**Giới thiệu chương:**

Tập trung trình bày về dự án phần mềm web bán hàng yame mô tả bài toán, giới thiệu về quy trình mô hình quy trình và công nghệ áp dụng trong phát triển YameWeb, công cụ hỗ trợ cũng như phương pháp phân tích về thiết kế phần mềm.

## 1.1. Giới thiệu chung

- Phần mềm sẽ do người lập trình tạo lên nhằm mục đích bán quần áo. Website giúp cho khách hàng dễ dàng mua sắm online những mặt hàng của shop cũng như nhân viên quản lí, nhân viên bán hàng có thể dễ dàng quản lí được số lượng hàng bán ra và số lượng hàng còn trong kho.

- GT: Yame là chuỗi cửa hàng thời trang dành cho thanh niên lứa tuổi từ 15 đến 25 tuổi mang phong cách hiện đại. Yame có địa chỉ tại Thành Phố Hồ Chí Minh YaMe và Kenshin Tea Quận Tân Phú: 189 Hòa Bình với hệ thống bán hàng gồm 36+ chi nhánh trên toàn quốc.

- Để ứng dụng thương mại điện tử và phát triển kênh bán hàng trực tuyến trong thời đại công nghệ 4.0, Yame đã có ý tưởng xây dựng phần mềm bán hàng trực tuyến trên nền tảng web (được gọi là YameWeb). Không những vậy, hệ thống còn cho phép Yame quản lý nhân lực và hàng hóa trong toàn bộ hệ thống của mình thông qua ứng dụng web. Điều này sẽ giúp Genz mở rộng thị trường và tiết kiệm được nhân công. Từ đó giúp Yame tăng thêm khả năng cạnh tranh trên thị trường và có chỗ đứng vững trãi trong thương mại điện tử.

## 1.2. Mô tả bài toán

Phần mềm webyame dùng để đáp ứng nhu cầu mua hàng của khách hàng đối với chuỗi cửa hàng yame, phần mềm quản lý khách hàng, nhân viên, sản phẩm.

### **1.2.1. Quản lý người dùng**

Mỗi người dùng hệ thống đều phải có một tài khoản bao gồm Account + Password để đăng nhập hệ thống.

**Các dạng người dùng bao gồm:**

* Người quản trị hệ thống.
* Người điều hành hệ thống.
* Nhân viên (Có thể có nhiều dạng nhân viên: bán hàng, kho, vận chuyển, bảo hành, …).
* Khách hàng thân thiết (có tài khoản được quản lý bởi Yame).
* Khách hàng (người dùng này là trường hợp không cần có Account + Password).

### **1.2.2. Quản lý hệ thống**

Quản lý hệ thống đều phải có một tài khoản bao gồm Account + Password để đăng nhập hệ thống.

Quản trị có thể xem các thông tin như: Quản trị sản phẩm, tài khoản, tìm kiếm, xem thông tin sản phẩm.

### **1.2.3. Quản lý khách hàng thân thiết**

Phần quản lý khách hàng thân thiết và nhân viên: Khách hàng thân thiết và nhân viên được cần đăng ký tài khoản và các thông tin khác:

* Khách hàng thân thiết: Đăng ký tài khoản đi kèm các thông tin phục vụ cho việc giao hàng và thanh toán (nếu có). Ngoài ra, hệ thống còn có các ưu đãi dành cho khách hàng thân thiết của Yame. Các thông tin bao gồm: Họ tên, tuổi, số điện thoại, địa chỉ thường trú, địa chỉ giao hàng, email.
* Nhân viên: Lưu trữ thông tin nhân viên bao gồm: Họ tên, tuổi, chức vụ, địa chỉ thường trú, số điện thoại, email, số chứng minh nhân dân, quê quán. Các nhân viên được giao nhiệm vụ bởi người điều hành.

### **1.2.4. Quản lý khách hàng thường**

Phần quản lý người quản trị hệ thống và người điều hành hệ thống: Người quản trị hệ thống và người điều hành hệ thống được cần đăng ký tài khoản và các thông tin khác:

* Người quản trị hệ thống, người điều hành hệ thống: Đăng ký tài khoản đi kèm các thông tin cá nhân phục vụ cho viện liên lạc, trao đổi với công ty tình hình của hệ thống. Các thông tin cá nhân bao gồm: Họ tên, tuổi, địa chỉ thường trú, số điện thoại, email, chức vụ, số chứng minh nhân dân, quê quán.

