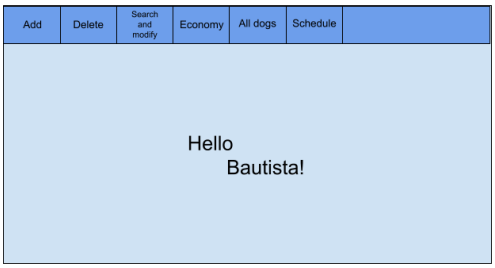
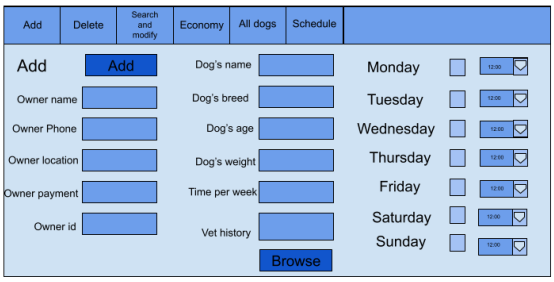
Criterion B:

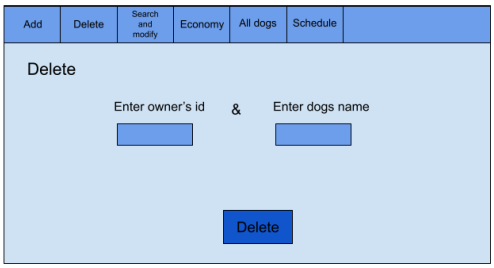
Test Plan:

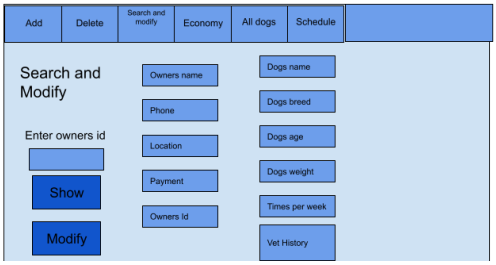
Test Type	Nature of Test	Example
It should be user-friendly	Everything will be computer-controlled. The only thing the user will have to do is to insert information.	An error will appear if the wrong string value is put on a number.
It should be able to store data after exiting the program.	The user will be able to leave and not worry about the information.	His program closes when it opens again, everything is still there.
It should be able to add dogs with all of the data.	The user should be able to enter all of the dog's information he requires.	Bautista enters a brand new dog into the system.
It should be able to tell the days where and when the dogs are taken out.	It should save specific days and time when the name is entered and where the house is located,	Bautista searches for a dog and the address appears.
Tell the user what the payment is monthly for a specific dog.	The software should be able to tell somewhere on the program how much is owed to him in monthly amounts.	Bautista searches for money he is owed, it appears.
It must be simple to work with monthly payments and modify them.	The software will allow the user to do simple math to calculate things like money and debts.	Bautista has a friend and gives him 20% off or rates change
There should be a way to store the image with the clinical vet information.	In the add, an area will be dedicated to entering the information about the clinical background of the dog.	Bautista saves a dog or modifies it, a picture can be inserted

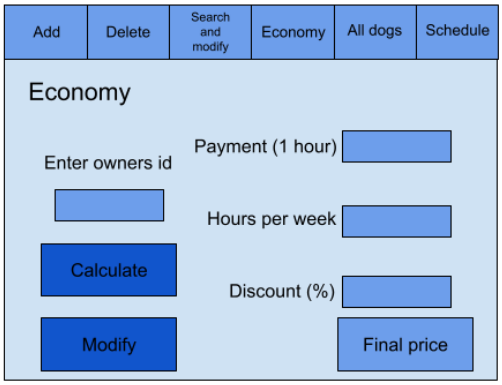
Prototypes:

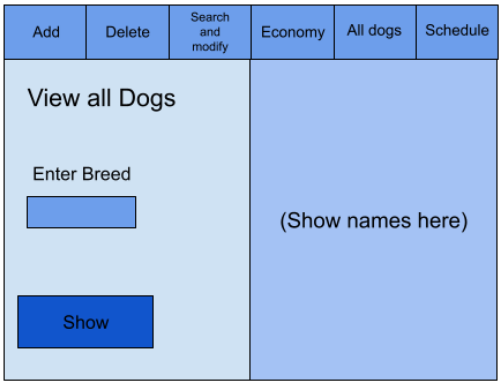

Add
Delete
Search and modify
Economy
All dogs
Schedule
Hello
Bautista!

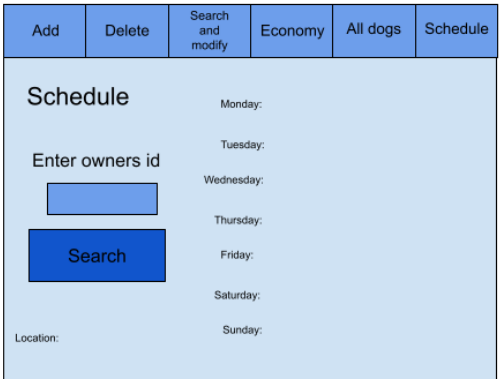

Add
Delete
Search and modify
Economy
All dogs
Schedule
Add
Add
Owner name
Owner Phone
Owner location
Owner payment
Owner id
Dog's name
Dog's breed
Dog's age
Dog's weight
Time per week
Vet history
Browse
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday


Add
Delete
Search and modify
Economy
All dogs
Schedule
Delete
Enter owner's id
&
Enter dogs name
Delete

