Pseudocode for driver.py (ENHANCED)

```
FROM initialize IMPORT Initialize
FROM intake IMPORT Intake
FROM reserve_animal IMPORT ReserveAnimal
FROM security IMPORT Security
IMPORT tkinter AS tk
FROM validation IMPORT Validation
IMPORT dashboard
IMPORT print_animals
IMPORT search

CLASS Driver:
    METHOD login()
        DISPLAY “Welcome to the Rescue Animal System. Please log in.”
        ASSIGN attempts TO 3

        WHILE attempts GREATER THAN 0
            DISPLAY “Username: “
                READ input ASSIGN username
            FROM Validation INIT null_validation ASSIGN TO username

            DISPLAY “Password: “
                READ input ASSIGN password
            FROM Validation INIT null_validation ASSIGN TO password

        FROM Initialize INIT get_user_from_db FOR username ASSIGN user

        IF user AND FROM Security INIT verify_password
            DISPLAY “Login Successful”
            RETURN True

            ELSE
                DISPLAY “Invalid username or password.”
                INCREMENT attempts MINUS 1
        ELSE
            DISPLAY “Too many failed attempts. Exiting.”
            RETURN False
        END WHILE


    METHOD main()
        IF login NOT True
            RETURN

        WHILE True
            FROM Initialize INIT connect_db
            INIT scnr AS input

            ASSIGN usr_input TO NOTHING

            WHILE usr_input IS NOT q
                DISPLAY menu
                ASSIGN usr_input TO scnr
```

```
                    IF usr_input EQUALS 1
                        DISPLAY “Beginning Dog Intake”
                        FROM Intake CALL intake_new_dog

                    ELSE IF usr_input EQUALS 2
                        DISPLAY “Beginning Monkey Intake”
                        FROM Intake CALL intake_new_monkey

                    ELSE IF usr_input EQUALS 3
                        DISPLAY “Beginning Animal Reservation”
                        FROM ReserveAnimal CALL reserve_animal

                    ELSE IF usr_input EQUALS 4
                        DISPLAY “Displaying List of dogs”
                        FROM print_animals CALL print_animals(1)

                    ELSE IF usr_input EQUALS 5
                        DISPLAY “Displaying List of monkeys”
                        FROM print_animals CALL print_animals(2)

                    ELSE IF usr_input EQUALS 6
                        DISPLAY “Displaying List of unreserved animals”
                        FROM print_animals CALL print_animals(3)

                    ELSE IF usr_input EQUALS 7
                        DISPLAY “Searching Database”
                        FROM search CALL search_and_update_animal

                    ELSE IF user_input EQUALS 8
                        DISPLAY “Adding New User”
                        FROM security INIT add_user

                    ELSE IF user_input EQUALS 9
                        DISPLAY “Displaying Dashboard”
                        FROM dashboard INIT Dashboard

                    ELSE IF usr_input EQUALS q
                        DISPLAY “Quitting.”
                        FROM Initialize INIT close_db
                        END

                    ELSE
                        DISPLAY “Invalid command.”

                    END IF
                END WHILE
            END IF

Static METHOD display_menu()
        DISPLAY 2 new line
        DISPLAY “Rescue Animal System Menu”
        DISPLAY “[1] Intake a new dog”
        DISPLAY “[2] Intake a new monkey”
        DISPLAY “[3] Reserve an animal”
        DISPLAY “[4] Print a list of all dogs”
        DISPLAY “[5] Print a list of all monkeys”
```

```
DISPLAY “[6] Print a list of all animals that are not reserved”
DISPLAY “[7] Search for animal, and update status”
DISPLAY “[8] Add a new user”
DISPLAY “[9] Display Dashboard”
DISPLAY “[q] Quit application”
DISPLAY “Enter a menu selection”
```

