**cuACS**

Requirements Analysis

**Table of contents**

1. Introduction
   1. Purpose of System
   2. Overview of Document
2. Proposed System
   1. Overview
   2. Functional requirements
   3. Non-functional requirements
   4. System models
      1. Use case model
      2. Object model
         1. Data dictionary
         2. Class diagrams
      3. Dynamic model
      4. User interface
3. Glossary
4. **Introduction**

**PURPOSE OF SYSTEM**

\* PROBLEM STATEMENT

\* Animal shelter animal-client automatic algorithmic matching service

\* Ensure optimal compatibility between matches

\* Including temperament, needs, expectations

\* Match ensuring optimal *set* of matches, where *all* animals’ interests are met, rather than {small subset}

\* Manage animal profiles

\* Manage client information and preferences

\* Run algorithm to generate an optimal set of matches

\* Support clients and shelter staff

\* {What is included in the animal profiles and client profiles – go into detail here}

**OVERVIEW OF DOCUMENT**

* Its purpose and organization

1. **Proposed System**

**2.1. Overview**

{Summary of this section?}

**2.2 Functional requirements**

{Functional “Table”}

**2.3 Non-functional requirements**

|  |  |  |
| --- | --- | --- |
| Usability | Supportability | Operations |
| Reliability | Implementation | Packaging |
| Performance | Interface | Legal |



1. Product requirements
   1. Usability
      1. ease of use, colour schemes
      2. online help, documentation
   2. Efficiency
      1. Performance
         1. response time, throughput
         2. availability
      2. Space
   3. Reliability
      1. ability to perform under certain conditions
      2. dependability (includes fault-tolerance and security)
   4. Supportability
      1. maintainability
      2. portability
2. Organizational requirements
   1. Delivery
   2. Implementation
   3. Standards
3. External
   1. Interoperability
   2. Ethical
   3. Legal
      1. Privacy
      2. Safety

?? Interface

?? Operations

?? Packaging

Usability

* Clear form fields for form data entry with accompanying confirmation page displaying the updated information
* All button objects need to be intuitive and have clear, concise text detailing their use (noun/verb)
* Required system configurations and installation instructions should be clearly stated in a README document and application wiki page
* Error messages should be readily understood by the average user and suggest possible actions to take
* Consistent UI and colour scheme

Performance

* Algorithm, regardless of number of comparisons made, should be efficient enough to provide results within 5 seconds

Reliability

* The system should handle the case where a different number of pets and clients are trying to be matched and note this in the results

Supportability

* Effectively separate Model, View, and Controller so the front-end may be easily switched out without any modifications of the back-end
* System will be written in C++ and should support systems that can execute C++ binaries

Packaging

* Executable
* README and wiki included

\*\* Handle case where

**2.4 System models**

{Explain system models, what models we are using, and their purpose}

**2.4.1 Use case model**

{Explain use case model, in the context of your system. Explain how this section is organized}

**Use Case Overview**

{Introduce your high-level use cases, depicted below}

{Introduce your detailed use cases with descriptions}

**Use Case Flow of Events**

{Introduce the table-based flow of events, for EACH use case}

**Object Model**

???

**Data Dictionary**

???

**Class Diagrams**

???

**Dynamic Model**

???

**User Interface**

???

1. **Glossary**