Designing phase

The design phase in ALIFY is an important step in translating the functional requirements into a well structured architecture . ALIFY is designed to adopt the client-server methodology since it has users (clients) interacting with a central server , and will consist of multiple components interacting with each other to provide a smooth experience for pet owners and vets. In this phase our team focus on establishing the overall system architecture ,data models and technical specifications .

system architecture

system architecture refers to high level of the application .It defines the components and how they interact with each other to form the entire system. And the building blocks (components) in ALIFY application are :

User Interface(UI): the front end where users (pet owners and vets) interact with the application. This includes screens for log in ,pet information , search, reminders and chat .

Admin Interface: a special interface for admins to make them able to monitor and manage the application data ,like (user profiles , pet information ,..etc).

\\ shown in figures below . or after finishing

Application Server : the back end components that processes and deals with user requests, like (searching for pets, creating reminders,..etc.) .it will handle business logic and communicate to database

Database : where all the application data (including user, vet, admin and pet information, vet chat content ,…etc) stored and retrieved.

Now that we defined the components , the interactions between them can be grouped in:

Users interactions with UI: different users will use UI in different manners to meet their needs ,like (pet owners will log in ,search and create reminders, vets will chat with different pet owners and admins will manage the whole information in the application).

The UI Communications with the Application Server: it is whenever the users performs an action ,the UI sends the request to the server .

Mutual Chatting : The chat functionality between pet owners and vets can be real-time (WebSocket) or message-based (stored in a database).

The application Server Interactions with Database : the server stores or retrieves the data from Database .

High security , communication protocol types and data management have to be considered To achieve the best architecture .

