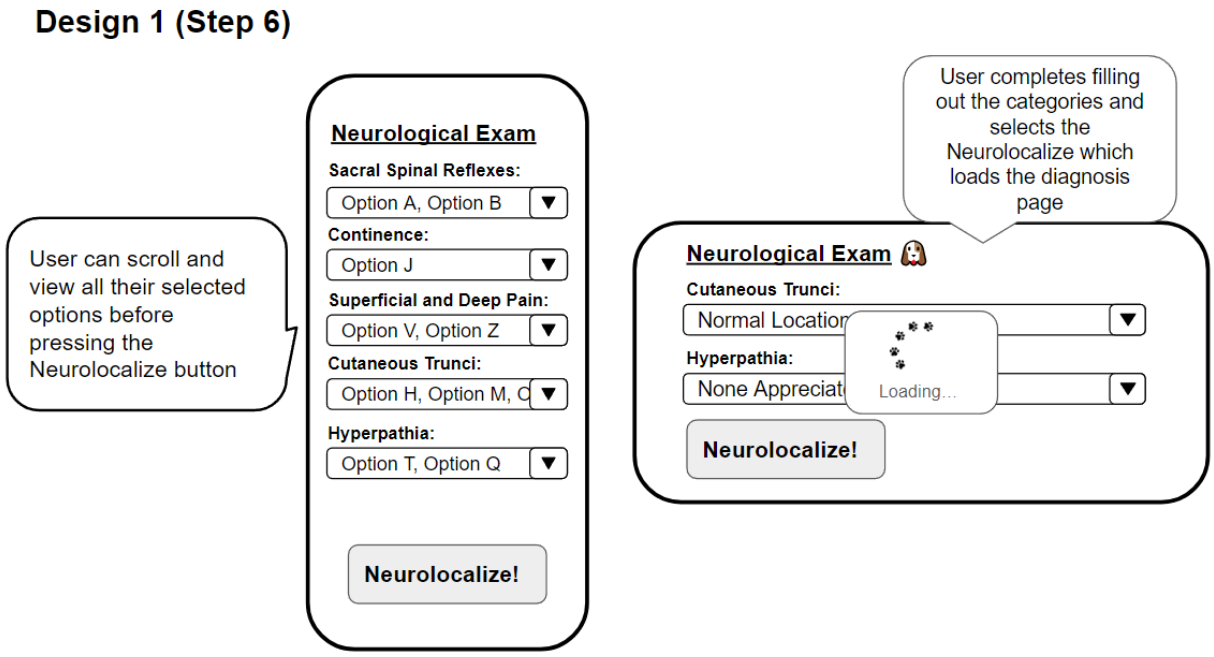


Figure 9: Sample Completed Input Page

5.3 Neurolocalization/Results Page

After all the inputs are collected, the back-end will display the neurolocalization that yielded the highest point from calculating those inputs' points. The neurolocalization page showcases the diagnosis found by the back-end and also gives out a plan to the user if needed (Figure 10). Currently, the results page is simplified to only showcase the top three neurolocalizations (Figure 13).