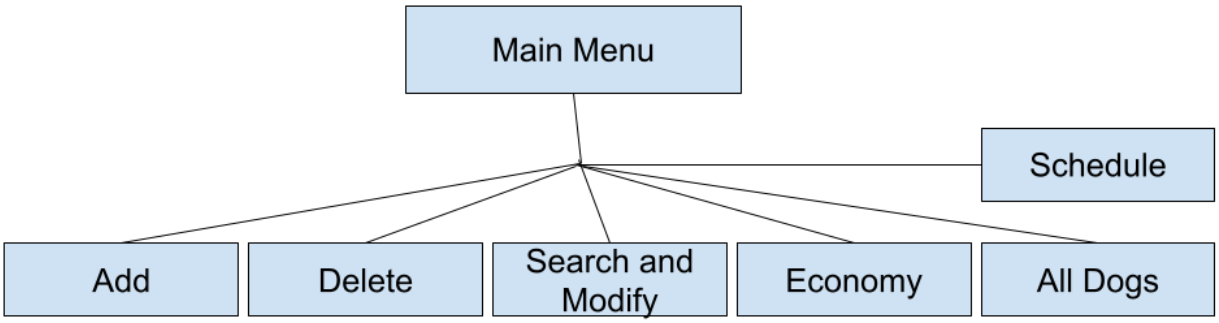

Add
Delete
Search and modify
Economy
All dogs
Schedule
Search and Modify
Enter owners id
Show
Modify
Owners name
Phone
Location
Payment
Owners Id
Dogs name
Dogs breed
Dogs age
Dogs weight
Times per week
Vet History


Add
Delete
Search and modify
Economy
All dogs
Schedule
Economy
Enter owners id
Payment (1 hour)
Hours per week
Calculate
Discount (%)
Modify
Final price


Add
Delete
Search and modify
Economy
All dogs
Schedule
View all Dogs
Enter Breed
Show
(Show names here)

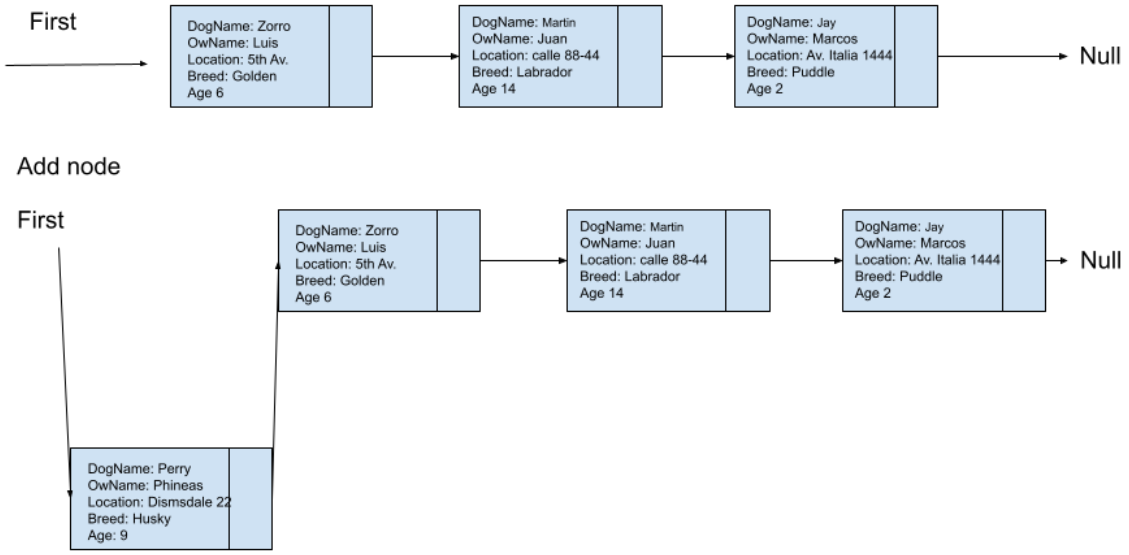

Add
Delete
Search and modify
Economy
All dogs
Schedule
Schedule
Enter owners id
Search
Location:
Monday:
Tuesday:
Wednesday:
Thursday:
Friday:
Saturday:
Sunday:

Modular Design:

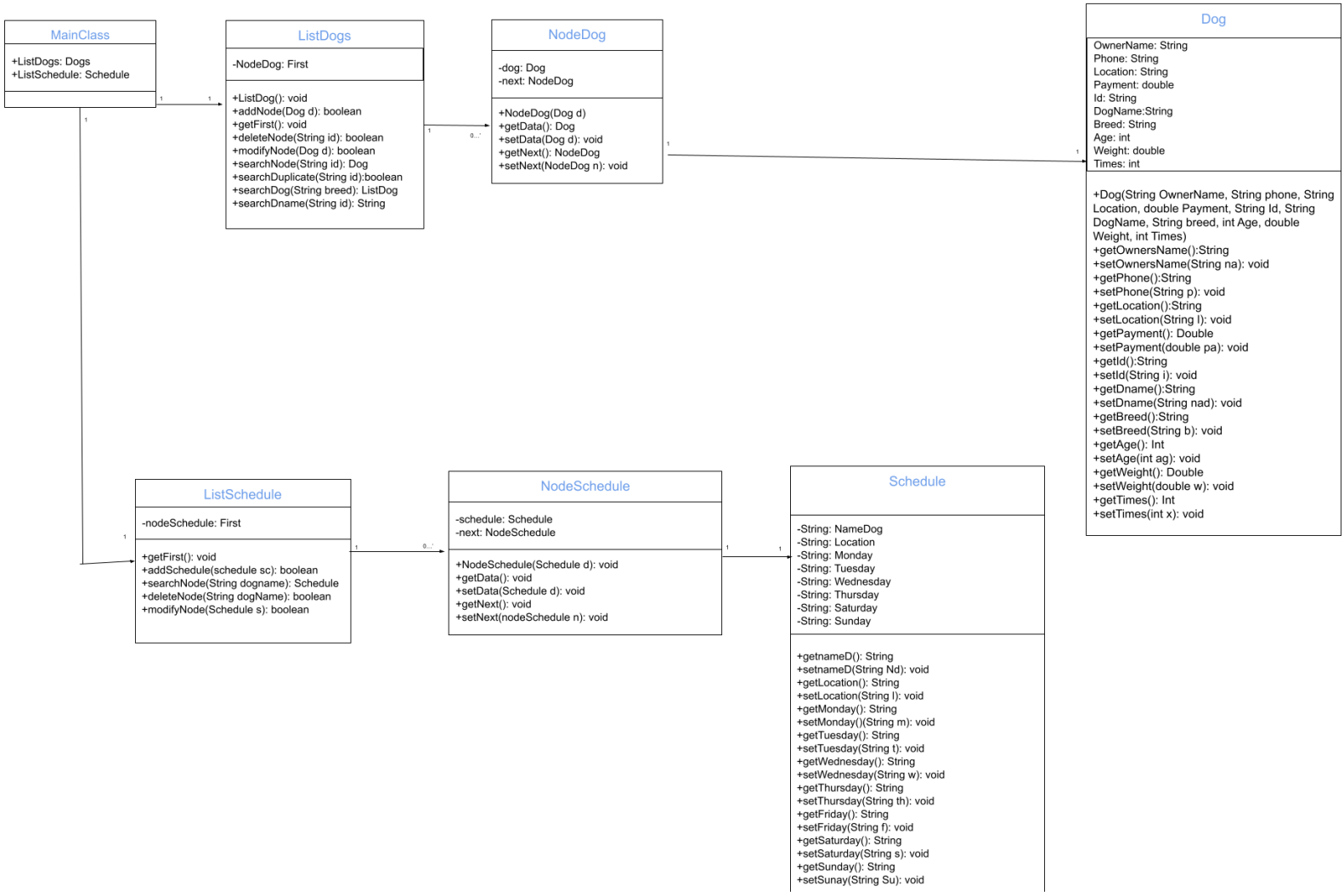


Data Structures:

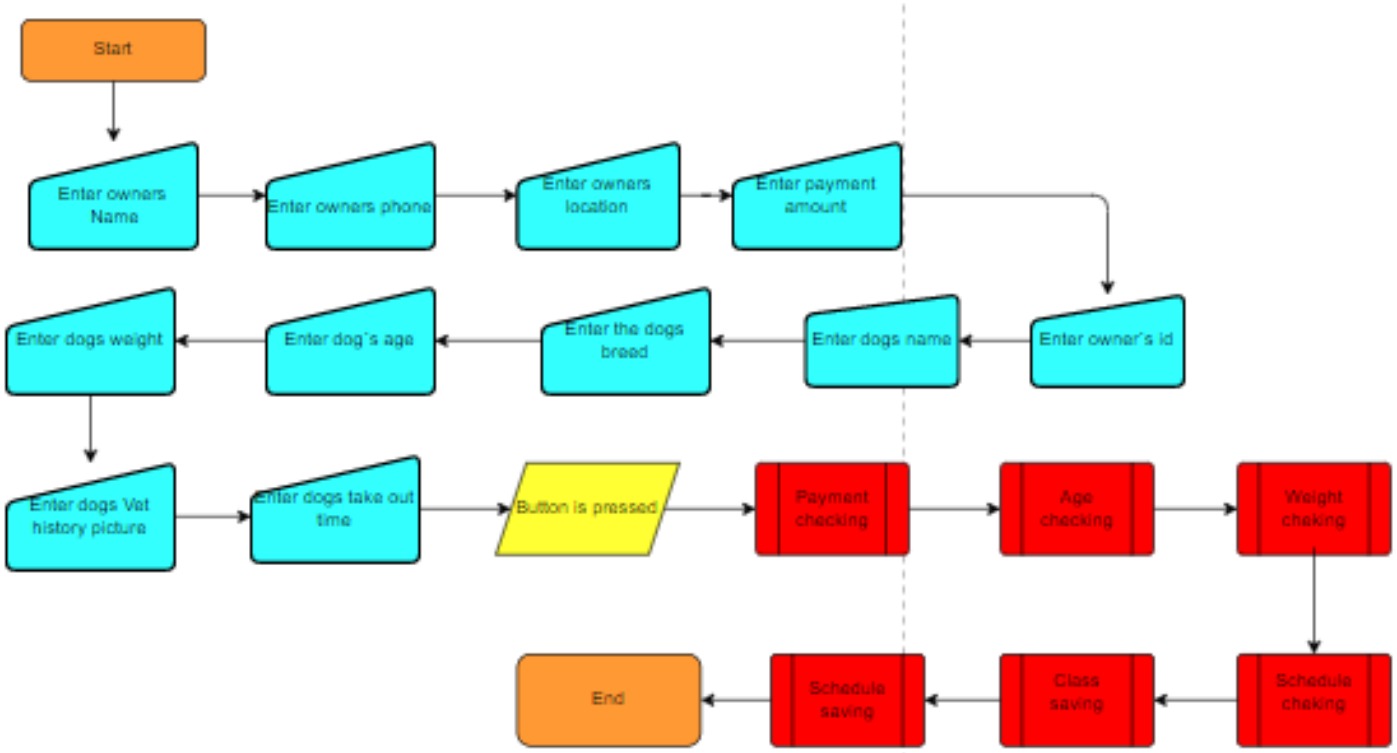
I will use linked lists for my data structures. Inside one, there will be an object named Dog, and in the other, the object will be scheduled. Inside these, there will be an object consisting of all the components, such as the dog's name, breed, and more. Since it's not a static data structure, I won't have to worry about the size. To tell my client which days the dogs need to be picked up, I will use a second linked list with two attributes from the previous list and a new one such as date and hour.

