**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC KINH TẾ CÔNG NGHIỆP LONG AN**

**KHOA KỸ THUẬT CÔNG NGHỆ**

**A blue and yellow logo

Description automatically generated**

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI**

**QUẢN LÝ TRANG WEB BÁN QUẦN ÁO, GIÀY**

**GIẢNG VIÊN HƯỚNG DẪN : ThS. NGUYỄN HỮU TÂM**

**SINH VIÊN THỰC HIỆN : LÊ PHAN NHƯ Ý- 2252010006**

**: TRẦN VĂN LỰC- 2252010012**

**: PHẠM THANH KHẢ- 2252010015**

**: HÀ ĐỖ NHẬT HUY- 2252010032**

**: NGUYỄN BÙI HỮU THỊNH- 2252010035**

**LONG AN, 07/2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC KINH TẾ CÔNG NGHIỆP LONG AN**

**KHOA KỸ THUẬT CÔNG NGHỆ**

**A blue and yellow logo

Description automatically generated**

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI**

**QUẢN LÝ TRANG WEB BÁN QUẦN ÁO, GIÀY**

**GIẢNG VIÊN HƯỚNG DẪN : ThS. NGUYỄN HỮU TÂM**

**SINH VIÊN THỰC HIỆN : LÊ PHAN NHƯ Ý- 2252010006**

**: TRẦN VĂN LỰC- 2252010012**

**: PHẠM THANH KHẢ- 2252010015**

**: HÀ ĐỖ NHẬT HUY- 2252010032**

**: NGUYỄN BÙI HỮU THỊNH- 2252010035**

**LONG AN, 07/2024**

**LỜI CẢM ƠN**

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép chúng em được bày tỏ lòng biết ơn đến tất cả các cá nhân và tổ chức đã tạo điều kiện hỗ trợ, giúp đỡ chúng em trong suốt quá trình học tập và nghiên cứu đề tài này. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô và bạn bè.

Với lòng biết ơn sâu sắc nhất, chúng em xin gửi đến quý Thầy Cô ở Khoa Công Nghệ đã truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Nhờ có những lời hướng dẫn, dạy bảo của các thầy cô nên đề tài nghiên cứu của chúng em mới có thể hoàn thiện tốt đẹp.

Một lần nữa, chúng em xin chân thành cảm ơn thầy/cô – người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn chúng em hoàn thành tốt bài báo cáo này trong thời gian qua.

Bài báo cáo thực tập thực hiện trong khoảng thời gian gần 3 tuần. Bước đầu đi vào thực tế của chúng em còn hạn chế và còn nhiều bỡ ngỡ nên không tránh khỏi những thiếu sót, chúng em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy/Cô để kiến thức của chúng em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiện bổ sung, nâng cao ý thức của mình.

Chúng em xin chân thành cảm ơn!

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

Long An, ngày … tháng … năm 2024

**Giảng viên hướng dẫn**

**ThS. Nguyễn Hữu Tâm**

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH, ĐỒ THỊ, BẢNG BIỀU iv](#_Toc173225338)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 1](#_Toc173225339)

[1.1. TÌM HIỂU VỀ CÔNG NGHỆ PHẦN MỀM 1](#_Toc173225340)

[1.1.1. KHÁI NIỆM VỀ CÔNG NGHỆ PHẦN MỀM: 1](#_Toc173225341)

[1.1.2. CÁC GIAI ĐOẠN PHÁT TRIỂN PHẦN MỀM: 1](#_Toc173225342)

[1.1.3. GIỚI THIỆU USE CASE 2](#_Toc173225343)

[1.2. TRELLO 7](#_Toc173225344)

[1.2.1. TRELLO LÀ GÌ? 7](#_Toc173225345)

[1.2.2. CƠ BẢN VỀ TRELLO 7](#_Toc173225346)

[1.3. GITHUB 17](#_Toc173225347)

[1.3.1. GIỚI THIỆU GITHUB 17](#_Toc173225348)

[1.3.2. TÍNH NĂNG CỦA GITHUB 18](#_Toc173225349)

[1.3.3. CÁCH THỨC HOẠT ĐỘNG TRÊN GITHUB LÀ GÌ? 22](#_Toc173225350)

[1.3.4. CÁC YẾU TỐ CẦN THIẾT CỦA GITHUB LÀ GÌ? 25](#_Toc173225351)

[1.3.5. GIỚI THIỆU VỀ GITHUB DESKTOP 29](#_Toc173225352)

[1.4. GITHUB TRONG VISUAL STUDIO 30](#_Toc173225353)

[1.4.1. MỞ GITHUB TRONG VISUAL STUDIO 30](#_Toc173225354)

[1.4.2. THAO TÁC VỚI NHÁNH (BRANCH) 32](#_Toc173225355)

[1.4.3. PULL VÀ PUSH DỮ LIỆU 33](#_Toc173225356)

[1.5. CÁC MỨC ĐỘ TEST LEVER TRONG KIỂM THỬ PHẦN MỀM 35](#_Toc173225357)

[1.6. UNIT TEST 35](#_Toc173225358)

[1.6.1. KHÁI NIỆM VỀ UNIT TEST 35](#_Toc173225359)

[1.6.2. ỨNG DỤNG CỦA UNIT TEST 36](#_Toc173225360)

[1.6.3. LỢI ÍCH CỦA UNIT TEST 37](#_Toc173225361)

[1.6.4. VÍ DỤ VỀ ỨNG DỤNG CỦA UNIT TEST 38](#_Toc173225362)

[1.7. INTEGRATION TEST: 42](#_Toc173225363)

[1.7.1. KHÁI NIỆM 42](#_Toc173225364)

[1.7.2. TẠI SAO INTEGRATION TESTING LÀ CẦN THIẾT 42](#_Toc173225365)

[1.7.3. VÍ DỤ VỀ ỨNG DỤNG SELENIMU IDE: 42](#_Toc173225366)

[1.8. SYSTEM TEST 45](#_Toc173225367)

[1.8.1. ĐẶC ĐIỂM CỦA SYSTEM TEST: 46](#_Toc173225368)

[1.8.2. TEST PERFORMANCE BẰNG JMETER 46](#_Toc173225369)