## 1.3. Tìm hiểu về quy trình phát triển phầm mềm, mô hình quy trình thác nước.

### **1.3.1. Quy trình phát triển phần mềm**

Vòng đời phát triển phần mềm (SDLC-Software Development Life Cycle) là một quá trình bao gồm một loạt các hoạt động được lên kế hoạch để phát triển hoặc thay đổi sản phẩm Phần mềm. SDLC còn được gọi là Quy trình phát triển phần mềm.

SDLC là một quy trình được ngành công nghiệp phần mềm sử dụng để thiết kế, phát triển và kiểm tra phần mềm chất lượng cao. SDLC nhằm mục đích tạo ra một phần mềm chất lượng cao đáp ứng hoặc vượt quá mong đợi của khách hàng, hoàn thành trong thời gian và chi phí ước tính.

SDLC là một quy trình theo sau cho một dự án phần mềm, trong một tổ chức phần mềm, bao gồm một kế hoạch chi tiết mô tả cách phát triển, bảo trì, thay thế và thay đổi hoặc nâng cao phần mềm cụ thể.

SDLC xác định một phương pháp luận để cải thiện chất lượng của phần mềm và quá trình phát triển tổng thể.

SDLC là một khuôn khổ xác định các tác vụ được thực hiện ở mỗi bước trong quy trình phát triển phần mềm.

**Các hoạt động cơ bản của quy trình phát triển phần mềm:**

**Stage 1:** Planning and Requirement Analysis (Lập kế hoạch và Phân tích yêu cầu)

**Stage 2:** Defining Requirements (Xác định/Định nghĩa yêu cầu)

**Stage 3:** Designing the Product Architecture (Thiết kế Kiến trúc Sản phẩm)

**Stage 4**: Building or Developing the Product (Xây dựng/phát triển sản phẩm)

**Stage 5:** Testing the Product (Kiểm tra sản phẩm)

**Stage 6:** Deployment in the Market and Maintenance (Triển khai/phát hành trên thị trường và duy trì/bảo trì)

### **1.3.2. Công cụ / kỹ thuật áp dụng**

**\*Phần mềm Rational Rose/ Draw.io**

Rational rose là phần mềm công cụ mạnh hỗ trợ phân tích, thiết kế hệ thống phần mềm theo đối tượng. Nó giúp ta mô hình hóa hệ thống trước khi viết mã chương trình. Rational rose hỗ trợ cho việc làm mô hình doanh nghiệp, giúp bạn hiểu được hệ thống của mô hình doanh nghiệp. Giúp chúng ta phân tích hệ thống và làm cho chúng ta có thể thiết kế được mô hình.

Mô hình Rational Rose là bức tranh của hệ thống từ những phối cảnh khác nhau nó bao gồm tất cả các mô hình UML, actors, use cases, objects, component, và development nodes v.v…, trong hệ thống. Nó mô tả chi tiết mà hệ thống bao gồm và nó sẽ làm việc như thế nào vì thế người lập trình có thể dùng mô hình như một bản thiết kế cho công việc xây dựng hệ thống.

[Draw.io](https://tinhte.vn/tag/drawio) là một công cụ [vẽ sơ đồ](https://tinhte.vn/tag/ve-so-do) rất mạnh mẽ, hỗ trợ nhiều hình khối, chạy online không cần cài đặt mà lại miễn phí và không bị giới hạn số biểu đồ như nhiều tool vẽ nền web khác. Bạn có thể vẽ [sơ đồ](https://tinhte.vn/tag/so-do) về mạng, điện, phác thảo vị trí các căn phòng trong nhà, hay vẽ các quy trình kinh doanh, vận hành, sản xuất. Anh em làm công nghệ cũng sẽ thích Draw.io vì nó cho phép bạn vẽ hàng tá sơ đồ thiết kế phần mềm, phần cứng và hệ thống. Draw.io có thư viện template rất phong phú để bạn có thể bắt đầu nhanh hơn, không phải tự mình vẽ lại hết tự đầu.

