cuACS

Carleton University Animal Care System

Requirements Analysis Document

Team 777

By:
Danny Barreto - 101044467
Zachary Zucco - 101043596
Foyin Ogbara - 100900459
Awab Safieldin -100909334

COMP3004 Winter 2019
Prof. Christine Laurendeau
School of Computer Science
Carleton University

Feb 12th 2019

Contents

1. Introduction.. 2
    1.1. Purpose of System... 2
    1.2. Overview of Document... 3
2. Proposed System.. 3
    2.1. Overview.. 3
    2.2. Functional Requirements.. 3
    2.3. Non-functional Requirements... 4
    2.4. System Models.. 6
        2.4.1. Use Case Model.. 6
        2.4.2. Object Model.. 19
3. Glossary.. 22

Figures

Figure 1 - High-Level Use Case Diagram 7
Figure 2 - Client Detailed Use Case Diagram 8
Figure 3 - Staff Detailed Use Case Diagram 8
Figure 4 - UML Class Diagram 20

Tables

Table 1 - Functional Requirements 3
Table 2 - Non-Functional Requirements 4
Table 3 - High-level Use Case Descriptions 7
Table 4 - Detailed Use Case Descriptions 9
Table 5: Entity Object Description 20
Table 6: Control Object Description 21

1. Introduction

1.1 Purpose

In its 2017 report on animal shelter statistics, Humane Canada is proud to state that the number of animals living in shelters is steadily declining year by year. Animal shelter staff are always happy to see an animal in their care be adopted into a loving forever home. A shelter's primary concern is the happiness of its animals, even after adoption, and so it seeks to maximise the potential of a great relationship between pet and owner. Likewise, humans looking to adopt want the best possible pet for them.

To this end, the Carleton University Animal Care System (henceforth *cuACS* or *the cuACS*) provides the means by which human adopters and animal adoptees may find the best match for one another. By examining animal traits and comparing them to client preferences, cuACS's matchmaking algorithm, dubbed the Animal-Client Matching (ACM) algorithm, provides shelter staff with an optimal set of matches between pet and human. The ACM algorithm works in such a way that the best interests of *all* animals within the shelter are considered, rather than generating a set with both very good matches and rather poor matches. As shelters care more about the quality of life of an animal than it being adopted, cuACS may opt to not match some animals if no client is decently compatible with them.

The cuACS is to be used by clients and shelter staff. The system stores two types of profiles, for animals and for clients. An animal's profile is managed by staff and contains the animal's attributes that are relevant to matching. These include but are not limited to its species, breed, and age, as well as behavioural descriptions and expected living conditions. In a client's profile, the client indicates their preferred attributes which they desire for a pet, as well as some relevant personal information. This personal information relates to the client's living situation, such as the size of their living space, a rough estimate of their income and the size of their family unit. This information is used by the ACM algorithm to ensure the client is able to properly care for an animal. Finally, a client profile also contains contact information.

Shelter staff oversee the creation of all profiles. Clients are able to edit their own profiles after it has been created by a staff member. Clients are able to browse a list of all animal profiles to help them tune their preferences to their animal of choice.

When all profiles are set up, staff may run the ACM algorithm. By examining the details specified within each profile, the algorithm attempts to create as many and as favorable matches between animal and human as possible. The algorithm provides staff with an ideal set of matches, which they may use to help along the adoption process.

1.2 Overview of Document

This document details the results of the requirements elicitation and analysis for the cuACS project. Its purpose is to formalize expectations for the system, specify core functionality and constraints, and to be a stepping stone from which further development can progress. The document can be divided into three different sections. First, the functional and non-functional requirements are presented. They are the result of consultation with the client and represents their needs and desires. The requirements specify the workings and constraints of the system. Second, the use case model explains all the specific functionality with which the users will interact and the relationships between use cases. Finally, the object model illustrates all classes and entities within the system, their attributes and their relations to one another.

2. Proposed System

2.1 Overview

The cuACS system stores client and animal profiles containing their attributes. Staff are able to run an algorithm to match clients and animals for adoption. This section details the results of requirements analysis.