Pseudocode for dog.py (REMOVED)

```
REMOVED
```

Pseudocode for monkey.py (REMOVED)

```
REMOVED
```

Pseudocode for rescueanimal.py (REMOVED)

```
REMOVED
```

Pseudocode for intake.py (ENHANCED)

```
FROM initialize IMPORT Initialize
FROM validation IMPORT Validation

CLASS Intake:
    METHOD intake_new_dog(scanner)
        FROM Initialize INIT fetch_all_dogs ASSIGN dogs

        DISPLAY “What is the dog’s name?”
            READ input ASSIGN TO name
        FROM Validation INIT null_validation ASSIGN TO name

        FOR EACH dog in dog_list
            IF dog.name EQUALS TO name
                DISPLAY “This dog is already in our system”
                RETURN
            END IF
        END FOR
        DISPLAY Adding “Dog:” ADD name

        DISPLAY “What is the dog’s breed?”
            READ input ASSIGN breed")
        FROM Validation INIT null_validation ASSIGN TO breed
        DISPLAY “Breed:” ADD breed
```

```
        DISPLAY “What is the dog’s gender?”
            READ input ASSIGN gender
        FROM Validation INIT gender_validation ASSIGN TO gender
        DISPLAY “Gender:” ADD gender

        DISPLAY “What is the dog’s age?”
            READ input ASSIGN age
        FROM Validation INIT positive_digit_validation ASSIGN TO age
        DISPLAY “Age:” ADD age

        DISPLAY “What is the dog's weight?”
            READ input ASSIGN weight
        FROM Validation INIT float_validation ASSIGN TO weight
        DISPLAY “Weight:” ADD weight

        DISPLAY "When was the dog acquired? Format: mm-dd-yyyy"
            READ input ASSIGN acquisition_date
        FROM Validation INIT date_validation ASSIGN TO acquisition_date
        DISPLAY "Dog Acquired:" ADD acquisition_date

        DISPLAY "What Country did the dog come from?"
            READ input ASSIGN acquisition_country
        FROM Validation INIT null_validation ASSIGN TO acquisition_country
        DISPLAY "Country of Origin:" ADD acquisition_country

        DISPLAY "What is the dog's training status?"
            READ input ASSIGN training_status
        FROM Validation INIT status_validation ASSIGN TO training_status
        DISPLAY "Training Status:" ADD training_status

        DISPLAY "Has this dog been reserved? (True or False)"
            READ input ASSIGN reserved
        FROM Validation INIT boolean_validation ASSIGN TO reserved
        DISPLAY "Reservation Status:" ADD reserved

        DISPLAY "What country is the dog in service?"
            READ input ASSIGN in_service_country
        FROM Validation INIT null_validation ASSIGN TO in_service_country
        DISPLAY "Country of Service:" ADD in_service_country

        FROM Initialize INIT conn.cursor ASSIGN cursor
            EXCUTE cursor INSERT INTO Dog ASSIGN name, breed, gender, age,
weight_float, acquisition_date, acquisition_country, training_status, reserved, in_service_country
            FROM Initialize INIT conn.commit
        DISPLAY "New Dog" ADD name ADD "added!"

METHOD intake_new_monkey(scanner)
        FROM Initialize INIT fetch_all_monkeys ASSIGN monkeys

        DISPLAY "What is the Monkey's name?"
            READ input ASSIGN name
        FROM Validation INIT null_validation ASSIGN TO name

        FOR EACH monkey in monkey_list
            IF monkey.name EQUALS TO name
                DISPLAY "This Monkey is already in our system"
```

```
        END IF
        RETURN
END FOR
DISPLAY "Adding Monkey:" ADD name


DISPLAY "What is the Monkey's species?"
        READ input ASSIGN species
FROM Validation INIT species_validation ASSIGN TO species
DISPLAY "Species:" ADD species

DISPLAY "What is the Monkey's gender?"
        READ input ASSIGN gender
FROM Validation INIT gender_validation ASSIGN TO gender
DISPLAY "Gender:" ADD gender

DISPLAY "What is the Monkey's age?"
        READ input ASSIGN age
FROM Validation INIT positive_digit_validation ASSIGN TO age
DISPLAY "Age:" ADD age

DISPLAY "What is the Monkey's weight?"
        READ input ADD weight
FROM Validation INIT float_validation ASSIGN TO weight
DISPLAY "Weight:" ADD weight

DISPLAY "What is the Monkey's tail length?"
        READ input ADD tail_length
FROM Validation INIT positive_digit_validation ASSIGN TO tail_length
DISPLAY "Tail Length:" ADD tail_length

DISPLAY "What is the Monkey's height?"
        READ input ASSIGN height
FROM Validation INIT positive_digit_validation ASSIGN TO height
DISPLAY "Height:" ADD height

DISPLAY "What is the Monkey's body length?"
        READ input ASSIGN body_length
FROM Validation INIT positive_digit_validation ASSIGN TO body_length
DISPLAY "Body Length:" ADD body_length

DISPLAY "What is the Monkey's torso length?"
        READ input ASSIGN torso_length
FROM Validation INIT positive_digit_validation ASSIGN TO torso_length
DISPLAY "Torso Length:" ADD torso_length

DISPLAY "What is the Monkey's skull length?"
        READ input ASSIGN skull_length
FROM Validation INIT positive_digit_validation ASSIGN TO skull_length
DISPLAY "Skull Length:" ADD skull_length

DISPLAY "What is the Monkey's neck length?"
        READ input ASSIGN neck_length
FROM Validation INIT positive_digit_validation ASSIGN TO neck_length
DISPLAY "Neck Length:" ADD neck_length
```

```
        DISPLAY "When was the Monkey acquired? Format: mm-dd-yyyy"
            READ input ASSIGN acquisition_date
        FROM Validation INIT date_validation ASSIGN TO acquisition_date
        DISPLAY "Monkey Acquired:" ADD acquisition_date

        DISPLAY "What Country did the Monkey come from?"
            READ input ASSIGN acquisition_country
        FROM Validation INIT null_validation ASSIGN TO acquisition_country
        DISPLAY "Country of Origin:" ADD acquisition_country

        DISPLAY "What is the Monkey''s training status?"
            READ input ASSIGN training_status
        FROM Validation INIT status_validation ASSIGN TO training_status
        DISPLAY "Training Status:" ADD training_status

        DISPLAY "Has this Monkey been reserved? (True or False)"
            READ input ASSIGN reserved
        FROM Validation INIT boolean_validation ASSIGN TO reserved
        DISPLAY "Reservation Status:" ADD reserved

        DISPLAY "What country is the Monkey in service?
            READ input ASSIGN in_service_country
        FROM Validation INIT null_validation ASSIGN TO in_service_country
        DISPLAY "Country of Service:" ADD in_service_country

        FROM Initialize INIT conn.cursor ASSIGN cursor
            EXCUTE cursor INSERT INTO Monkey ASSIGN name, species, gender, age,
    weight_float, tail_length_float, height_float, body_length_float,
                torso_length_float, skull_length_float, neck_length_float, acquisition_date,
    acquisition_country, training_status, reserved, in_service_country
            FROM Initialize INIT conn.commit
        DISPLAY "New Monkey" ADD name ADD "added!"
```

