# Carleton University

cuACS

Deliverable 1

**Team QuackJaws**

Jake Bauer

Ashlee Foureyes

Skyler Gubbels

Will Watt

Submitted to:

Dr. Christine Laurendeau

COMP3004 – Object-Oriented Software Engineering

School of Computer Science

Carleton University

February 12, 2019

# Contents

1. Introduction

1.1 Purpose of cuACS

1.2 Document Overview

2. Proposed System

2.1 Overview

2.2 Functional Requirements

2.3 Non-Functional Requirements

2.4 System Models

2.4.1 Use Case Model

3. Object Model

3.1 Data Dictionary

3.2 Class Diagrams

# 1. Introduction

## 1.1 Purpose of cuACS

Animal shelters are tasked with the responsibility to provide care to animals in need while they await adoption into a loving home. Humans that are seeking the companionship of a pet can visit a shelter and choose an animal to adopt. However, with this process an issue often arises where a pet is adopted by a human whom they are not fully compatible with. This mismatch can be due to a variety of reasons, including temperament, conflicting lifestyle requirements, and physical and non-physical needs of both the pet and the potential adopter.

Carleton University Animal Care System (cuACS) aims to alleviate the issue of mismatching by providing a tool that automatically matches a pet to a potential owner based on compatibility. This compatibility measurement is based on numerous physical and non-physical traits, as specified by the potential owner. cuACS enables a smooth adoption process and ensures the experience will prove positive and match an animal that fulfills a clients expectations and suits their lifestyle.

## 1.2 Document Overview

This document will outline and describe the functional and non-functional requirements of cuACS. The functional requirements will describe the interactions between the system and its environment independent of its implementation. The non-functional requirements describe the components of the system that are not directly related to its functional behaviour. This document will also present an in-depth look into the functional requirements by providing a use case model.

# 2. Proposed System

## 2.1 Overview

## 2.2 Function Requirements

## 2.3 Non-Functional Requirements

## 2.4 System Models

## 2.4.1 Use Case Model

# 3. Object Model

## 3.1 Data Dictionary

## 3.2 Class Diagrams