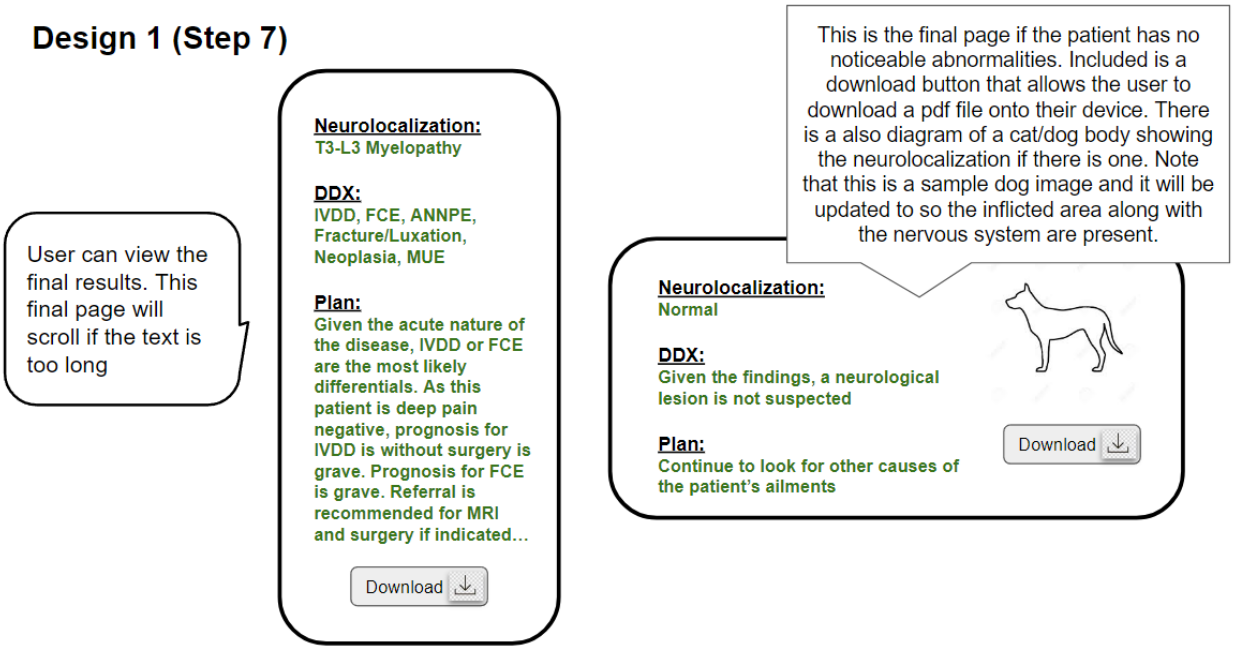


Figure 10: Neurolocalization/Results Page

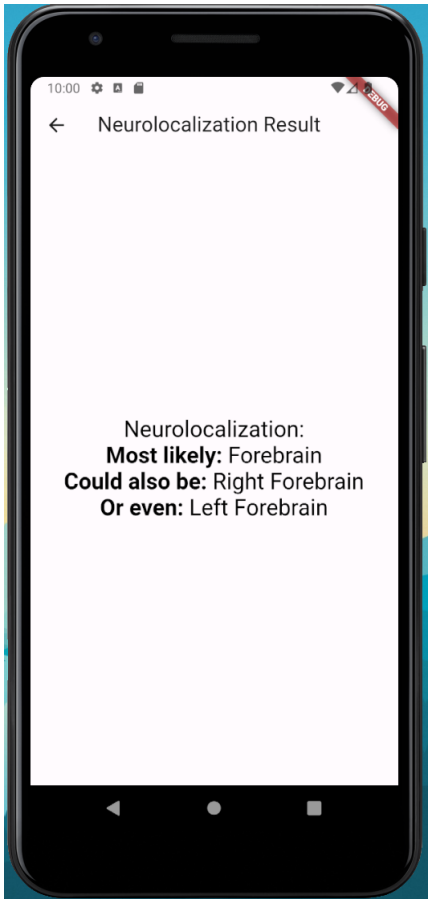


Figure 11: Simplified Results Page

At the very bottom of this page, there is a download button. This button will convert all the inputs and the neurolocalization with its plan into a PDF file that the user can save on their device. This file will follow a similar structure to the table given in the paper neurological exam (seen in Table 1 for a sample PDF showing the results of a normal neurolocalization).

Table 1: Sample PDF Table of a Normal Neurolocalization

Neurological Exam –							
Behavior:		Quiet					
Mentation:		Alert					
Posture:		Normal					
Gait:		No ataxia or paresis appreciated					
Cranial Nerves (0,1,2)		Left	Right	Cranial Nerves (0,1,2)		Left	Right
I	Smell:			V	Muscle Sym:	2	2
II	Vision:	2	2	V/VII	Palpebral:	2	2
II/VII	Menace:	2	2	VII	Facial Sym:	2	2
III	Pupil Size:	2	2		Sensory:	2	2
II/III	PLR Direct:	2	2	VIII	Oculocephalic	2	2
	PLR Indirect:	2	2		Nystagmus	2	2
III/IV/VI	Strabismus:	2	2		Hearing	2	
V/VI	Globe Retract:			IX/X	Gag/Swallow:	2	
V	Motor:	2	2	XI	Trapezius	2	2
	Sensory:	2	2	XII	Tongue:	2	2
Thoracic Limbs	Left	Right	Pelvic Limbs	Left	Right		
Proprioception (0, 1, 2)			Proprioception (0, 1, 2)				
Paw Placement	2	2	Paw Placement	2	2		
Hopping	2	2	Hopping	2	2		
Wheelbarrow	2	2	Extensor Thrust	2	2		
Spinal Reflexes (0, 1, 2, 3, 4)			Spinal Reflexes (0, 1, 2, 3, 4)				
Biceps	2	2	Patellar	2	2		
Triceps	2	2	Gastroc. / Sciatic	2	2		
Extensor Carpi	2	2	Cranial Tibial	2	2		
Withdrawal	2	2	Withdrawal	2	2		
Crossed Extensor	-	-	Crossed Extensor	-	-		
Muscle Tone	2	2	Muscle Tone	2	2		
Muscle Atrophy	None	None	Muscle Atrophy	None	None		
Perineal Reflex/Anal & Tail Tone:				Intact			
Continence:				Not Evaluated			
Superficial/Deep pain sensation:				Not Necessary to Evaluate			
Cutaneous Trunci:				Normal Location			
Hyperpathia:				None Appreciated			
Neurolocalization: Normal Neurological Exam							
DDX: Given the findings, a neurological lesion is not suspected							
Plan: Continue to look for other causes of the patient's ailments							