[1.8.3. TEST CASE 53](#_Toc173225370)

[CHƯƠNG 2: GIỚI THIỆU ĐỀ TÀI BÁN QUẨN ÁO, GIÀY 54](#_Toc173225371)

[2.1. GIỚI THIỆU ĐỀ TÀI 54](#_Toc173225372)

[2.1.1. Ý TƯỞNG ĐỀ TÀI 54](#_Toc173225373)

[2.1.2. NỘI DUNG ĐỀ TÀI 54](#_Toc173225374)

[2.2. YÊU CẦU KHÁCH HÀNG 55](#_Toc173225375)

[2.3. ĐẶC TẢ YÊU CẦU HỆ THỐNG 55](#_Toc173225376)

[2.3.1. MÔ TẢ PHẦN MỀM 55](#_Toc173225377)

[2.4. NHÓM CÔNG VIỆC TIN HỌC 55](#_Toc173225378)

[CHƯƠNG 3: XÂY DỰNG PHẦN MỀM QUẢN LÝ KINH DOANH QUẦN ÁO, GIÀY 56](#_Toc173225379)

[3.1. THIẾT KẾ CƠ SỞ DỮ LIỆU 56](#_Toc173225380)

[3.1.1. SƠ ĐỒ ERD 56](#_Toc173225381)

[3.1.2. DATATABLES DIAGRAM 57](#_Toc173225382)

[3.1.3. BẢNG TB\_CATEGORY 57](#_Toc173225383)

[3.1.4. BẢNG TB\_NEWS 58](#_Toc173225384)

[3.1.5. BẢNG TB\_PRODUCTCATEGORY 58](#_Toc173225385)

[3.1.6. BẢNG TB\_PRODUCTS 59](#_Toc173225386)

[3.1.7. BẢNG TB\_ORDER 59](#_Toc173225387)

[3.1.8. BẢNG TB\_ORDERDETAILS 60](#_Toc173225388)

[3.1.9. BẢNG TB\_PRODUCTIMAGE 60](#_Toc173225389)

[3.2. CÁC SƠ ĐỒ THIẾT KẾ KHÁC 61](#_Toc173225390)

[3.2.1. USE CASE DIAGRAM 61](#_Toc173225391)

[3.2.2. CLASS DIAGRAM 62](#_Toc173225392)

[3.2.3. SEQUENCE DIAGRAM 62](#_Toc173225393)

[3.2.4. ACTIVITY DIAGRAM 63](#_Toc173225394)

[3.3. XÂY DỰNG WEBSITE 64](#_Toc173225395)

[3.3.1. GIAO DIỆN NGƯỜI DÙNG 64](#_Toc173225396)

[3.3.2. GIAO DIỆN DASHBOARD 67](#_Toc173225397)

[3.3.3. CODE 68](#_Toc173225398)

[CHƯƠNG 4: KẾT LUẬN 79](#_Toc173225399)

[TÀI LIỆU THAM KHẢO iv](#_Toc173225400)

[BẢNG PHÂN CÔNG CÔNG VIỆC v](#_Toc173225401)

DANH MỤC HÌNH ẢNH, ĐỒ THỊ, BẢNG BIỀU

[Hình 1.1- Giai đoạn Review 5](#_Toc173225266)

[Hình 1.2- Giới thiệu Trello 7](#_Toc173225267)

[Hình 1.3- Task 8](#_Toc173225268)

[Hình 1.4- Bảng 9](#_Toc173225269)

[Hình 1.5- Danh sách 9](#_Toc173225270)

[Hình 1.6- Thẻ 10](#_Toc173225271)

[Hình 1.7- Tải Trello 10](#_Toc173225272)

[Hình 1.8- Đăng ký tài khoản Trello 11](#_Toc173225273)

[Hình 1.9- Tạo bảng trong Trello 12](#_Toc173225274)

[Hình 1.10- Tạo bảng trong Trello 12](#_Toc173225275)

[Hình 1.11- Tạo bảng trong Trello 13](#_Toc173225276)

[Hình 1.12- Tạo nhóm trong Trello 13](#_Toc173225277)

[Hình 1.13- Tạo danh sách trong Trello 14](#_Toc173225278)

[Hình 1.14- Tạo danh sách trong Trello 15](#_Toc173225279)

[Hình 1.15- Tạo thẻ trong Trello 15](#_Toc173225280)

[Hình 1.16- Tạo thẻ trong Trello 16](#_Toc173225281)

[Hình 1.17- Tạo thẻ trong Trello 16](#_Toc173225282)

[Hình 1.18- Giới thiệu về GitHub 18](#_Toc173225283)

[Hình 1.19- Giới thiệu về GitHub 18](#_Toc173225284)

[Hình 1.20- GitHub-Dễ dàng quản lý mã nguồn 19](#_Toc173225285)

[Hình 1.21- Markdown 20](#_Toc173225286)

[Hình 1.22- GitHub Action 21](#_Toc173225287)

[Hình 1.23- GitHub Package Registry 22](#_Toc173225288)

[Hình 1.24- Cách thức hoạt động của GitHub 23](#_Toc173225289)

[Hình 1.25- Cách thức hoạt động của GitHub 24](#_Toc173225290)

[Hình 1.26- GitHub Repository 25](#_Toc173225291)

[Hình 1.27- GitHub Branch 26](#_Toc173225292)

[Hình 1.28- GitHub commits 27](#_Toc173225293)

[Hình 1.29- GitHub Pull Request 28](#_Toc173225294)

[Hình 1.30- GitHub trong Visual Studio 30](#_Toc173225295)

[Hình 1.31- GitHub trong Visual Studio 31](#_Toc173225296)

[Hình 1.32- GitHub trong Visual Studio 31](#_Toc173225297)

[Hình 1.33- Thao tác với nhánh 32](#_Toc173225298)

[Hình 1.34- Thao tác với nhánh 32](#_Toc173225299)

[Hình 1.35- Thao tác với nhánh 33](#_Toc173225300)

[Hình 1.36- Thao tác với nhánh 33](#_Toc173225301)

[Hình 1.37- Pull dữ liệu 34](#_Toc173225302)

