

Project Title: Pets’ Care System

Course Name: Introduction to Software Engineering

Supervisor: Dr. Rana Yousef  
Names of Students:

Salma Mohammad Alzubaidi – 0186541

Sara Maen Jarrar – 0187803

Asma Adel Ayyoub – 0187904

Razan Abdullah Alowedat - 0173400

Contents

[1.0 Introduction 7](#_heading=h.1fob9te)

[1.1 Background 7](#_heading=h.3znysh7)

[1.2 Problem definition 7](#_heading=h.2et92p0)

[1.3 Issues 8](#_heading=h.tyjcwt)

[1.4 Solutions 8](#_heading=h.3dy6vkm)

[1.5 Requirements 8](#_heading=h.1t3h5sf)

[1.6 Constraints 8](#_heading=h.2s8eyo1)

[1.7 Proposed Solution 9](#_heading=h.17dp8vu)

[1.8 Outline 9](#_heading=h.3rdcrjn)

[2.0 Feasibility Study 10](#_heading=h.26in1rg)

[2.1 Technical Feasibility 10](#_heading=h.lnxbz9)

[2.2 Operational Feasibility 10](#_heading=h.35nkun2)

[2.3 Economic Feasibility 10](#_heading=h.1ksv4uv)

[2.3.1 Costs: 10](#_heading=h.44sinio)

[2.3.2 Benefits: 12](#_heading=h.3j2qqm3)

[2.3.3 Cost-Benefit-Analysis: 12](#_heading=h.4i7ojhp)

[2.4 Schedule Feasibility 13](#_heading=h.1ci93xb)

[2.4.1 Gantt Chart 15](#_heading=h.2bn6wsx)

[2.4.2 Pert Diagram 15](#_heading=h.qsh70q)

[3.0 Software Requirements Document 16](#_heading=h.3as4poj)

[3.1 System Stakeholders and Requirements Sources 16](#_heading=h.1pxezwc)

[3.2 System Architecture 20](#_heading=h.49x2ik5)

[3.3 User Requirement Definition 20](#_heading=h.147n2zr)

[3.4 System Functional Requirement Specification 21](#_heading=h.3o7alnk)

[3.5 Non-Functional Requirements 24](#_heading=h.23ckvvd)

[3.6 Requirements Scenarios 25](#_heading=h.32hioqz)

[4.0 System Models and Design 34](#_heading=h.1hmsyys)

[4.1 Context Diagram 34](#_heading=h.41mghml)

[4.2 Use-case Model 35](#_heading=h.2grqrue)

[4.3 Scenarios 36](#_heading=h.3fwokq0)

[4.4 Sequence Diagram 47](#_heading=h.3l18frh)

[4.5 Class Diagram 48](#_heading=h.206ipza)

[4.6 Database Design 48](#_heading=h.2zbgiuw)

[4.7 Graphical User Interface Design 49](#_heading=h.3ygebqi)

[Appendix A 55](#_heading=h.3q5sasy)

[Links and references 61](#_heading=h.1x0gk37)

Table of Figures

[Figure 1. Gantt Chart](about:blank) 14

[Figure 2. Pert Diagram](about:blank) 14

[Figure 3: capture of the questionnaire](about:blank) 15

[Figure 4: capture of the questionnaire cont.](about:blank) 16

[Figure 5: capture of the questionnaire cont.](about:blank) 16

[Figure 6](about:blank) 16

[Figure 7: capture of the questionnaire cont.](about:blank) 17

[Figure 8: capture of the questionnaire cont.](about:blank) 17

[Figure 9: capture of the questionnaire cont.](about:blank) 18

[Figure 10: capture of the questionnaire cont.](about:blank) 18

[Figure 11: System Architecture. 19](#_heading=h.2p2csry)

[Figure 12: context diagram](about:blank) 33

[Figure 13: use-case model 34](#_heading=h.vx1227)

[Figure 14: sequence diagram](about:blank) 46

[Figure 15: class diagram 47](#_heading=h.4k668n3)

[Figure 16: database design 47](#_heading=h.1egqt2p)

[Figure 17: the entry of the system](about:blank) 48

[Figure 18: sign up page](about:blank) 48

[Figure 19: main menu of the system](about:blank) 49

[Figure 20: sign in page 49](#_heading=h.sqyw64)

[Figure 21: the user profile page Figure 22: the list of the owner’s pets 50](#_heading=h.3cqmetx)

[Figure 23: entering the pet’s information Figure 24: entering the pet’s information cont. 51](#_heading=h.1rvwp1q)

[Figure 25:the pet's QR code Figure 26: the pet's page 52](#_heading=h.4bvk7pj)

[Figure 27: QR code scanning page Figure 28: upcoming alarms page 53](#_heading=h.2r0uhxc)