5.4 Video Demo

To demonstrate the current version of NeuroVeTele in a more interactive manner, a video was created for users to follow along. The sequence of the video is further described by Table 2 and Figure 12.

Table 2: Timetable of the Video Demo

Timestamp:	Description:
0:00 - 0:07	User starts on the main page and selects the "Dog" option; this brings them to the General Page.
0:07 - 0:25	User selects options via drop-down menus for each category in the General Page.
0:25 - 0:31	Currently, only the General Page contains the points for the options, so the user clicks on the 'Neurolocalize' button which brings up the Results Page.
0:31 - 1:20	If the user is an admin, they can access the Dashboard from the Main Page, where they can change option points through text box inputs and save them by clicking the floppy disk icon on the top right corner.
1:20 - 1:28	With the points changed, the same options selected before will yield different neurolocalizations on the Results Page.

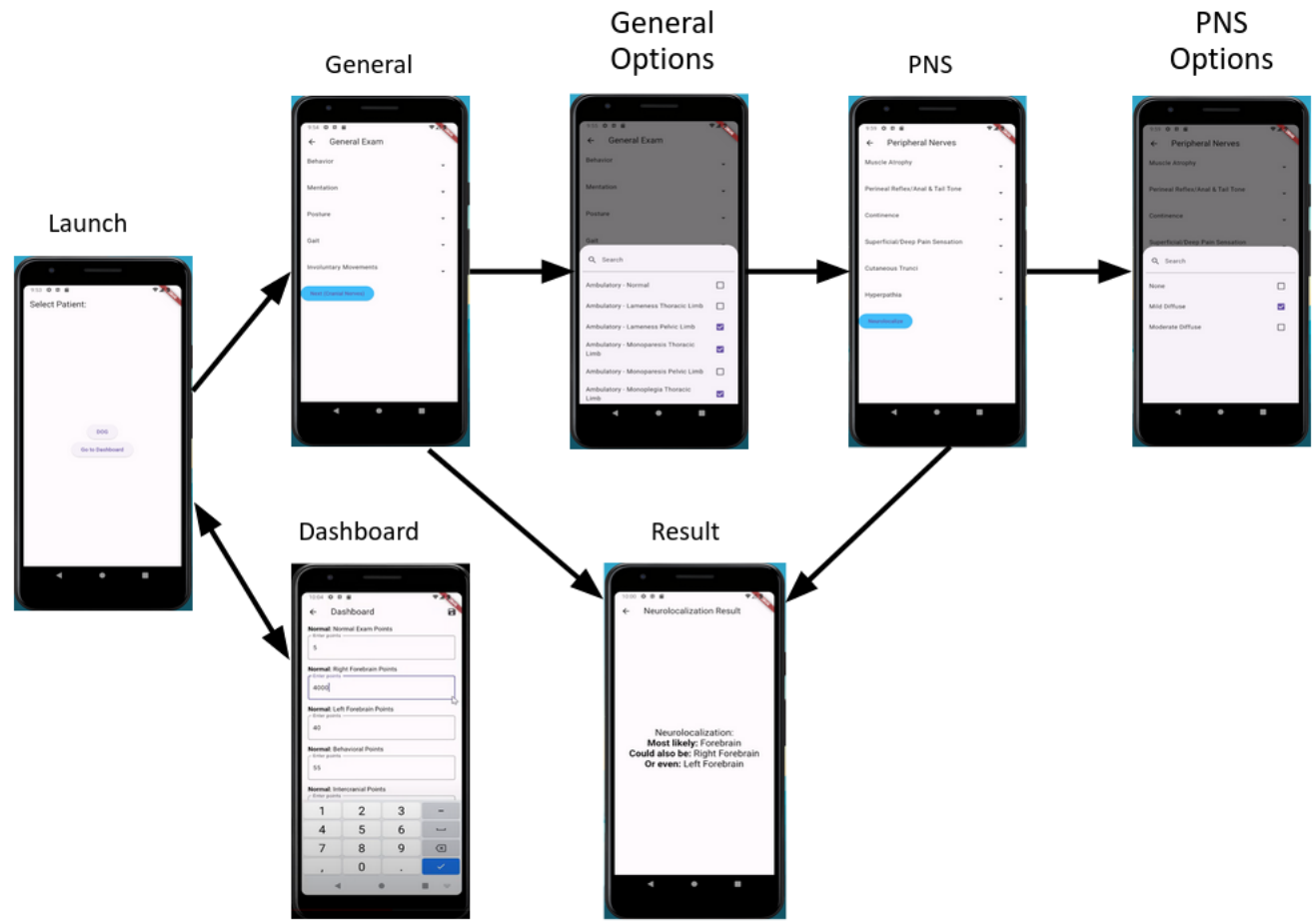


Figure 12: Wireframe of the Demo Sequence

Figure 12 is a wireframe of how the demo was sequenced. The demo began with the Launch Page where the user pressed the "Dog" button, navigating to the General Page. There, the user selected options pertaining to categories from the General Page. From here, they could press the "Neurolocalize" button to go to the Result page. They could also navigate from the General Page to other input pages like the PNS Page. Similar to the General Page,the user selected options and then press the "Neurolocalize" button to go to the Result page. At the Launch Page, if the user is an admin, they can press the "Dashboard" button to go to the Dashboard page to change the point values. From here the user can repeat the previous steps to verify the changes made in the dashboard. See the

accompanying file NeuroVeTeleDemo.mp4 for the video demo.

6 Developer's Manual

6.1 Prerequisites

Before further development and installment of NeuroVeTele, it is important that one's local machine follows some standards to ensure NeuroVeTele is able to run efficiently. The system should follow these recommended prerequisites:

- Operating System: Windows 8, 10, or 11
- Processor: Intel Core i5 or equivalent