2.2 Functional Requirements

The functional requirements describe features available to users that are core to fulfilling the system's purpose. The primary function of cuACS is the animal-client matching algorithm. The requirements describe the actions necessary for the ACM to function as intended, namely the creation, viewing and modification of profiles.

Table 1 - Functional Requirements

ID	Description
F-1	User can log into Client account.
F-2	User can log into Staff account
F-3	Clients can view their own profiles
F-4	Clients can edit their own profiles
F-5	Clients can view a list of animals
F-6	Clients can view individual animal profiles
F-7	Staff can view a list of animals
F-8	Staff can view individual animal profiles
F-9	Staff can add new animal profiles

F-10	Staff can edit animal profiles
F-11	Staff can view a list of clients
F-12	Staff can view individual client profiles
F-13	Staff can add new clients
F-14	Staff can launch the Animal-Client Matching algorithm

2.3 Non-functional Requirements

The non-functional requirements describe the user's ability to interact with the system, expectations in regards to performance and management, constraints on the system, and other details about the experience of operating the software rather than the features of that software itself.

Table 2: Non-Functional Requirements

ID	Category	Description
NF-1-1	Usability	The system adheres to the WCAG specification level A.
NF-1-2	Usability	Documentation is provided for installation and operation.
NF-1-3	Usability	UI is designed such that users shouldn't need to consult documentation post-installation.
NF-1-4	Usability	Users are asked for confirmation before saving/quitting.
NF-1-5	Usability	All error messages should be descriptive and suggest appropriate solutions.
NF-2-1	Reliability	The program should never crash, freeze or require a reset.
NF-2-2	Reliability	Data is never lost without explicit deletion, especially in the event of an unexpected termination.
NF-2-3	Reliability	Animal/Client profiles are saved after every modification.
NF-2-4	Reliability	Invalid values in input fields should be rejected without issue.
NF-3-1	Performance	Saving data after creation, deletion or modification of profiles should take less than 10 seconds.
NF-3-2	Performance	Running the ACM should take less than 1 minute.
NF-3-3	Performance	Displaying lists of profiles or individual profiles should be near instantaneous (less than 3 seconds).
NF-3-4	Performance	System updates should happen overnight - thus there should be no downtime when the system is used during the day.

NF-4-1	Supportability	Supports the addition of new attributes to client/animal profiles without requiring significant changes to the ACM.
NF-4-2	Supportability	All updates should be backwards compatible.
NF-4-2-A		If an update does not change the database schema, it can be applied with no hassle.
NF-4-2-B		If an update does change the database schema, then an interface should be provided to update old databases to the new structure.
NF-4-3	Supportability	Updates are downloaded from the cuACS website.
NF-4-4	Supportability	Translations to new languages can be added through the addition or modification of a single ‘language’ or ‘strings’ file. No changes are required in source code. Meaning the source code does not rely on strings being in English.
NF-5-1	Implementation	Code is written in C++.
NF-5-2	Implementation	The system can be installed and run on the Linux Ubuntu 18.04 OS.
NF-5-3	Implementation	A SQLite3 database is used to store data in persistent storage.
NF-6-1	Interface	Staff can print data by connecting to a printer. Printers can be connected to the system via USB or through wireless connection.
NF-6-2	Interface	Staff can export/import data to/from file formats such as .csv, .json or .xlsx for use with other data management software.
NF-6-3	Interface	An interface is provided to update old databases to be compatible with updated versions of the software, if required.
NF-7-1	Operations	All staff members have administrative privileges and can run tasks such as system updates.
NF-7-2	Operations	The system is operated by staff from a computer within an animal shelter office.
NF-7-3	Operations	Staff can choose when to update, or set the system to update automatically.
NF-7-4	Operations	Staff can set a schedule for automatic tasks to be performed, including automatic updates, shutdown, reboots, and more.
NF-8-1	Packaging	The system is distributed through online download.
NF-8-2	Packaging	Installation can done through a single command on the command line.
NF-8-3	Packaging	Users can opt into using a installer wizard to allow for more installation options, such as the creation of a desktop shortcut.
NF-9-1	Legal	Users must agree to an End User License Agreement, which outlines their rights and obligations regarding the use and distribution of the software.