[Figure 29: Alarm that appears on the user's phone 54](#_heading=h.1664s55)

[Figure 30: capture of the questionnaire’s responses. 55](#_heading=h.25b2l0r)

[Figure 31: capture of the questionnaire’s responses cont. 55](#_heading=h.kgcv8k)

[Figure 32: capture of the questionnaire’s responses cont. 56](#_heading=h.34g0dwd)

[Figure 33: capture of the questionnaire’s responses cont. 57](#_heading=h.1jlao46)

[Figure 34: capture of the questionnaire’s responses cont. 58](#_heading=h.43ky6rz)

[Figure 35: capture of the questionnaire’s responses cont. 59](#_heading=h.2iq8gzs)

[Figure 36: capture of the questionnaire’s responses cont. 60](#_heading=h.xvir7l)

[Figure 37: capture of the questionnaire’s responses cont. 60](#_heading=h.3hv69ve)

List of Tables

[Table 1. Development Costs 10](#_heading=h.2jxsxqh)

[Table 2. Operating Costs 11](#_heading=h.z337ya)

[Table 3. Tangible Benefits 12](#_heading=h.1y810tw)

[Table 4. Cost-Benefit Analysis 13](#_heading=h.2xcytpi)

[Table 5. Project Schedule 14](#_heading=h.3whwml4)

[Table 6: Non-Functional Requirements. 24](#_heading=h.ihv636)

[Table 7: use-case scenatio1 36](#_heading=h.1v1yuxt)

[Table 8: use-case scenatio2 37](#_heading=h.4f1mdlm)

[Table 9 :use-case scenatio3 38](#_heading=h.2u6wntf)

[Table 10 :use-case scenatio4 39](#_heading=h.19c6y18)

[Table 11 :use-case scenatio5 40](#_heading=h.3tbugp1)

[Table 12: use-case scenatio6 40](#_heading=h.28h4qwu)

[Table 13: use-case scenatio7 41](#_heading=h.nmf14n)

[Table 14: use-case scenatio8 42](#_heading=h.37m2jsg)

[Table 15: use-case scenatio9 43](#_heading=h.1mrcu09)

[Table 16: use-case scenatio10 44](#_heading=h.46r0co2)

[Table 17: use-case scenatio11 45](#_heading=h.2lwamvv)

[Table 18: use-case scenatio12 46](#_heading=h.111kx3o)

# Introduction

# Background

People have always been fond of having pets in their households. According to a recent survey, 67% of U.S. households own a pet. Whether it was common domestic animals like cats and dogs, or even exotic and possibly dangerous like spiders and snakes, pets are everywhere. This means that there are many people out there who need to take care of their pets properly, and in this ever so increasingly busy world, one would tend to forget to feed their pet, or perhaps forget to give them the scheduled vaccination they require, etc.

# Problem definition

Because of the human tendency to forget, caring for pets might prove difficult, especially when you have other things on your mind. People need a system that could remind them of the many things a pet needs to stay happy and healthy, and sometimes, even provide them with tips and information that they lack regarding dealing with their pets. They require a schedule that could be set for them, with reminders to perform the daily, weekly, or monthly tasks related to their pet’s nutrition, health, and happiness. Scheduled visits to the vet, taking the animal for a walk, intervals for feeding their pet, which differs from one animal to another, and such customized reminders and advices fit for the type of pet they have.

# Issues

1. Complicated user interfaces in some current systems contain many buttons and branched lists.
2. Spending a part of time on searching about pets’ care information.
3. Forgetting to do many tasks which are related to the pets’ care at a specific time.

# Solutions

1. Creating a Simple user interface that provides all the required functions (Easy interaction).
2. Attaching short tips and important information about pets’ care.
3. Setting an alarm clock that reminds the user to do the event which is required for pets.

# Requirements

* The system should be easy to use for nontechnical people.
* The system must allow users to create an unlimited list of reminders as well as specifying the exact time for each reminder to appear on their phone.
* The system must be able to remind user of what they want at the specified time.
* The system must have an interactive interface.
* The system must be simple, fast and must have low error rate.

# Constraints

* Development cost must not exceed 2000 JD.
* The system plan must be delivered due 22/May/2021.

# Proposed Solution

Application for smart phone operating systems. The aim of its establishment is to provide pet owners with services that help them to take care of their pets. The system enables users to select the type of their pets and enter the basic information about them, creates a default list of tasks that should be accomplished towards their pets, specifies the default time for this task, all based on the type of the pet, so the system will send a reminder to the user’s phone screen with the name of the pet and the task to be performed at this time, as well as providing users with some tips and advices to take care of their pet.