[Hình 1.38- Push dữ liệu 34](#_Toc173225303)

[Hình 1.39- Số lần commits 35](#_Toc173225304)

[Hình 1.40- Các mức độ test lever trong kiểm thử phần mềm 35](#_Toc173225305)

[Hình 1.41- Giới thiệu về Unit Test 36](#_Toc173225306)

[Hình 1.42- Tạo Unit test 38](#_Toc173225307)

[Hình 1.43- Run Unit test 41](#_Toc173225308)

[Hình 1.44- Xem kết quả Unit test 41](#_Toc173225309)

[Hình 1.45- Tạo Selenium IDE 43](#_Toc173225310)

[Hình 1.46- Xem kết quả 45](#_Toc173225311)

[Hình 1.47- Giới thiệu Test Performance bằng JMeter 46](#_Toc173225312)

[Hình 1.48- Khời chạy JMeter 47](#_Toc173225313)

[Hình 1.49- Tạo thread group 48](#_Toc173225314)

[Hình 1.50- Các thành phần của Thread Group 49](#_Toc173225315)

[Hình 1.51- Test case 53](#_Toc173225316)

[Hình 3.1- Sơ đồ ERD 56](#_Toc173225317)

[Hình 3.2- Bảng TB\_CATEGORY 57](#_Toc173225318)

[Hình 3.3- Bảng TB\_NEWS 58](#_Toc173225319)

[Hình 3.4- Bảng TB\_PRODUCTCATEGORY 58](#_Toc173225320)

[Hình 3.5- Bảng TB\_PRODUCTS 59](#_Toc173225321)

[Hình 3.6- Bảng TB\_ORDER 59](#_Toc173225322)

[Hình 3.7- Bảng TB\_ORDERDETAILS 60](#_Toc173225323)

[Hình 3.8- Bảng TB\_PRODUCTIMAGE 60](#_Toc173225324)

[Hình 3.9- Use case diagram 61](#_Toc173225325)

[Hình 3.10- Class Diagram 62](#_Toc173225326)

[Hình 3.11- Sequence Diagram 62](#_Toc173225327)

[Hình 3.12- Activity Diagram 63](#_Toc173225328)

[Hình 3.13- Trang chủ 64](#_Toc173225329)

[Hình 3.14- Trang sản phẩm 64](#_Toc173225330)

[Hình 3.15- Trang chi tiết sản phẩm 65](#_Toc173225331)

[Hình 3.16- Trang giỏ hàng 65](#_Toc173225332)

[Hình 3.17- Thực hiện thanh toán 66](#_Toc173225333)

[Hình 3.18- Đặt hàng thành công 66](#_Toc173225334)

[Hình 3.19- Giao diện quản lý danh sách sản phẩm 67](#_Toc173225335)

[Hình 3.20- Giao diện quản lý đơn hàng 67](#_Toc173225336)

[Hình 3.21- Giao diện quản lý tin tức 68](#_Toc173225337)

# CƠ SỞ LÝ THUYẾT

## TÌM HIỂU VỀ CÔNG NGHỆ PHẦN MỀM

Công nghệ phần mềm là lĩnh vực nghiên cứu, phát triển và ứng dụng các phần mềm máy tính. Đây là một lĩnh vực rất quan trọng và phát triển nhanh chóng trong thế giới công nghệ hiện đại. Dưới đây là một số điểm cơ bản về công nghệ phần mềm:

### KHÁI NIỆM VỀ CÔNG NGHỆ PHẦN MỀM:

Công nghệ phần mềm bao gồm các phương pháp, công cụ và quy trình để phát triển phần mềm từ các yêu cầu ban đầu đến sản phẩm hoàn chỉnh.

### **CÁC GIAI ĐOẠN PHÁT TRIỂN PHẦN MỀM**:

Bao gồm thu thập yêu cầu, thiết kế, lập trình, kiểm thử, triển khai và bảo trì.

* **Giai đoạn 1: Lập kế hoạch và phân tích (Analysis)**

Trước khi bạn bắt đầu xây dựng phần mềm bước đầu tiên là nên lập kế hoạch và phân tích các yêu cầu của người dùng hoặc các bên liên quan đối với sản phẩm phần mềm đang xây dựng. Nghiên cứu thị trường nên được thực hiện để xác định chức năng mà phần mềm cung cấp cho người dùng và để đảm bảo rằng người dùng thấy phần mềm hữu ích.

* **Giai đoạn 2: Thiết kế phần mềm (Design)**

Từ các yêu cầu và thông số kỹ thuật được cung cấp trong bước 1 của quy trình phát triển phần mềm, nhà phát triển phần mềm thiết kế kiến ​​trúc tổng thể cần thiết để xây dựng phần mềm. Ngoài ra, các yếu tố như ngân sách, thời gian, công nghệ ứng dụng và mức độ rủi ro cũng được xác định rõ ràng.

* **Giai đoạn 3: Thực hiện (Development)**

Trong bước này, nhà phát triển phần mềm viết mã và bắt đầu triển khai các tham số thiết kế được chỉ định trong bước 2. Các nhà phát triển front-end tạo giao diện phần mềm. Các nhà phát triển viết mã trên máy chủ sử dụng các ngôn ngữ và khuôn khổ lập trình để làm việc với dữ liệu cùng với các quản trị viên trên cơ sở dữ liệu.

* **Giai đoạn 4: Kiểm thử phần mềm (Testing)**

Sau khi hoàn thành phần lập trình phần mềm, sản phẩm vẫn được bàn giao cho các người kiểm thử phần mềm. Các trường hợp thực hiện kiểm thử phần mềm. Mục đích của kiểm thử phần mềm là để kiểm tra chất lượng của sản phẩm và đảm bảo rằng nó đúng như mong đợi. Sau khi kiểm thử, người kiểm tra cập nhật lỗi trong công cụ quản trị và báo lỗi cho nhà phát triển. Trong bước này, người kiểm tra ngồi lại với các nhà phát triển để sửa các lỗi hiện có và cập nhật hệ thống quản lý lỗi.