NF-9-2	Legal	The software - when used as intended - must not violate the intellectual property laws of the jurisdiction where the software is to be used and distributed.
NF-9-3	Legal	Any images used as part of the system, for example as profile pictures, must be part of the public domain, owned by the end user or otherwise used with permission.
NF-9-4	Legal	Handling of client data must adhere to the Personal Information Protection and Electronic Documents Act.

2.4 System Models

2.4.1 Use Case Model

A single use case describes a single bit of functionality. The use case model details which users can access specific use cases within the system, as well as the relations between the use cases. It shows what actions are available to both staff and clients. In the following section, each use case is illustrated in a diagram to demonstrate its relation to actors and to other use cases. Each use case is also described more thoroughly within its individual tables.
First, the high-level use cases are detailed. These are the first actions a client or staff member must partake. All other functionality stems from the completion of the high-level use cases.

Use Case Overview

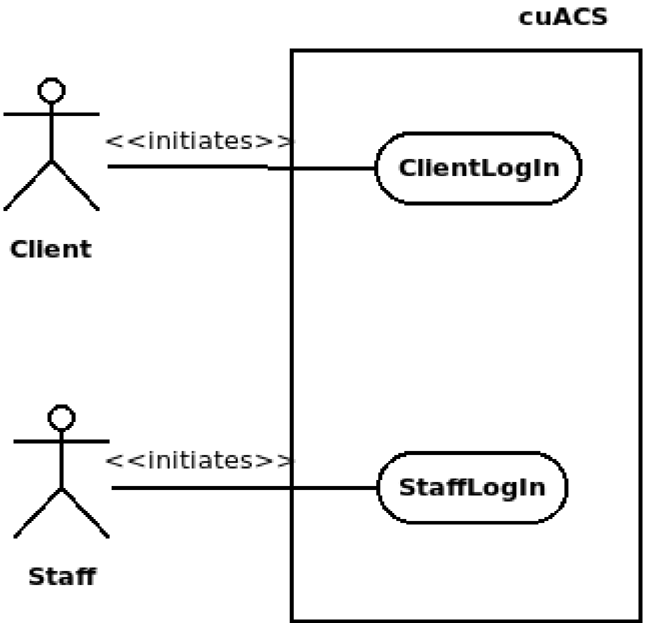

Figure 1 - High-Level Use Case Diagram

Table 3 - High-level Use Case Descriptions

UC-01	**ClientLogIn**	Log into cuACS system with a Client account and allows access to Client commands.
UC-02	**StaffLogIn**	Log into cuACS system with a Staff account and allows access to Staff commands.