- RAM: At least 4 GB
- Disk Space: At least 50 GB of free space
- Internet Connection: Stable internet connection used for downloading and running NeuroVeTele

6.2 Applications

To fully implement the code of NeuroVeTele on one's system, certain applications must be downloaded:
Download Flutter SDK:

- Visit `https://docs.flutter.dev/get-started/install/windows/mobile?tab=download` and download the Flutter SDK from the main blue button on that site.
- Extract the downloaded zip file and move it to an accessible directory.
- Search and click "Edit environment variables for your account" in your taskbar.
- Edit the user variable for path by adding the path to the directory containing the installed Flutter SDK.

- Run “flutter doctor” on CMD to verify the tools have been installed (optional).

Download Android Studio:

- Visit `https://developer.android.com/studio` and download the latest version for Windows with default settings.
- After installation, open Android Studio and install the Flutter plugin found in the Android Plugins section.
- Install Dart when prompted and restart the system.

Download Visual Studio:

- Visit `https://visualstudio.microsoft.com/downloads/` and download the latest version for Windows with default settings.

Download Visual Studio Code:

- Visit `https://code.visualstudio.com/Download` and download the latest version for Windows with default settings.
- Make a new folder in VSCode to store NeuroVeTele.

6.3 GitHub

For source control along with general communication amongst coders, GitHub was used as the main control center through its git features like clone, push, and commit. Developers can access the README and source code from this GitHub. Though cloning is allowed for other developers to study with, it is imperative that it cannot be used for reasons that could negatively affect the integrity of our project. To obtain the web URL of NeuroVeTele for cloning, developers should contact Dr. Richard Shinn (shinndvm@vt.edu).

6.4 Running in VSCode

To run NeuroVeTele in VSCode after cloning the program to their local device, developers should follow these steps:

- Open main.dart and run it without debugging.
- When prompted with options of devices on which to showcase NeuroVeTele, choose the one most desirable to you.

