5. SOFTWARE ARCHITECTURE:

Với Frameword Django ta thấy rằng mô hình cũng tương tự như các framework cũng sử dụng architecture MVC(Model-View-Controller), tuy nhiên điểm khác biệt ở đây thì các nhà phát triển lại sử dụng architecture là MVT(Model-View-Template), nhưng cốt lõi vẫn là MVC, vì chính bản Django đã là controller .

Dựa vào các yêu cầu về chi phí, phi chức năng – hoạt động, hiệu suất, bảo mật, …

để phục vụ cho việc phát triển ứng dụng web của nhóm thì mô hình MVC (Model – View –

Controller) đáp ứng được các tiêu chí:

● Dễ dàng tiếp cận.

● Trình tự xử lý rõ ràng.

● Không phụ thuộc môi trường, nền tảng xây dựng hay ngôn ngữ lập trình phát triển.

● Mô hình đơn giản, dễ hiểu và dễ dàng triển khai với các dự án vừa hoặc nhỏ.

● Thuận lợi trong việc phát triển, quản lý, vận hành, bảo trì.

A screenshot of a computer

Description automatically generated with low confidence● Tạo được các chức năng chuyên biệt hóa đồng thời kiểm soát được luồng xử lý

Mô hình MVC được chia làm 3 lớp xử lý gồm **Model – View – Controller :**

**Model** : là nơi chứa những nghiệp vụ tương tác với dữ liệu hoặc hệ quản trị cơ sở dữ liệu (mysql, mssql,sqlite… ); nó sẽ bao gồm các class/function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm – xóa – sửa dữ liệu…

**View** : là nơi chứa những giao diện như một nút bấm, khung nhập, menu, hình ảnh… nó đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.

**Controller** : là nơi tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/ function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp View.

**Mô hình MVT:**

Dù tuân thủ theo MVC nhưng Django vẫn có những quy ước riêng của nó trong đó các lập trình viên sẽ sử dụng mô hình MVT để phát triển (có thể xem Django chính là bản thân Controller)

**Model**: Cũng giống như Model trong MVC, ở đây nó cũng có chức năng tương tự là cung cấp giao diện cho dữ liệu được lưu trữ trong cơ sở dữ liệu

**View**. Trong Django, View hoạt động như một liên kết giữa data từ Model và giao diện, cũng giống như controller trong MVC, các View trong Django MVT chịu trách nhiệm xử lý tất cả các logic nghiệp vụ đằng sau ứng dụng web. Nó hoạt động như một cầu nối giữa Model và Template. Nó xem yêu cầu của người dùng, truy xuất dữ liệu thích hợp từ database , sau đó kết xuất lại template cùng với data đã truy xuất, do đó không có Controller riêng trong kiến ​​trúc Django MVT và mọi thứ đều dựa trên Model -View Template và do đó có tên MVT .

**Template** : Cũng giống như View trong MVC, Django sử dụng các template trong khuôn khổ của nó. Template chịu trách nhiệm hoàn toàn cho toàn bộ giao diện người dùng. Nó xử lý tất cả các phần tĩnh của trang web cùng với HTML,CSS mà người dùng truy cập trang web sẽ thấy.

**Kiến trúc hệ thống của MVT:**

Diagram

Description automatically generated

1. Người dùng gửi yêu cầu URL cho một tài nguyên tới Django.
2. Sau đó, Controller của Django sẽ tìm kiếm tài nguyên URL.
3. Nếu đường dẫn URL liên kết đến một View, thì view của chế độ đó được gọi.
4. Sau đó, View sẽ tương tác với Model và lấy dữ liệu thích hợp từ database.
5. Sau đó, Chế độ xem kết xuất lại một form thích hợp cùng với dữ liệu đã truy xuất cho người dùng.

**Kiến trúc ứng dụng của MVT:**

Diagram

Description automatically generated