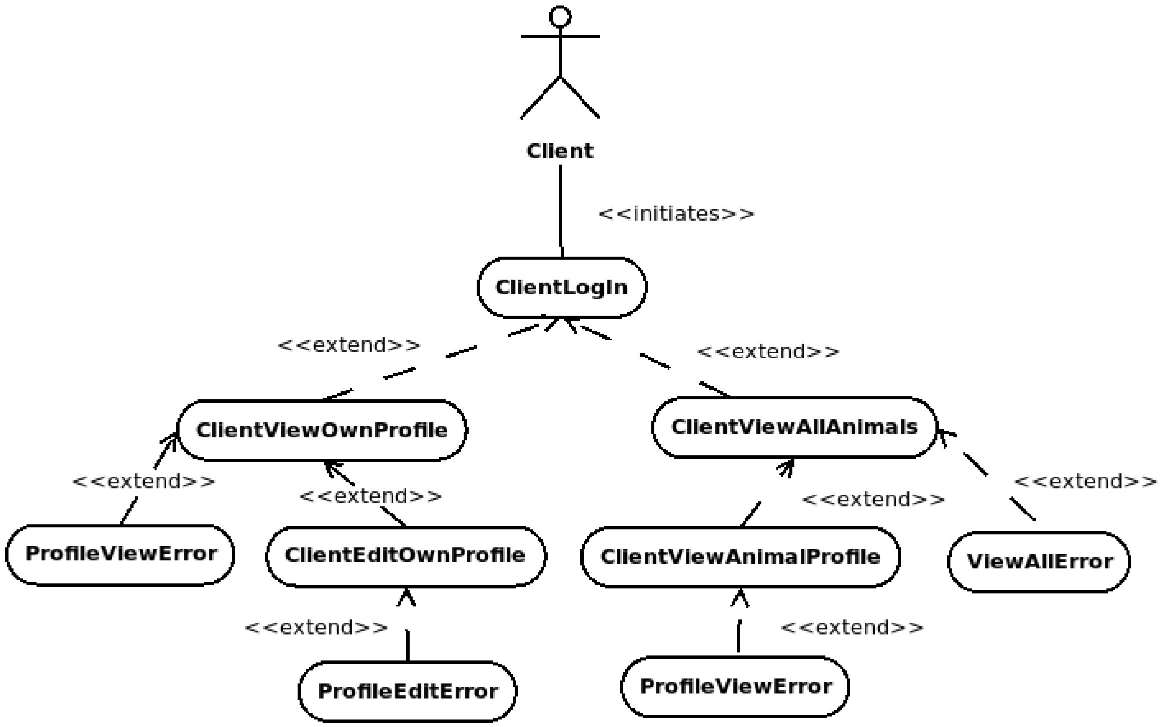

Figure 2 - Client Detailed Use Case Diagram

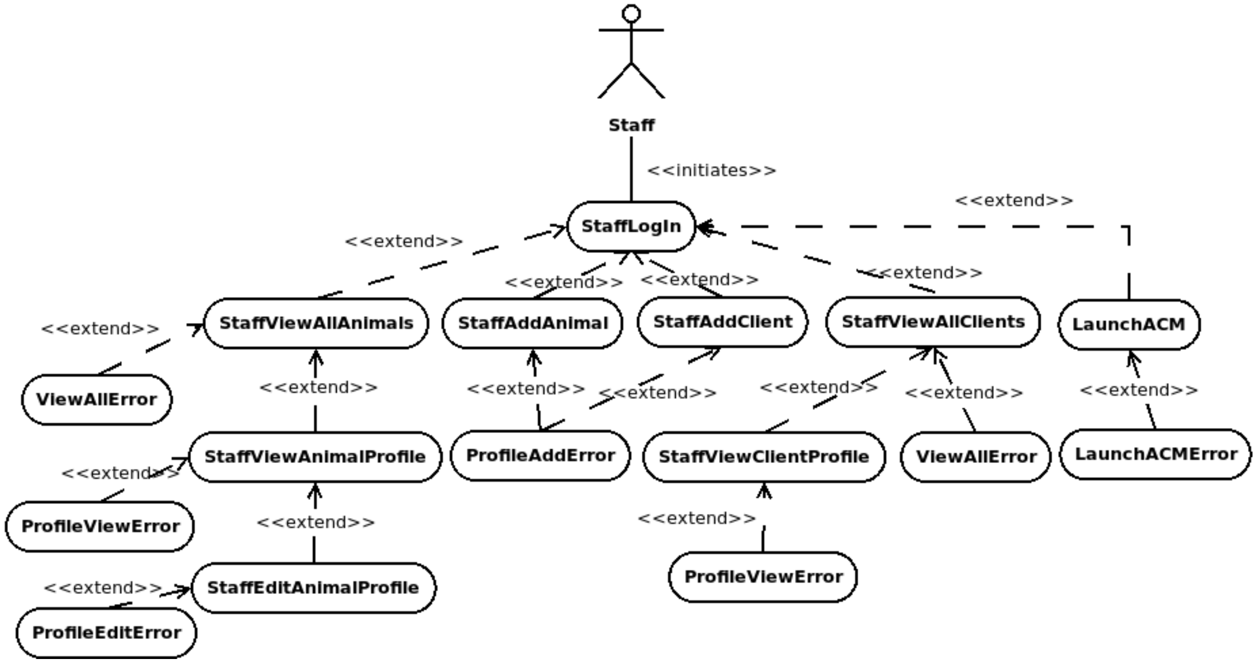

Figure 3 - Staff Detailed Use Case Diagram

Table 4 - Detailed Use Case Descriptions

UC-03	**ClientViewAllAnimals**	The Client can view a list of animals available for adoption.

