**PetPal**

**Virtual Pet Care Web App**

**Project Description:**

"PetPal" is a virtual pet care web app that allows users to adopt and care for a virtual pet. The pet needs regular feeding, playtime, and sleep, and it responds to the user’s actions. The goal is to keep the pet healthy and happy, with visual indicators showing its current mood, energy level, and hunger status. This project is ideal for students, as it introduces them to interactive features and gamification, making the development process enjoyable.

1. **Questions & Answers:**
2. **Do we need a database for the PetPal project?**  
   **Answer:** Yes, the database is required to store user accounts, pet details, and their attributes like food, thirst, sleep, and happiness.
3. **Should users be able to create pets of different types?**  
   **Answer:** Yes, users should be able to choose from common pets like dogs, cats, and rabbits.
4. **Should there be a limit on the number of pets a user can create?**  
   **Answer:** Yes, users should be able to manage up to three pets at a time for simplicity and balance.
5. **Do pets need individual attributes like food, thirst, sleep, and happiness?**  
   **Answer:** Yes, each pet should have its own attributes to make the experience engaging and realistic.
6. **Do users need to feed pets with different types of food?**  
   **Answer:** No, a single type of food is sufficient for this project.
7. **Should performing actions like eating or drinking increase the respective pet attributes?**  
   **Answer:** Yes, actions should directly impact the corresponding attributes (e.g., eating increases food, drinking increases thirst).
8. **Do you want pets to have unique images for different actions (e.g., eating, playing)?**  
   **Answer:** Yes, each pet type should have specific images for actions to enhance the visual appeal.
9. **Should pets' attributes decrease automatically over time?**  
   **Answer:** Yes, attributes like food, thirst, and sleep should decrease in real-time to simulate the responsibility of caring for a pet.
10. **Do you want pets to lose multiple attributes when performing actions like playing?**  
    **Answer:** Yes, some actions (e.g., playing) should decrease multiple attributes, such as food, thirst, and sleep.
11. **Should pets have sound effects for actions?**  
    **Answer:** Yes, each action (e.g., eating, drinking, playing) should have a sound effect to improve engagement.
12. **Do pets need an idle state with a default image when no action is being performed?**  
    **Answer:** Yes, pets should have an idle state with a default image to display when they are not performing actions.
13. **Do users need to log in and register to access the PetPal platform?**  
    **Answer:** Yes, login and registration are necessary to provide a personalized experience and ensure user data security.
14. **Should users be able to delete a pet they no longer want to care for?**  
    **Answer:** Yes, users should have the option to delete pets if they wish to create new ones.
15. **Do pets need to interact differently based on their type (e.g., dogs play differently than cats)?**  
    **Answer:** No, all pets will follow the same interaction logic for simplicity and balance.
16. **Should there be an exit button during pet interaction?**  
    **Answer:** Yes, users should have the ability to exit to the dashboard at any time to manage other pets.
17. **Requirements**

**User Requirements**

1. Users should be able to:

* Register and log in to the system securely.
* Create and manage up to three virtual pets.
* Choose pet types (e.g., dog, cat, rabbit) during creation.
* Perform actions like feeding, drinking, playing, and sleeping for each pet.
* View real-time updates on pet attributes such as food, thirst, sleep, and happiness.
* Delete pets they no longer want to manage.
* Exit from pet interaction to the dashboard at any time.

1. The interface should be intuitive, with clear buttons and visual feedback (e.g., action images, sound effects).

**System Requirements**

1. **Functional Requirements**:
   * **Database**:
     + Store user information (name, email, password).
     + Store pet information (type, name, gender, attributes).
   * **Actions**:
     + Feeding increases the pet's food attribute.
     + Drinking increases the pet's thirst attribute.
     + Playing increases happiness but decreases food, thirst, and sleep.
     + Sleeping fully replenishes the sleep attribute.
   * **Attribute Management**:
     + Food decreases by 4 every 2 seconds.
     + Thirst decreases by 6 every 2 seconds.
     + Sleep decreases by 3 every 2 seconds.
     + Happiness decreases by 3 every 2 seconds.
2. **Hardware/Software Requirements**:
   * **Server**:
     + PHP-enabled web server (e.g., XAMPP, LAMP).
     + MySQL or MariaDB for the database.
   * **Client**:
     + Modern web browser (e.g., Chrome, Firefox) with JavaScript enabled.
   * **Development Tools**:
     + HTML, CSS, Bootstrap for the frontend.
     + JavaScript, jQuery, and AJAX for interactivity.
     + PHP for backend logic.

**Priorities**

1. **High Priority**:

* User login and registration with a secure password.
* Creating and managing pets with real-time attribute management.
* Implementing core actions (eat, drink, sleep, play) and their impact on attributes.
* Displaying pet information (name, gender, type) and dynamic images for actions.
* Limiting users to three pets.

