# Java Web Development Basics

# Exam Preparation – Fluffy Duffy Munchkin Cats

**FDMC** (**Fluffy Duffy Munchkin Cats**) is an application for kitten trading. You have been tasked to implement this application for an unusually low price, by an unusually rich client. There are several requirements you must follow in the implementation.

## Technological Requirements

* Use the Javache Web Server
* Use the Broccolina and Toyote request handlers
* Use the Summer Framework
* Use Hibernate native (no Spring Data)

The Technological Requirements are **ABSOLUTE**. If you **do not follow** them, you will **NOT** be scored for other Requirements.

Now that you know the Technological Requirements, let us see what are the Functional Requirements.

## Database Requirements

The **Database** of the **FDMC** application needs to support **2 entities**:

### User

* Has an Username
* Has a Password
* Has an Email

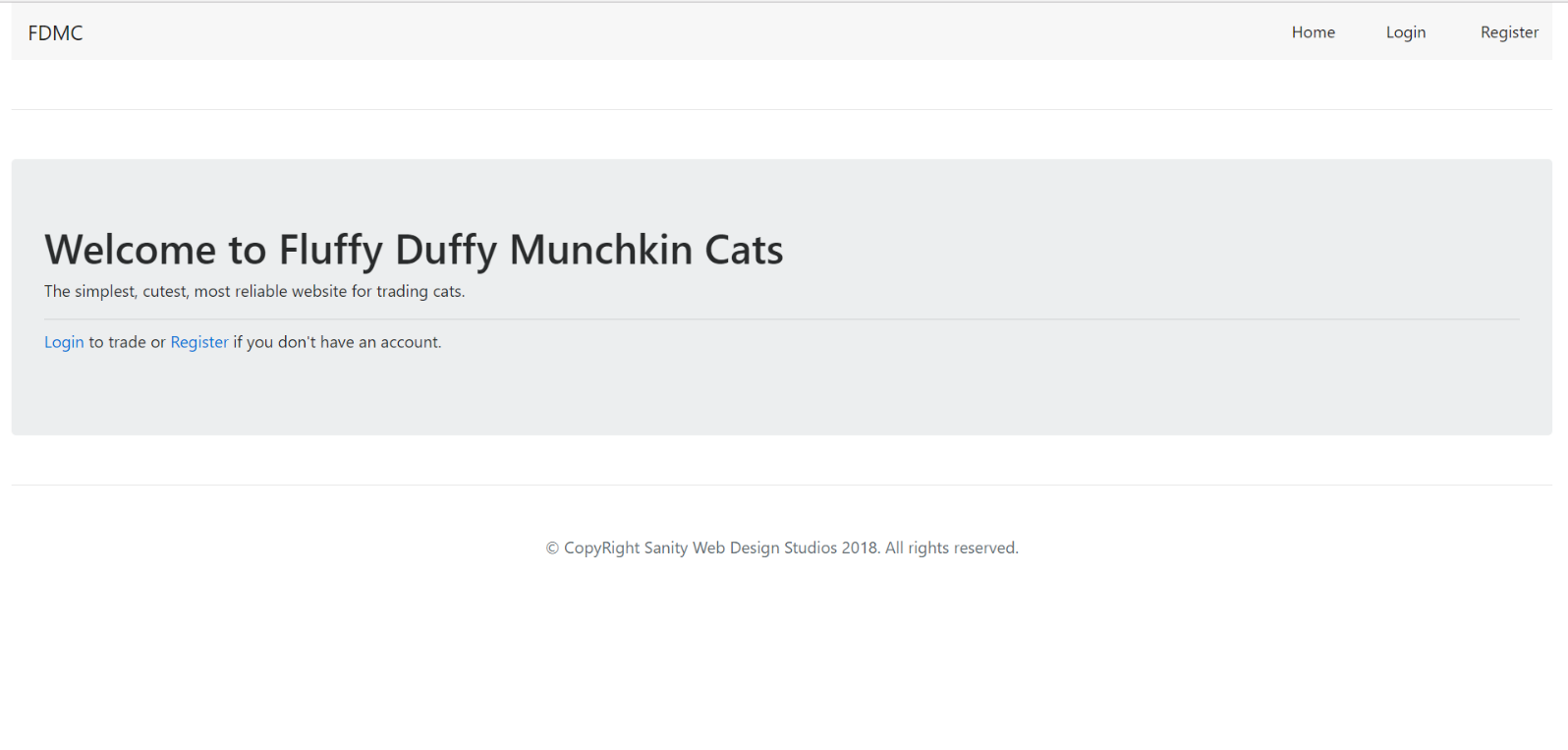
### Kitten

* Has a Name
* Has an Age
* Has a Breed

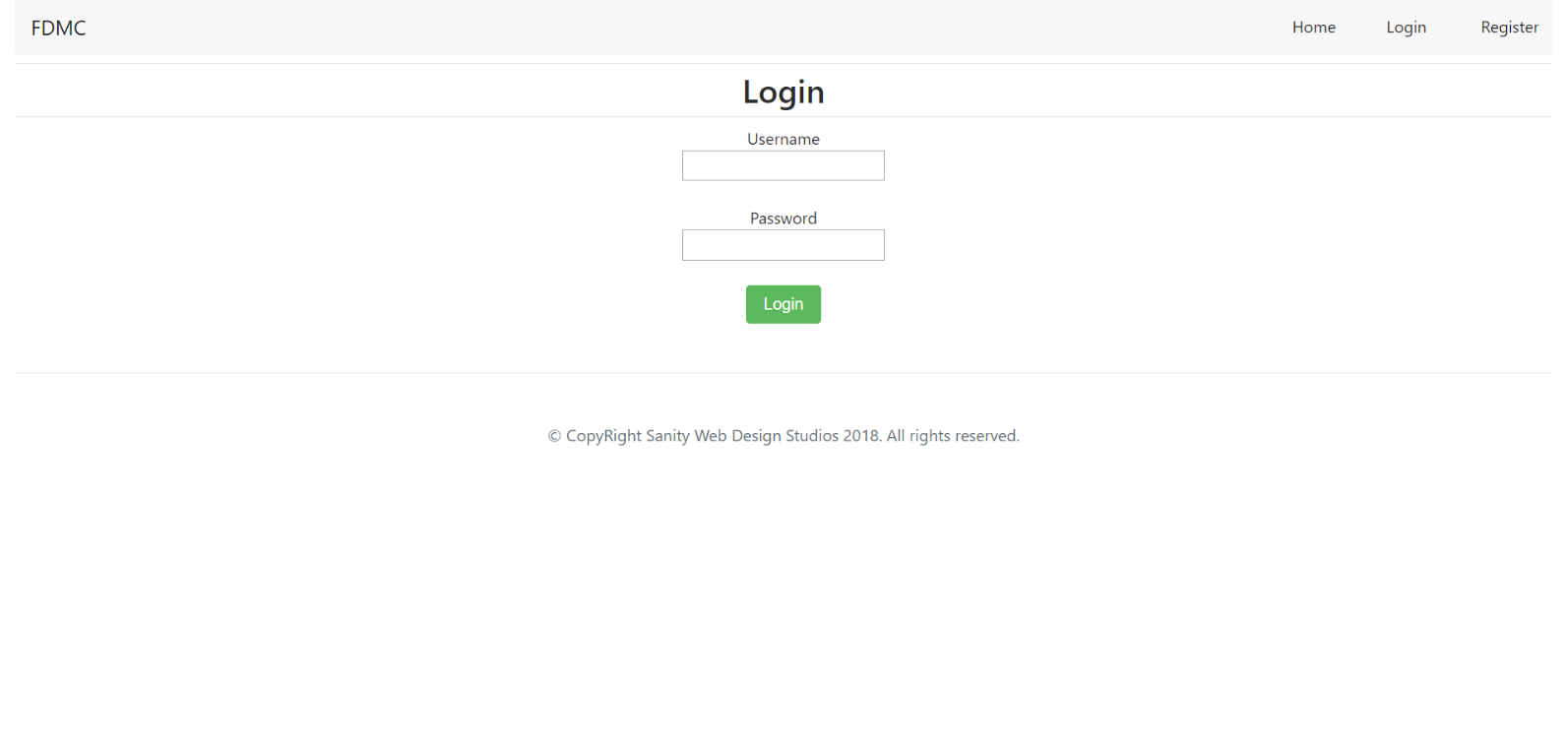
Implement the entities with the **correct datatypes**, and implement **repositories** for them.