# Outline

In chapter 2.0 the project will discuss the feasibility study of the system by 20/March/2021. In chapter 3.0 the project will identify the human information requirements by 10/April/2021. In chapter 4.0 DFD, ERD and context diagrams will be designed, and the analysis of the new system will be made by 1/May/2021. In chapter 5.0 project will include the GUI design by 22/May/2021.

# Feasibility Study

# Technical Feasibility

The system is technically feasible. Given the current technical resources it is possible to develop the system and design an interface that is easy to use and understandable. The project team members have enough skills to accomplish the objectives of the system.

# Operational Feasibility

The system is operationally feasible. This system is effective in its environment because it facilitates the pats’ care process which benefits the user in many ways, so the system will be acceptable from most users.

# Economic Feasibility

### Costs:

1. Development costs:

*Table 1. Development Costs*

|  |  |
| --- | --- |
| Hardware & Software | |
| Development Server | 650 JD |
| DBMS Server Software | 350 JD |
| DBMS Client Software | 350 JD |
| Total | 1350 JD |

1. Operating costs:

*Table 2. Operating Costs*

|  |  |
| --- | --- |
| Personnel | |
| System Analyst | 3000 JD |
| Programmer | 2000 JD |
| GUI Designer | 700 JD |
| Database Specialist | 400 JD |
| Maintenance | |
| Maintenance of DBMS Software | 600 JD |
| Maintenance of Servers | 600 JD |
| Total | 7,300 JD |

### Benefits:

1. Tangible Benefits:

*Table 3. Tangible Benefits*

|  |  |
| --- | --- |
| Cost Reduction | 2,800 JD |
| Increased flexibility | 2,900 JD |
| Reduced Inventories | 2,500 JD |
| Labor Reduction | 1,800 JD |
| Total | 10,000 JD |

* + 1. Cost-Benefit-Analysis:  
       The discount rate is 0.7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Year | 0 | 1 | 2 | 3 | 4 |
| Costs (JD) | | | | | |
| Development Costs | 1,350 | 1,000 | 900 | 950 | 1,200 |
| Operating Costs | 7,300 | 7,000 | 6,500 | 6,800 | 7,200 |
| Total Costs | 8,650 | 8,000 | 7,400 | 7,750 | 8,400 |
| Discount | 1.000 | 0.588 | 0.346 | 0.203 | 0.12 |
| NPV of Costs | 8,650 | 4,704 | 2,560.4 | 1,573.25 | 1,008 |
| Accumulated Costs | 8,650 | 13,354 | 15,914.4 | 17,487.65 | 18,495.65 |
| Benefits (JD) | | | | | |
| Tangible Benefits | 0 | 10,000 | 12,500 | 30,000 | 80,000 |
| NPV of Benefits | 0 | 5,880 | 4,325 | 6,090 | 9,600 |
| Accumulated Benefits | 0 | 5,880 | 10,205 | 16,295 | 25,895 |

*Table 4. Cost-Benefit Analysis*

When the system has launched the cost were 8,650, with discount ratio 0.7. In conclusion, this project has a Payback Period of slightly less than 5 years.

Lifetime ROI = 40%

Annual ROI = 8%

NPV = 7,399.35

# Schedule Feasibility

The system is scheduled in a feasible way; the due dates of each chapter of the project during the semester are reasonable and convenient to deliver a proper working plan for the pets’ care system.

*Table 5. Project Schedule*

|  |  |  |  |
| --- | --- | --- | --- |
| Activity | Detailed Activity | Predecessor | Duration |
| Data Gathering | **A** Conduct Interviews | None | 2 |
| **B** Administer Questionnaires | A, C | 3 |
| **C** Read Company Reports | None | 2 |
| **D** Introduce Prototype | B, E | 4 |
| **E** Observe Reactions Prototype | G | 2 |
| Data Flow and Decision Analysis | **F** Analyze Data Flow | B, E | 6 |
| Proposal Preparation | **G** Perform Cost/Benefit Analysis | None | 2 |
| **H** Prepare Proposal | D | 1 |
| **I** Present Proposal | H | 1 |

### Gantt Chart



### Pert Diagram

# Software Requirements Document

# System Stakeholders and Requirements Sources

System Stakeholders:

Pet Owners: Pet owners will enter the required information to sign in/sign up and the information of their pet.

Requirements Elicitation Methods:

* 1. Brainstorming
  2. Prototyping: a high-fidelity prototype will be used
  3. Questionnaire

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generatedTable

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

# System Architecture

Diagram

Description automatically generated

*Figure 11: System Architecture.*

# User Requirement Definition

* + 1. Pet Owners:

3.3.1.1 Sign up:

User can create an account to be part of the system.

* + - 1. Log in validation:

User will use their sign in information to access the system and use the services provided by it.

* + - 1. Add a pet:

User will fill a form that requires information about the pet (can add one pet or more).

* + - 1. Modify reminders:

User can make modifications on the default reminders that are set by the system.