UC-04	**ClientViewAnimalProfile**	While viewing a list of animals, the Client can choose an animal's profile and will be displayed the profile of that animal.
UC-05	**ClientViewOwnProfile**	The Client will be displayed their own profile.
UC-06	**ClientEditOwnProfile**	While viewing their own profile, the Client can make changes to the information within their own profile.
UC-07	**StaffViewAllAnimals**	The Staff can view a list of animals available for adoption.
UC-08	**StaffViewAnimalProfile**	While viewing a list of animals, the Staff can choose an animal's profile and will be displayed the profile of that animal.
UC-09	**StaffEditAnimalProfile**	While viewing an animal's profile, the Staff can make changes to the information within that animal's profile.
UC-10	**StaffAddAnimalProfile**	The Staff can add a new animal profile to the system.
UC-11	**StaffViewAllClients**	The Staff can view a list of all Clients in the system.
UC-12	**StaffViewClientProfile**	While viewing a list of Clients, the Staff can choose a Client and will be displayed the profile of that Client.
UC-13	**StaffAddClient**	The Staff can add a new Client profile to the system.
UC-14	**LaunchACM**	The Staff can launch the ACM algorithm.
UC-15	**ProfileViewError**	The system reports that the selected profile could not be viewed.
UC-16	**ProfileEditError**	The system reports that the selected profile could not be edited.
UC-17	**ViewAllError**	The system reports that it could not display the list of profiles.
UC-18	**ProfileAddError**	The system reports that the profile could not be added.
UC-19	**LaunchACMError**	The system reports that it could not launch the ACM algorithm.

Use Case Flow of Events

Use Case Identifier	UC-01
Use case name	ClientLogin
Participating actors	Initiated by Client
Flow of events	1. Client logs into cuACS system using a Client account. 2. cuACS system grants the Client access to Client privileges.
Entry condition	
Exit condition	- The Client logs out.
Quality Requirements	- Logging into a Client account should take less than 5 seconds.
Traceability	F-1

Use Case Identifier	UC-02
Use case name	StaffLogin
Participating actors	Initiated by Staff
Flow of events	1. Staff logs into cuACS system using a Staff account. 2. cuACS system grants the Staff access to Staff privileges.
Entry condition	
Exit condition	- The Staff logs out.
Quality Requirements	- Logging into a Staff account should take less than 5 seconds.
Traceability	F-2

Use Case Identifier	UC-03
Use case name	ClientViewAllAnimals
Participating actors	Initiated by Client
Flow of events	1. Client requests to view all animals available for adoption 2. cuACS responds by displaying all animals available for

	adoption.
Entry condition	- Must be logged into cuACS system with a Client account.
Exit condition	- The Client has received a list of all animals available for adoption, OR - The Client has received an error indicating why the list of animals could not be displayed
Quality Requirements	- The list of all animals available for adoption should be displayed in no longer than 5 seconds.
Traceability	F-5

Use Case Identifier	UC-04
Use case name	ClientViewAnimalProfile
Participating actors	Initiated by Client
Flow of events	1. Client requests to view an individual animal's profile. 2. cuACS responds by displaying the animal's profile.
Entry condition	- Must be logged into cuACS system with a client account, AND - Must be viewing list of animals available for adoption.
Exit condition	- The Client has been displayed the animal's profile, OR - The Client has received an error explaining why the animal's profile could not be viewed.
Quality Requirements	- Animal's profile should be displayed in no longer than 3 seconds
Traceability	F-6

Use Case Identifier	UC-05
Use case name	ClientViewOwnProfile
Participating actors	Initiated by Client
Flow of events	1. Client requests to view their own profile. 2. cuACS system responds by displaying the client's profile.
Entry condition	- Must be logged into cuACS system with a client account
Exit condition	- The Client has been displayed their own profile, OR - The Client has received an error explaining why their profile

	could not be viewed
Quality Requirements	- Client’s profile should be displayed in no longer than 3 seconds
Traceability	F-3

Use Case Identifier	UC-06
Use case name	ClientEditOwnProfile
Participating actors	Initiated by Client
Flow of events	1. Client requests to edit their own profile. 2. cuACS system gives the Client permission to edit their own profile. 3. Client makes the changes to their profile and then submits the changes to the cuACS system to be saved. 4. cuACS system receives the changes to the profile and saves the changes and then notifies the Client that the changes have been saved.
Entry condition	- Must be logged into cuACS system with a client account, AND - Must be viewing Client’s own profile.
Exit condition	- The Client’s profile changes have been saved by the cuACS system and the Client has been notified, OR - The Client has received an error explaining why they were not allowed to edit their profile.
Quality Requirements	- Saving changes to profile should take no longer than 5 seconds.
Traceability	F-4