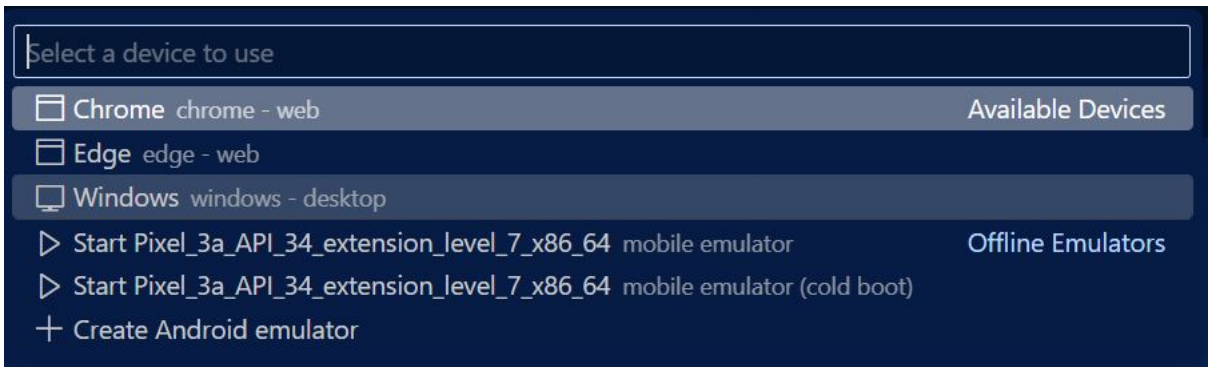


Figure 13: Options of Devices

- NeuroVeTele will be displayed in the form of the chosen device through your local device.
- Remember to "flutter clean" through a PowerShell terminal after runs to wipe away previous data.

6.5 Dashboard

The Dashboard is specifically used and accessed by admins and developers. This page can be retrieved through the Main Page, which is where the type of pet patient is picked. In the Dashboard, admins are able to adjust the points to options provided by the points system. This is done so by putting in values into their respective text boxes (Figure 14). To save

these values, the admin clicks on the floppy disk symbol located on the top right corner to save all (as seen in Figure 14). This will update the initial default values via global variables.

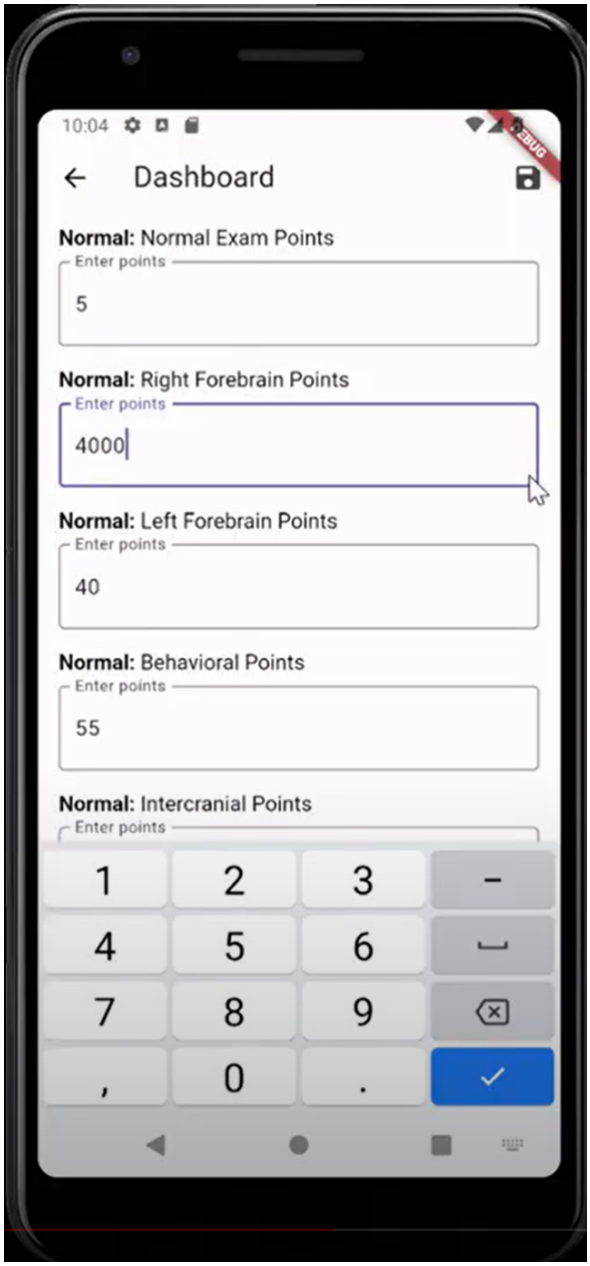


Figure 14: Sample Dashboard

7 Lessons Learned

- **Effective client communication:** We began each week by planning with our client the time and place of the next meeting. When dealing with clients who are not computer science majors, we have found that offline communication tends to be more efficient. We use body language, whiteboards, and other tools to explain algorithms and user interfaces to clients offline, and clients can easily demonstrate how users will use the application on-site, which enhances the developer's understanding of the user's needs. We also use real-time email to report progress to the client, so that both parties can better understand each other's needs through this process.

