ALIFY Application

Phase1 :

Introduction

Lots of applications are developed each day .But not every application serves its users properly . Only useful application are needed by users. Useful applications are characterized by their ability to solve daily problems faced by several people and make their life easier.

From this standpoint our team focused on developing an application (ALIFY)that solves a common problem faced by many people**.**

This phase will talk about a general analysis of the application including the purpose of developing and creating it, the problems that the application is meant to solve, and the proposed audience that will use the application.in addition to the necessary requirements must exist.

2.1 Application purpose:

ALIFY application is made to ease the life of pet owners, which helps them take care and raise their pets properly by giving appropriate information about their lifestyle.

2.2 The problems it solves:

Contributes to solve the problems faced by pet lovers raising pets and provides them sufficient expertise and knowledge to care of them, especially in areas lacking veterinary centers.

The proposed audience :

ALIFY is made for pet owners , people interested in pets in general and for the contributed vets.

Application Requirements

a study was conducted to determine the basic requirements needed to exist in the application. so it includes a set of functional and non functional requirements that make it achieve the purpose it was created for.

Functional requirements:

These are services the application should provide based on user needs ,which specify the different actions taken by the application to make users interact with . including:

The application Should allow the creation of a new account for each user containing his or her personal information: name, email, and password.

The application should allow users to search for the type of animal and information related to the specific animal.

The application Should allow information about the animals to be provided.

The application Should allow the creation of sub-accounts for each pet the user owns, including detailed pet data

The application Should allow a special interface to be provided to the administrator.

The application should allow users to create reminders for various pet care activities

The application Should allow a communication with veterinarians

Non functional requirements:

These are the functions that have to be in the application by default to ease the process of using the application and give users a seamless experience. Including :

The application should be compatible with android and iOS operating systems.

The application should make noise when sending notifications.

The application should work on the internet.

The application information should be edited only by the admin.

The application should be quick to response(5 sec max)

These requirements will ensure the application will work consistently and ensure a great user experience.

Requirements refining

in addition to previous requirements and to make sure being in touch with user real needs, a survey was conducted to gather other requirement users want to be in the application . and based on the responses -see in figure 1&2 below – a requirement refining was made to include the following :