* + - 1. Add reminders:

User can make customized reminders in addition to the default reminders that are set by the system.

* + - 1. Add a pet by invitation code:

User can have the access to other user’s pet’s information and reminders using an invitation code but cannot make any modifications.

* + - 1. Remove the pet:

User can remove can select a pet or more to remove from the pets list.

* + - 1. Log out:

User can log out of the system to log in using any other account.

# System Functional Requirement Specification

* + 1. Pet Owners:

3.4.1.1 Sign up

3.4.1.1.1 Access the main page of the system.

3.4.1.1.2 Press “Sign up” button.

3.4.1.1.3 Enter name, email, and password.

3.4.1.1.4 Press “OK” button.

3.4.1.1.5 Validate email.

3.4.1.1.6 If valid email, open the “Profile Page”.

3.4.1.1.7 If invalid email, return the message “Incorrect Email”.

3.4.1.2 Log in Validation

3.4.1.2.1 Access the home page of the system.

3.4.1.2.2 Press the “Log In” button.

3.4.1.2.3 Enter the required information.

3.4.1.2.4 Validate the email and password against the “Pet Owners” database.

3.4.1.2.5 If valid user, open the “Profile” Page.

3.4.1.2.6 If invalid user, return the message “Incorrect Email or Password”.

3.4.1.3 Add a pet:

3.4.1.3.1 Access the “Pets List”.

3.4.1.3.2 Press on the “Add a Pet” button.

3.4.1.3.3 Fill the pet’s information.

3.4.1.3.4 Press on “OK” button to save the information and view the “Pet’s Profile”.

3.4.1.4 Modify reminders:

3.4.1.4.1 Access the “Pet’s Profile”.

3.4.1.4.2 Click on the reminder to access the reminder’s details.

3.4.1.4.3 Click on “Edit Reminder” option.

3.4.1.4.3 Make modifications on the reminder.

3.4.1.4.4 Press “OK ‘button to save the modifications and return to the “Pet’s Profile”.

3.4.1.5 Add reminders:

3.4.1.5.1 Access the “Pet’s Profile”.

3.4.1.5.2 Click on “Add a Reminder’ option to access the “Add a Reminder” page.

3.4.1.5.3 Fill the required information about the reminder.

3.4.1.5.4 Press the “OK” button to save the reminder and return to the “Pet’s Profile” page.

3.4.1.6 Add a pet by invitation code:

3.4.1.6.1 Access the “Pets List”.

3.4.1.6.2 Press on the “Add a Pet” button.

3.4.1.6.3 Select “Add by invitation code” option to access the code validation page.

3.4.1.6.4 Paste the invitation code in the provided place.

3.4.1.6.5 Press “Submit” button to validate the code.

3.4.1.6.6 If valid code, add the “Pet’s Profile”.

3.4.1.6.7 If invalid code, return the message “Incorrect Code”.

3.4.1.7 Remove the pet:

3.4.1.7.1 Access the “Pet’s Profile”.

3.4.1.7.2 Select “Remove the Pet” option.

3.4.1.7.3 Press “Confirm” to complete the process and return to the “Pets List”.

3.4.1.8 Log out:

3.4.1.8.1 Access the “Profile” page.

3.4.1.8.2 Select the “Log Out” option.

3.4.1.8.3 Press the “OK” button to log out of the account and return to the main page of the system.

# Non-Functional Requirements

*Table 6: Non-Functional Requirements.*

|  |  |  |
| --- | --- | --- |
| NFR# | Goal | Description |
| 1 | Security | The system should protect users’ personal information and should be protected against the risk of hacking and all types of damages. |
| 2 | Usability | The system should have user-friendly interface, so the user gets high satisfaction from users by getting the efficiency of using and accomplish their goal. |
| 4 | Performance | System should respond immediately, there should be no delay in giving the required information and actions, and the operations must be done quickly. |
| 5 | Availability | System must be available 24/7 hours for receiving information required. |
| 6 | Storage | The ability of the system to receive and store a lot of information. |

