Documentation for MDS app

1. Initial project requirements

We used Trello for managing the stages of this project. The User Stories we initially planned were:

✅ 1. As a user, I want to share photos of my pet.
☑️ 2. As a user, I want to be able to see the number of likes on a post and also the list of users that liked it.

- ★ This user story wasn't completely achieved due to the lack of time (bad allocation of time), you are able to see the number of likes but you don't have access to the list of users who liked it. We plan to develop the full functionality in the following development phase. We plan to enforce a more efficient time allocation system in order to ensure that the user story will be achieved.

✅ 3. As a user, I want to be able to comment on other people's posts.
✅ 4. As a user, I want to be able to like a post and to be notified when my posts are liked.
✅ 5. As a user, I want to customize my pet's profile.
✅ 6. As a user, I want to be able to like comments and to delete comments on my posts.
☑️ 7. As a user, I want to be able to edit my comments on other's posts.

- ★ The User Story was achieved, but there is a bug that allows you to delete other people's comments. We will fix this bug in the next development phase. Our plan is to have each person fix one bug per sprint in order to ensure we don't run out of time.

☑️ 8. As a user, I want to be able to share stories (by this, we mean blog posts) about my pets.

- ★ The User Story was achieved, but there is a bug that allows you to delete other people's posts.We will fix this bug in the next development phase. Our plan is to have each person fix one bug per sprint in order to ensure we don't run out of time.t.

✅ 9. As a user, I want to be able to talk about subjects related to taking care of pets (tips & tricks).
✅ 10. As a user, I want to be able to delete my posts.
✅ 11. As a user, I want to be able to put tags on my stories so that people can find them easier.
✅ 12. As a user, I want to search by tags/title posts.
✅ 13. As a user, I want to be able to search other pets accounts.
✅ 14. As a user, I want to be able to create multiple pet accounts.
✅ 15. As a user, I want to be able to follow other people and see their pet's posts.
✅ 16. As an admin, I want to be able to delete inappropriate posts.
☑️ 17. As an admin, I want to be able to ban users that have suspicious/inappropriate activity.

- ★ This user story wasn't achieved at all as it wasn't a critical part of our project. We plan to make the application more accessible, so we will add this user

story as a requirement to the next development phase. We will add it with a medium priority tag to ensure we will achieve it.

☑️ 18. As a visitor, I want to be able to create an account for my pets.

★ We decided that this user story doesn't fit the purpose of our app. Only registered users should create accounts for their pets. We don't plan to develop this user story in the following phase.

2. Team description

The project was implemented by a team of 3 members, their names and roles being:

- Ana-Maria Panait - Full Stack Developer, implemented the posts and explore pages
- George-Tiberiu Teodorescu - Full Stack Developer, implemented the users and pets profile pages
- Maria Neaga-Budoiu - Full Stack Developer, implemented the search and notification pages

For the Petbook 2.0, the members changed, as we are now 4 full stack developers with better defined responsibilities, to better cover the requirements left and to add new features. The new team and roles:

- Ana-Maria Panait - Full Stack Developer, UX Designer
- George-Tiberiu Teodorescu - Full Stack Developer, Product Manager
- Alexandra-Diana Ciocan - Full Stack Developer, Team Lead
- Miruna-Bianca Georgescu - Full Stack Developer, UX Designer

3. Software Architecture Report

The app is developed using the C# programming language and the ASP.NET Core MVC framework, providing a scalable and secure architecture. SQL Server is utilized as the database management system, ensuring efficient storage and retrieval of user data.

We used the Model-View-Controller (MVC) Pattern for the following reasons:

- Development of the application becomes fast
- It's easy for multiple developers to collaborate and work together, as every feature has a different model, view and controller
- It's easier to Update the application
- It's easier to debug large-scale applications as multiple levels are structurally defined and properly written in the application.

We also have the Observer Design Pattern. The observer design pattern is used by default in MVC projects. Its goal is to create this one-to-many relationship between the subject and all of the observers waiting for data so they can be updated (relationship between a model and all the views associated with it).

We used the following coding style for the code: https://github.com/dotnet/runtime/blob/main/docs/coding-guidelines/coding-style.md.

Besides the aforementioned bugs the app also presents other unintended behaviors such as the personal feed page displaying all the posts from the platform. These problems will be solved using the system described earlier (4 bugs per sprint) and we plan to prevent future problems by using more rigorous testing practices.

We plan to improve the logic behind our application in order to achieve better time and resources performance.

Link to MDS repository: https://github.com/anamariapanait10/Petbook