Use Case Identifier	UC-07
Use case name	StaffViewAllAnimals
Participating actors	Initiated by Staff
Flow of events	1. Staff requests to view a list of all animals available for adoption. 2. cuACS system responds by displaying all animals available for adoption.
Entry condition	- Must be logged into cuACS system with a Staff account.

Exit condition	- The Staff has received a list of all animals available for adoption, OR - The Staff has received an error indicating why the list of animals could not be displayed.
Quality Requirements	- List of animals should be displayed in no longer than 5 seconds.
Traceability	F-7

Use Case Identifier	UC-08
Use case name	StaffViewAnimalProfile
Participating actors	Initiated by Staff
Flow of events	1. Staff requests to view an animal's profile. 2. cuACS system responds by displaying the animal's profile.
Entry condition	- Must be logged into cuACS system with a Staff account, AND - Must be viewing list of animals available for adoption.
Exit condition	- The Staff has been displayed the animal's profile, OR - The Staff has received an error explaining why the profile could not be displayed.
Quality Requirements	- The animal's profile should be displayed in no longer than 5 seconds
Traceability	F-8

Use Case Identifier	UC-10
Use case name	StaffAddAnimalProfile
Participating actors	Initiated by Staff
Flow of events	1. Staff requests to add a new animal profile. 2. cuACS responds by presenting a form to the Staff. 3. Staff fills out the form by entering *(all the animal details needed for the animal*). Once the form is completed, Staff submits the form. 4. cuACS receives the form and creates and adds the new animal profile. cuACS system also sends Staff confirmation that the animal profile has been added.
Entry condition	- Must be logged into cuACS system with a Staff account

Exit condition	- The Staff has received confirmation that the animal profile has been added, OR - The Staff has received an error indicating why the new profile could not be added.
Quality Requirements	- Adding a new animal should take no longer than 5 seconds.
Traceability	F-9

Use Case Identifier	UC-09
Use case name	StaffEditAnimalProfile
Participating actors	Initiated by Staff
Flow of events	1. Staff requests to edit an animal profile. 2. cuACS gives permission to Staff to edit animal profile. 3. Staff makes changes to the animal's profile and then submits it to the cuACS system. 4. cuACS system receives the changes to the profile and saves them and notifies the Staff that the changes have been saved.
Entry condition	- Must be logged into cuACS system with a Staff account, AND - Must be viewing an individual animal's profile.
Exit condition	- The animal's profile changes have been saved by the cuACS system and Staff has received confirmation of the changes, OR - The Staff has received an error explaining why their changes could not be saved, OR - The Staff has received an error explaining why they cannot change the animal's profile.
Quality Requirements	- Saving changes to the profile should take no longer than 5 seconds.
Traceability	F-10

Use Case Identifier	UC-11
Use case name	StaffViewAllClients
Participating actors	Initiated by Staff
Flow of events	1. Staff requests to view a list of all Clients.

	2. cuACS system responds by displaying all Clients.
Entry condition	- Must be logged into cuACS system with a Staff account.
Exit condition	- The Staff has received a list of all Clients, OR - The Staff has received an error explaining why the list of Clients could not be displayed
Quality Requirements	- List of Clients should be displayed in no longer than 5 seconds.
Traceability	F-11

Use Case Identifier	UC-13
Use case name	StaffAddClient
Participating actors	Initiated by Staff
Flow of events	1. Staff requests to add new Client Profile 2. cuACS system responds by presenting a form to the Staff 3. Staff fills out the form by entering *(all the client details needed for the client). Once the form is completed, Staff submits the form. 4. cuACS receives the form and creates and adds the new Client profile. cuACS system also sends the Staff Confirmation of the Client profile being added.
Entry condition	- Must be logged into cuACS system with a Staff account.
Exit condition	- The Staff has received confirmation that the Client profile has been added, OR - The Staff has received an error indicating why the new profile could not be added, OR - The Staff has received an error indicating why they cannot add a new Client profile.
Quality Requirements	- Adding a new Client should take no longer than 5 seconds.
Traceability	F-13

Use Case Identifier	UC-12
Use case name	StaffViewClientProfile
Participating actors	Initiated by Staff
Flow of events	1. Staff requests to view Client’s profile.