Pseudocode for initialize.py (ENHANCED)

```
IMPORT psycopg2

CLASS Initialize
    INIT conn AS NOTHING
    METHOD connect_db()
        TRY
            CONNECT psycopg2 AS dbname EQUALS TO RescueAnimal AND user
    EQUALS TO postgres AND password EQUALS TO password AND host EQUALS TO localhost
    ASSIGN conn
            INIT create_tables()
        EXCEPT Exception ASSIGN e
            DISPLAY “Failed to connect to the database:” ADD e
            ASSIGN conn AS NOTHING

    METHOD close_db()
        INIT conn.close()

    METHOD create_tables():
```

```
        INIT conn.cursor ASSIGN cursor
                EXECUTE cursor IF TABLE does not exist CREATE TABLE Dog ASSIGN id
SERIAL PRIMARY KEY, name VARCHAR(50), breed VARCHAR(50), gender VARCHAR(10),
age VARCHAR(10),
                        weight VARCHAR(10), acquisition_date VARCHAR(20),
                acquisition_country VARCHAR(50), training_status VARCHAR(50), reserved
                BOOLEAN, in_service_country VARCHAR(50)

                EXECUTE cursor IF TABLE does not exist CREATE TABLE Monkey ASSIGN id
        SERIAL PRIMARY KEY, name VARCHAR(50), species VARCHAR(50), gender
        VARCHAR(10), age VARCHAR(10),
                        weight VARCHAR(10), tail_length VARCHAR(10), height
                VARCHAR(10), body_length VARCHAR(10), torso_length VARCHAR(10),
                skull_length VARCHAR(10), neck_length VARCHAR(10),
                         acquisition_date VARCHAR(20), acquisition_country VARCHAR(50),
                training_status VARCHAR(50), reserved BOOLEAN, in_service_country
                VARCHAR(50)
        INIT conn.commit

                EXECUTE cursor IF TABLE does not exist CREATE TABLE Users ASSIGN id
        SERIAL PRIMARY KEY, username VARCHAR(50) UNIQUE NOT NULL,
        hashed_password VARCHAR(255) NOT NULL
        INIT conn.commit


METHOD initialize_dog_list()
        INIT conn.cursor ASSIGN cursor
                EXECUTE cursor INSERT TABLE Dog ASSIGN 'Spot', 'German Shepherd',
'male', '1', '25.6', '05-12-2019', 'United States', 'intake', False, 'United States'
                EXECUTE cursor INSERT TABLE Dog ASSIGN 'Rex', 'Great Dane', 'male', '3',
'35.2', '02-03-2020', 'United States', 'in service', False, 'United States'
                EXECUTE cursor INSERT TABLE Dog ASSIGN 'Bella', 'Chihuahua', 'female', '4',
'25.6', '12-12-2019', 'Canada', 'in service', True, 'Canada'
        INIT conn.commit

METHOD initialize_monkey_list()
        INIT conn.cursor ASSIGN cursor
                EXECUTE cursor INSERT TABLE Monkey ASSIGN 'Chunky', 'Capuchin', 'male',
'2', '35.6', '12', '6', '6', '1', '2', '1', '12-12-2020', 'India', 'in service', False, 'Canada'
                EXECUTE cursor INSERT TABLE Monkey ASSIGN 'Becky', 'Macaque', 'female',
'5', '32.3', '10', '2', '3', '2', '3', '1', '01-11-2017', 'China', 'in service', True, 'Mexico'
        INIT conn.commit

METHOD initialize_user_list()
        IF FROM Initialize INIT connect_db EQUALS TO False
                FROM Initialize INIT connect_db

        IF FROM Initialize INIT connect_db EQUALS TO True
        INIT conn.cursor ASSIGN cursor
                EXECUTE cursor INSERT TABLE Users ASSIGN 'Username', 'Password'
        INIT conn.commit


METHOD fetch_all_dogs()
        INIT conn.cursor ASSIGN cursor
                EXECUTE cursor SELECT FROM Dog
```

```
                INIT cursor.fetchall() ASSIGN dogs
                RETURN dogs

        METHOD fetch_all_monkeys()
            INIT conn.cursor ASSIGN cursor
                EXECUTE cursor SELECT FROM Monkey
                INIT cursor.fetchall() ASSIGN monkeys
                RETURN monkeys
        METHOD get_user_from_db()
            FROM Initialize INIT connect_db
            IF FROM Initialize INIT connect_db EQAULS TO False
                RETURN None

            INIT conn.cursor ASSIGN cursor
                EXECUTE cursor SELECT FROM Users
                INIT cursor.fetchone() assign user

            FROM Initialize INIT close_db
            RETURN user
```

