

| STT | Họ và Tên      | Mã Sinh Viên | Lớp |
|-----|----------------|--------------|-----|
| 1   | Phạm Lê Sơn    | 20211090     | IT4 |
| 2   | Trương Văn Tài | 20211078     | IT4 |
| 3   | Vũ Tiến Quân   | 20210951     | IT4 |

## ĐỀ TÀI: PHÂN TÍCH, ĐẶC TẢ, THIẾT KẾ VÀ PHÁT TRIỂN ỨNG DỤNG QUẢN LÝ CỬA HÀNG QUẦN ÁO

### Các tool hỗ trợ:

#### 1. Pha lập kế hoạch (Planning Phase)



##### 1.1 Công cụ: Trello / Jira

- **Trello** là một công cụ trực quan và dễ sử dụng cho việc quản lý dự án, giúp các thành viên trong nhóm theo dõi các nhiệm vụ và tiến độ công việc một cách dễ dàng. **Jira** là công cụ quản lý dự án mạnh mẽ, thường được dùng trong các quy trình Agile, cho phép quản lý các nhiệm vụ phức tạp, lộ trình phát triển (roadmap), và sprints.

**Trello** phù hợp cho các nhóm muốn một giao diện thân thiện với người dùng, trong khi **Jira** cung cấp nhiều tính năng chi tiết hơn cho các dự án phức tạp.

##### 1.2 Công dụng:

- Tạo một **board** cho dự án quản lý cửa hàng.
- Tạo các **card** cho từng nhiệm vụ như: phân tích yêu cầu khách hàng, thiết kế giao diện, lên danh sách sản phẩm.
- Gán các thành viên trong nhóm vào nhiệm vụ, đặt **deadline**, và sử dụng **checklist** để kiểm tra tiến độ.

- Sử dụng **Jira** cho các dự án lớn, quản lý backlog, tạo các sprint cho từng chu kỳ phát triển, và theo dõi tiến độ qua **burndown chart**.

### 1.3 Cách sử dụng công cụ

#### A. Công cụ: Trello

- Tạo tài khoản Trello: Truy cập trang Trello và đăng ký tài khoản miễn phí hoặc đăng nhập nếu bạn đã có tài khoản.
- Tạo board (bảng):
  1. Nhấp vào "Create new board" trên giao diện chính.
  2. Đặt tên cho dự án (ví dụ: "Dự án quản lý cửa hàng") và chọn màu nền hoặc hình nền.
- Tạo các danh sách (list):
  1. Trong board vừa tạo, thêm các danh sách theo các trạng thái công việc, như "To Do" (Việc cần làm), "In Progress" (Đang làm), và "Done" (Hoàn thành).
- Tạo thẻ (card):
  1. Trong từng danh sách, nhấp vào "Add a card" để thêm nhiệm vụ cụ thể (ví dụ: "Phân tích yêu cầu khách hàng", "Thiết kế giao diện").
  2. Sau khi tạo thẻ, bạn có thể nhấp vào thẻ để thêm chi tiết: mô tả, deadline, gán người thực hiện (Assign members), và tạo checklist.
- Theo dõi tiến độ:
  - Khi một nhiệm vụ chuyển trạng thái, kéo thẻ từ danh sách "To Do" sang "In Progress" hoặc "Done" để cập nhật tiến độ.

#### B. Công cụ: Jira

- Tạo tài khoản Jira: Truy cập Jira và đăng ký tài khoản.
- Tạo dự án mới:
  1. Tạo một dự án mới từ bảng điều khiển (dashboard) bằng cách nhấp vào "Create project".
  2. Chọn loại dự án (ví dụ: Scrum project nếu bạn theo phương pháp Agile/Scrum).
- Tạo backlog:
  1. Trong bảng điều khiển của dự án, bạn có thể thêm các nhiệm vụ vào backlog (kho công việc chưa thực hiện).
  2. Mỗi nhiệm vụ có thể được gán độ ưu tiên, thành viên thực hiện và thời gian ước tính.

- Tạo sprint:

1. Trong bảng "Backlog", bạn có thể chọn các nhiệm vụ và thêm vào một sprint.
2. Xác định thời gian cho sprint (thường từ 1 đến 2 tuần).

- Theo dõi tiến độ:

- Sau khi bắt đầu sprint, bạn có thể theo dõi tiến độ qua biểu đồ burndown (giúp xem số lượng công việc hoàn thành theo thời gian).

## 2. Pha phát triển (Development Phase)



### 2.1 Công cụ: Git / GitHub

- **Giới thiệu:**

**Git** là hệ thống kiểm soát phiên bản phân tán phổ biến, cho phép nhiều nhà phát triển cùng làm việc trên một mã nguồn mà không gây xung đột. Các nền tảng như **GitHub** cung cấp khả năng lưu trữ mã nguồn trên đám mây, giúp quản lý mã, review code, và triển khai (deployment) dễ dàng hơn.

