**Low-Level Architecture and Data Models**

**P01:PetsWala**

| **Student ID** | **Name** |
| --- | --- |
| **22100300** | **Muhammad Aaish Javed** |
| **22100098** | **Muhammad Ibrahim Bhalli** |
| **22100200** | **Muhammad Tayyab** |
| **20100049** | **Syed Raza Abbas** |

**Table of Contents**

[1.](#_gjdgxs) Introduction 3

[2.](#_30j0zll) System Architecture 4

[2.1 Architecture Diagram—](#_1fob9te)As it is in the prototype code 4

[2.2 Architecture Diagram—](#_3znysh7)As it should-be 4

[3.](#_2et92p0) Data Models 5

[4.](#_tyjcwt) Tools and Technologies 6

[5.](#_3dy6vkm) Who Did What? 7

[6.](#_1t3h5sf) Review checklist 7

# Introduction

**1.1 Product Scope:**

We will be developing a web application that registers pet owners, vets, other pet service and accessories providers, and possibly pet experts and allows them to interact based on their needs. For instance, pet service providers and vets would be able to advertise their services and products and pet owners would be able to contact these service providers as well as connect with other pet owners for any purpose such as breeding, purchasing, or selling.

**1.2 Potential Users:**

As mentioned earlier, potential users of the web application will be key characters in the pets’ communities:

· Pet owners

· Service providers

· Rescue Services

· Vets

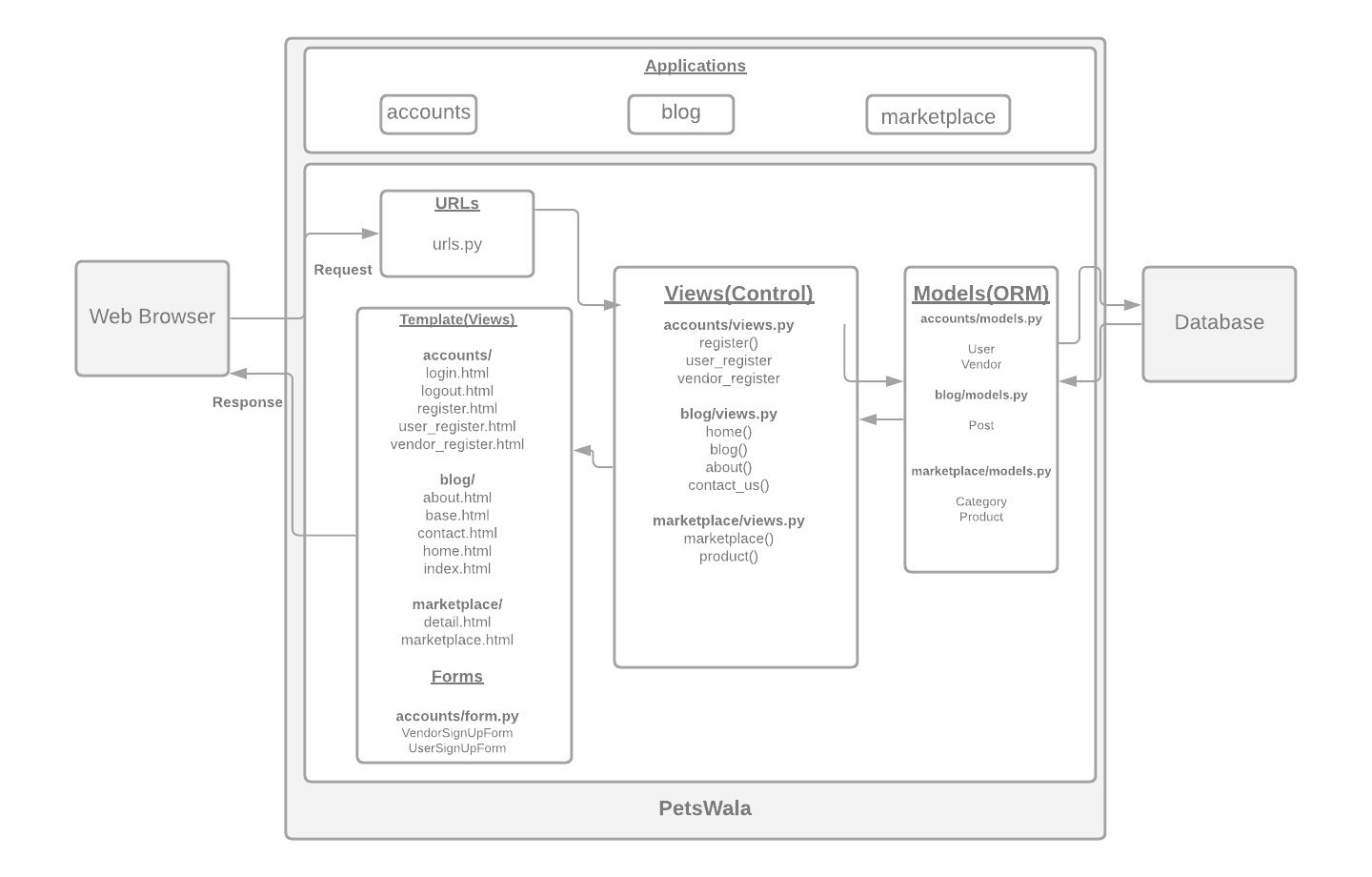
· Pet Buyers

· Pet Sellers

· Business administrators

# System Architecture

## Architecture Diagram—As it is in the prototype code



## Architecture Diagram—As it should-be

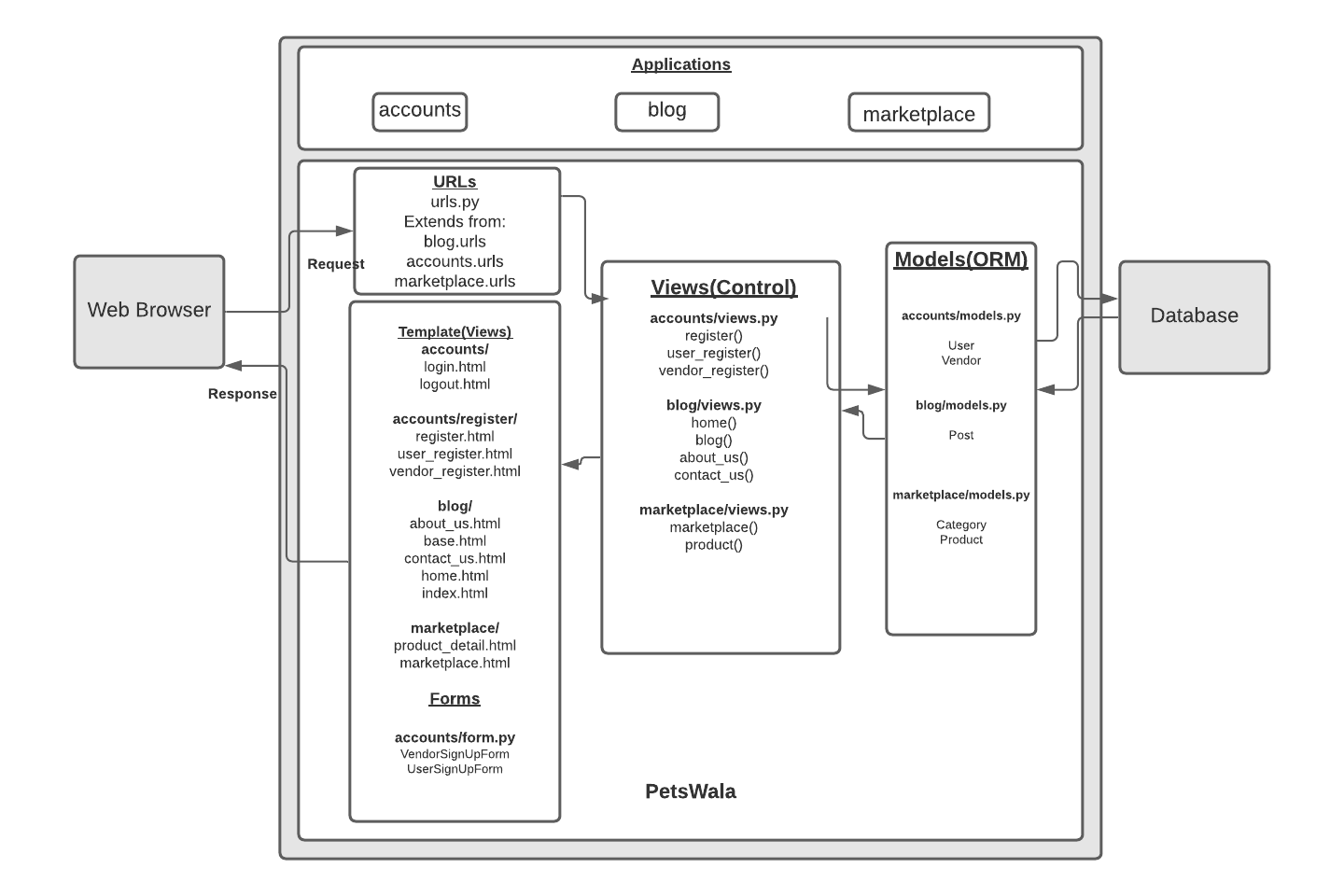
<Draw an updated diagram of the above system architecture. The diagram must reflect the architecture of the code as it should be. For instance, you may have learnt during prototyping that you should structure your code in a different way to make it, *maintainable*, for instance. This diagram should reflect this updated understanding. When you rethink your system architecture, keep the following important design goals in mind:

* Maintainability
* Reusability
* Extensibility
* Separation of concerns

In a couple of paragraphs, discuss how your updated architecture will help in achieving design goals listed above.