* **Giai đoạn 5: Giai đoạn triển khai (Deployment stage)**

Sau khi kiểm tra xong, phần mềm không có lỗi và nhà phát triển triển khai sản phẩm vào môi trường sản xuất (môi trường chứa các ứng dụng thực chạy trên người dùng thực và dữ liệu thực) và giao thành phẩm cho khách hàng. Bước này cũng yêu cầu chủ đầu tư phải lập các kế hoạch dự phòng để chủ động đối phó với các sự cố không mong muốn.

* **Giai đoạn 6: Duy trì (Maintenance)**

Sau khi phần mềm được phát hành chính thức và khách hàng bắt đầu sử dụng ở mức chất lượng cao nhất, bước tiếp theo của quy trình phát triển phần mềm là bảo trì sản phẩm. Công ty sẽ thành lập một nhóm chuyên bảo trì và quản lý các vấn đề người dùng gặp phải khi sử dụng sản phẩm. Quản lý và giải quyết các vấn đề cho tất cả người dùng. Đồng thời, phần mềm cũng được cập nhật sau khi triển khai để sửa lỗi và cải thiện hiệu suất.

### GIỚI THIỆU USE CASE

**Use Case** là một thuật ngữ được sử dụng trong lĩnh vực phát triển phần mềm và quản lý dự án để mô tả một tình huống hoặc kịch bản cụ thể về cách hệ thống sẽ được sử dụng. Nó định nghĩa các chức năng, hoạt động và tương tác giữa hệ thống và người dùng hoặc các thành phần khác trong một tình huống cụ thể.

Một Use Case được sử dụng để hiểu rõ các yêu cầu và đặc điểm của một hệ thống, xác định các hành vi và kịch bản sử dụng của người dùng, và xác định các yếu tố quan trọng trong việc phát triển phần mềm. Nó cung cấp một cách tiếp cận hợp lý để phân tích và mô tả các tác động của hệ thống.

Use Case thường được biểu diễn dưới dạng các biểu đồ, ví dụ như biểu đồ Use Case UML (Unified Modeling Language). Mỗi Use Case mô tả một tình huống cụ thể, bao gồm các hành vi, tương tác và kết quả mong đợi. Các Use Case này sau đó có thể được sử dụng như cơ sở để thiết kế, phát triển và kiểm thử hệ thống.

#### CẤU TRÚC CỦA USE CASE

Cấu trúc cơ bản của một Use Case bao gồm các thành phần sau:

- Tên Use Case: Đây là tên gọi ngắn gọn và mô tả tình huống cụ thể mà Use Case đang mô tả. Ví dụ: "Xem danh sách sản phẩm", "Đặt hàng", "Quản lý tài khoản người dùng",...

- Mô tả: Đây là phần mô tả chi tiết về mục đích và chức năng của Use Case. Nó diễn tả các hoạt động và tương tác giữa hệ thống và người dùng hoặc các thành phần khác trong tình huống được mô phỏng. Mô tả này nên làm rõ các bước, luồng làm việc và kết quả mong đợi.

- Tác nhân (Actor): Đây là thành phần bên ngoài tương tác với hệ thống trong Use Case. Tác nhân có thể là người dùng cuối, một hệ thống khác, hoặc bất kỳ thực thể nào có tương tác với hệ thống. Mỗi Use Case có ít nhất một tác nhân liên quan đến nó.

- Tiền điều kiện (Precondition): Đây là trạng thái hoặc điều kiện mà hệ thống phải đáp ứng trước khi Use Case được thực hiện. Điều kiện này thường liên quan đến trạng thái ban đầu của hệ thống hoặc các hoạt động trước đó.

- Các bước (Steps): Đây là phần mô tả chi tiết về các bước cần thiết để thực hiện Use Case. Các bước này phải được mô tả theo thứ tự logic và dễ hiểu.

- Kết quả mong đợi (Expected Outcome): Đây là mô tả về kết quả mong đợi sau khi Use Case được thực hiện thành công. Kết quả này có thể là trạng thái mới của hệ thống, thông báo cho người dùng, hoặc bất kỳ tác động nào khác từ Use Case.

- Hậu điều kiện (Postcondition): Đây là trạng thái hoặc điều kiện mà hệ thống phải đạt được sau khi Use Case được thực hiện thành công. Điều kiện này thường liên quan đến trạng thái cuối cùng của hệ thống hoặc các hoạt động kế tiếp.

Cấu trúc Use Case có thể được biểu diễn bằng nhiều cách, bao gồm biểu đồ Use Case UML, văn bản mô tả hoặc các hình thức biểu diễn khác tùy thuộc vào quy ước và phong cách của dự án hoặc tổ chức cụ thể.

#### CÁCH PHÂN LOẠI USE CASE

Use Case có thể được phân loại theo các tiêu chí khác nhau, tùy thuộc vào ngữ cảnh và mục đích sử dụng của dự án. Dưới đây là một số cách phân loại phổ biến:

**Primary Use Case (Use Case chính):** Đây là các Use Case quan trọng nhất và mang tính chất cốt lõi của hệ thống. Primary Use Case thường mô tả các chức năng và hoạt động chính của hệ thống và đáp ứng trực tiếp nhu cầu của người dùng.

**Supporting Use Case (Use Case hỗ trợ):** Đây là các Use Case hỗ trợ hoặc bổ sung cho Primary Use Case. Chúng cung cấp các chức năng phụ hoặc tương tác hỗ trợ cho các Use Case chính. Supporting Use Case có thể giúp tăng cường hiệu quả hoặc hoàn thiện các chức năng của hệ thống chính.

**Extension Use Case (Use Case mở rộng):** Đây là các Use Case mở rộng hoặc mở rộng các chức năng của Use Case chính. Extension Use Case xảy ra khi một Use Case chính được mở rộng bằng cách thêm các chức năng bổ sung hoặc các luồng làm việc tùy chọn.

**Alternative Use Case (Use Case thay thế):** Đây là các Use Case thay thế cho các chức năng hoặc hoạt động của Use Case chính trong một tình huống cụ thể. Alternative Use Case xảy ra khi có nhiều phương pháp hoặc lựa chọn khác nhau để đạt được mục tiêu tương tự.

