### JAVA coding standardS

### Introduction

This document contains coding standards and guidelines for the development of the Greyhound Pets of America Web Application. Unless specified otherwise, these guidelines apply to the Java programming language. The purpose of these standards is to promote a consistent and uniform code base. These standards will also help improve the readability and maintainability of the code.

### Coding Standards

This project will follow Oracle’s [Code Conventions for the Java Programming Language](http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html).

### Coding Guidelines

The following sections include guidelines that should be followed, but contain good advice rather than hard rules.

## Security

Security is an important issue to keep in mind while writing code.

1. Ensure that usernames and passwords are not transmitted in clear text.
2. Ensure that any stored passwords are encrypted and salted.
3. Ensure that parameterized SQL statements are used to prevent sql injection.

## UI Guidelines

The user interface design is just as important as the underlying code that drives the application. In general, interface design should attempt to conform to the eight golden rules of interface design.

1) Use a consistent design and commands when developing the user interface.

2) Include shortcuts and options for frequent or power users to increase the pace of user interaction.

3) The interface should provide informative feedback for users in response to their actions.

4) Sequences of actions performed by users should be grouped and designed to provide closure to a user upon completing those actions.

5) Design the system to protect users from committing critical errors and include mechanisms for handling any errors that may occur.

6) Allow users to easily reverse their actions in the event that they make a mistake.

7) Design the user interface so that the user feels in control of the system.

8) Reduce the short-term memory load on users by keeping the design simple and using common menu options and shortcuts.

## Complexity

Modularize programs as much as possible. The McCabe complexity metric (National Institute of Standards and Technology special publication 500–99) can be used as a guideline for measuring the effectiveness of modularization (with case statements allocated a complexity of 1 plus the complexity of the most complex branch, regardless of the number of cases, as suggested by NBS 500–99). A rule of thumb is to keep the complexity of any given unit under ten.

Also, perform refactoring any function containing over 200 lines of code into multiple sub-functions. This improves the readability and testability of the code as long functions tend to be complex.

## Coupling AND Cohesion

Code developed should be highly cohesive and loosely coupled. These properties make code more readable and can reduce the amount of time spent locating and fixing defects in the code.

Cohesion is the measure of how well parts in a module fit together. For example, the methods in the java Math class are all used to calculate mathematical functions. If a module contains many parts that are not related to each other, it may worthwhile to use code refactoring techniques to create multiple new modules.

Coupling is the degree of interaction between different modules in a system. Generally, encapsulation should be used to prevent modules from reaching into each other. Limiting the use of global data and keeping modules from being dependent on another’s implementation details will help achieve loose coupling.