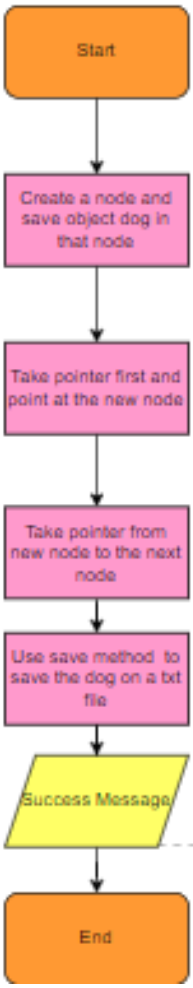


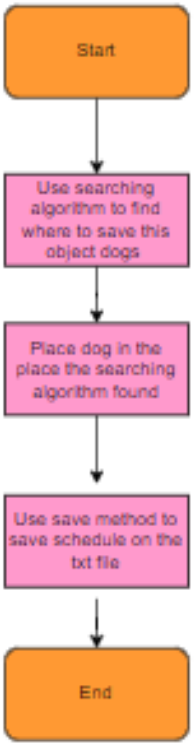
UML Diagram:

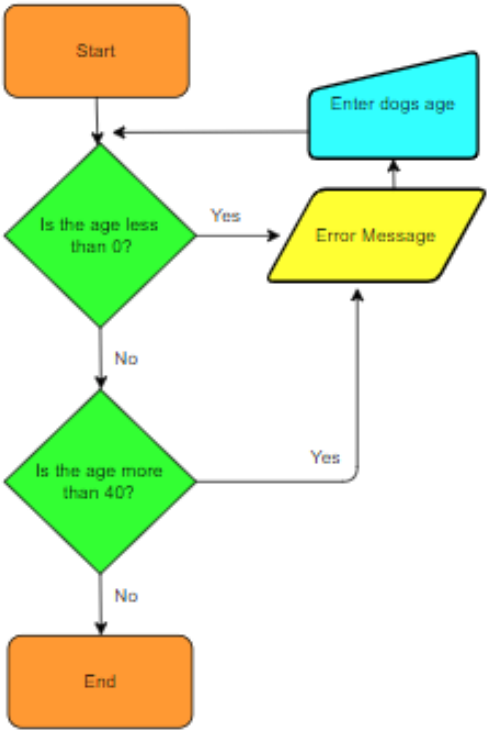

MainClass
+ListDogs: Dogs
+ListSchedule: Schedule
ListDogs
-NodeDog: First
+ListDog(): void
+addNode(Dog d): boolean
+getFirst(): void
+deleteNode(String id): boolean
+modifyNode(Dog d): boolean
+searchNode(String id): Dog
+searchDuplicate(String id):boolean
+searchDog(String breed): ListDog
+searchDname(String id): String
NodeDog
-dog: Dog
-next: NodeDog
+NodeDog(Dog d)
+getData(): Dog
+setData(Dog d): void
+getNext(): NodeDog
+setNext(NodeDog n): void
Dog
OwnerName: String
Phone: String
Location: String
Payment: double
Id: String
DogName:String
Breed: String
Age: int
Weight: double
Times: int
+Dog(String OwnerName, String phone, String Location, double Payment, String Id, String DogName, String breed, int Age, double Weight, int Times)
+getOwnersName():String
+setOwnersName(String na): void
+getPhone():String
+setPhone(String p): void
+getLocation():String
+setLocation(String l): void
+getPayment(): Double
+setPayment(double pa): void
+getId():String
+setId(String i): void
+getDname():String
+setDname(String nad): void
+getBreed():String
+setBreed(String b): void
+getAge(): Int
+setAge(int ag): void
+getWeight(): Double
+setWeight(double w): void
+getTimes(): Int
+setTimes(int x): void
ListSchedule
-nodeSchedule: First
+getFirst(): void
+addSchedule(schedule sc): boolean
+searchNode(String dogname): Schedule
+deleteNode(String dogName): boolean
+modifyNode(Schedule s): boolean
NodeSchedule
-schedule: Schedule
-next: NodeSchedule
+NodeSchedule(Schedule d): void
+getData(): void
+setData(Schedule d): void
+getNext(): void
+setNext(nodeSchedule n): void
Schedule
-String: NameDog
-String: Location
-String: Monday
-String: Tuesday
-String: Wednesday
-String: Thursday
-String: Saturday
-String: Sunday
+getnameD(): String
+setnameD(String Nd): void
+getLocation(): String
+setLocation(String l): void
+getMonday(): String
+setMonday()(String m): void
+getTuesday(): String
+setTuesday(String t): void
+getWednesday(): String
+setWednesday(String w): void
+getThursday(): String
+setThursday(String th): void
+getFriday(): String
+setFriday(String f): void
+getSaturday(): String
+setSaturday(String s): void
+getSunday(): String
+setSunay(String Su): void

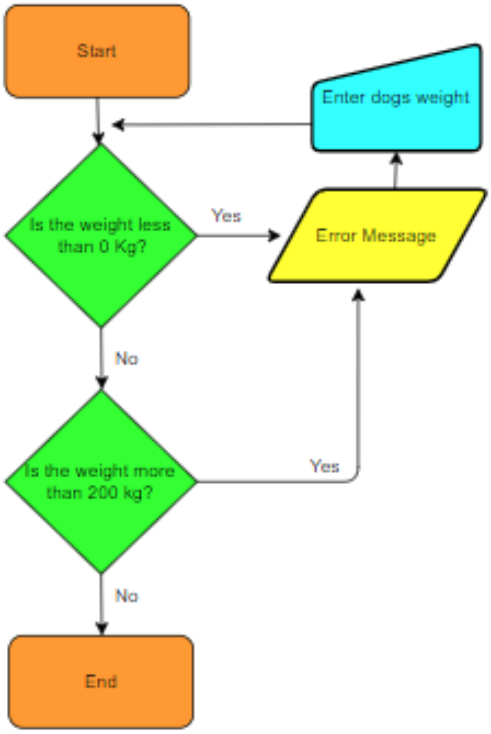
Flowchart:

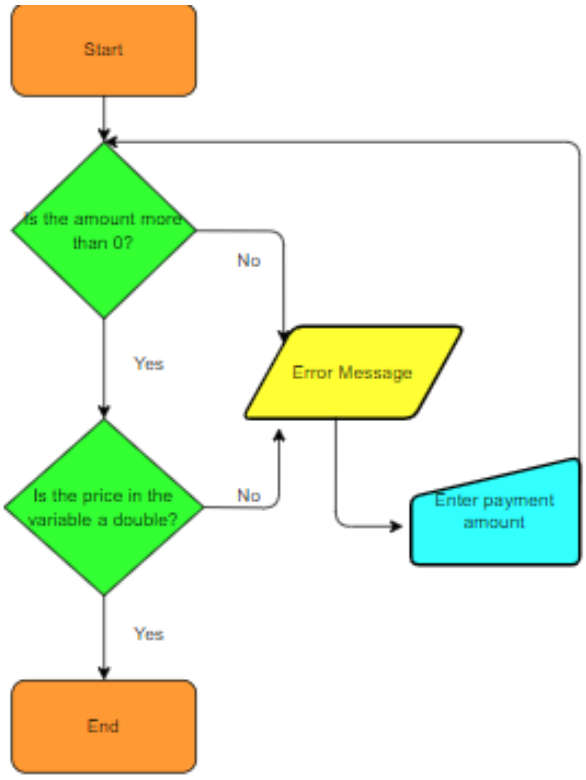

Start
Enter owners Name
Enter owners phone
Enter owners location
Enter payment amount
Enter owner's id
Enter dogs name
Enter the dogs breed
Enter dog's age
Enter dogs weight
Enter dogs Vat history picture
Enter dogs take out time
Button is pressed
Payment checking
Age checking
Weight cheking
Schedule cheking
Class saving
Schedule saving
End

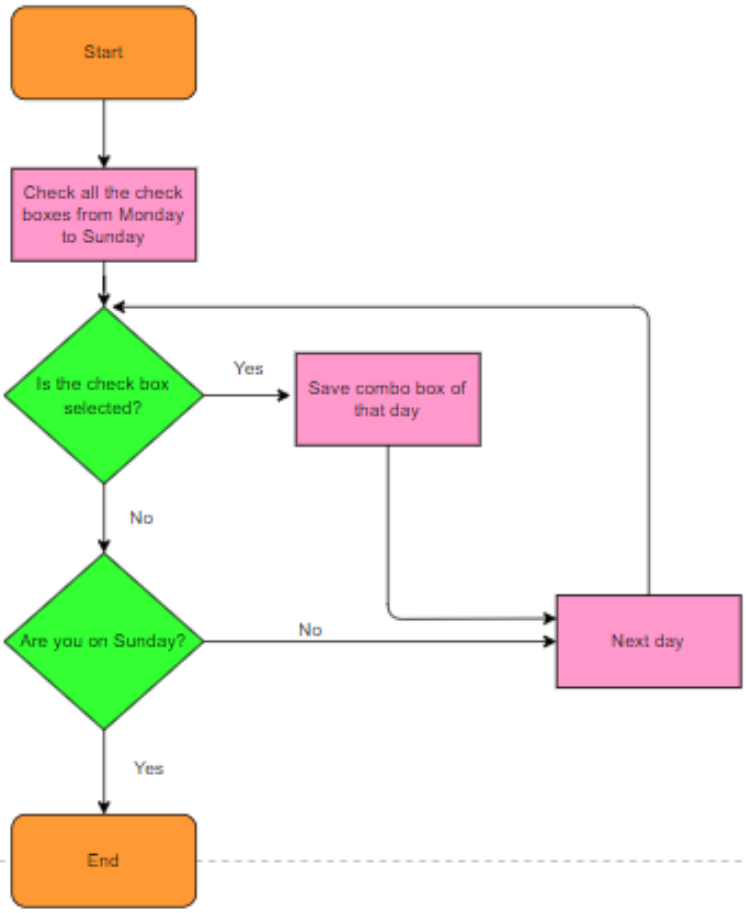

Start
Create a node and save object dog in that node
Take pointer first and point at the new node
Take pointer from new node to the next node
Use save method to save the dog on a txt file
Success Message
End