**System Use Case (Use Case hệ thống):** Đây là các Use Case tương tác giữa hệ thống và các thành phần hệ thống khác nhau, bao gồm các hệ thống bên ngoài, cơ sở dữ liệu, dịch vụ web, vv. System Use Case mô tả các tương tác giữa hệ thống và các thành phần khác.

**User Use Case (Use Case người dùng):** Đây là các Use Case tương tác giữa người dùng và hệ thống. User Use Case mô tả các hoạt động và tương tác của người dùng với hệ thống để đạt được mục tiêu cụ thể.

Cách phân loại Use Case có thể linh hoạt và có thể điều chỉnh tùy thuộc vào yêu cầu và ngữ cảnh dự án cụ thể. Một số dự án có thể sử dụng các phân loại khác nhau hoặc kết hợp các loại Use Case để phù hợp với mục đích và tổ chức của họ.

#### CÁC BƯỚC XÂY DỰNG BẢN VẼ USE CASE

**GIAI ĐOẠN MÔ HÌNH HÓA:**

* Bước 1: Thực hiện thiết lập ngữ cảnh của hệ thống.

Đầu tiên, xác định các Use Case chính trong hệ thống. Đây là những Use Case quan trọng nhất và mang tính cốt lõi của hệ thống. Xác định các chức năng và hoạt động chính mà hệ thống cung cấp và mô phỏng.

* Bước 2: Xác định các Actor.
* Bước 3: Xác định các Use Case.
* Bước 4: Định nghĩa các mối quan hệ giữa Actor và Use Case.
* Bước 5: Đánh giá các mối quan hệ đó để tìm cách chi tiết hóa.

**GIAI ĐOẠN CẤU TRÚC:**

* Bước 6: Đánh giá các Use Case cho quan hệ Include.
* Bước 7: Đánh giá các Use Case cho quan hệ Extend.
* Bước 8: Đánh giá các Use Case cho quan hệ Generalization .

**GIAI ĐOẠN REVIEW:**

* Bước 9: Kiểm tra (verification): Đảm bảo hệ thống đúng với tài liệu đặc tả.
* Bước 10: Thẩm định (validation): Đảm bảo hệ thống sẽ được phát triển là thứ mà khách hàng cuối thực sự cần thiết.

A diagram of a website

Description automatically generated

Hình 1.1- Giai đoạn Review

Để xây dựng bản vẽ Use Case, bạn có thể tuân theo các bước sau đây:

1. **Xác định các Use Case chính:** Đầu tiên, xác định các Use Case chính trong hệ thống. Đây là những Use Case quan trọng nhất và mang tính cốt lõi của hệ thống. Xác định các chức năng và hoạt động chính mà hệ thống cung cấp và mô phỏng.
2. **Xác định tác nhân (Actors):** Xác định các tác nhân liên quan đến mỗi Use Case. Tác nhân có thể là người dùng cuối, hệ thống bên ngoài, hoặc bất kỳ thực thể nào khác tương tác với hệ thống. Ghi lại các tác nhân này để hiểu rõ người hoặc thành phần nào liên quan đến mỗi Use Case.
3. **Vẽ biểu đồ Use Case:** Sử dụng biểu đồ Use Case UML để biểu diễn các Use Case và mối quan hệ giữa chúng. Vẽ một hình chữ nhật cho mỗi Use Case, ghi tên Use Case bên trong hình chữ nhật đó. Sử dụng các mũi tên để chỉ ra mối quan hệ giữa các Use Case và tác nhân. Ví dụ, mũi tên từ tác nhân đến Use Case để chỉ tác nhân tham gia vào Use Case đó.
4. **Kết nối các Use Case:** Sử dụng mũi tên để kết nối các Use Case với nhau. Điều này thể hiện mối quan hệ giữa các Use Case và sự tương tác giữa chúng. Ví dụ, một Use Case có thể kế thừa từ một Use Case khác, hoặc một Use Case có thể mở rộng một Use Case khác.
5. **Ghi chú và miêu tả:** Bổ sung ghi chú và miêu tả cho mỗi Use Case để cung cấp thông tin bổ sung và giải thích chi tiết. Ghi chú có thể cung cấp thông tin về các ràng buộc, điều kiện tiền điều kiện hoặc bất kỳ thông tin nào quan trọng khác liên quan đến Use Case.
6. **Kiểm tra và điều chỉnh:** Kiểm tra và rà soát biểu đồ Use Case để đảm bảo tính logic, đầy đủ và chính xác. Đảm bảo rằng biểu đồ Use Case phản ánh đúng yêu cầu và chức năng của hệ thống. Rà soát và điều chỉnh biểu đồ để đảm bảo nó.

## TRELLO

### TRELLO LÀ GÌ?

Trello là một công cụ quản lý dự án, hỗ trợ các nhóm lên ý tưởng, lập kế hoạch để cùng nhau hợp tác và hoàn thành công việc hiệu quả nhất. Ứng dụng có nguyên lý hoạt động dựa trên phương pháp Kanban.

Trello có thiết kế tương tự như một bảng với các ghi chú của những dự án và nhiệm vụ được phân chia bằng cách sắp xếp thành cột và di chuyển dễ dàng để chỉ ra quy trình làm việc, quyền sở hữu dự án và tình trạng hoàn thiện.



Hình 1.2- Giới thiệu Trello

Sự nổi tiếng của Trello phần lớn đến từ sự đơn giản. Trong khi các công cụ quản lý dự án truyền thống được thiết kế cho các nhà quản lý dự án chuyên nghiệp đôi khi có phần phức tạp, thì Trello nhấn mạnh tính dễ sử dụng. Giúp hầu hết mọi người có thể nắm được những kiến thức cơ bản về Trello trong vài phút và lập tức sử dụng phục vụ công tác làm việc và học tập.