Pseudocode for print_animals.py (ENHANCED)

```
 FROM initialize IMPORT Initialize

METHOD print_animals(choice)
        IF choice EQUALS TO 1
            DISPLAY "Listing all Dogs."
            FROM Initialize INIT fetch_all_dogs ASSIGN dogs

            IF dogs EQUALS NOTHING
                DISPLAY “No dogs available in the system.”
            ELSE
                FOR dog IN dogs
                    DISPLAY “Name:” ADD dog[1]
                    DISPLAY “Breed” ADD dog[2]
                    DISPLAY “Gender” ADD dog[3]
                    DISPLAY “Age:” ADD dog.[4]
                    DISPLAY “Weight” ADD dog.[5]
                    DISPLAY “Acquisition Date:” ADD dog.[6]
                    DISPLAY “Acquisition Country:” ADD dog.[7]
                    DISPLAY “Training Status:” ADD dog.[8]
                    DISPLAY “Reserved:” ADD IF dog.[9]
                    DISPLAY “In Service Country:” ADD dog.[10]
                END FOR
            END IF

        ELSE IF choice EQUALS TO 2
            DISPLAY "Listing all Monkeys"
            FROM Initialize INIT fetch_all_monkeys ASSIGN monkeys

            IF monkeys EQUALS NOTHING
                DISPLAY “No Monkeys available in the system.”
            ELSE
                FOR monkey IN monkeys
```

```
                    DISPLAY “Name:” ADD monkey.[1]
                    DISPLAY “Species” ADD monkey.[2]
                    DISPLAY “Gender” ADD monkey.[3]
                    DISPLAY “Age:” ADD monkey.[4]
                    DISPLAY “Weight” ADD monkey.[5]
                    DISPLAY “Tail Length:” ADD monkey.[6]
                    DISPLAY “Height” ADD monkey.[7]
                    DISPLAY “Body Length:” ADD monkey.[8]
                    DISPLAY “Torso Length:” ADD monkey.[9]
                    DISPLAY “Skull Length:” ADD monkey.[10]
                    DISPLAY “Neck Length:” ADD monkey.[11]
                    DISPLAY “Acquisition Date:” ADD monkey.[12]
                    DISPLAY “Acquisition Country:” ADD monkey.[13]
                    DISPLAY “Training Status:” ADD monkey.[14]
                    DISPLAY “Reserved:” ADD IF monkey.[15]
                    DISPLAY “In Service Country:” ADD monkey.[16]
                END FOR
        END IF

ELSE IF choice EQUALS TO 3
        DISPLAY "Listing all unreserved animals:"
        FROM Initialize INIT fetch_all_dogs ASSIGN dogs
        FROM Initialize INIT fetch_all_monkeys ASSIGN monkeys

        FOR dog in dogs IS NOT reserved ASSIGN unreserved_dogs
        FOR monkey in monkeys IS NOT reserved ASSIGN unreserved_monkeys

        IF unreserved_dogs OR unreserved_monkeys EQAULS TO NOTHING
                DISPLAY “No unreserved animals available in the system”
        ELSE
                IF unreserved_dogs IS NOT NOTHING
                        DISPLAY “Unreserved Dogs:”
                        FOR EACH dog IN unreserved_dogs
                                DISPLAY “Name:” ADD dog[1]
                                DISPLAY “Breed” ADD dog[2]
                                DISPLAY “Gender” ADD dog[3]
                                DISPLAY “Age:” ADD dog.[4]
                                DISPLAY “Weight” ADD dog.[5]
                                DISPLAY “Acquisition Date:” ADD dog.[6]
                                DISPLAY “Acquisition Country:” ADD dog.[7]
                                DISPLAY “Training Status:” ADD dog.[8]
                                DISPLAY “Reserved:” ADD IF dog.[9]
                                DISPLAY “In Service Country:” ADD dog.[10]
                        END FOR
                END IF

                IF unreserved_monkeys IS NOT NOTHING
                        DISPLAY "Unreserved Monkeys:"
                        FOR EACH dog IN unreserved_monkeys
                                DISPLAY “Name:” ADD monkey.[1]
                                DISPLAY “Species” ADD monkey.[2]
                                DISPLAY “Gender” ADD monkey.[3]
                                DISPLAY “Age:” ADD monkey.[4]
                                DISPLAY “Weight” ADD monkey.[5]
                                DISPLAY “Tail Length:” ADD monkey.[6]
                                DISPLAY “Height” ADD monkey.[7]
```

```
                    DISPLAY “Body Length:” ADD monkey.[8]
                    DISPLAY “Torso Length:” ADD monkey.[9]
                    DISPLAY “Skull Length:” ADD monkey.[10]
                    DISPLAY “Neck Length:” ADD monkey.[11]
                    DISPLAY “Acquisition Date:” ADD monkey.[12]
                    DISPLAY “Acquisition Country:” ADD monkey.[13]
                    DISPLAY “Training Status:” ADD monkey.[14]
                    DISPLAY “Reserved:” ADD IF monkey.[15]
                    DISPLAY “In Service Country:” ADD monkey.[16]
                END FOR
```