1. **Medium Priority**:

* Adding sound effects for actions.
* Implementing the "Remember Me" feature for login.
* Including an exit button to return to the dashboard.

1. **Low Priority**:

* Aesthetic customizations like color-coded bars and animations.
* Advanced features such as adding more pet types or customizations.

**Non-Functional Requirements**

1. **Performance**:

* The system should handle up to 50 concurrent users without performance degradation.
* Pet attribute updates (decrease or increase) should reflect in under 1 second.

1. **Usability**:

* The interface should be user-friendly and intuitive for all age groups.
* Buttons, bars, and actions should be clearly labeled.

1. **Scalability**:

* The system should support adding more pet types or features in the future.

1. **Reliability**:

* User and pet data should persist across sessions using the database.
* Cookies should securely handle "Remember Me" functionality without data loss.

1. **Security**:

* Passwords must be securely hashed using PHP's password\_hash() function.
* Users should not access pets or actions belonging to other accounts.

1. **Accessibility**:

* The interface should work on both desktop and mobile browsers.

1. **UI Prototype in Figma**

**A cartoon dog standing on a rug in a room

Description automatically generated**

A screenshot of a video game

Description automatically generated

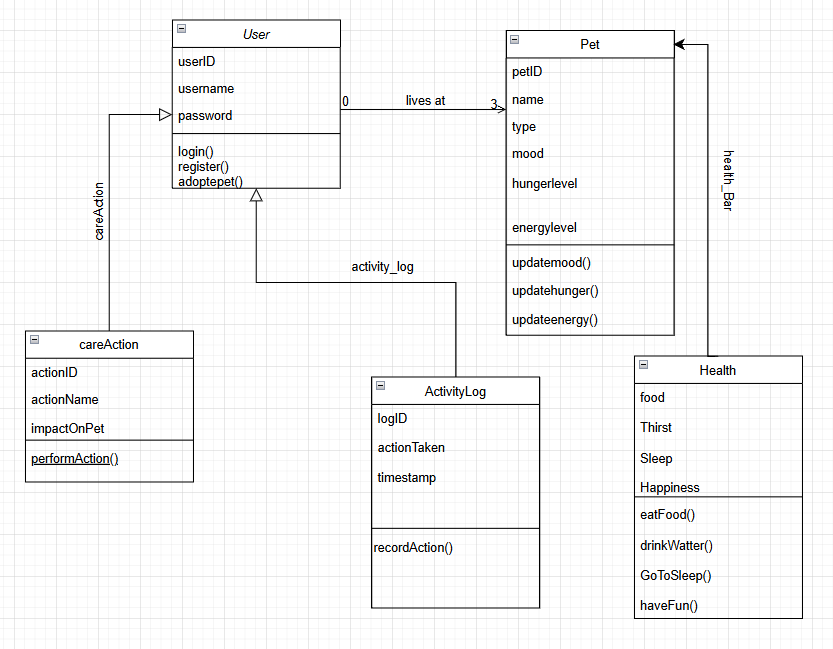
A screenshot of a login form

Description automatically generated

1. **UML Diagram**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Elements** | **Attributes** | | | **Operations** | | | **Relationships** | |
| User | | userID  username  password | | | | login();  register();  adoptpet(); | | User can have 1 to 3 pets. | |
| Pet | | petID  name  type  mood  hungerLevel  updateMood()  updateHunger() | | | | updateHunger  updateMood()  updateHunger() | | Check every bar level of the pet | |
| careAction | | actionID  actionName  impactOnPet | | | | performAction() | |  | |
| ActivityLog | | logID  actionTaken  timestamp | | | | recordAction() | |  | |
| Health | | | eatFood();  Food: get decrease by 4 every two second and get + 25 when we press eat  Thirst();  Thirst: get decrease by 6 every two second and get + 30 when drink  GoToSleep();  Sleep: get decrease by 3 every two second and get +100 when sleep  HaveFun();  Happiness: get decrease by 3 every two second and get +60 when play | | recordAction() | | | Make sure your pet eats,  drink water, go to sleep,  and have fun to take care  of your pet's health. | | |

**UML Diagram from draw.io**



1. **User Evaluation**

[Questionnaire](https://docs.google.com/forms/d/e/1FAIpQLSddx9QNlBzqY1Wtji5vT0IO8kKfsH6K0vH3SmVyl4zqv5SGOA/viewform?usp=dialog)

A screenshot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated

A pie chart with different colored circles

Description automatically generated

**Team members**  
Vladimiros Orfanov st022675  
Giannis Poursanidis st022296Charalampos Theodosidis st023373

**How to set up:**

To set up the **PetPal** game, create a database named petpal in **phpMyAdmin** and import the provided .sql file to set up the tables. Open the config.php file and update the database connection details (servername, username, password, dbname) with your local setup. Place all files in the htdocs folder (if using XAMPP), start Apache and MySQL, and access the game at http://localhost/petpal. The game will now connect to the database and be ready to use.