### CƠ BẢN VỀ TRELLO

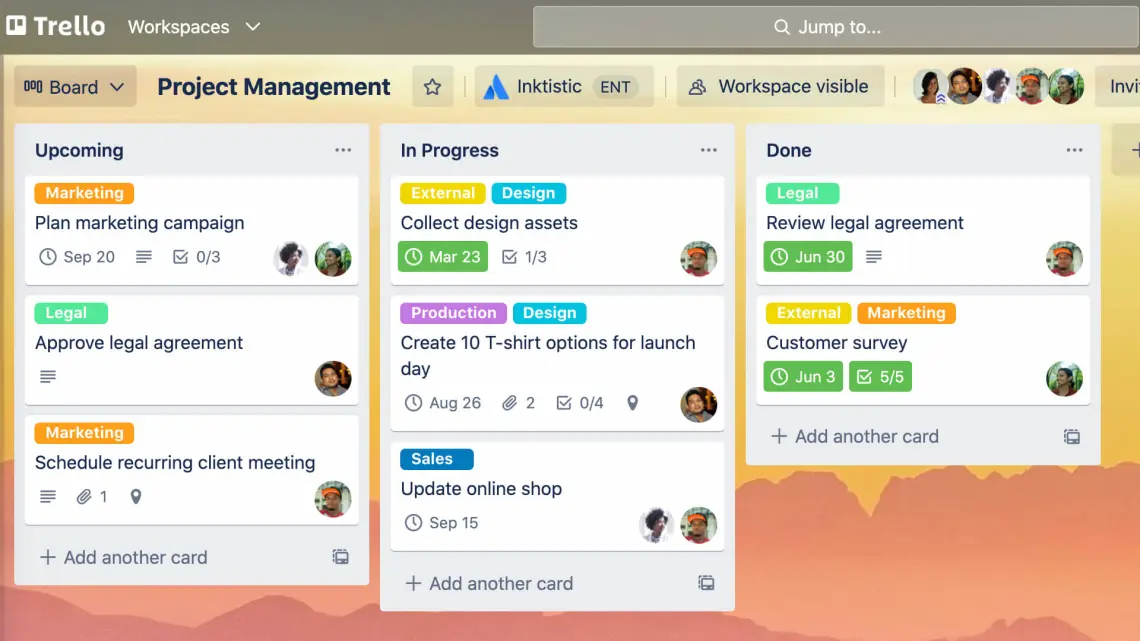
#### CÁC THÀNH PHẦN TRONG TRELLO

Mặc dù là ứng dụng miễn phí nhưng Trello rất hữu dụng bởi sự đa dụng và có những tính năng thiết thực như Task, Bảng, Danh sách và thẻ. Dưới đây, chúng ta sẽ tìm hiểu chi tiết từng thành phần này.

**TASK**

Có thể chứa nhiều thông tin bao gồm hình ảnh và tệp đính kèm, ngoài ra chúng có thể tùy chỉnh được thời hạn và dữ liệu theo dõi trạng thái khác. Task cũng có thể chứa các ghi chú, cho phép các thành viên trong nhóm theo dõi tiến trình và giao việc cụ thể bằng cách “gắn cờ” để tiện phân biệt.

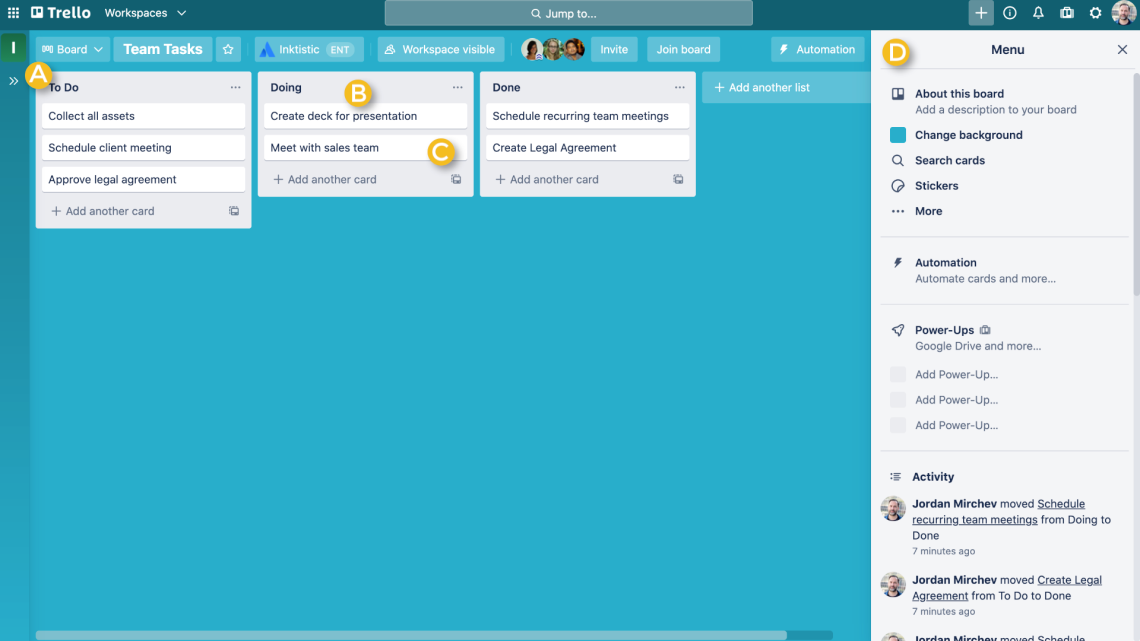
Trello cũng hỗ trợ tích hợp với hàng trăm ứng dụng và dịch vụ của bên thứ ba, chẳng hạn như Google Drive, OneDrive, GitHub, Slack, Jira,…



Hình 1.3- Task

**BẢNG**

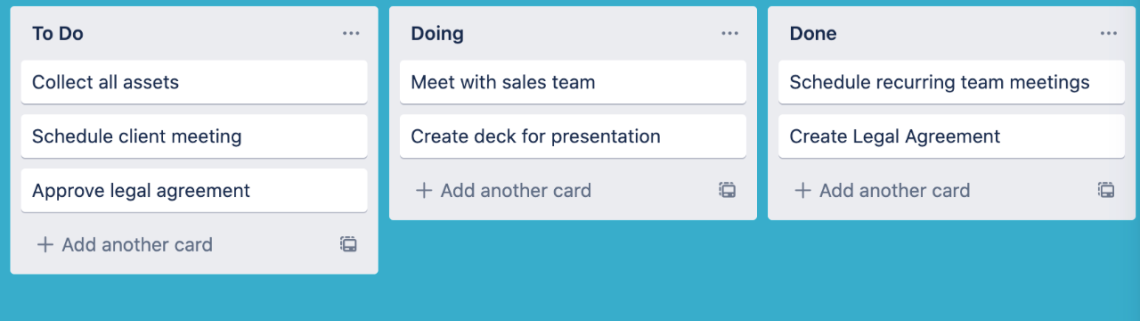
Không gian làm việc tổng thể là bảng và nó có thể chứa bất kỳ số lượng danh sách và thẻ nào.