**\*Phần mềm StarUML**

StarUML là một UML công cụ mô hình nguồn mở hỗ trợ khả năng tạo ra các thiết kế phần mềm từ các khái niệm cơ bản để giải mã. Các StarUML là để xây dựng một công cụ mô hình phần mềm và một nền tảng mà có thể thay thế công cụ UML thương mại như Rational Rose, hoặc cùng nhau. Công cụ này là phức tạp hơn nhiều so với một công cụ chỉnh sửa sơ đồ UML đơn giản. StarUML được viết chủ yếu ở Delphi, nhưng thực sự là một dự án đa ngôn ngữ. Nó không phải được gắn với một ngôn ngữ lập trình cụ thể.

**\* Subline Text**

Sublime Text là một trình soạn thảo mã nguồn thương mại. Nó hỗ trợ nhiều ngôn ngữ lập trình và ngôn ngữ đánh dấu. Người dùng có thể mở rộng chức năng của nó bằng các plugin, thường do cộng đồng xây dựng và duy trì theo giấy phép phần mềm miễn phí. Để tạo điều kiện cho các plugin, Sublime Text có API Python.

**\* Visual Studio Code**

Visual Studio Code là một trình biên tập mã được phát triển bởi [Microsoft](https://vi.wikipedia.org/wiki/Microsoft) dành cho [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux) và [macOS](https://vi.wikipedia.org/wiki/MacOS). Nó hỗ trợ chức năng debug, đi kèm với [Git](https://vi.wikipedia.org/wiki/Git_(ph%E1%BA%A7n_m%E1%BB%81m)), có syntax highlighting, tự hoàn thành mã thông minh, [snippets](https://vi.wikipedia.org/w/index.php?title=Snippets&action=edit&redlink=1), và [cải tiến mã nguồn](https://vi.wikipedia.org/wiki/C%E1%BA%A3i_ti%E1%BA%BFn_m%C3%A3_ngu%E1%BB%93n). Nó cũng cho phép tùy chỉnh, do đó, người dùng có thể thay đổi theme, phím tắt, và các tùy chọn khác. Nó miễn phí và là [phần mềm mã nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_t%E1%BB%B1_do_ngu%E1%BB%93n_m%E1%BB%9F) theo giấy phép MIT, mặc dù bản phát hành của [Microsoft](https://vi.wikipedia.org/wiki/Microsoft) là theo [giấy phép phần mềm](https://vi.wikipedia.org/wiki/Gi%E1%BA%A5y_ph%C3%A9p_ph%E1%BA%A7n_m%E1%BB%81m) miễn phí.

Visual Studio Code được dựa trên [Electron](https://vi.wikipedia.org/wiki/Electron), một nền tảng được sử dụng để triển khai các ứng dụng [Node.js](https://vi.wikipedia.org/wiki/Node.js) [máy tính cá nhân](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh_c%C3%A1_nh%C3%A2n) chạy trên động cơ bố trí Blink. Mặc dù nó sử dụng nền tảng Electron nhưng phần mềm này không phải là một bản khác của [Atom](https://vi.wikipedia.org/wiki/Atom), nó thực ra được dựa trên trình biên tập của [Visual Studio Online](https://vi.wikipedia.org/wiki/Microsoft_Visual_Studio).

### **1.3.3. Phương pháp phân tích và thiết kế phần mềm**

**Thiết kế hướng đối tượng:**

* Hệ thống được nhìn nhận như 1 bộ các đối tượng tương tác với nhau, đối tượng gồm dữ liệu + thao tác.
* Một lớp được xác định = thuộc tính + phương thức, có tính kế thừa cao.
* Các đối tượng liên lạc với nhau bằng các thông điệp

Gồm các bước:

* Xác định kiến trúc của hệ thống.
* Sắp thứ tự ưu tiên các gói.
* Với mỗi gói thiết kế ca sử dụng tương ứng.
* Xây dựng biểu đồ tương tác.
* Thiết kế chi tiết các lớp.
* Phân tích và hoàn thiện biểu đồ lớp dựa trên mẫu thiết kế.

