**PET APPLICATION SYSTEM**

Petpeers is a web based application, which should provide features to pet lovers like enter pet information and view the details. They shall also be able to login and buy pets and list pets owned by them. The application must be flexible enough to adopt new features in future.

The following are the Use Cases for Pet Peers

1. User Registration – New users can register themselves by entering a **unique** user name and password.
2. User Login – Users should be able to login into the application by inputting correct user name and password
3. View List of all Pets – This functionality lists all pets and also indicates whether if a Pet is ‘Sold Out’ or available for purchase. If available, provides a link to buy that Pet.
4. Add a Pet – A user can enter a new Pet to the list
5. Buy a Pet – If a User likes an available Pet, she / he can buy the same by clicking on ‘Buy’ button
6. List of owned Pets – A loggin user can view the list of Pets owned by het / him by clicking on ‘My Pets’ button
7. User Logout – A loggedin user can logout of the application by clicking on ‘Logout’ button. She / he can access the application functional only after login again.

The trainee will be provided with a partially implemented eclipse web application project template and the design specification (this document) that contains instruction to develop an application. The trainee is expected to create the necessary objects and components in the given project template.

The template project structure follows the Maven web application structure.

# System Components

The expected petpeers application has been divided into 3 components. Such as,

1. Web UI
2. Business Components
3. Data Access Components

The tools/technologies for developing these components are as follows,

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Component | Technology/Tools | Notes |
| 1. | Web UI | View pages (implemented by JSP using HTML, CSS, JS etc) | Empty JSP pages are included in eclipse template project |
| 2. | Web UI | Spring MVC Components – Controller, Model classes for data handling and Validators for user input | Controller to handle the request, invoke server components and send response back to user. Template classes and configuration are included |
| 3. | Business Service Components | Business Service components to be invoked by Controller/s implemented as Spring Beans in Java | Classess with empty implementations included. |
| 4. | Data Access Components | DAO components for data persistence and retrieval implemented using Hibernate ORM framework | Partial Configuration and empty classes included in eclipse project template |