	2. cuACS system responds by displaying the Client’s profile.
Entry condition	- Must be logged into cuACS system with a Staff account.
Exit condition	- The Staff has been displayed the Client’s profile, OR - The Staff has received an error indicating why the Staff cannot view the Client’s profile.
Quality Requirements	- Viewing Client’s profile should take no longer than 3 seconds.
Traceability	F-12

Use Case Identifier	UC-14
Use case name	LaunchACM
Participating actors	Initiated by Staff
Flow of events	1. Staff requests to launch ACM algorithm. 2. cuACS system receives request and informs the Staff that the ACM algorithm will run. The system then runs the ACM algorithm and once it has completed, it will return the results to the Staff.
Entry condition	- Must be logged into cuACS system with a Staff account.
Exit condition	- The Staff has received the results of the ACM algorithm, OR - The Staff has received an error indicating why the LaunchACM failed.
Quality Requirements	- The ACM algorithm should complete and return results to the Staff in less than two minutes.
Traceability	F-14

Use Case Identifier	UC-15
Use case name	ProfileViewError
Participating actors	Client or Staff
Flow of events	1. The cuACS system informs the Client or the Staff that there was an error when trying to view a profile.
Entry condition	- Failed to view the profile of an animal or a client.

Exit condition	- The requested operation is aborted.
Quality Requirements	
Traceability	NF-1-5

Use Case Identifier	UC-16
Use case name	ProfileEditError
Participating actors	Client or Staff
Flow of events	1. The cuACS system informs the Client or the Staff that there was an error when trying to edit a profile.
Entry condition	- Failed to edit the profile of an animal or a client.
Exit condition	- The requested operation is aborted.
Quality Requirements	
Traceability	NF-1-5

Use Case Identifier	UC-17
Use case name	ViewAllError
Participating actors	Client or Staff
Flow of events	1. The cuACS system informs the Client or the Staff that there was an error when trying to display a list of Clients or animals.
Entry condition	- Failed to display the list of available animals or Clients.
Exit condition	- The requested operation is aborted
Quality Requirements	
Traceability	NF-1-5

Use Case Identifier	UC-18
Use case name	ProfileAddError
Participating actors	Staff

Flow of events	1. The cuACS system informs the Staff that there was an error when trying to add a new animal or Client to the system.
Entry condition	- Failed to add a new animal or Client profile.
Exit condition	- The requested operation is aborted.
Quality Requirements	
Traceability	NF-1-5

Use Case Identifier	UC-19
Use case name	LaunchACMError
Participating actors	Staff
Flow of events	1. The cuACS system informs the Staff that there was an error when trying to launch the ACM algorithm.
Entry condition	- Failed to launch ACM
Exit condition	- Requested operation is aborted
Quality Requirements	
Traceability	NF-1-5

2.4.2 Object Model

Table 5: Entity Object Description

Entity Object	Attributes and Associations	Description
Client	● Name ● ID ● Phone number ● Address ● Home square footage ● interests	Customer who wants to adopt an animal he/she find a good match with
Staff	● Name ● ID ● Address ● Phone number ● Department	Member of staff who helps client find match and manages client and animal information

Animal	• Species • Breed • Gender • Age • Colour • Shedding (none, some, lots) • Aggressiveness • Height • Weight	Animal in shelter to be adopted

