CA 1

VALUE: 15%

COMPLETION & UPLOAD DATE: MONDAY 3RD NOVEMBER 9AM

You are required to design and implement a solution in C# that will be used by an animal shelter.

The following user stories have been developed to model the required functionality:

1. As a shelter manager, I want to be able to enter all new animals into the system.
2. As a vet, I want to be able to view all information about a particular animal and add important information to their records e.g. vaccination status.
3. As a potential fosterer, I want to be able to see the list of all animals that need a foster home.
4. As a potential pet owner, I want to be able to view a list of all animals available for adoption, and be able to filter this list (e.g. by animal i.e., cat, dog, rabbit, etc., by vaccination status, by location, etc.)

Note: You are not required to create a database (but you can if you wish) – you may keep test data in your code, or in a text/json/xml file. You are not required to develop login/logout functionality (again, you can if you wish).

How you structure your project is up to you but you should display your knowledge of the following:

- Classes and objects, including abstract classes and interfaces
- Suitable collections – arrays, lists, dictionaries, etc.
- Reading/writing data to and from a file.

User interface: You can choose whether to develop a menu structure for a console application, create a windows forms application and present the user with a GUI, or to develop a Web Page as a front end for your application.

Marks will be awarded as follows:

1.	User Interface (GUI or menu)	10%
2.	Complete functionality & use of specified constructs	40%
3.	Class structure & organisation	10%
4.	Extra functionality*	15%
5.	Testing (Unit Tests & more)	10%
6.	Overall Impression/code quality	15%

*Extra functionality is where you show off – marks are awarded for challenging or innovative features.

Submission Checklist

Make sure to submit all the following to Moodle or your submission will not be marked:

GitHub Link Regular commits, relevant commit messages, minimum 2 branches Must be either public, or have invited your lecturer as a collaborator	
Screencast 5 minutes – talk through a running demo of your project including extra functionality, point out relevant code	
Coversheet Signed	
Zipped project Include everything needed to run the project, including test data.	

Academic Integrity

The assignment must be entirely the work of each student – in your own words. Students are not permitted to share any pseudocode or source code from their solution with any other student in the class. Students may not distribute the source code of their solution to any other student in any format (i.e., electronic, verbal, or hardcopy transmission). Any suspected plagiarism will be investigated, pursued, and reported to the Plagiarism Committee.

Generative artificial intelligence (AI) tools cannot be used in this assessment task. In this assessment, **you must not use** generative artificial intelligence (AI) (ChatGPT, ChatSonic, Bing Chat, Lex, DALL-E 2, or other tools) to generate any materials or content in relation to the assessment task.

The DkIT Academic Integrity Policy and Procedures, https://www.dkit.ie/about-dkit/policies-and-guidelines/academic-policies.html) states the following:

"Using generative artificial intelligence tools (e.g. ChatGPT) in an assessment unless explicitly permitted to do so and with proper acknowledgement, is a form of plagiarism".

Late Submissions

The institute's standard policy on marking of late submissions will be applied:

- Submissions received up to 1 week late will have a 20% grade reduction applied.
- Submissions received up to 2 weeks late will have the above penalty applied and will be capped at 40%.
- Submissions received more than 2 weeks late cannot be graded.