## Template Requirements

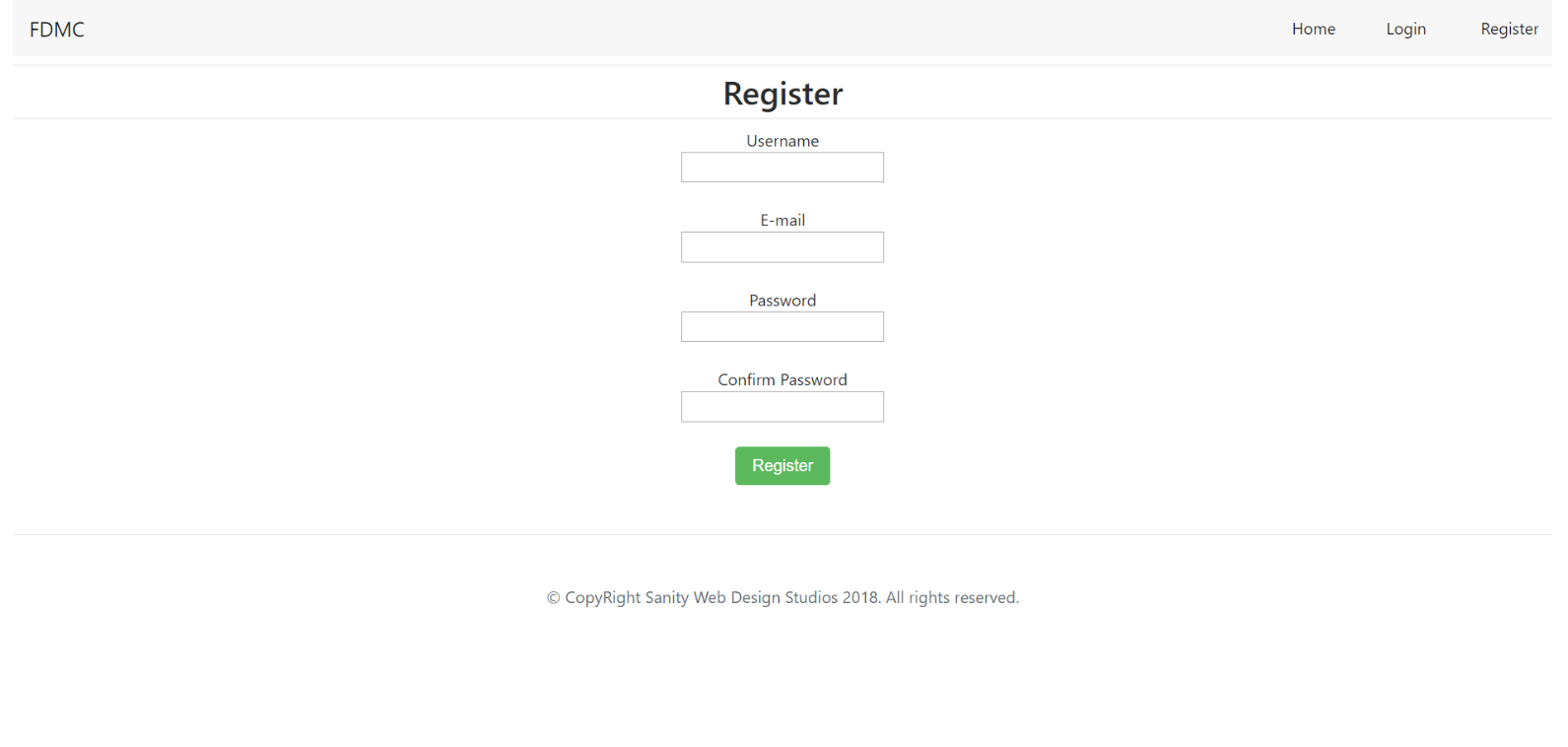
### Index Template (route = “/”) (logged out user)