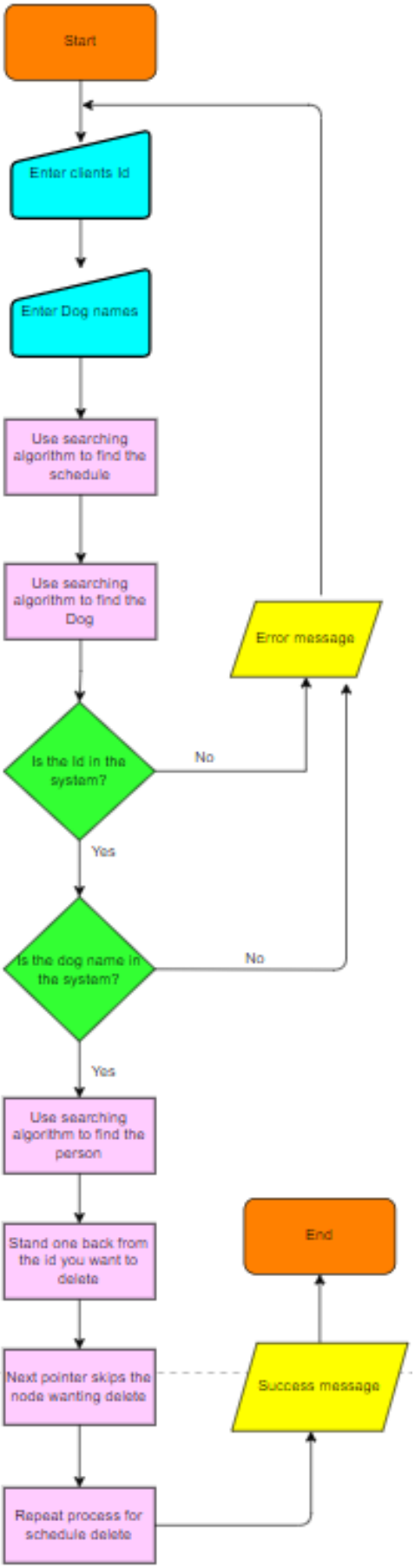

Start
Use searching algorithm to find where to save this object dogs
Place dog in the place the searching algorithm found
Use save method to save schedule on the txt file
End

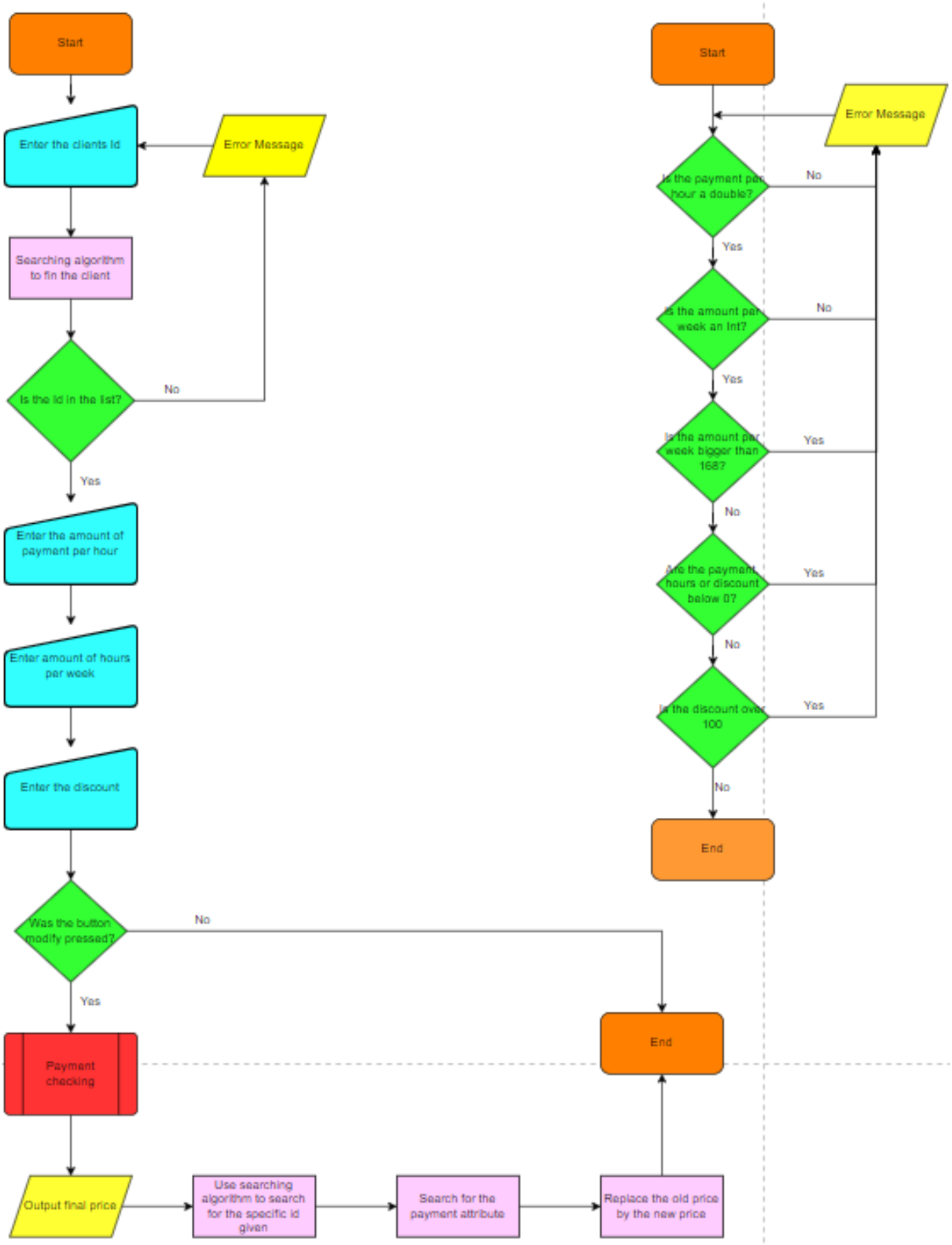

Start
Enter dogs age
Is the age less than 0?
Yes
Error Message
No
Is the age more than 40?
Yes
No
End