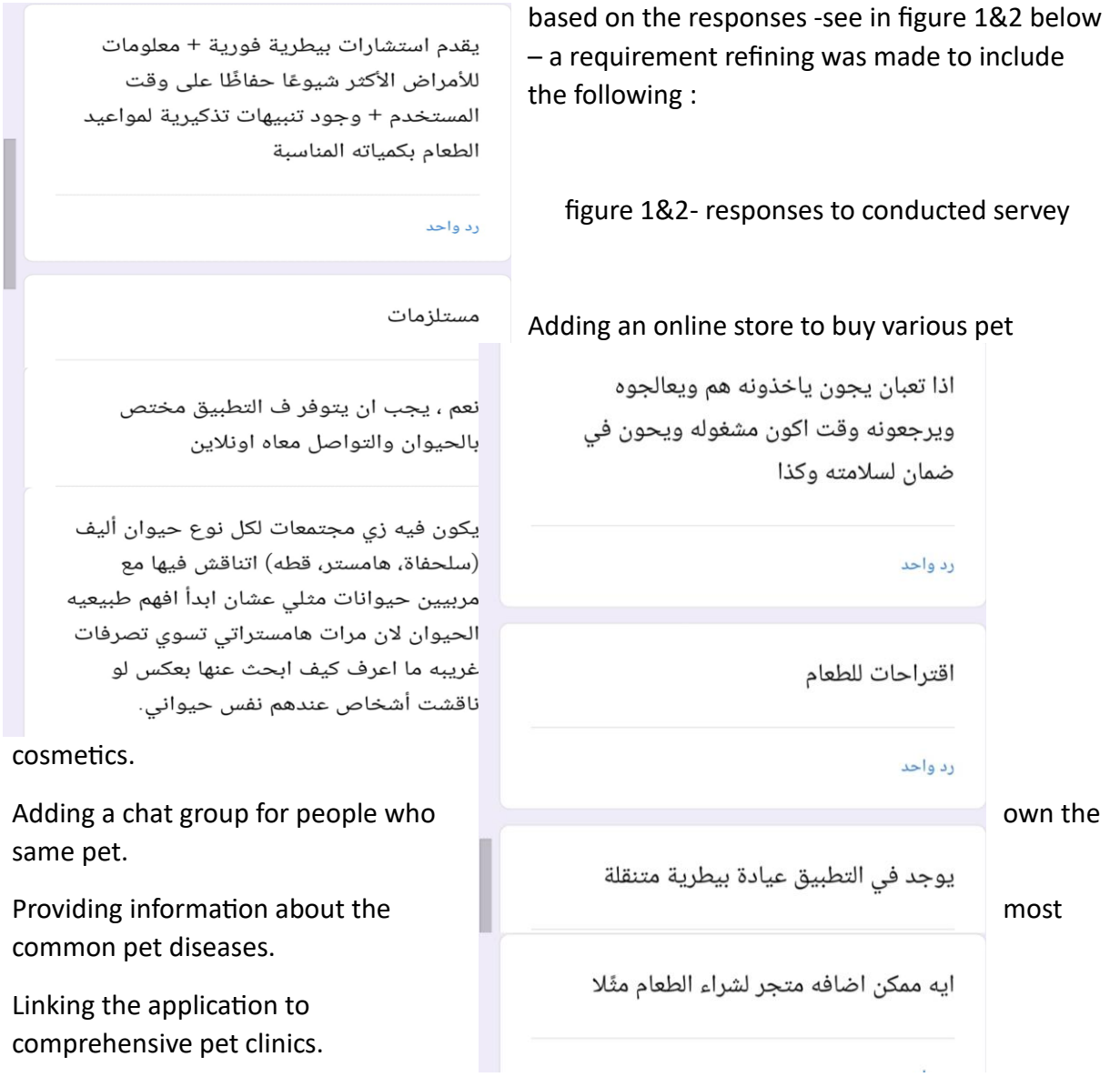

figure 1&2- responses to conducted servey

Adding an online store to buy various pet cosmetics.

Adding a chat group for people who own the same pet.

Providing information about the most common pet diseases.

Linking the application to comprehensive pet clinics.

Set notifications for sending funny information about pets.

Comparison of pet care applications:

After refining the functional and non-functional requirements of the ALIFY application, and incorporating valuable feedback from the conducted user survey, it is important to contextualize

these features within the existing landscape of similar applications. By comparing ALIFY with other pet care applications , we can better understand the unique strengths in ALIFY design.

WATCH PET**:**

Is a pet care application that Focuses on providing a virtual pet care experience and used by people who enjoy raising virtual pets on their smartphones. Providing a fun interactive experience related to caring for a virtual pet.

Advantages and disadvantages of watch pet application:

Advantages:

- It allows users to have an entertaining interactive experience of caring , feeding and playing with pets.
- Those who cannot have pets and would like to have one can get a virtual one.
- It offers fun challenges of caring a pet and keeping it healthy and happy

disadvantages:

- May not provide a deep realistic experience.
- Requires constant interaction with the pet via the screen, which affects the user's health.
- It is fun, but does not provide many benefits about raising a pet in real life.

ALIFY comes to solve these disadvantages by aiming to provide assistance in raising real pets, and offering the user all essential information- including practical tips and solutions on how to take care of pets- ensuring a more interactive and enjoyable experience for pet owners.

Phase 2

introduction

this phase will include the developed design of ALIFY. which 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).

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 .

UI prototype:

Prototypes are an optimal way to present a high abstracted view of systems . our team considered the importance of it and some User Interfaces were deigned to show how the application looks like- see in next figures 2&3&4 below.

