# Tài liệu Hướng dẫn Phát triển

## Tiêu chuẩn Mã hóa và Hướng dẫn Phát triển

\*\*Tiêu chuẩn Mã hóa và Hướng dẫn Phát triển\*\*  
  
\*\*1. Giới thiệu\*\*  
  
Tài liệu này mô tả các tiêu chuẩn mã hóa và hướng dẫn phát triển cho dự án, dựa trên tài liệu 'Configuration\_Management\_Plan.docx' và 'Project\_Plan.xml'. Vì không cung cấp nội dung cụ thể của hai tài liệu này, nên các tiêu chuẩn được thiết lập dựa trên các thực tiễn tốt nhất trong ngành. Giả sử ngôn ngữ lập trình chính là Python và framework là Django.  
  
\*\*2. Quy tắc đặt tên\*\*  
  
\* \*\*Biến:\*\* Sử dụng snake\_case (ví dụ: `my\_variable`, `user\_id`).  
\* \*\*Hàm:\*\* Sử dụng snake\_case (ví dụ: `calculate\_total`, `get\_user\_data`).  
\* \*\*Lớp:\*\* Sử dụng CamelCase (ví dụ: `UserData`, `OrderProcessor`).  
\* \*\*Module:\*\* Sử dụng snake\_case (ví dụ: `user\_module`, `order\_processing`).  
\* \*\*Hằng số:\*\* Sử dụng toàn chữ in hoa với dấu gạch dưới (ví dụ: `MAX\_VALUE`, `DATABASE\_URL`).  
  
  
\*\*3. Cấu trúc thư mục\*\*  
  
Sử dụng cấu trúc thư mục logic và dễ hiểu. Ví dụ:  
  
```  
project\_name/  
├── app1/  
│ ├── models.py  
│ ├── views.py  
│ ├── urls.py  
│ └── ...  
├── app2/  
│ ├── models.py  
│ ├── views.py  
│ ├── urls.py  
│ └── ...  
├── templates/  
│ ├── app1/  
│ │ └── ...  
│ └── app2/  
│ └── ...  
├── static/  
│ ├── css/  
│ ├── js/  
│ └── img/  
└── ...  
```  
  
\*\*4. Quy ước comment\*\*  
  
\* Sử dụng comment để giải thích mục đích của mã, không phải là cách thức hoạt động của nó (đã rõ ràng trong mã).  
\* Viết comment ngắn gọn, súc tích và dễ hiểu.  
\* Sử dụng docstrings cho các hàm và lớp để tạo tài liệu tự động.  
  
\*\*5. Thực hành tốt nhất (Python/Django)\*\*  
  
\* Sử dụng `virtualenv` để quản lý môi trường ảo.  
\* Sử dụng Git để quản lý phiên bản mã nguồn.  
\* Thực hiện kiểm thử đơn vị (unit testing) đầy đủ.  
\* Tuân thủ nguyên tắc DRY (Don't Repeat Yourself).  
\* Sử dụng các pattern thiết kế phù hợp (ví dụ: MVC).  
\* Thực hiện code review trước khi merge code.  
  
  
\*\*6. Component Diagram (Graphviz DOT)\*\*  
  
Giả sử ứng dụng có 3 module chính: `UserModule`, `ProductModule`, `OrderModule`. `UserModule` tương tác với cả `ProductModule` và `OrderModule`.

## Component Diagram