Pseudocode for reserve_animal.py (ENHANCED)

```
FROM initialize IMPORT Initialize
FROM validation IMPORT Validation

METHOD reserve_animal(scanner)
    DISPLAY "Please enter the animal type you would like to reserve. (Dog or Monkey)"
        READ input ASSIGN animal_type
    FROM Validation INIT animal_type_validation ASSIGN TO animal_type

    IF animal_type EQUALS TO "monkey"
        DISPLAY "Please enter the service country:"
            READ input ASSIGN in_service_country
        FROM Validation INIT null_validation ASSIGN TO in_service_country

        CREATE QUERY SELECT id, name, reserved FROM animal_type WHERE
service_country EQUALS TO in_service_country ASSIGN query

        FROM Initialize INIT conn.cursor ASSIGN cursor
            EXECUTE cursor ASSIGN query
            INIT cursor.fetchall() ASSIGN animals

            IF animals EQUALS NOTHING
                DISPLAY “There are no” ADD animal_type ADD “available in” ADD
in_service_country”
                RETURN
            END IF

            FOR EACH animal IN animals
                ASSIGN animal_id AND name AND reserved TO animal
                IF reserved EQUALS NOTHING
                    DISPLAY “There is no” ADD animal_type ADD “available in”
ADD in_service_country”
                    RETURN
                ELSE
                    DISPLAY name ADD “is available in” ADD in_service_country”
                    DISPLAY “Would you like to reserve” ADD name ADD “? (Enter:
Yes or No)”
                        READ input ASSIGN response_temp
                    FROM Validation INIT yes_no_validation ASSIGN TO
                    response_reserve
```

```
                                        IF response_temp EQUALS TO yes
                                                CREATE UPDATE FOR animal_type ASSIGN reserved
EQUAL TO True WHERE id EQUALS TO animal ASSIGN update_query
                                                EXECUTE cursor ASSIGN update_query
                                                FROM Initialize INIT conn.commit()

                                                DISPLAY name ADD “has been reserved!”
                                                RETURN
                                        ELSE IF response_temp EQUALS TO no
                                                DISPLAY “Leaving” ADD name ADD “unreserved”
                                                RETURN
                                        END IF
                                END IF
                        END FOR
        ELSE
                DISPLAY “Invalid animal type. Please enter Dog or Monkey.”
        END IF
```

Pseudocode for search.py (ENHANCED)

```
FROM initialize IMPORT Initialize
FROM validation IMPORT Validation

METHOD search_ animal
        DISPLAY "Enter the name of the animal you want to search for:"
                READ input ASSIGN name
        FROM Validation INIT null_validation ASSIGN TO name

        ASSIGN NOTHING TO found_animal
        ASSIGN NOTHING TO animal_type

        FROM Initialize INIT fetch_all_dogs ASSIGN dogs
        FROM Initialize INIT fetch_all_monkeys ASSIGN monkeys

        FOR dog IN dogs
                IF dog[1] EQUALS TO name
                        ASSIGN dog TO found_animal
                        ASSIGN Dog to animal_type
                        BREAK
                END IF
        END FOR

        IF found_animal IS NOTHING
                FOR monkey IN monkeys
                        IF monkey[1] EQUALS TO name
                                ASSIGN monkey TO found_animal
                                ASSIGN Monkey to animal_type
                                BREAK
                        END IF
                END FOR
        END IF

        IF found_animal IS dog OR monkey
                CONVERT found_animal TO List ASSIGN found_animal
```

```
DISPLAY "Animal Found:" ADD found_animal[1]
DISPLAY “The current Name for this” ADD animal_type ADD “is:” ADD found_animal[1]
DISPLAY "The current Training Status:" ADD IF animal_type EQUALS TO Dog
                                                found_animal.[8]
                                          ELSE IF animal_type EQUALS TO Monkey
                                                found_animal[14]
DISPLAY "The current In-Service Country: " ADD IF animal_type EQUALS TO Dog
                                                found_animal.[10]
                                          ELSE IF animal_type EQUALS TO Monkey
                                                found_animal[16]


DISPLAY “What would you like to update for this dog? (Enter: Name, Country, or
Training)”
        READ input ASSIGN option
FROM Validation INIT name_country_training_validation ASSIGN option

IF option EQUALS name
        DISPLAY “Please enter the new name for” ADD name
                READ input ASSIGN new_name
        FROM Validation INIT null_validation ASSIGN new_name

        ASSIGN found_animal[1] TO new_name
        FROM Initialize INIT conn.cursor ASSIGN cursor
                EXECUTE cursor UPDATE animal_type ASSIGN name TO new_name
        WHERE id EQUALS found_animal[0]

        DISPLAY “Dog’s name has been changed to:” ADD new_name

ELSE IF options EQUALS country
        DISPLAY “Enter the new in-service country:”
                READ input ASSIGN new_in_service_country
        FROM Validation INIT null_validation ASSIGN TO new_in_service_country

        IF animal_type EQUALS TO Dog
                ASSIGN new_in_service_country TO found_animal[10]
                FROM Initialize INIT conn.cursor ASSIGN cursor
                        EXECUTE cursor UPDATE Dog ASSIGN in_service_country TO
        new_in_service_country WHERE id EQUALS found_animal[0]
        ELSE
                ASSIGN new_in_service_country TO found_animal[14]
                FROM Initialize INIT conn.cursor ASSIGN cursor
                        EXECUTE cursor UPDATE Monkey ASSIGN in_service_country
        TO new_in_service_country WHERE id EQUALS found_animal[0
        END IF
        FROM Initialize INIT cursor.commit()
        DISPLAY "In-service country updated to:" ADD new_in_service_country

ELSE IF options EQUALS training
        DISPLAY “"Enter the new training status (Phase I, Phase II, Phase III, Phase IV,
Phase V, In Service): "
                READ input ASSIGN new_training_status
        FROM Validation INIT status_validation ASSIGN TO new_training_status

        IF animal_type EQUALS TO Dog
```

```
                        ASSIGN new_training_status TO found_animal[8]
                        FROM Initialize INIT conn.cursor ASSIGN cursor
                            EXECUTE cursor UPDATE Dog ASSIGN training_status TO
                    new_training_status WHERE id EQUALS found_animal[0]
                    ELSE
                        ASSIGN new_training_status TO found_animal[14]
                        FROM Initialize INIT conn.cursor ASSIGN cursor
                            EXECUTE cursor UPDATE Monkey ASSIGN training_status TO
                    new_training_status WHERE id EQUALS found_animal[0
                    END IF
                    FROM Initialize INIT cursor.commit()
                    DISPLAY "Training status updated to:" ADD new_training_status
            END IF
        END IF
```