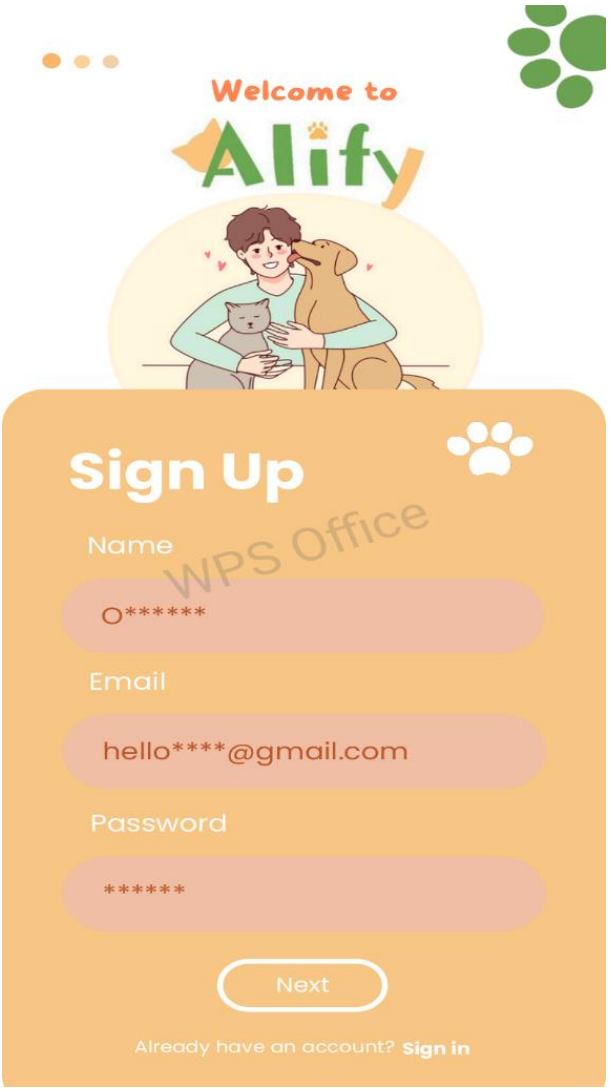

Figure 2-sign up interface

Figure 3-Home interface

Figure 3-pet account interface

Data models

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

Use Case Diagram for ALIFY Application : 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, pet search, pet data management, creating reminders, communicating with veterinarians, and more shown in figure 3 below

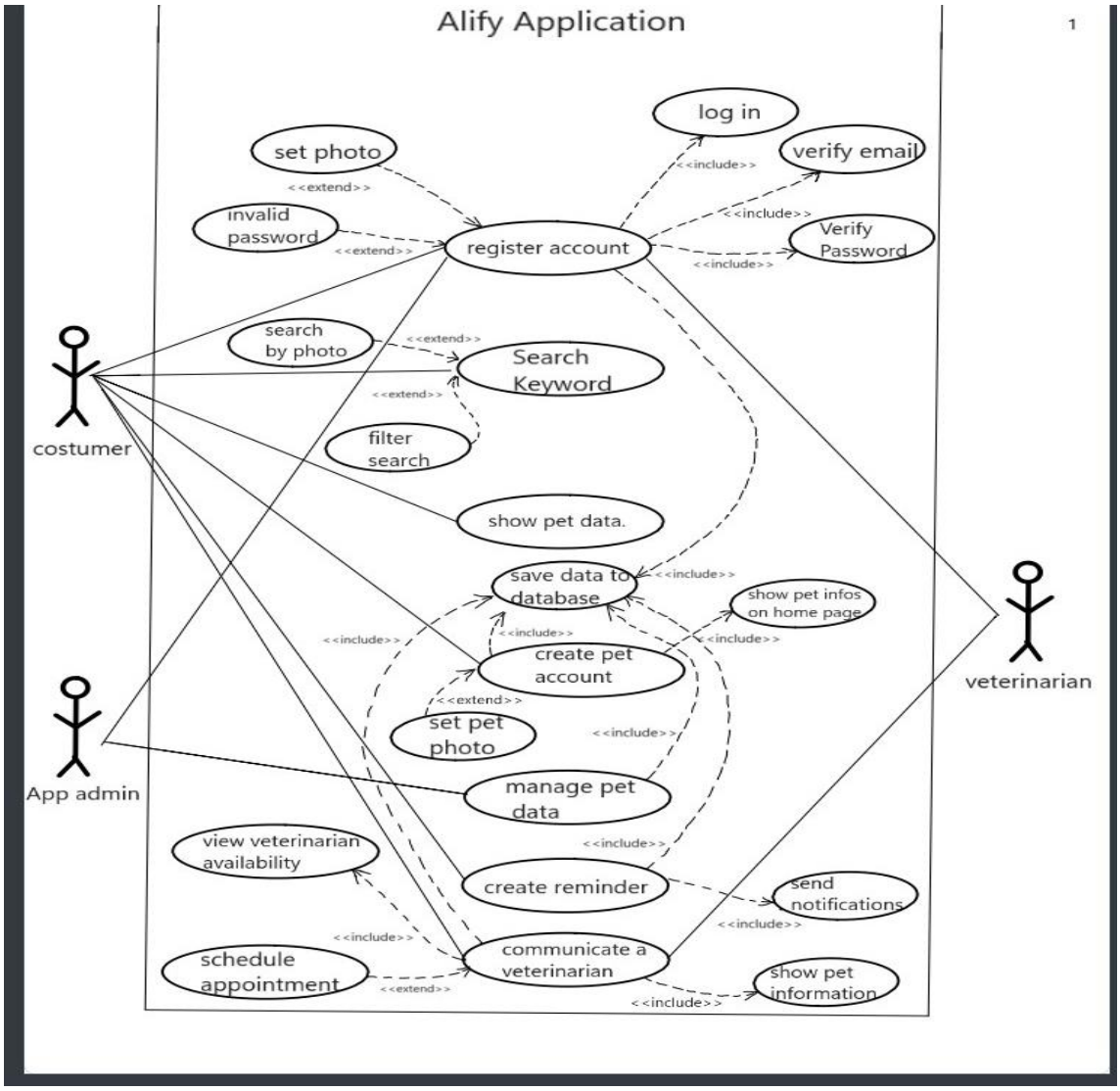

Figure 3- ALIFY use case

Class Diagram for ALIFY Application:

We will use class Diagram to show the integrated components consisting the application and their relationships as follows:

App User: this class manages user account functions like registration and log in.