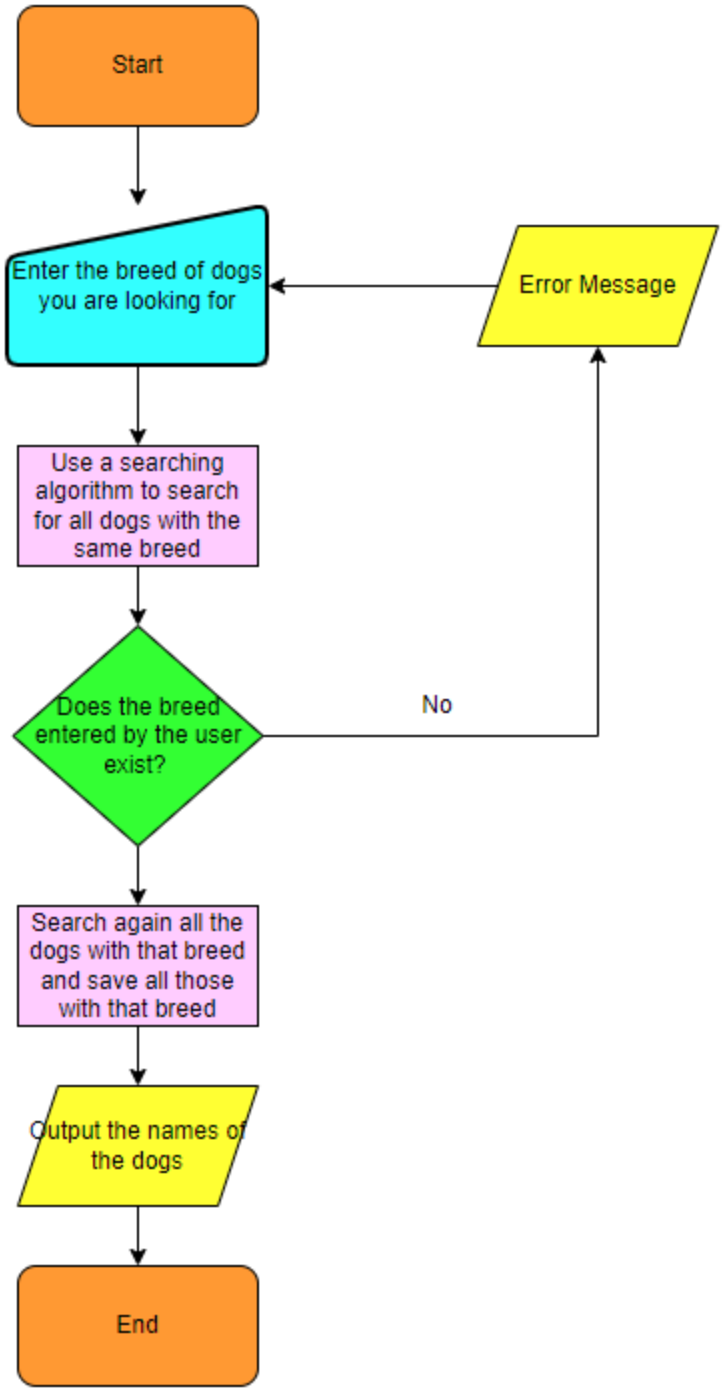

Start
Enter dogs weight
Is the weight less than 0 Kg?
Yes
Error Message
No
Is the weight more than 200 kg?
Yes
No
End

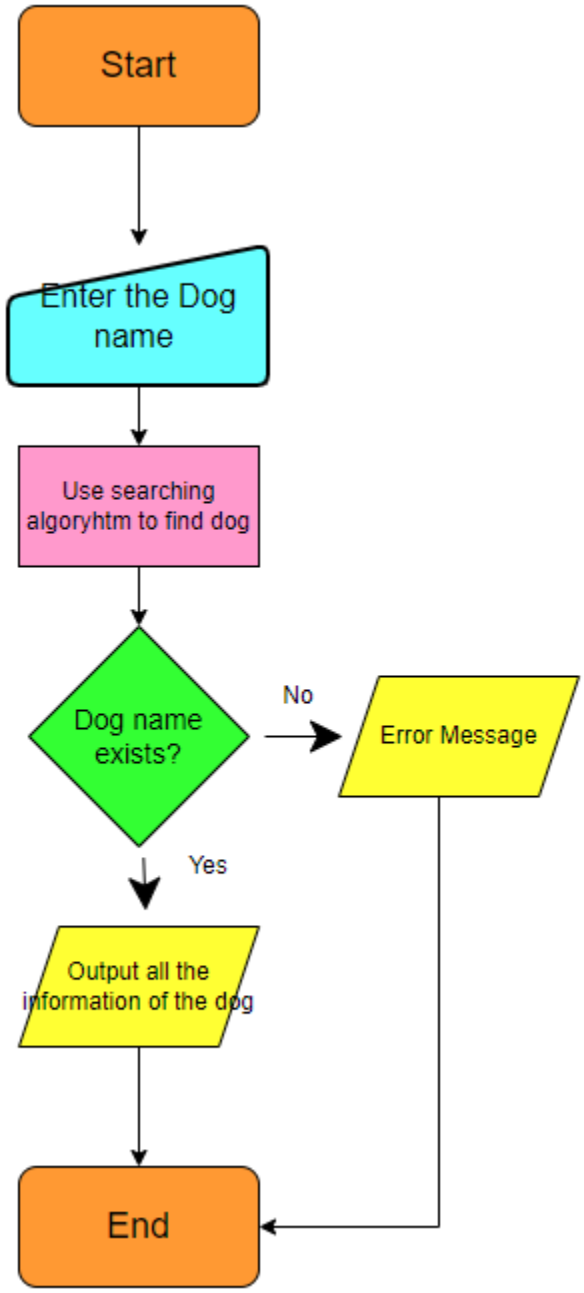

Start
Is the amount more than 0?
No
Yes
Error Message
Is the price in the variable a double?
No
Enter payment amount
Yes
End