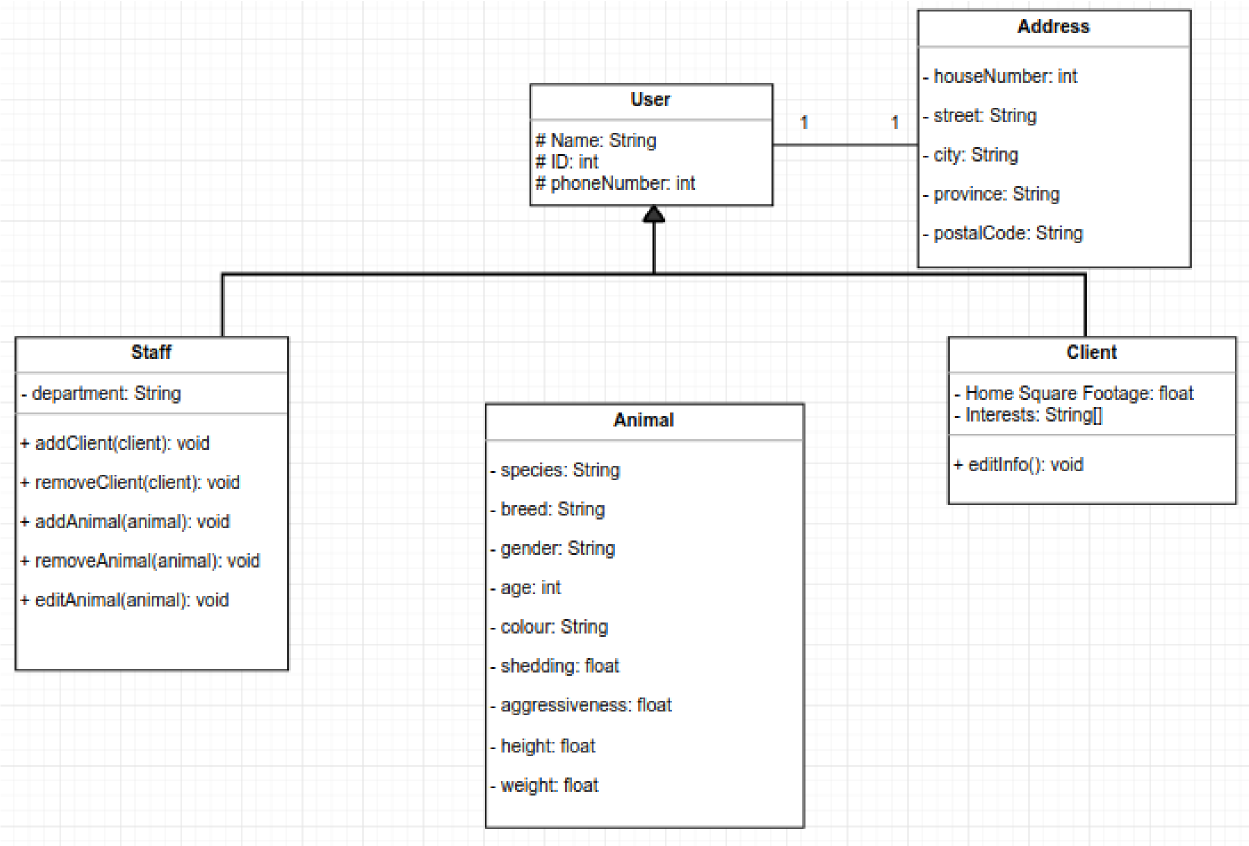

Figure 4: UML Class Diagram

Table 5: Boundary Object Description

Boundary Object	Description
addClient	Adds client to client list
removeClient	Removes client from client list
editClientInfo	Edits client info
clientList	List of clients displayed to the staff

addAnimal	Adds animal to animal list
removeAnimal	Removes animal from animal list
editAnimalInfo	Edits animal info
AnimalList	List of animals displayed to the staff or client
viewAnimalListOption	UI object that triggers an clients or staff request to view the list of animals in the shelter.
viewClientListOption	UI object that triggers an staff request to view the list of clients.
viewReply	List of animals / clients returned by system.

Table 6: Control Object Description

Control Object	Description
systemControl	Manages communications with system.
viewListControl	Initiates the View animal / client List control flow, contacts the system to obtain the appropriate list information, and displays it to the appropriate actor.

Glossary:

ACM: The ‘Animal-Client Matching’ algorithm which will match animals and prospective clients.

Animal: An animal in the shelter that is placed in the cuACS to be adopted.

cuACS: The Carleton University Animal Care System.

Client: An individual who is placed in the cuACS with purpose of adopting an animal.

Staff: Individuals who work at the shelter and will manage the adoption process using cuACS.