Customer: Controls creation, update and deletion of pet data. So pet owners can create ,modify, and remove pet profiles.

Pet Account: Contains details of pet accounts and linked to the its owner profile for easy access and management.

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

Veterinarian: Takes care of medical history and appointments by viewing and modifying medical records and schedules

Admin: Manages pet data in general ,he has the access to all parts of the system

Database: manages connecting, disconnecting and data backup.

Search: Searches data and photos of pets

Reminder: Creates, edits and deletes reminders for each pet

Appointment: Manages and schedules appointments between vets and pet owners

-See the structured class diagram in figure 4 below-

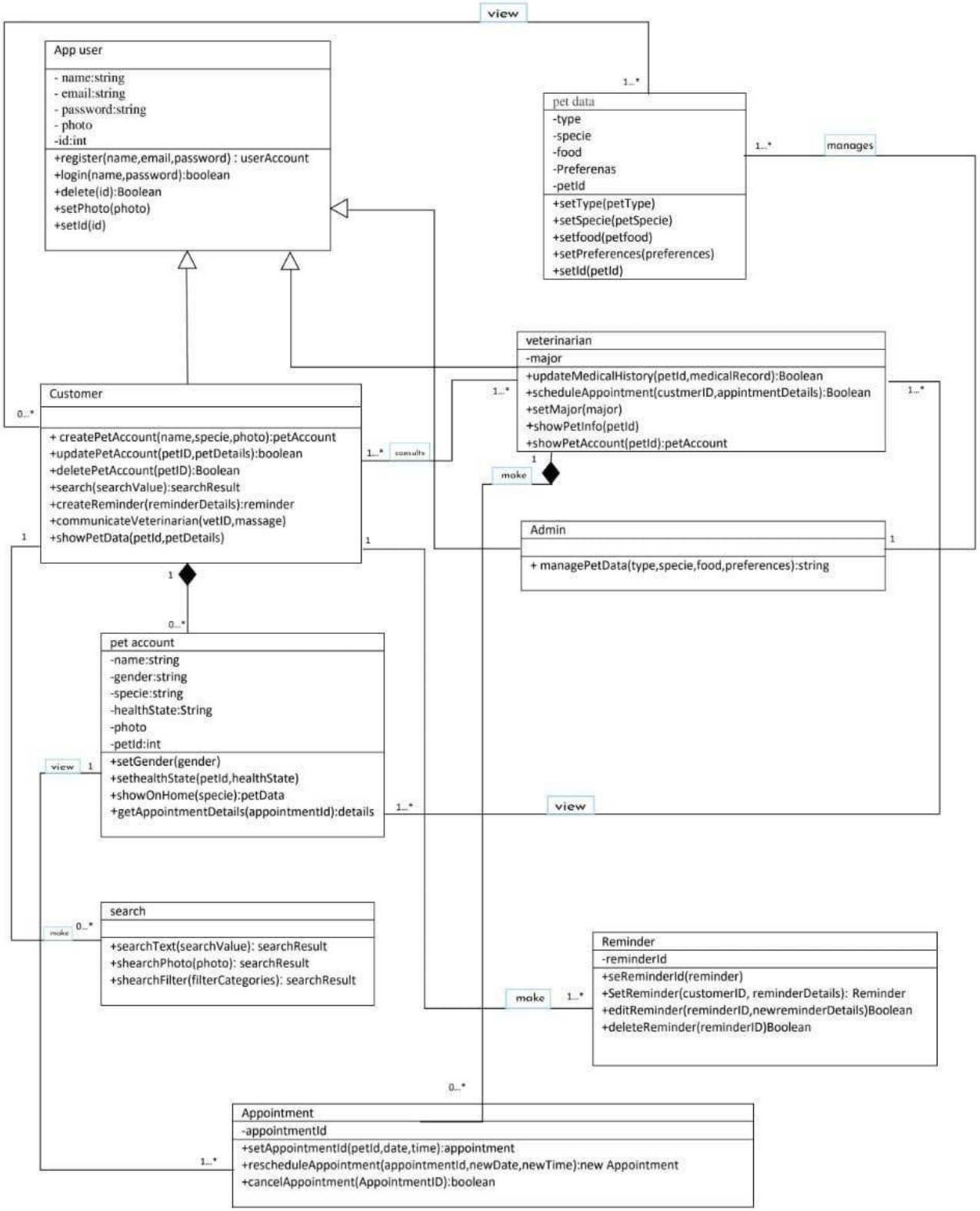

Figure 4 - class diagram of ALIFY

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