<Dog shelter>

Analysis and Design Document

Student: Catargiu Andreea-Alexandra

**Group: 30238**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

# The project is about abandoned dogs or dogs that their owners cannot take care of them

# anymore. There will be users of the application that can interact with each other and the staff

# who run the adoptions.

When a user wants to sign in, he has to choose whether he want to adopt or do give a dog to adoption (he can choose both or just one of them). If he choose the option to adopt, he has to fill some questions about himself, like where he lives (house or apartment), if he even had a dog or even a pet, why he wants to adopt, and to give some arguments according to which the staff will choose him.

Once a user is signed in, he is presented with the option for dogs or accessories for dogs. If he chooses dogs and is looking to adopt, then he can filter the dogs by their breed, size, age or gender. Each dog will have a description, because some people prefer to have a dog with energy, while other prefer to have a dog more calm. If he wants to give a dog for adoption, he will have to give the dog breed, age, size, genre and a short description of the dog. He also has to check a box, where it shows how many days after the posting, if the dog in not taken, wants to be brought to the shelter. He can choose 10 days, 1 month, 3 months, 5 months, 1 year, or he can choose not to be brought. After the owner completes there requests, he has to wait for the staff to give him the access for the dog to be posted on the site.

After log in, if the user chooses the accessories, then he can see the accessories available on the site. The user can filter them by free or with payment. He is given the possibility to add new accessories, but it can only be free.

Any user can donate money for the shelter at any time.

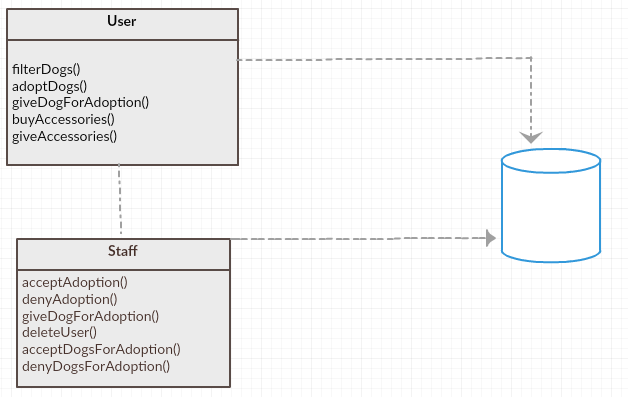
This application will be managed by the staff. They will choose which people will give the dogs available for adoption. They will accept the request of adding another dog or other accessories. Also, they are given the possibility do delete the users who have more deviations, and add dogs for adoption.

# Elaboration – Iteration 1.1

# Domain Model

A domain model is conceptual model of the domain that incorporates both behavior and data. It describes selected aspects of a sphere of knowledge, influence or activity. The model can be used to solve problems related to the domain. The domain model is a representation of meaningful real-world concepts pertinent to the domain that need to be modeled in software. The concepts include the data involved in the business and rules the business uses in relation to that data.

A domain model generally uses the vocabulary of the domain, thus allowing a representation of the model to be communicated to non-technical stakeholders. It should not refer to any technical implementations suah as database or software components that are being designed.



*[Define the domain model and create the conceptual class diagrams]*

# Architectural Design

## Conceptual Architecture

The architectural pattern used in this project, will be the Model-View-Controller pattern. The Model is the central component of the pattern. It is the application’s dynamic data structure, independent of the user interface. It directly manages the data, logic and rules of the application. It does not depend on the controller or the view.

The view helps with the presentation of the model in a particular format. It displays the model data and sends user actions to the controller.

The controller responds to the user input and performs interactions on the data model objects. The controller receives the input, optionally validates it and then passes the input to the model.

Specific for this project, the **model** will be used for **getters and setters** of the objects, like name of the users, name of the dogs, their age, gender, size.

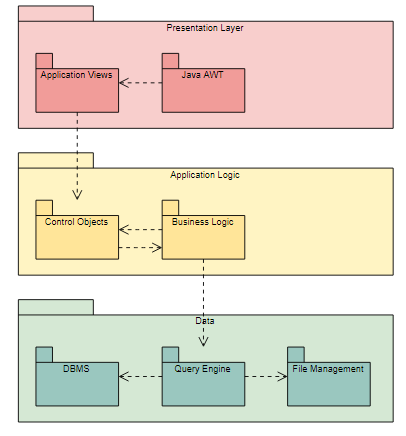
The **view** will be used to **create the necessary interfaces** for the project. We will have an interface for user/staff, log in/ sign in, adopt/give for adoption, get/give accessories.

The **controller** will be used to **implement** delete, insert and update for the necessary classes and also, we will have the **methods** used in the final application.

The MVC (Model-View-Controller), is an architectural pattern which is ease of modification and future development, because of the separation of responsibilities. Also, it enables logical grouping of related actions on a controller together. The view for a specific model are also grouped together.

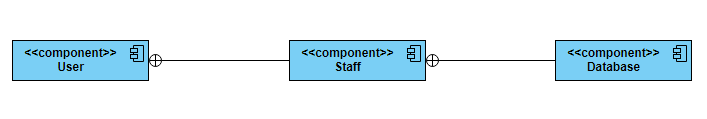
*[Define the system’s conceptual architecture; use an architectural style and pattern - highlight its use and motivate your choice.]*

## Package Design



*[Create a package diagram]*

## Component and Deployment Diagrams



*[Create the component and deployment diagrams.]*

# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

# Data Model

*[Create the data model for the system.]*

# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

Future improvements will be to add a chat box between the users which the adoption takes place, for them to talk more, to see if they are compatible with the dog.

For the dog description, to can add a photo of the dog.

*[Present future improvements for the system]*

# Bibliography

[1] <https://online.visual-paradigm.com/diagrams.jsp?fbclid=IwAR0jPUJ1QLTNAcuY5t4kVZRMV8KKhbCDGtTSjbQ-5M2I_p3itQP5fytMkq8#diagram:proj=0&type=PackageDiagram&gallery=/repository/65e92382-98d6-4066-bd1d-660475ee8ccb.xml&name=MVC%20Structure>

[2] <https://en.wikipedia.org/wiki/Domain_model>

[3] <https://searchsoftwarequality.techtarget.com/answer/Use-cases-scenarios-and-user-goals>

[4] <https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm>

[5] <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>