# Requirements Scenarios

* Sign Up
* **Initial assumption**: Each user owns a phone or tablet on which they installed the app and already have an email.
* **Normal**: The user is asked to sign up upon opening the app. As the user chooses the option ‘sign up’, they are prompted to choose a username, input their email, and pick a password. They then click ‘sign up’ and their account is created but awaits validation via a link sent to their email.
* **What can go wrong**:
* Users might choose a username that matches another user; therefore, they must be unique, and are shown a message that informs them if the username they wrote in the username field is already used, and another must be chosen.
* The user might already have an account and try to create another using the same email. A message would appear to inform them that the email already has an existing account and would not let them create the account with the same email.
* The user leaves a field empty when signing up. They sign up button wouldn’t activate until they fill all required fields.
* The user puts a very short or weak password. There would be a note under password field depicting what a password should include or not include. They would be given an error if a rule is broken.
* A user might submit a password they didn’t mean and typed wrong. A ‘confirm password’ field would be added to ensure that doesn’t happen and would display a warning message that the two passwords don’t match in case of a slip up.
* The user wouldn’t have a valid account to login with, because they didn’t validate the email yet. So, the user must be taken after signing up to a screen that tells them to validate the email first.
* **Other activities:** A validation email with a link is sent to the email the user signed up with and must be visited to take them back to the app and inform them that their account is validated.
* **System state on completion**: The user is shown back to the login screen, and their account is added to the database upon validation.
* Login
* **Initial assumption**: A user has an existing account on the app.
* **Normal**: The user is taken to the login screen. The user is prompted to enter either their username or email and enter their password. They then click ‘login’ and they are taken to their account.
* **What can go wrong**:
* Username or email entered by the user has no existing account. The user hadn’t made an account or didn’t validate it. A message would notify the user that there’s no account with that email or username upon entering it in the field, as it’s compared to the accounts in the database.
* The username/email and password don’t match. A message should appear and warn the user that they don’t match.
* The user forgot their password. There should be an option ‘Forgot password?’ that the user can click, where they put in their email for password recovery. It should also denote whether the entered email address has an existing account.
* **Other activities:** The credentials entered are compared with the ones in the database and if they match the user is permitted to access the account. In case of a password recovery, a password reset link is sent via email to the user.
* **System state on completion**: The user is logged onto their account and can view their profile.
* Create a new pet profile
* **Initial assumption**: Each user owns a pet or more that they need to add to the app. They have created an account on the app and logged in successfully.
* **Normal**: The user chooses ‘add new pet’, and they’re given a window that displays certain options to set up. They are prompted to write the name of the added pet, the age of the pet, the type of pet from the available options, and have an optional field to state the pet’s breed. Upon completion, the pet profile is added to their app homepage. A code of symbols and numbers would be assigned to the pet profile.
* **What can go wrong**:
* The assigned code overlaps with another for a different user. Number generated should be unique and long enough.
* The user clicked cancel by mistake and didn’t save anything they input. A confirmation message should be added in both cases, to make sure a user really wanted to cancel, or confirm adding the pet profile.
* **Other activities:** Each pet profile upon completion would be assigned a code of numbers and symobls that would be included with the profile information, so users can share it with other.
* **System state on completion**: User is logged on. An icon that represents the added pet profile would appear on homepage screen in the space depicted for pet profiles. The profile would contain the supplied information.
* Add an existing pet profile from someone else’s account to theirs.
* **Initial assumption:** The user already has an account and is logged onto it. They have a pet profile code from someone else who gave it to them to care for their pet.
* **Normal**: The user chooses ‘add existing pet’, and they’re given a window that prompts entering a pet profile code, checks that code and adds the requested pet profile to the user’s list.
* **What can go wrong**:
* A user’s code was used who didn’t give it. The original user should get a message if they would allow a certain user (name provided) to add their pet, and they can confirm or reject it.
* The user who entered the code would be left waiting. A message upon entering the code should inform them to await confirmation of their request. A notification would be sent to them about acceptance or rejection once the owner confirmed or rejected.
* The user might mix up the pet with their own pets. A label should be added to the icon to note that this pet is not owned but transferred.
* The code entered doesn’t exist. An error message informing the user of no such code should be given.
* **Other activities:** The request by a user would be processed, the code for pet profile should be located and its original owner’s account, and a message would be sent to that account to seek permission. Confirmation would be pending, and if responded to, the response would return a message informing the other user, with a confirmation and the pet profile added if permission was granted, otherwise they would receive a rejection message with no action taken.
* **System state on completion**: A new pet profile is added to the homepage of the app with the label. It matches all the information from the original owner’s account contains the supplied information.
* View and edit pet profile
* **Initial assumption:** The user already has an account and is logged onto it. They have an existing pet profile they want to view or modify.
* **Normal**: The user clicks on the icon for the pet’s profile, and they’re taken to a screen that displays all the information chosen and entered when the profile was added. It also contains a default schedule that’s dependent on the assigned pet type. A schedule would contain different categories, each when clicked views a different screen. Some of the categories would differ depending on the pet type, and the categories are things such as food, playtime, vaccination, etc. Each category contains its own sub schedule with certain times set. Alarms are associated with each time. The user can view any of these. They can also modify the profile information, as well as the default schedule assigned for the pet time. They can add, edit, or delete alarms, customizing the sub schedules. The user can click save or cancel if they edit anything, and any modification would be saved or dismissed, returning them to homepage.
* **What can go wrong**:
* The user has a customized schedule for a pet profile, then changed the pet type, which requires a different schedule. The user should not be permitted to change the pet type field of a profile.
* The user disabled sound for alarms on their device. A notification should be accompanied with the alarm sound so in this case they can still see the message.
* The user adds an alarm that overlaps with another of the same category. They should be warned with a message that there’s already an activity at this time, and that they should choose another.
* The user deleted default schedule by accident. There should be a reset schedule to default button.
* The user closed the pet profile without saving. A message should ask the user whether they want to save first or quit.
* **Other activities:** When save button is clicked, any modified information would overwrite existing data. If cancel was clicked after changing some fields, changes are discarded.
* **System state on completion**: The user is returned to the homepage in their profile, and the pet profile information was changed in case of modification and saving.
* Delete pet profile
* **Initial assumption:** The user already has an account and is logged onto it. They have an existing pet profile, and the want to delete it.
* **Normal**: The user selects the icon of the pet profile they intend to remove by long tapping, and an option shows up to delete the pet. They would press delete and it would be deleted. They also get to cancel. When the user deletes the pet profile, it should no longer appear on their homepage.
* **What can go wrong**:
* The user deleted the pet profile by clicking the delete button by accident. There should be a confirmation message that pops up for the user, asking them to confirm before deleting.
* The user mixed things up or realized they shouldn’t have deleted the profile. There should be a history that keeps track of deleted pet profiles, to allow recovery after the profile had been removed. The user could go there and click the restore button for the deleted profile.
* **Other activities:** The deleted profile would still be stored somewhere but locked, and upon recovery the profile would be brought back to its original place and available to modify.
* **System state on completion**: The pet profile is removed from the homepage and sent to history.
* View tips
* **Initial assumption:** The user already has an account and is logged onto it. They would like to view tips about a certain type of pets.
* **Normal**: The user clicks on a tab for tips, which has a screen with a button and a little menu field. The user selects from the menu a pet type to search for tips about, which filters the results, and then they press the button to generate a random tip about the chosen pet type. Each click would result in another tip or advice.
* **What can go wrong**: The user didn’t select a pet type before clicking the button. The button has to be disabled until a pet type is selected.
* **Other activities:** When an option is selected from the menu, a dictionary type file with all the tips associated with each option would be filtered, so the only tips chosen from are the ones tied to that specific option. A random process selects one of the tips and displays it to the user on the tips page each time the button is clicked.
* **System state on completion**: A tip shows on screen for the user to read.