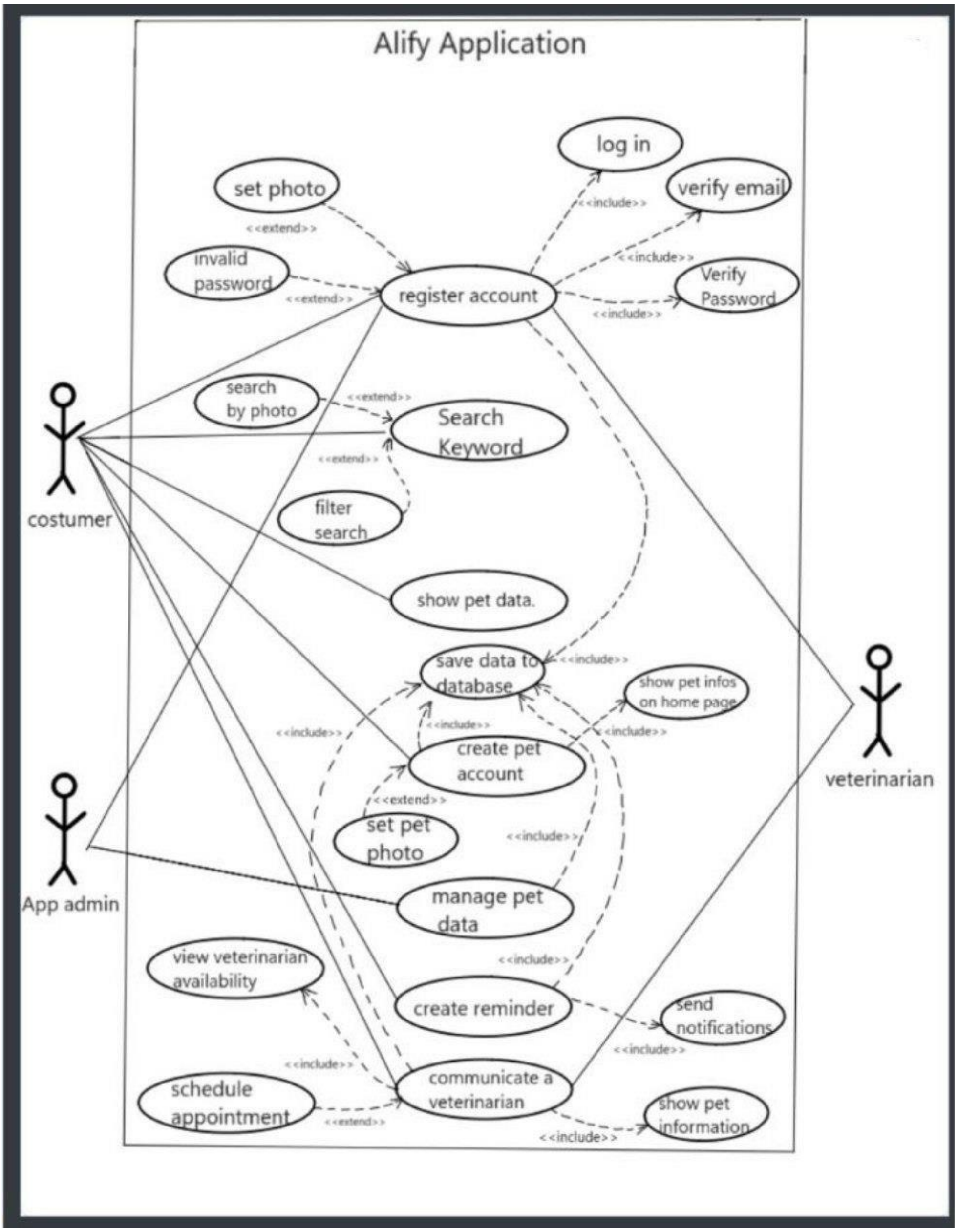
Data models

Now that we've defined the basic components of the system and how they interact with each other, we'll go into detail about a set of models that explain how the Alify app works.

Use Case Diagram for the Alify App: Here we will provide an overview of the interactions between different users (customer, app administrator, and veterinarian) and the different functions they can perform within the app.

These functions include:

Account registration, login, email verification, pet search, pet data management, creating reminders, communicating with veterinarians, and more.



Class Diagram for Alify App:

We will use class Diagram to manage pet data and it will contain:

1. App User: Manages user accounts

2. Customer: Controls creation, update and deletion of pet data

3. Pet Account: Contains details of pet accounts

4. Pet Data: Manages pet information such as gender and preferences

5. Veterinarian: Takes care of medical history and appointments

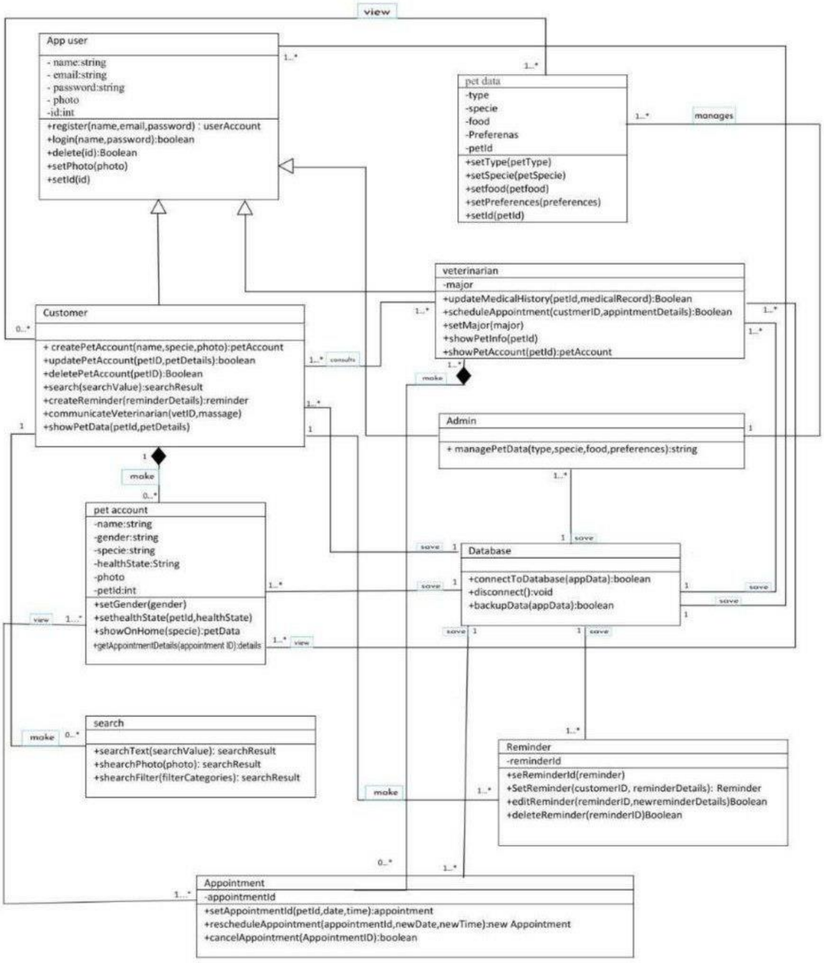
6. Admin: Manages pet data in general

7. Database: Takes care of connecting, disconnecting and data backup

8. Search: Searches data and photos

9. Reminder: Creates, edits and deletes reminders

10. Appointment: Manages and schedules appointments


view
App user
- name:string
- email:string
- password:string
- photo
-id:int
+register(name,email,password) : userAccount
+login(name,password):boolean
+delete(id):Boolean
+setPhoto(photo)
+setId(id)
1...*
1...*
pet data
-type
-specie
-food
-Preferenas
-petId
+setType(petType)
+setSpecie(petSpecie)
+setfood(petfood)
+setPreferences(preferences)
+setId(petId)
1...*
manages
veterinarian
-major
+updateMedicalHistory(petId,medicalRecord):Boolean
+scheduleAppointment(custmerID,appintmentDetails):Boolean
+setMajor(major)
+showPetInfo(petId)
+showPetAccount(petId):petAccount
1...*
1...*
1...*
1...*
make
Customer
0...*
+ createPetAccount(name,specie,photo):petAccount
+updatePetAccount(petID,petDetails):boolean
+deletePetAccount(petID):Boolean
+search(searchValue):searchResult
+createReminder(reminderDetails):reminder
+communicateVeterinarian(vetID,massage)
+showPetData(petId,petDetails)
1...*
consults
1...*
1
1
Admin
+ managePetData(type,specie,food,preferences):string
1
1...*
1
make
0...*
pet account
-name:string
-gender:string
-specie:string
-healthState:String
-photo
-petId:int
+setGender(gender)
+sethealthState(petId,healthState)
+showOnHome(specie):petData
+getAppointmentDetails(appointment ID):details
view
1...*
1...*
view
1...*
1 save
save 1
Database
save 1
+connectToDatabase(appData):boolean
+disconnect():void
+backupData(appData):boolean
1
save
save
1
save 1
1 save
search
make
0...*
+searchText(searchValue): searchResult
+shearchPhoto(photo): searchResult
+shearchFilter(filterCategories): searchResult
1...*
Reminder
-reminderId
+seReminderId(reminder)
+SetReminder(customerID, reminderDetails): Reminder
+editReminder(reminderID,newreminderDetails)Boolean
+deleteReminder(reminderID)Boolean
make
1...*
0...*
1...*
1...*
Appointment
-appointmentId
+setAppointmentId(petId,date,time):appointment
+rescheduleAppointment(appointmentId,newDate,newTime):new Appointment
+cancelAppointment(AppointmentID):boolean

System technical specifications

The technical specifications of ALIFY application outline the key technologies, tools and configurations required for the successful development and deployment of the system. These specifications ensure that the application meets the functional requirements and operates in a professional manner.

This section covers the necessary hardware, software, and network requirements, as well as scalability options.

Hardware requirements:

1.5 GHz or higher possessor

minimum 8 GB RAM

100-200 MB for Disk Space

Software requirements :

Windows 10 operating system or later

Java or C programming language

MySQL for relational Database

Network and communication:

TCP/IP communication protocols for reliable communication

HTTPS for secure communication

SSL/TLS for encrypting transferred data to maintain security

Scalability:

Adding more servers to distribute the load

Using indexing to handle growing data volume