- **Use iterative thinking:** From the very beginning of the software interface design, we planned to design the user interface iteratively. After discussions with the client, we have iterated the user interface for two rounds to meet the client's needs. Instead of planning a lot at the beginning of the project, we can present each generation of the design and communicate with the client to make changes.

- **Use of resources:** At the beginning of the project, all of us were worried about the underlying algorithms. However, after talking with our mentors, they gave us a lot of effective algorithms based on their experience. Instead of trial and error, it is better to utilize the resources of the computer profession to solve the problem more efficiently.

- **Use of git:** As the project was a team effort, in git, the team members completed their code and tested it in their respective branches, eventually merging the project into a single branch. There were many uses of git that will be indispensable in future development or not, and the project increased our proficiency in git.

- **Dart**: The project is implemented with Dart (Lougheed, 2024). Learning Dart begins with mastering its syntax and setup, then exploring its integration with Flutter for developing applications on multiple platforms from one codebase. Dart's advantages

include both Just-In-Time and Ahead-Of-Time compilation for quick development and high performance. Its strong typing system enhances code reliability, while features like async and await simplify asynchronous programming. Dart's robust tooling and supportive community also make it a great choice for developers.

- **UX**: We learned that to develop easily usable interfaces in software development involves understanding the principles of user experience (UX) design. This includes studying how users interact with applications and designing interfaces that are intuitive and easy to navigate. Key aspects include consistency in design, clear navigation cues, and responsive layouts that adapt to various devices. We also focus on accessibility, ensuring that applications are usable by people with different abilities. Practical experience through projects, user testing, and feedback is crucial for refining interface designs.

7.1 Encountered Problems

Throughout the development of the NeuroVeTele app, our team has encountered several significant challenges. One of the primary hurdles was cross-platform testing for iOS devices. Initially, our lack of access to a Mac interface, which is crucial for iOS app development and testing, hindered our ability to ensure the app's functionality across different platforms. This issue was resolved when the Virginia Tech Computer Science Department generously provided us with a Mac computer. This equipment has been instrumental in our ongoing efforts to test and optimize the app for iOS, thus ensuring a uniform user experience across all platforms.

Another significant challenge was the development of the back-end algorithm. We engaged Dr. Cao, a domain expert in data structures and biological computations, to refine our approach. Through consultations and collaborative meetings focused on the algorithm's structure and implementation, we developed a solution that is both scientifically sound and

practically viable for clinical use.

Additionally, we faced the challenge of enabling our client, Dr. Shinn, to modify the neurolocalization weights directly without interacting with the codebase. Initially, we attempted to implement a CSV parser to import data from Dr. Shinn's specifically formatted Excel sheets. However, the variability in formatting proved too complex to manage effectively. As a workaround, we developed a Dashboard Page, allowing Dr. Shinn to directly edit these weights within the app. The ongoing challenge is to ensure that these changes are permanently saved.

After every meeting with the client, there are always new requirements and we need to add these features to the project. This often interrupts our plans, so integrating the product iteration with the original plan is a big challenge. That's why we were behind schedule, but in fact, we had more features in the original plan.

During development, we encountered huge datasets that required developers to manually translate them into code. This process is tedious but necessary. We tried to use external tools to speed up this process, but we have only handled a fraction of it so far, and there are still thousands of rows of excel spreadsheets that we need to work with in the future.

While testing with the client, we realized that the client had a need to update the points system themselves. This requirement required manual debugging, which was time consuming for the client as well as the developer. Currently we have only managed to get the client to send the updated configuration file to the developer and the developer to upload it manually.

Another problem is that the dataset given to us by the client was not sufficient for the entire project. It is a huge points system, and even though the client has been updating us

with data for four weeks in a row, it currently only covers about a third of the entire test. But the completeness of the project depends heavily on the data, so the problems that will be encountered in future development are unknown. Only after we have processed the data can we discover additional problems. Such issues include UI design, user experience, test accuracy, and so on. So, this will be a long term project and will require a lot of debugging.

7.2 Timeline

Jan. 31: Brainstormed the initial concept, and started consultation with client Dr. Richard Shinn.

Feb. 15: Finalized the design mock-up with the client and began implementing it on Figma (see Figure 15 in the Appendix).