Hình 1.4- Bảng

**DANH SÁCH**

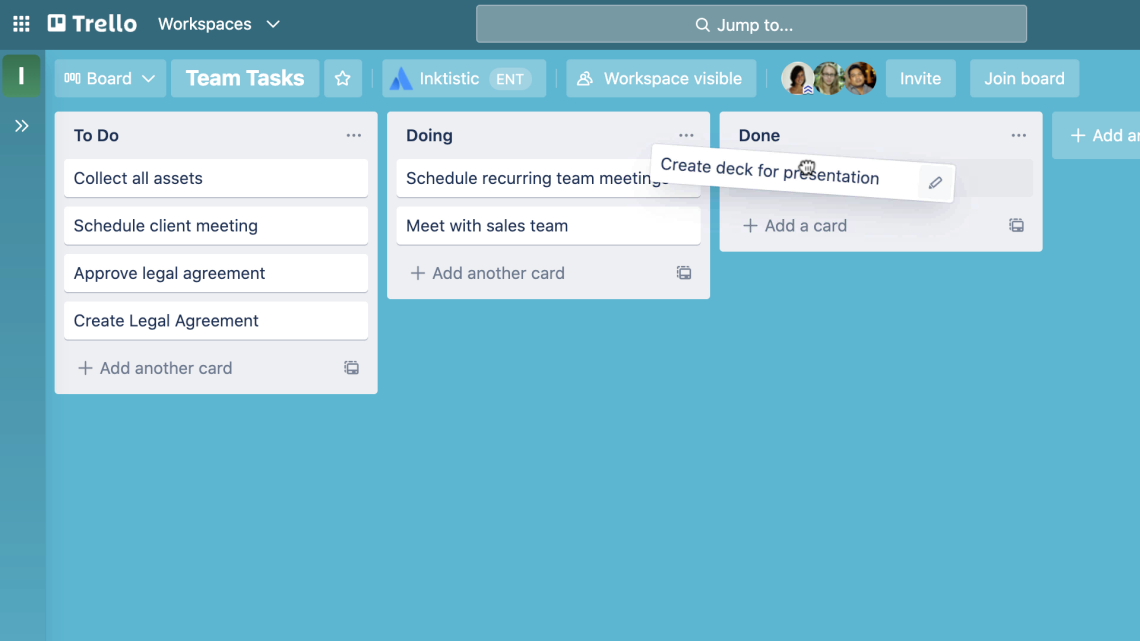
Danh sách trong Trello là một cột chứa các thẻ. Cách bạn sử dụng và tổ chức danh sách hoàn toàn tùy thuộc vào bạn. Ví dụ: mỗi danh sách có thể là một người trong nhóm và hội đồng quản trị theo dõi nhiệm vụ của mọi người. Hoặc bạn có thể tổ chức danh sách dưới dạng quy trình làm việc, trong đó mỗi thẻ được chuyển từ danh sách này sang danh sách khác khi các tác vụ được xử lý và hoàn thành.



Hình 1.5- Danh sách

**THẺ**

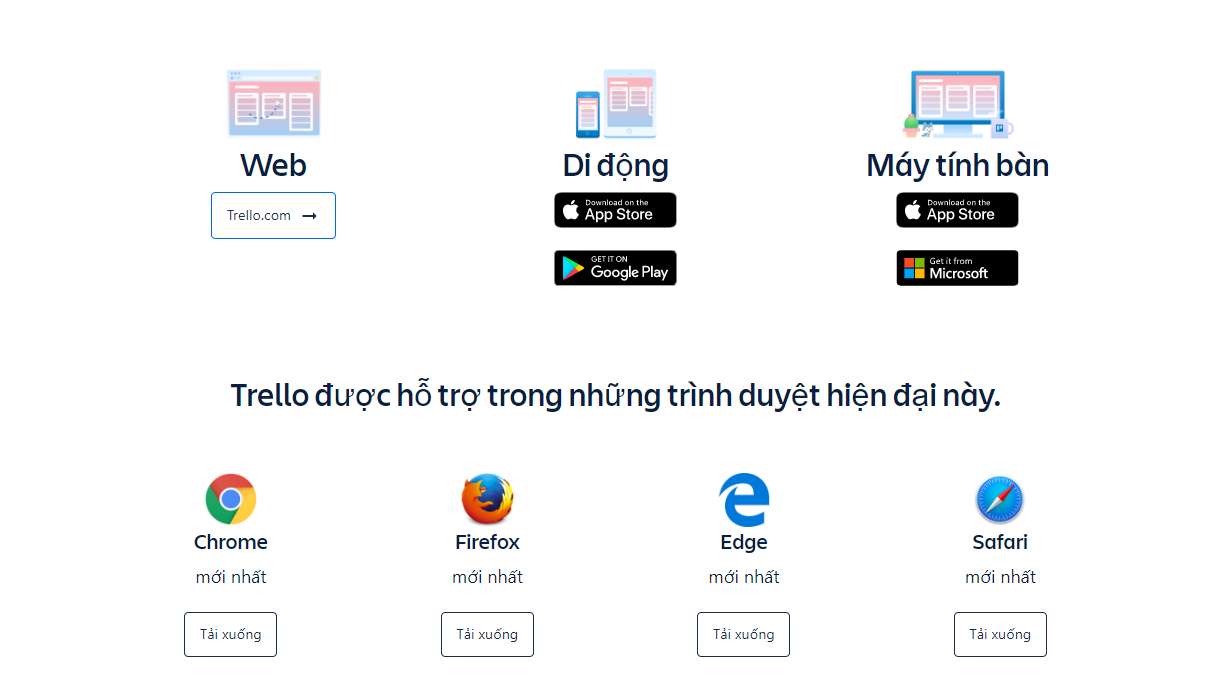
Thẻ là đơn vị nhỏ nhất trong Trello và thường được sử dụng để xác định các nhiệm vụ cần được thực hiện. Thẻ có thể chứa tệp đính kèm, hình ảnh, ghi chú,…Thẻ có thể được mở và chỉnh sửa bằng một cú nhấp chuột và việc di chuyển thẻ giữa các danh sách dễ dàng như kéo và thả.