If you are following layered architecture, your diagram must distinguish all layers along with classes/modules in each layer. You must mention exact names of classes/modules inside the layers as they are in your prototype code. Similar statements hold if you are following another architectural pattern. If there are too many classes, your diagram must show classes/modules involved in at least two use cases.

Add a brief description of the classes/modules in the architecture diagram.



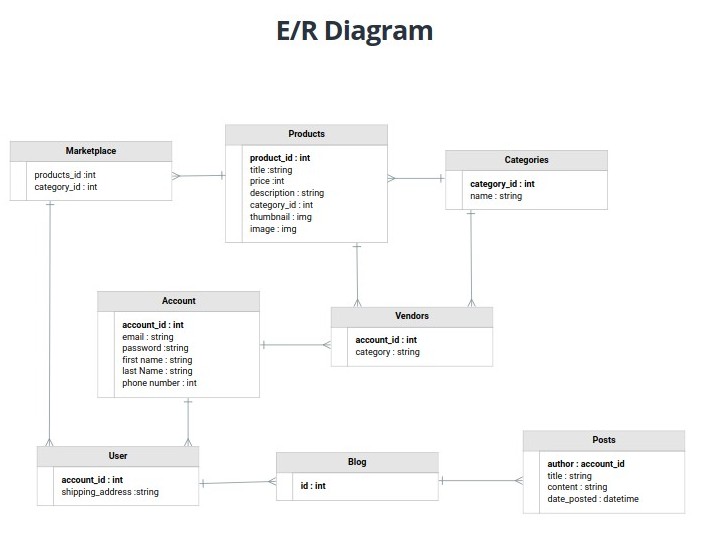
The updated URLs extend from the URLs of specific applications. This way every application along with its URLs is modular. We only have to make changes within the local URLs of the blog, for example, and they will automatically be extended to the main urls.py file. This also keeps the main urls.py file from being too unmanageable, minimizing the lines of code in it.

The templates containing the HTML files are already modular and in separate applications, and every application only contains its respective files. This also helps in the reusability of code as every application only contains its own HTML files. The updated register folder in the accounts template will improve the maintainability of the signup process.

The views are already limited to their respective applications which satisfy our goal of maintainability, reusability, and extensibility. The same goes for the models.

We need to update our code with comments and proper documentation. We somewhat have Pascal’s case naming convention, but we need to extend it to the whole codebase.

# Data Models



**Brief Description:**

* **Products:** This entity has data of all the products that the vendor will publish on the marketplace.
* **Categories:** This entity has all the categories that vendors will add.
* **Account:** This entity has data of all the accounts made on the website.
* **Vendors:** This entity is inherited from the account and will have the data of vendors.
* **User:** This entity will keep data of general users who will benefit from services provided on the website.
* **Blog:** This entity will keep a record of all the posts inherited from the entity named **“Posts”.**
* **Marketplace:** This entity inherits data from entities “**Products**” and “**Categories**” and offers it to “**User”.**

# Tools and Technologies

We will be using the following Tools and Technologies for development and deployment:

* Django(v3.0 or above)
* Python(v3.7 or above)
* PostgreSQL (v12.3)
* AWS CLI (v2)
* HTML5
* CSS (v2.1)
* Trello
* Javascript (ES2015)

# Who Did What?

| **Name of the Team Member** | **Tasks done** |
| --- | --- |
| Muhammad Aish Javed | Nothing |
| M. Ibrahim Bhalli | Updated Architecture Diagram and description |
| Muhammad Tayyab | Data Model (E/R diagram, description), Introduction, Tools, and technologies. |
| Syed Raza Abbas | Architecture Diagram |

# Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

| **Section** **Title** | **Reviewer Name(s)** |
| --- | --- |
| Nothing | Muhammad Aish Javed |
| Tools and Technologies, Data Models | M. Ibrahim Bhalli |
| System Architecture | Muhammad Tayyab |
| Data Models | Syed Raza Abbas |