Feb. 29: Fully completed UML (see Figure 16 in the Appendix) and partially implemented the UI (basic framework of front-end completed).

Mar. 15: Implemented the data structure. Used arrays for the inputs and point system. Determined the algorithm used in the back-end.

Mar. 31: Completed the basic backbone of back-end and front-end. Began adding data from the points system to back-end. Initialized the early stages of alpha testing.

April 15: Further proliferation of the back-end with additional data from the points system provided by our client. Created the Dashboard Page to streamline the process of adding data to the back-end.

April 30: Checked the potential features that could be added to the app to iterate the product. Prepared future improvements to the app, specifically to the Cranial Nerves page.

7.3 Future Plans

After completing this course, team members will continue to develop the app. Over the summer, we plan to conduct extensive testing with a user group comprising Dr. Shinn's

students and colleagues. We will expand the app's capabilities by incorporating additional data from Dr. Shinn, which will enhance NeuroVeTele's point system. The user interface of NeuroVeTele is slated for a significant redesign to improve visual appeal, making it suitable for display on app stores and for end-user interaction.

In addition to aesthetic improvements, we aim to enhance user interaction by introducing sound effects for buttons and background music. We are also planning to extend the app's features, including diagrams that illustrate animal gaits and additional examination options that cover consciousness, vision, and seizures. Once the framework for dog examinations is established, we will add functionalities for other common animals in veterinary practice, such as cats and horses.

Looking forward, we intend to develop more comprehensive functionalities and integrate them with existing veterinary medical systems to expand the app's application and explore commercialization avenues. Dr. Shinn envisions NeuroVeTele not only as a tool for veterinarians to conduct neurolocalizations but also as a platform for pet owners to remotely connect with veterinarians via video.

To support these updates, we are planning to establish a server-based infrastructure that will securely store and manage the app's enhancements. This setup will ensure that changes made through the Dashboard Page are consistently maintained and integrated. Additionally, we are developing a feature to automatically convert Excel files into CSV format, simplifying data management and improving operational efficiency.

Further enhancements include the implementation of the General Page, where the system will display an error message if incompatible options are selected, guiding users toward valid choices. This error handling is pending as the necessary data is not yet fully available. We

will also implement the Cranial Nerve Page and finish adding all options to the PNS page once the required data is received, using a similar implementation approach to the General page.

In the Results page, plans are underway to include differential diagnoses (ddx) and recommended plans as soon as the relevant data becomes available, thereby ensuring accuracy and improving the user experience.

8 Acknowledgements

We would like to acknowledge our client, Dr. Richard Shinn, for providing us this opportunity in utilizing our discipline of computer science in the neurological field. His expertise and constant help has made this project ever more flourishing. We would also like to acknowledge Dr. Yang Cao for aiding us in our journey of creating the most efficient and effective back-end algorithm for this project scope. Additionally, we are ever more thankful for Dr. Edward A. Fox in guiding use throughout our project's timeline. Further, we would also like to thank Mr. Satvik Chekuri for helping us secure the needed resources that made our product versatile.

Dr. Richard Shinn, Assistant Professor in Neurology
Department of Small Animal Clinical Sciences
shinndvm@vt.edu

Dr. Yang Cao, Associate Professor in Computational Biology
ycao@vt.edu

Dr. Edward A. Fox, Professor
Director, Digital Library Research Laboratory
Executive Director, Networked Digital Library of Theses and Dissertations
fox@vt.edu