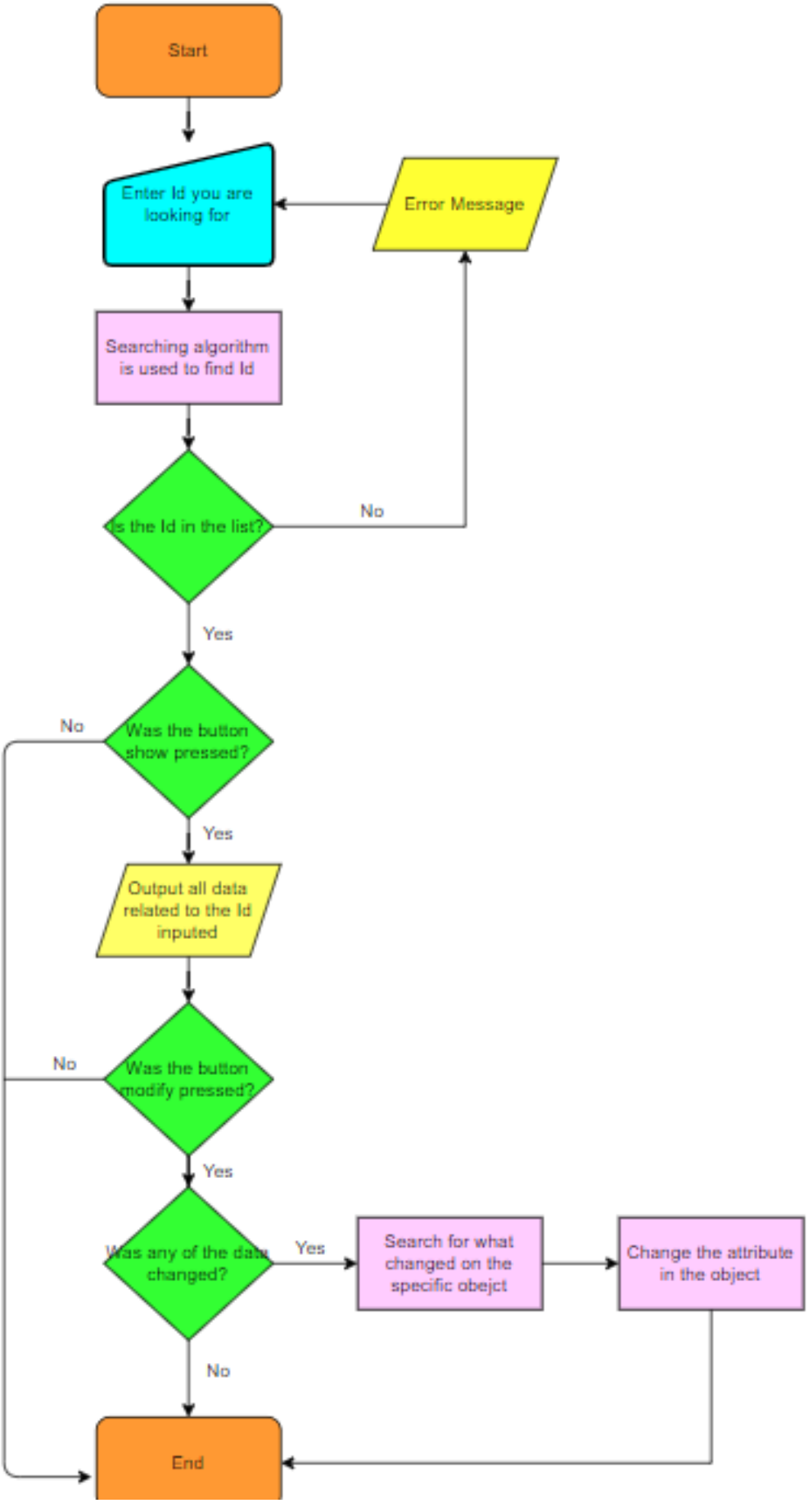

Start
Check all the check boxes from Monday to Sunday
Is the check box selected?
Yes
Save combo box of that day
No
Are you on Sunday?
No
Next day
Yes
End


Start
Enter clients Id
Enter Dog names
Use searching algorithm to find the schedule
Use searching algorithm to find the Dog
Error message
Is the Id in the system?
No
Yes
Is the dog name in the system?
No
Yes
Use searching algorithm to find the person
Stand one back from the id you want to delete
End
Next pointer skips the node wanting delete
Success message
Repeat process for schedule delete


Start
Enter the clients Id
Error Message
Searching algorithm to fin the client
Is the Id in the list?
No
Yes
Enter the amount of payment per hour
Enter amount of hours per week
Enter the discount
Was the button modify pressed?
No
Yes
Payment checking
Output final price
Use searching algorithm to search for the specific id given
Search for the payment attribute
Replace the old price by the new price
End
Start
Error Message
Is the payment per hour a double?
No
Yes
Is the amount per week an Int?
No
Yes
Is the amount per week bigger than 168?
Yes
No
Are the payment hours or discount below 0?
Yes
No
Is the discount over 100
Yes
No
End


Start
Enter the breed of dogs you are looking for
Error Message
Use a searching algorithm to search for all dogs with the same breed
Does the breed entered by the user exist?
No
Search again all the dogs with that breed and save all those with that breed
Output the names of the dogs
End


Start
Enter the Dog name
Use searching algoryhtm to find dog
Dog name exists?
No
Error Message
Yes
Output all the information of the dog
End


Start
Enter Id you are looking for
Error Message
Searching algorithm is used to find Id
Is the Id in the list?
No
Yes
Was the button show pressed?
No
Yes
Output all data related to the Id inputed
Was the button modify pressed?
No
Yes
Was any of the data changed?
Yes
Search for what changed on the specific obejct
Change the attribute in the object
No
End

Word count: 394