**GitHub** hỗ trợ CI/CD (Continuous Integration/Continuous Deployment), giúp tự động hóa quá trình tích hợp và triển khai ứng dụng, giúp phát triển và đưa sản phẩm ra thị trường nhanh chóng hơn.

### 2.2 Công dụng:

- Cài đặt **Git** và tạo kho lưu trữ (repository) trên **GitHub** để lưu mã nguồn của ứng dụng.
- Tạo các **branch** riêng cho mỗi tính năng hoặc phần sửa lỗi. Ví dụ: bạn có thể tạo branch "add-login-function" để thêm tính năng đăng nhập.
- Sau khi hoàn thành một tính năng, tạo **pull request** để so sánh, kiểm tra mã trước khi hợp nhất (merge) vào nhánh chính (main branch).

- Sử dụng các tích hợp CI/CD của **GitHub** để tự động kiểm tra và triển khai mã khi có thay đổi mới.

## 2.3 Cách sử dụng công cụ

### - Cài đặt Git:

1. Tải Git từ [git-scm.com](https://git-scm.com) và cài đặt trên máy tính.
2. Mở terminal và chạy lệnh `git --version` để kiểm tra cài đặt Git thành công.

### - Tạo kho lưu trữ (repository) trên GitHub:

1. Truy cập [GitHub](https://github.com), đăng ký tài khoản và nhấp vào "New repository".
2. Đặt tên kho, chọn công khai (public) hoặc riêng tư (private), sau đó tạo repository.

### - Kết nối repository với Git trên máy:

1. Trong terminal, chuyển đến thư mục dự án bằng lệnh `cd /path/to/your/project`.
2. Chạy lệnh `git init` để khởi tạo Git.
3. Thêm kho từ xa (remote repository) bằng lệnh:

```
git remote add origin https://github.com/username/repository.git
```

### - Tạo branch và commit thay đổi:

1. Tạo nhánh mới cho tính năng hoặc sửa lỗi bằng lệnh:

```
git remote add origin https://github.com/username/repository.git
```

2. Sau khi thực hiện thay đổi, commit bằng lệnh:

```
git add .  
git commit -m "Thêm tính năng đăng nhập"
```

### - Tạo pull request (PR):

1. Đẩy nhánh lên GitHub bằng lệnh:

```
git push origin feature/add-login
```

2. Trên GitHub, tạo pull request để yêu cầu review và hợp nhất (merge) vào nhánh chính.

### 3. Pha kiểm thử (Testing Phase)



#### 3.1 Công cụ: Selenium / Postman

- **Giới thiệu:**  
**Selenium** là một công cụ tự động hóa kiểm thử giao diện người dùng (UI) phổ biến cho các ứng dụng web. Nó cho phép bạn mô phỏng các hành động của người dùng và đảm bảo rằng mọi chức năng trên giao diện đều hoạt động đúng cách.
- **Postman** là công cụ phổ biến để kiểm thử API, giúp đảm bảo rằng các endpoint của ứng dụng hoạt động đúng theo yêu cầu.  
**JUnit** là framework kiểm thử phổ biến cho Java, thường được sử dụng để kiểm tra logic và các phương thức của back-end.

#### 3.2 Công dụng:

- Với **Selenium**, viết các kịch bản kiểm thử tự động cho giao diện như: kiểm tra luồng đăng ký người dùng, thêm sản phẩm vào giỏ hàng, thanh toán.
- Dùng **Postman** để kiểm thử các API endpoint, thực hiện các request GET, POST, PUT, DELETE, và kiểm tra phản hồi từ server (thường là JSON).

#### 3.3 Cách sử dụng công cụ

##### Công cụ: Selenium

- Cài đặt Selenium:

1. Cài đặt Selenium WebDriver cho ngôn ngữ lập trình của bạn (ví dụ: Java, Python).
2. Tải xuống ChromeDriver hoặc GeckoDriver (nếu dùng Chrome/Firefox) từ trang chính thức.

- Viết kịch bản kiểm thử:

1. Dùng ngôn ngữ lập trình bạn chọn để viết các bài kiểm thử. Ví dụ trong Java:

```
WebDriver driver = new ChromeDriver();  
driver.get("https://yourwebsite.com");  
WebElement loginButton = driver.findElement(By.id("login"));  
loginButton.click();
```

2. Mô phỏng các thao tác như nhấp chuột, nhập liệu, kiểm tra kết quả trên giao diện.

### **Công cụ: Postman**

- Tạo request:

1. Mở Postman và tạo một collection mới.
2. Thêm yêu cầu (request) mới, chọn loại request (GET, POST, PUT, DELETE).
3. Nhập URL của API và thêm các tham số nếu cần (body, headers).

- Kiểm tra phản hồi (response):

1. Sau khi gửi request, Postman sẽ hiển thị phản hồi từ server.
2. Kiểm tra mã trạng thái HTTP (200, 404, 500) và nội dung phản hồi (thường là JSON hoặc XML).