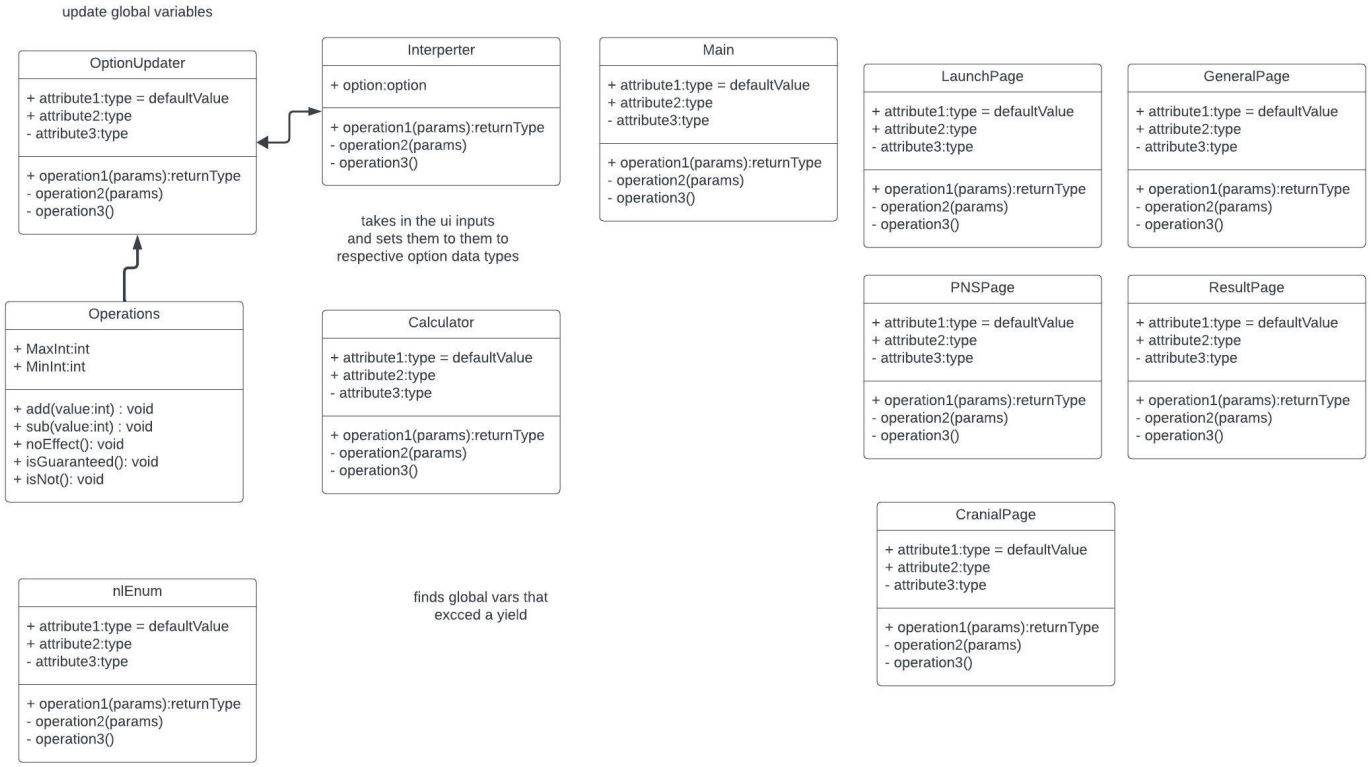
Mr. Satvik Chekuri, Ph.D. Student
Department of Computer Science

9 References

1. Berg J. Problems in neurolocalization. Probl Vet Med. 1989 Jul-Sep;1(3):358-65. PMID: 2520120.
2. Bush, William. "Guide to Neurolocalization." Delaware Valley Academy of Veterinary Medicine, Delaware Valley Academy of Veterinary Medicine, 15 Mar. 2019, delawarevalleyacademyvm.org/pdfs/mar15/DVA3-15Notes(33).pdf. Accessed 30 April 2024.
3. Lougheed, Parker. "Building User Interfaces with Flutter." Flutter, Google, 4 Apr. 2024, docs.flutter.dev/ui. Accessed 30 April 2024.
4. Nagendran A, José López R, Suñol A, Brocal J, Gonçalves R. The value of a head turn in neurolocalization. J Vet Intern Med. 2024; 38(1): 268-276. doi:10.1111/jvim.16957
5. Rylander, Helena. "The Neurologic Examination in Companion Animals, Part 2: Interpreting Abnormal Findings." Today's Veterinary Practice, Today's Veterinary Practice, 17 Feb. 2022, todaysveterinarypractice.com/neurology/the-neurologic-examination-in-companion-animals-part-2-interpreting-abnormal-findings/. Accessed 30 April 2024.
6. Schubert, Thomas. "The Neurologic Evaluation in Cats - Cat Owners." Merck Veterinary Manual, Merck & Co., Inc., Aug. 2018, www.merckvetmanual.com/cat-owners/brain,-spinal-cord,-and-nerve-disorders-of-cats/the-neurologic-evaluation-in-cats. Accessed 30 April 2024.

10 Appendix

Figure 15: Figma Implementation



User Scenario:
A veterinarian wanted to speed up his busy checkup schedule and decided to use NeuroVeTele. After entering the software, the veterinarian entered the checkup parameters into NeuroVeTele as the dog went through the checkup, and the software immediately displayed the results of the checkup, which the veterinarian confirmed, and then submitted the checkup report to the dog's owner.

Figure 16: Simplified UML of NeuroVeTele. On the left are functions. On the right are details for the 6 pages.