### **1.3.4. Mô hình quy trình thác nước**

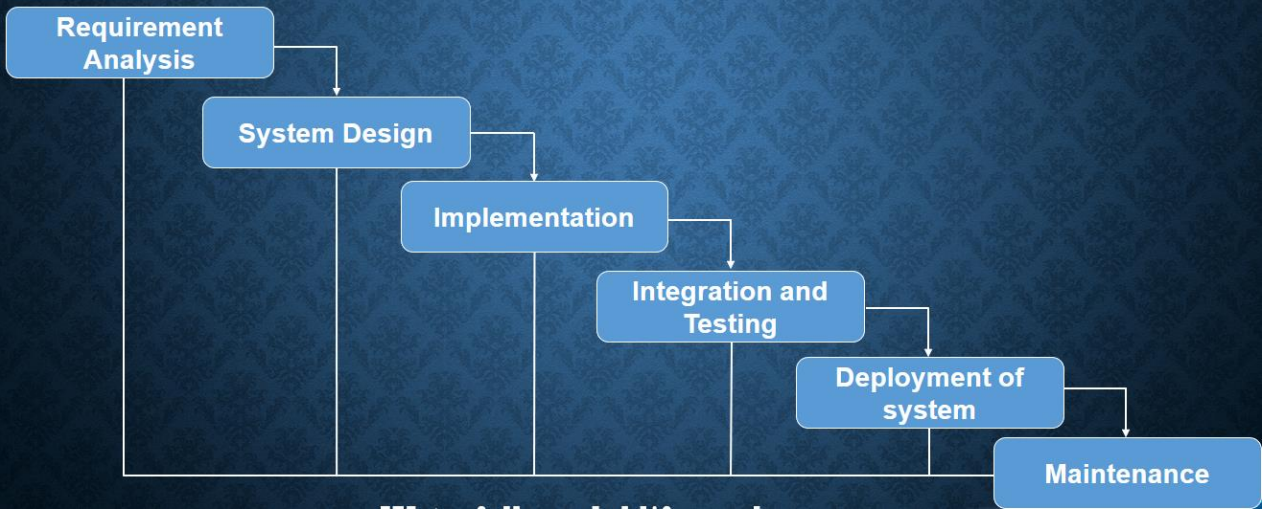
**1.3.4.1. Giới thiệu mô hình**

Mô hình thác nước (Waterfall model) là Mô hình quy trình đầu tiên được giới thiệu. Nó cũng được gọi là mô hình vòng đời tuần tự tuyến tính. Nó rất đơn giản để hiểu và sử dụng. Trong mô hình thác nước, mỗi giai đoạn phải được hoàn thành trước khi giai đoạn tiếp theo có thể bắt đầu và không có sự chồng chéo trong các giai đoạn.

Waterfall là mô hình quy trình đầu tiên và nổi tiếng nhất được giới thiệu, với cách tiếp cận Vòng đời phát triển hệ thống (SDLC) lâu đời nhất, được sử dụng để phát triển phần mềm.

Mô hình thác nước minh họa quá trình phát triển phần mềm theo dòng tuần tự tuyến tính. Điều này có nghĩa là bất kỳ giai đoạn nào trong quá trình phát triển chỉ bắt đầu nếu giai đoạn trước đó hoàn thành. Trong mô hình thác nước, các pha không chồng lên nhau.

**1.3.4.2. Các pha trong mô hình**



Hình 1.3.4. 2: Các pha trong mô hình thác nước

Hình 1.3.4.2. minh họa về các bước trong mô hình Waterfall, bao gồm:

Thu thập và phân tích yêu cầu (Requirement Analysis): Tất cả các yêu cầu có thể có của hệ thống được phát triển đều được ghi lại trong giai đoạn này và được ghi lại trong tài liệu đặc tả yêu cầu để phục vụ cho các giai đoạn sau:

Thiết kế hệ thống (System Design): Các thông số kỹ thuật yêu cầu từ giai đoạn đầu được nghiên cứu trong giai đoạn này và thiết kế hệ thống được chuẩn bị. Thiết kế hệ thống này giúp xác định các yêu cầu phần cứng và hệ thống cũng như giúp xác định kiến trúc hệ thống tổng thể thực hiện.

Thực hiện(Implementation): Với đầu vào từ thiết kế hệ thống, hệ thống được phát triển đầu tiên trong các chương trình nhỏ hoặc code gọi là đơn vị, được tích hợp trong giai đoạn tiếp theo. Mỗi đơn vị được phát triển và kiểm tra chức năng của nó (hay còn có thể được gọi là Kiểm thử đơn vị).

Tích hợp và Kiểm thử (Integration and Testing): Tất cả các đơn vị được phát triển trong giai đoạn triển khai được tích hợp vào một hệ thống sau khi thử nghiệm của mỗi đơn vị. Sau khi tích hợp, toàn bộ hệ thống được kiểm tra xem có bất kỳ lỗi và hỏng hóc nào không (giai đoạn này còn được gọi là kiểm thử tích hợp).