# 4.0 System Models and Design

# 4.1 Context Diagram

Diagram

Description automatically generated

# 4.2 Use-case Model

Diagram

Description automatically generated

*Figure 13: use-case model*

# 4.3 Scenarios

*Table 7: use-case scenatio1*

|  |  |
| --- | --- |
| Use-Case Name | Log in |
| Description | this case allows the user to enter the system and use it services on condition that he have an account and valid email and password to enter it. |
| Primary Actor | Pets owner |
| Secondary Actor | None |
| Include Use-Case | 1. Verify email & password 2. Display login error |
| Precondition | Pet owner must have an account  Valid email and password |
| Main flow | 1. Access the home page of the system. 2. Press the “Log In” button. 3. Enter the required information. 4. validate the email and password against the “Pet Owners” database. 5. if valid user, open the “Profile” Page. 6. If invalid user, return the message “Incorrect Email or Password”. |

*Table 8: use-case scenatio2*

|  |  |
| --- | --- |
| Use-Case Name | Add a pet |
| Description | can add a new pet on the user profile by two ways ,first one is using a code the second is to fill the pet's information |
| Primary Actor | Pets owner |
| Secondary Actor | None |
| Include Use-Case | 1. Create pet's profile 2. Display incorrect code |
| Precondition | The used code and information about the pet’s must be correct |
| Main flow | 1. Access the “Pets List”. 2. Press on the “Add a Pet” button. 3. Fill the pet’s information. 4. Press on “OK” button to save the information and view the “Pet’s Profile”. |

*Table 9 :use-case scenatio3*

|  |  |
| --- | --- |
| Use-Case Name | Modify reminders |
| Description | the ability of edit on the reminder (by default) , by the name and the time for the reminder |
| Primary Actor | Pets owner |
| Secondary Actor | None |
| Include Use-Case | Edit alarm |
| Precondition | the edited reminder must be existed from the beginning  the edited information must be correct |
| Main flow | 1. Access the “Pet’s Profile”. 2. Click on the reminder to access the reminder’s details. 3. Click on “Edit Reminder” option. 4. Make modifications on the reminder. 5. Press “OK ‘button to save the modifications and return to the “Pet’s Profile”. |