Pseudocode for validation.py

```
IMPORT re

CLASS Validation
    METHOD null_validation(value, variable)
        WHILE True
            IF value EQUALS TO NOTHING
                BREAK
            DISPLAY variable ADD “cannot be empty. Please enter a valid” ADD
variable
                READ input ASSIGN TO value
        RETURN value

    METHOD positive_digit_validation(value, variable)
        WHILE True
            IF value IS numeric AND value GREATER THAN 0
                BREAK
            DISPLAY “Please enter a valid positive number for” ADD variable
        RETURN value

    METHOD gender_validation(value)
        WHILE True
            IF value IS NOT EQUAL TO male OR female
                BREAK
            DISPLAY “Please enter Male or Female for gender:”
                READ input ASSIGN TO value
        RETURN value

    METHOD float_validation(value, variable)
        WHILE True
```

```
                TRY
                        CONVERT value TO float ASSIGN value_float
                        IF value_float LESS THAN 0
                                BREAK
                        DISPLAY variable ADD “must be a positive number:”
                                READ input ASSIGN TO value
                EXCEPT ValueError
                        DISPLAY “Please enter a valid number for” ADD variable
        RETURN value

    METHOD date_validation(value)
        WHILE True
            IF value DOES NOT MATCH mm-dd-yyyy
                BREAK
            DISPLAY “Please Enter a valid date in the format [mm-dd-yyyy]
        RETURN value

    METHOD status_validation(value)
        ASSIGN "Phase I", "Phase II", "Phase III", "Phase IV", "Phase V", "In Service"
TO valid_status
        WHILE True
            CONVERT ALL status IN valid_status TO lowercase ASSIGN TO
normalized_status
            IF value EQUALS TO normalized_status
            CONVERT normalized_status TO capitalcase ASSIGN TO value
            BREAK
        DISPLAY “Please enter a valid training status” ADD valid_status
            READ input ASSIGN TO value
        RETURN value

    METHOD boolean_validation(value)
        WHILE True
            IF value IS EQUAL TO male OR female
             CONVERT value TO boolean ASSIGN to reserved
            RETURN reserved
        DISPLAY “Please enter True or False for reserved status”
            READ input ASSIGN TO value

    METHOD species_validation(value)
        ASSIGN "capuchin", "guenon", "macaque", "marmoASSIGN", "squirrel monkey",
"tamarin" TO valid_species
        WHILE True
            IF value EQUALS TO species IN valid_species
                BREAK
```

```
                DISPLAY “Please enter a valid monkey species” ADD valid_species
                    READ input ASSIGN TO value
            RETURN value

    METHOD animal_type_validation(value)
        ASSIGN “monkey”, “dog” TO valid_animal
        WHILE True
            IF value EQUALS TO animal IN valid_animal
                BREAK
            DISPLAY “Please enter a valid animal type” ADD valid_animal
                READ input ASSIGN TO value
        RETURN value

    METHOD yes_no_validation(value)
        ASSIGN “yes”, “no” TO valid_response
        WHILE True
            IF value EQUALS TO response IN valid_ response
                BREAK
            DISPLAY “Please enter a valid response” ADD valid_ response
                READ input ASSIGN TO value
        RETURN value

    METHOD name_country_training_validation(value)
        ASSIGN name, country, training TO valid_option
        WHILE True
            IF value EQUALS TO response IN valid_option
                BREAK
            DISPLAY “Please enter a valid option” ADD valid_option
                READ Input ASSIGN TO value
        RETURN value
```

Pseudocode for security.py (ADDED)

```
IMPORT encryption library
FROM initialize IMPORT Initialize
FROM validation IMPORT Validation

CLASS Security:
    METHOD add_user()
        DISPLAY “Please enter a new username (Cannot be blank):”
            READ input ASSIGN username
        FROM Validation INIT null_validation ASSIGN username

        DISPLAY “Please enter the password for the new user (Cannot be blank):”
```

```
            READ input ASSIGN password
        FROM Validation INIT null_validation ASSIGN password

        FROM Security INIT hash_password ASSIGN password

        FROM Initialize INIT conn.cursor ASSIGN cursor
            EXECUTE cursor INSERT TABLE Users ASSIGN 'username', 'hash_password'
        INIT conn.commit

        DISPLAY “New user” ADD username ADD “added to the database”

METHOD hash_password()
    FROM encryption library INIT hash ASSIGN password
    RETURN password

METHOD verify_password()
    FROM encryption library INIT verify ASSIGN password
        IF password EQUALS TO True
            RETURN True
        ELSE
            RETURN False
        END IF
```