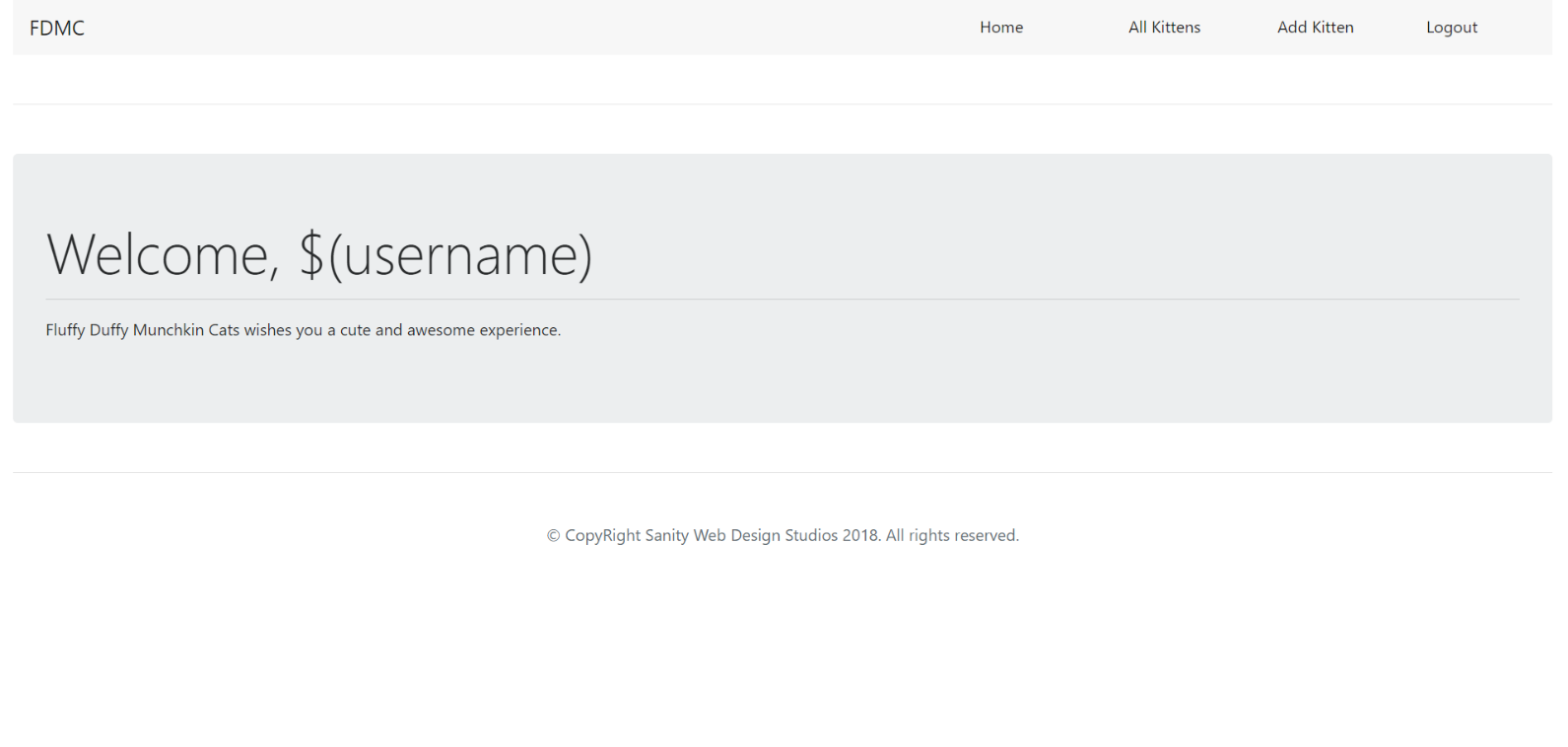
### Login Template (route = “/login”) (logged out user)