*Table 10 :use-case scenatio4*

|  |  |
| --- | --- |
| Use-Case Name | Add reminder |
| Description | The use have the ability to add a new reminder , in case if there is anything happened , or if he just needed to add a reminder . |
| Primary Actor | Pets owner |
| Secondary Actor | None |
| Include Use-Case | Create new alarm |
| Precondition | The added reminder must not be at the same time of another reminder |
| Main flow | 1. Access the “Pet’s Profile”. 2. Click on “Add a Reminder’ option to access the “Add a Reminder” page. 3. Fill the required information about the reminder. 4. Press the “OK” button to save the reminder and return to the “Pet’s Profile” page. |

*Table 11 :use-case scenatio5*

|  |  |
| --- | --- |
| Use-Case Name | Remove the pet |
| Description | The ability to delete the pet's profile, The system does not provide the data return service, but it confirms the deletion request more than once |
| Primary Actor | Pets owner |
| Secondary Actor | None |
| Include Use-Case | Delete pet’s Profile |
| Precondition | the pet profile must be existed |
| Main flow | 1. Access the “Pet’s Profile”. 2. Select “Remove the Pet” option. 3. Press “Confirm” to complete the process and return to the “Pets |

*Table 12: use-case scenatio6*

|  |  |
| --- | --- |
| Use-Case Name | Verify email & password |
| Description | The system checks the email and password by returning to  its  database |
| Primary Actor | Pets owner |
| Secondary Actor | None |
| Extend Use-Case | Log in |
| Include Use-Case | None |
| Precondition | Have an account |
| Main flow | 1. The system matches the e-mail with the database 2. The system matches the password with the database 3. If the match happens in the both of them , it will enter directly . 4. If no match is occurred , Case (Display login error) occurs . |

*Table 13: use-case scenatio7*

|  |  |
| --- | --- |
| Use-Case Name | 1. Delete pet’s Profile |
| Description | In case that the pet`s profile were deleted the system doesn`t provide the data return service , but it confirms the deletion request more than once |
| Primary Actor | Pets owner |
| Secondary Actor | None |
| Extend Use-Case | Remove the pet |
| Include Use-Case | None |
| Precondition | Profile must be existed |
| Main flow | 1. Access the “Pet’s Profile”. 2. Select “Remove the Pet” option. 3. Press “Confirm” to complete the process and return to the “Pets |

*Table 14: use-case scenatio8*

|  |  |
| --- | --- |
| Use-Case Name | Display login error |
| Description | If the email or password is entered incorrectly, the system displays a message indicating the wrong entry one of them and prompts the user to repeat the entry process correctly. |
| Primary Actor | Pets owner |
| Secondary Actor | None |
| Extend Use-Case | Log in |
| Include Use-Case | None |
| Precondition | The email or password does not match the database |
| Main flow | 1. After the wrong entry of the password or the e-mail 2. The system sends a message to alert the user to the wrong entry 3. Re-enter the email 4. Re-enter the password |

*Table 15: use-case scenatio9*

|  |  |
| --- | --- |
| Use-Case Name | Create pet's profile |
| Description | The system creates a new pet profile by either using a code sent by another user or by filling in information about the new pet. |
| Primary Actor | Pets owner |
| Secondary Actor | None |
| Extend Use-Case | Add a pet |
| Include Use-Case | None |
| Precondition | Enter the pet information correctly  Enter a valid code |
| Main flow | 1. The system opens an input window (either through the code or filling in the information) 2. If it is via the code, the code number is entered 3. The code number is matched if it is false , Case (Display incorrect code) occurs 4. If the user chooses to enter by information 5. The user fills in the required information such as name, type, etc. |

*Table 16: use-case scenatio10*

|  |  |
| --- | --- |
| Use-Case Name | Display incorrect code |
| Description | If a wrong code is entered, the system will display a message to alert the user that the code that has been entered is a wrong code, so the user returns to enter the correct code number |
| Primary Actor | Pets owner |
| Secondary Actor | None |
| Extend Use-Case | Add a pet |
| Include Use-Case | None |
| Precondition | Incorrect code |
| Main flow | 1. After a wrong entry 2. The system displays a message explains  to the user that the input process is wrong 3. The code number is requested again |

*Table 17: use-case scenatio11*

|  |  |
| --- | --- |
| Use-Case Name | Edit alarm |
| Description | The system provides a service to modify the pre-existing alarm, in order to meet the different needs of the user according to the circumstances in which the user passes, by modifying (time, name, etc.) |
| Primary Actor | Pets owner |
| Secondary Actor | None |
| Extend Use-Case | Modify reminders |
| Include Use-Case | None |
| Precondition | Alarm must be existed |
| Main flow | 1. The system receives the modifications made by the user 2. Adjusts the alarm 3. The modification is stored in the database |