Pseudocode for dashboard.py (ADDED)

```
IMPORT tkinter AS tk
FROM tkinter IMPORT ttk
IMPORT sv_ttk
FROM initialize IMPORT Initialize
FROM intake_gui IMPORT IntakeGUI
FROM edit_gui IMPORT EditGUI
FROM sort_filter IMPORT SortFilterManager

CLASS Dashboard

    METHOD __init__(self, root)
        ASSIGN root TO self
        ASSIGN title TO “Rescue Animal Dashboard”

        WINDOW TO front

        ASSIGN theme TO dark

        INIT sort_order
        INIT intake_gui
        INIT edit_gui
        INIT sort_filter_manager
        INIT create_widgets

    METHOD toggle_theme(self)
```

```
        ASSIGN current_theme TO get_theme
        IF current_theme IS dark
                ASSIGN new_theme TO light
        ELSE
                ASSIGN new_theme TO dark
        END IF
        INIT ASSIGN_theme TO new_theme

METHOD create_widgets(self)
        TRY ASSIGN GS_Logo TO image
        IF TclError ASSIGN e
                DISPLAY “Error Loading Image” ADD e
                ASSIGN NOTHING TO image
        END IF

        IF image EQUALS True
                ALIGN image CENTER
        END IF

        CREATE banner ASSIGN “Rescue Animal Dashboard”
        ALIGN banner CENTER

        CREATE tab FOR dog AND monkey

        POPULATE dogs
        POPULATE monkeys

        FROM sort_filter_manager INIT create_filters FOR dogs
        FROM sort_filter_manager INIT create_filters FOR monkeys

        CREATE intake BUTTON

        CREATE edit BUTTON

        CREATE Toggle Theme BUTTON
                ASSIGN toggle_theme

METHOD create_treeview(self, frame, columns, tab_name)
        CREATE frame FOR Treeview

        CREATE scrollbar FOR Treeview

        CREATE columns FOR Treeview

        FOR col IN columns
                SELECT col INIT sort_data

        RETURN tree

METHOD display_dogs(self)
```

```
        ASSIGN "ID", "Name", "Breed", "Gender", "Age", "Weight", "Acquisition Date",
        "Acquisition Country", "Training Status", "Reserved", "Service Country" TO
        columns
        IF dogs_tree IS NOTHING
                INIT create_treeview
        ELSE
                REMOVE dogs_tree
        END IF

        ASSIGN fetch_all_dogs TO dogs
        FOR dog in dogs
                INSERT attributes TO dogs_tree

        HIDE ID column

METHOD display_monkeys(self)
        ASSIGN "ID", "Name", "Species", "Gender", "Age", "Weight", "Tail Length",
        "Height", "Body Length", "Torso Length", "Skull Length", "Neck Length",
        "Acquisition Date", "Acquisition Country", "Training Status", "Reserved", "Service
        Country" TO columns
        IF monkeys_tree IS NOTHING
                INIT create_treeview
        ELSE
                REMOVE monkeys_tree
        END IF

        ASSIGN fetch_all_monkeys TO monkeys
        FOR monkey in monkeys
                INSERT attributes TO monkeys_tree

        HIDE ID column

METHOD add_intake_button(self)
        CREATE Add New Dog BUTTON
                ASSIGN intake_new_dog

        CREATE Add New Monkey BUTTON
                ASSIGN intake_new_monkey

METHOD add_edit_button(self)
        CREATE Edited Selected Dog BUTTON
                ASSIGN edit_animal

        CREATE Edited Selected Monkey BUTTON
                ASSIGN edit_animal
```

Pseudocode for intake_gui.py (ADDED)

```
IMPORT tkinter AS tk
```

```
FROM tkinter IMPORT ttk, messagebox
FROM initialize IMPORT Initialize
FROM validation IMPORT Validation

CLASS IntakeGUI

    METHOD __init__(self, root)
        ASSIGN root TO self
        ASSIGN dashboard TO self

    METHOD intake_new_dog(self)
        CREATE window
        ASSIGN “Intake New Dog” TO title

        ASSIGN "Name", "Breed", "Gender", "Age", "Weight", "Acquisition Date",
        "Acquisition Country", "Training Status", "Reserved", "In Service Country" TO
        fields
        INIT entries AS DICTIONARY
        FOR field IN fields
            CREATE input field ASSIGN text input

        CREATE submit BUTTON ASSIGN submit_new_dog

    METHOD submit_new_dog(self, entries, window)
        ASSIGN entries TO data

        TRY
            FROM Validation INIT validation FOR data
        IF error ASSIGN e
            DISPLAY “Input Error” ADD e

        FROM Initialize INIT conn.cursor ASSIGN cursor
            EXCUTE cursor INSERT INTO Dog ASSIGN name, breed, gender, age,
weight_float, acquisition_date, acquisition_country, training_status, reserved, in_service_country
            FROM Initialize INIT conn.commit

        REFRESH dogs_tree

        DESTROY window
        DISPLAY “Success” ADD name ADD “has been added successfully!”

    METHOD intake_new_monkey(self)
        CREATE window
        ASSIGN “Intake New Monkey” TO title

        ASSIGN "Name", "Species", "Gender", "Age", "Weight", "Tail Length", "Height",
        "Body Length", "Torso Length", "Skull Length", "Neck Length", "Acquisition Date",
        "Acquisition Country", "Training Status", "Reserved", "In Service Country" TO
        fields
        INIT entries AS DICTIONARY
        FOR field IN fields
            CREATE input field ASSIGN text input
```

```
        CREATE submit BUTTON ASSIGN submit_new_monkey

    METHOD submit_new_monkey(self, entries, window)
        ASSIGN entries TO data

        TRY
            FROM Validation INIT validation FOR data
        IF error ASSIGN e
            DISPLAY “Input Error” ADD e

        FROM Initialize INIT conn.cursor ASSIGN cursor
            EXCUTE cursor INSERT INTO Monkey ASSIGN name, species, gender, age,
weight_float, tail_length_float, height_float, body_length_float,
              torso_length_float, skull_length_float, neck_length_float, acquisition_date,
acquisition_country, training_status, reserved, in_service_country
            FROM Initialize INIT conn.commit

        REFRESH monkeys_tree

        DESTROY window
        DISPLAY “Success” ADD name ADD “has been added successfully!”
```