Triển khai hệ thống (Deployment of system): Sau khi kiểm tra chức năng và phi chức năng được thực hiện; sản phẩm được triển khai trong môi trường.

Bảo trì (Maintenance): Có một số vấn đề xảy ra trong môi trường khách hàng.

Để khắc phục những vấn đề đó, các bản vá lỗi được phát hành. Ngoài ra để nâng cao sản phẩm một số phiên bản tốt hơn được phát hành. Bảo trì được thực hiện để mang lại những thay đổi này trong môi trường khách hàng.

**1.3.4.3. Đánh giá**

Mô hình thác nước minh họa quá trình phát triển phần mềm theo dòng tuần tự tuyến tính. Điều này có nghĩa là bất kỳ giai đoạn nào trong quá trình phát triển chỉ bắt đầu nếu giai đoạn trước đó hoàn thành. Tất cả các giai đoạn này được xếp tầng với nhau, trong đó tiến trình được xem như chảy đều đặn xuống dưới (giống như thác nước) qua các giai đoạn. Giai đoạn tiếp theo chỉ được bắt đầu sau khi đạt được tập hợp mục tiêu đã xác định cho giai đoạn trước và nó được ký kết, vì vậy có tên "Mô hình thác nước". Trong mô hình này, các giai đoạn không chồng chéo lên nhau.

- **Ưu điểm:** Cho phép thiết lập một lịch trình với các thời hạn cho từng giai đoạn phát triển và một sản phẩm, và có thể tiến hành từng giai đoạn của mô hình quy trình phát triển. Sự phát triển di chuyển từ ý tưởng, thông qua thiết kế, thực hiện, thử nghiệm, cài đặt, khắc phục sự cố và kết thúc là vận hành và bảo trì. Mỗi giai đoạn phát triển diễn ra theo thứ tự nghiêm ngặt. Một số ưu điểm chính của Mô hình thác nước như sau:

* Đơn giản, dễ hiểu và sử dụng
* Dễ dàng quản lý do độ cứng của mô hình.
* Mỗi giai đoạn có các phân phối cụ thể và một quy trình xem xét.
* Các giai đoạn được xử lý và hoàn thành cùng một lúc.
* Hoạt động tốt cho các dự án nhỏ hơn, nơi các yêu cầu được hiểu rất rõ.
* Các giai đoạn được xác định rõ ràng.
* Các mốc quan trọng được hiểu rõ.
* Dễ dàng sắp xếp các công việc.
* Quá trình và kết quả được ghi chép đầy đủ.

- **Nhược điểm:** Nó không cho phép phản ánh hoặc sửa đổi nhiều. Một khi ứng dụng đang trong giai đoạn thử nghiệm, rất khó để quay lại và thay đổi một cái gì đó không được ghi chép đầy đủ hoặc được nghĩ đến trong giai đoạn khái niệm.

Những nhược điểm chính của Mô hình thác nước như sau:

* Không có phần mềm đang hoạt động nào được sản xuất cho đến cuối vòng đời.
* Lượng rủi ro cao và không chắc chắn.
* Không phải là một mô hình tốt cho các dự án phức tạp và hướng đối tượng.
* Mô hình kém cho các dự án dài và đang diễn ra. Không phù hợp với các dự án mà các yêu cầu có nguy cơ thay đổi từ trung bình đến cao.
* Vì vậy, rủi ro và sự không chắc chắn là cao với mô hình quy trình này.
* Rất khó để đo lường sự tiến bộ trong các giai đoạn.
* Không thể đáp ứng các yêu cầu thay đổi.

- **Ứng dụng:** Mỗi phần mềm được phát triển đều khác nhau và đòi hỏi phải tuân theo một cách tiếp cận SDLC phù hợp dựa trên các yếu tố bên trong và bên ngoài.

* Một số dự án phần mềm thích hợp với mô hình Waterfall như:
* Các yêu cầu được ghi chép rất đầy đủ, rõ ràng và cố định.
* Mô tả (định nghĩa) về sản phẩm ổn định.
* Công nghệ được hiểu và không mang tính năng động.
* Không có yêu cầu mơ hồ.
* Có sẵn các nguồn lực dồi dào với kiến thức chuyên môn cần thiết để hỗ trợ sản phẩm.
* Dự án ngắn hạn.