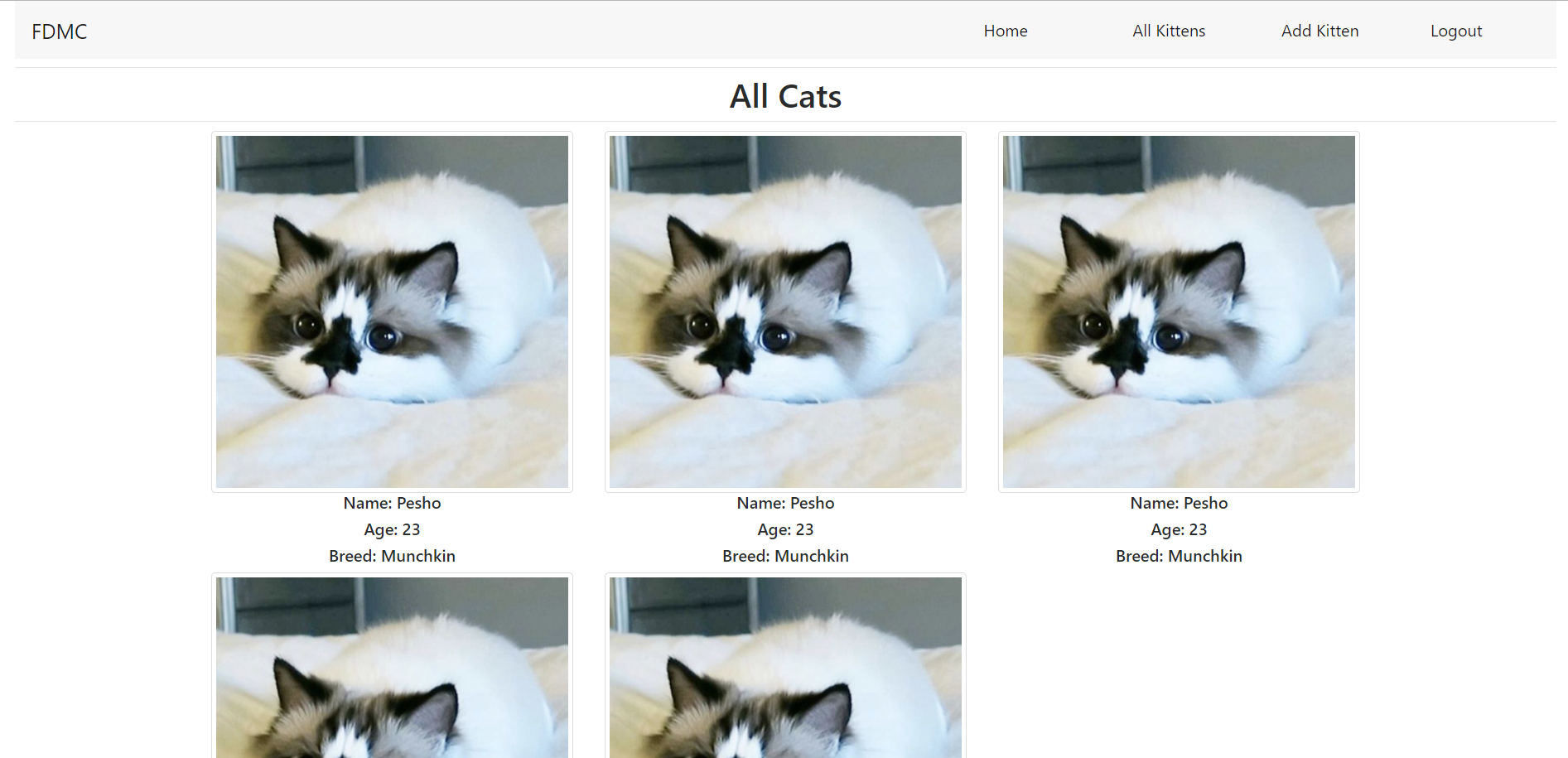
### Register Template (route = “/register”) (logged out user)