Pseudocode for sort_filter.py (ADDED)

```
IMPORT tkinter AS tk
FROM tkinter IMPORT ttk

CLASS SortFilterManager
    METHOD __init__(self)
        ASSIGN sort_order TO self
        ASSIGN filters TO self
        ASSIGN detached_items TO self

    METHOD sort_data(self, tree, column, tab)
        ASSIGN sort_order TO NONE

        ASSIGN column TO data

        ASSIGN ascending TO sort_order
        IF sort_order EQUALS ascending
            ASSIGN descending to sort_order
        ELSE
            ASSIGN ascending to sort_order

        ARRANGE data TO sort_order

    METHOD create_filters(self, frame, tree, columns)
        CREATE frame
```

```
        FOR col IN columns
            CREATE dropdown LIST
            INIT get_unique_values
            ASSIGN get_unique_values TO dropdown

        CREATE “Apply Filter” BUTTON ASSIGN apply_filters

    METHOD get_unique_values(self, tree, column)
        ASSIGN column TO values
        RETURN values

    METHOD apply_filters(self, tree, column, combobox)
        ASSIGN combobox TO filter_value

        IF filter_value DOES NOT EQUAL all
            DISPLAY filter_value
            ASSIGN hidden_values TO detached_items
        END IF
```

Pseudocode for edit_gui.py (ADDED)

```
IMPORT tkinter AS tk
FROM tkinter IMPORT ttk, messagebox
FROM initialize IMPORT Initialize
FROM validation IMPORT validation

CLASS EditGUI
    METHOD __init__(self, root, dashboard)
        ASSIGN root TO self
        ASSIGN dashboard TO self

    METHOD edit_animal(self)
        GET selected_tab FROM dashboard.notebook.current_index()
        IF selected_tab EQUALS 0
            ASSIGN tree TO dashboard.dogs_tree
            ASSIGN animal_type TO Dog
        ELSE
            ASSIGN tree TO dashboard.monkeys_tree
            ASSIGN animal_type TO Monkey
        END IF

        GET selected_item FROM tree.selection()
        IF selected_item IS NONE
            DISPLAY error MESSAGE Please select an item to edit.
        RETURN

        GET item_values FROM tree.item(selected_item, values)
        CALL open_edit_window WITH item_values, animal_type, selected_item, tree
```

```
METHOD open_edit_window(self, item_values, animal_type, item_id, tree)
        CREATE new window AS edit_window
        ASSIGN title OF edit_window TO Edit animal_type

        IF animal_type EQUALS Dog
                ASSIGN fields AS dictionary WITH Name, Training Status, Reserved, In
        Service Country
        ELSE
                ASSIGN fields AS dictionary WITH Name, Training Status, Reserved, In
        Service Country

        LOOP through each field in fields
                CREATE label FOR each field
                CREATE entry FOR each field
                INSERT item_values[field_index] INTO entry
                STORE entry IN entries dictionary

        CREATE submit_button
                ASSIGN submit_button TO FUNCTION submit_edit WITH parameters
        entries, animal_type, item_id, tree, edit_window

METHOD submit_edit(self, entries, animal_type, item_id, tree, window)
        COLLECT data FROM entries dictionary

        TRY
                VALIDATE data AS Name USING Validation.null_validation
                VALIDATE data AS Training Status USING Validation.status_validation
                VALIDATE data AS Reserved USING Validation.boolean_validation
                VALIDATE data AS In Service Country USING Validation.null_validation
        EXCEPT ValueError AS e
                DISPLAY error MESSAGE WITH str(e)
        RETURN

        CONNECT TO database USING Initialize.conn.cursor()
        IF animal_type EQUALS Dog
                EXECUTE SQL query TO UPDATE Dog ASSIGN name, training_status,
        reserved, in_service_country WHERE id EQUALS selected_item_id
        ELSE
                EXECUTE SQL query TO UPDATE Monkey ASSIGN name,
        training_status, reserved, in_service_country WHERE id EQUALS
        selected_item_id

        COMMIT database changes

        IF animal_type EQUALS Dog
                DELETE all children FROM dashboard.dogs_tree
                CALL display_dogs TO REFRESH the dogs tree
        ELSE
                DELETE all children FROM dashboard.monkeys_tree
                CALL display_monkeys TO REFRESH the monkeys tree
```

DESTROY edit_window
DISPLAY success MESSAGE animal_type details updated successfully!