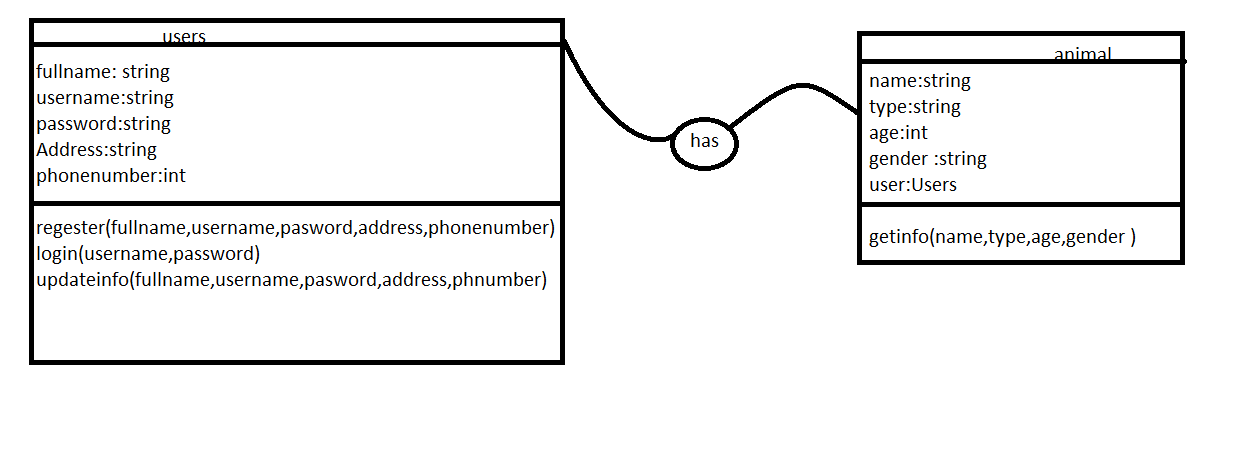
*Table 18: use-case scenatio12*

|  |  |
| --- | --- |
| Use-Case Name | Create new alarm |
| Description | The system offers a service to add a new alarm, in order to meet the different needs of the user according to the circumstances in which the user passes, by adding   (time, name, etc.) |
| Primary Actor | Pets owner |
| Secondary Actor | None |
| Extend Use-Case | Add reminder |
| Include Use-Case | None |
| Precondition | the edit reminder must not be at the same time to another reminder |
| Main flow | 1. The system receives the alarm information (from the user( 2. Adjusts the alarm 3. The information is stored in the database |

# 4.4 Sequence DiagramDiagram Description automatically generated



# 4.5 Class Diagram



*Figure 15: class diagram*

# 4.6 Database Design

Graphical user interface, application

Description automatically generated

As shown in Figure 18, the database in this project is not complex, the owner is only associated with his pet

*Figure 16: database design*

# 4.7 Graphical User Interface Design

# Application Description automatically generatedGraphical user interface, text, application Description automatically generated

Graphical user interface

Description automatically generatedGraphical user interface, text

Description automatically generated

*Figure 20: sign in page*

Graphical user interface, text, application, chat or text message

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

*Figure 21: the user profile page Figure 22: the list of the owner’s pets*

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generated

*Figure 23: entering the pet’s information Figure 24: entering the pet’s information cont.*

Graphical user interface

Description automatically generated with low confidenceQr code

Description automatically generated

*Figure 25:the pet's QR code Figure 26: the pet's page*

Text

Description automatically generatedGraphical user interface, application

Description automatically generated

*Figure 27: QR code scanning page Figure 28: upcoming alarms page*

Graphical user interface, application, icon

Description automatically generated

*Figure 29: Alarm that appears on the user's phone*

# Appendix A

The analysis of the questionnaire responses.

Graphical user interface, text, application, chat or text message

Description automatically generated

*Figure 30: capture of the questionnaire’s responses.*

Chart, pie chart

Description automatically generated

*Figure 31: capture of the questionnaire’s responses cont.*

A picture containing graphical user interface

Description automatically generated

*Figure 32: capture of the questionnaire’s responses cont.*

Chart, pie chart

Description automatically generated

*Figure 33: capture of the questionnaire’s responses cont.*

Chart, pie chart

Description automatically generated

*Figure 34: capture of the questionnaire’s responses cont.*

Graphical user interface, chart, application, pie chart

Description automatically generated

*Figure 35: capture of the questionnaire’s responses cont.*

Chart, pie chart

Description automatically generated

*Figure 36: capture of the questionnaire’s responses cont.*

Chart, bar chart

Description automatically generated

*Figure 37: capture of the questionnaire’s responses cont.*

# References

1. Creately [https://creately.com/](https://creately.com/?fbclid=IwAR1VuLOgA8M_nfALcrafv1zOxqpjuShHqjDtu2P2zTtpx5lAK8wZgHGINOw)