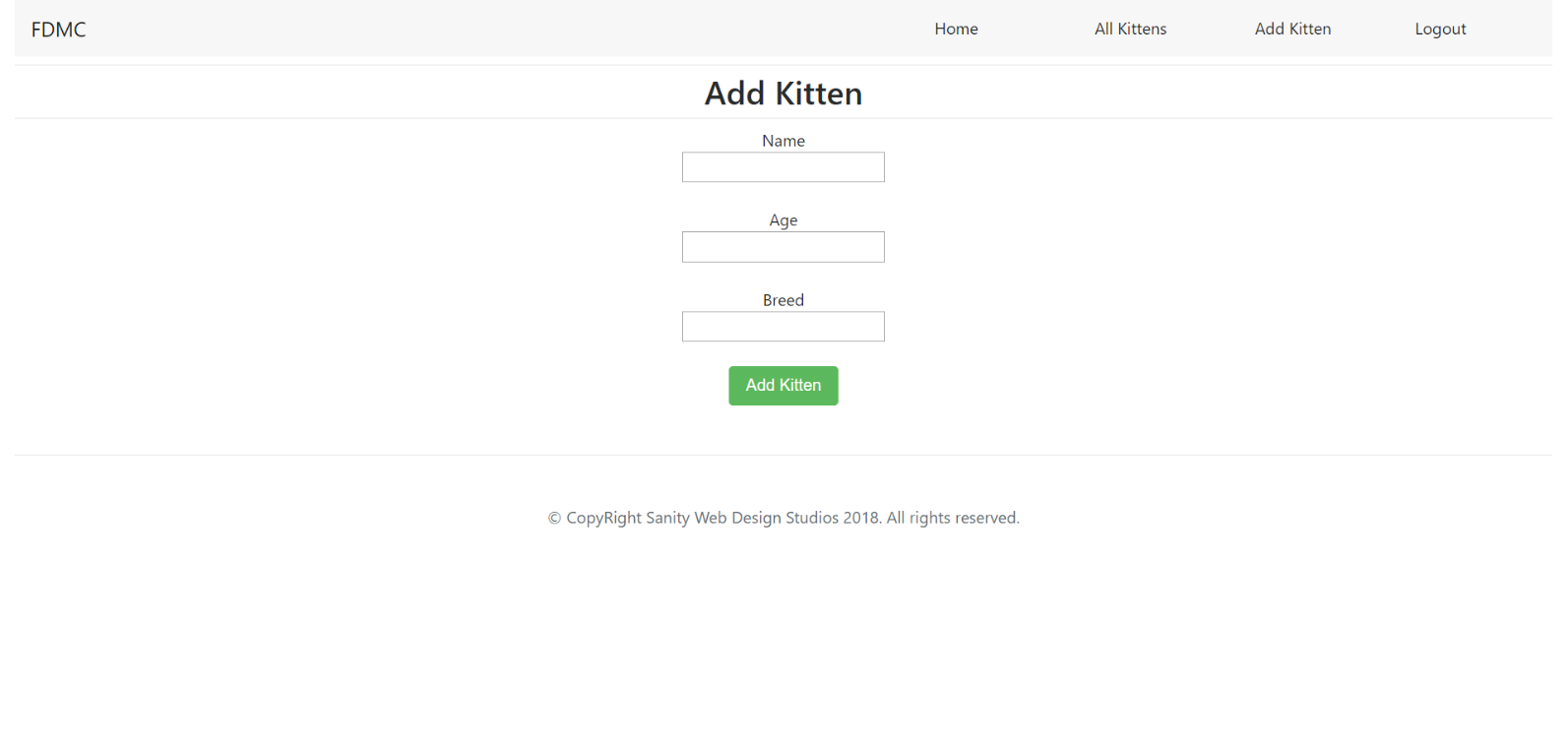
### LoggedIn Template (route=”/”) (logged in user)



### All Kittens Template (route=”/kittens/all) (logged in user)



### Add Kitten Template (route=”/kittens/add”) (logged in user)



Some of the templates have been given to you in the application skeleton, but the others will be for you to implement, so make sure you implement them correctly. You can use the given ones as helpers.

**NOTE**: The templates should look **EXACTLY** as shown above.

**NOTE**: The templates do **NOT** **require** **additional** **CSS**. Only **bootstrap** is enough.

## Functional Requirements

The Functionality Requirements describe the functionality that the **Application** must support.

The **application** should provide Guest (not logged in) users with the functionality to login, register and **view** the Index page.

The **application** should provide Users (logged in) with the functionality to logout, add a kitten, and view all kittens.

The **application** should provide **functionality** for **adding kittens** with **only** **4 breeds** – “Street Transcended”, “American Shorthair”, “Munchkin”, “Siamese”.

The **application** should **store** its **data** into a MySQL database, using Hibernate native.

## Security Requirements

The Security Requirements are mainly access requirements. Configurations about which users can access specific functionalities and pages.

* Guest (not logged in) users can access Index page.
* Guest (not logged in) users can access Login page.
* Guest (not logged in) users can access Register page.
* Users (logged in) can access LoggedIn page.
* Users (logged in) can access All Kittens page.
* Users (logged in) can access Add Kittens page.
* Users (logged in) can access Logout functionality.

## Framework Requirements

Extend the given skeleton (Javache Web Server, Broccolina / Toyote Request Handlers, Summer Framework) by **adding** an **additional annotation**.

The annotation should be named @PreAuthorize. It should be in the scope of Methods, as it should be used on Controller Actions.

The **annotation’s main purpose** should be to **tell an action** that it **requires** a **logged in user**. The **framework** should **scan** the **actions** and If an **action** has this **annotation**, it should **only be executed** if there is a **logged in user**.

## Scoring

This section describes how the scoring of the Exam Preparation will be made.

### Database Requirements – 20 points.

### Template Requirements – 15 points.

### Functional Requirements – 35 points.

### Security Requirements – 10 points.

### Framework Requirements – 20 points.