Hình 1.6- Thẻ

**CÁCH TẢI TRELLO NHANH CHÓNG**

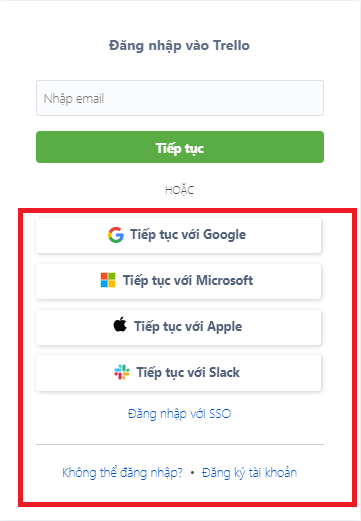
Để tải Trello, bạn cần vào trang web Trello [tại đây](http://trello.com/vi/platforms) và chọn nền tảng phù hợp. Sau đó tải về và cài đặt.



Hình 1.7- Tải Trello

**CÁCH ĐĂNG KÝ TÀI KHOẢN TRELLO**

Hiện tại, Trello có hỗ trợ đăng ký bằng các tài khoản từ Google, Microsoft, Apple hoặc Slack. Bạn có thể chọn liên kết với các tài khoản này để tạo tài khoản nhanh chóng. Hoặc bạn có thể chọn **Đăng ký tài khoản**và nhập bất kì mail nào để hoàn thành đăng ký.



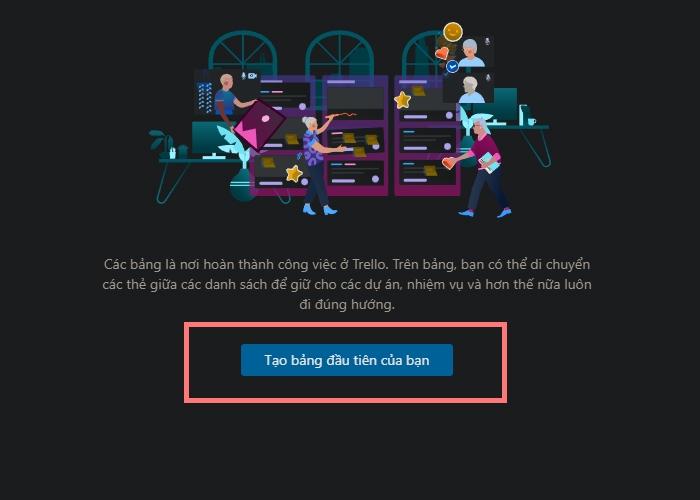
Hình 1.8- Đăng ký tài khoản Trello

#### LÀM VIỆC VỚI TRELLO

**1. TẠO BẢNG TRONG TRELLO**

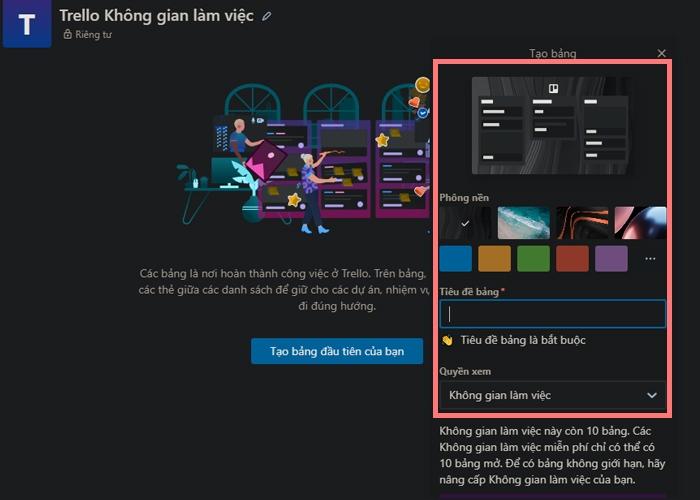
Để tạo một bảng, hãy làm theo các bước sau:

**Bước 1:** Bật máy tính mở ứng dụng Trello. Trong Bảng cá nhân, hãy nhấp vào hộp có nội dung **Tạo bảng đầu tiên của bạn**…



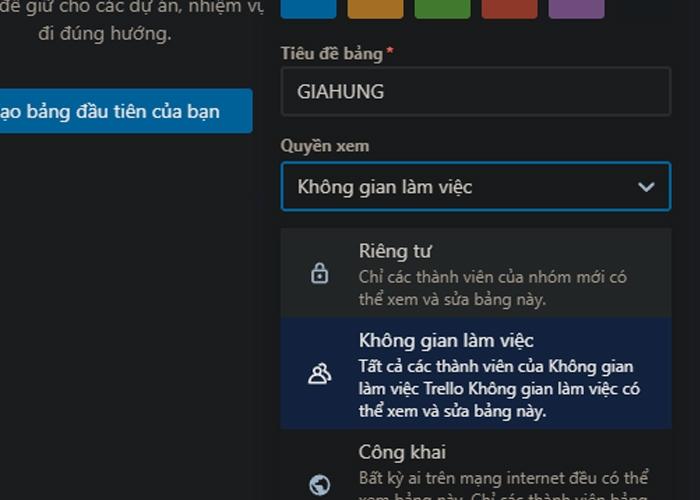
Hình 1.9- Tạo bảng trong Trello

**Bước 2:** Đặt tiêu đề cho bảng. Bạn cũng có thể chọn màu nền hoặc mẫu theo sở thích cá nhân.



Hình 1.10- Tạo bảng trong Trello

**Bước 3:** Nếu bạn có nhiều nhóm, hãy chọn nhóm mà bạn muốn cấp quyền truy cập vào bảng. Hoặc chưa có nhóm, bạn có thể đến phần tiếp theo để tạo nhóm nhé.



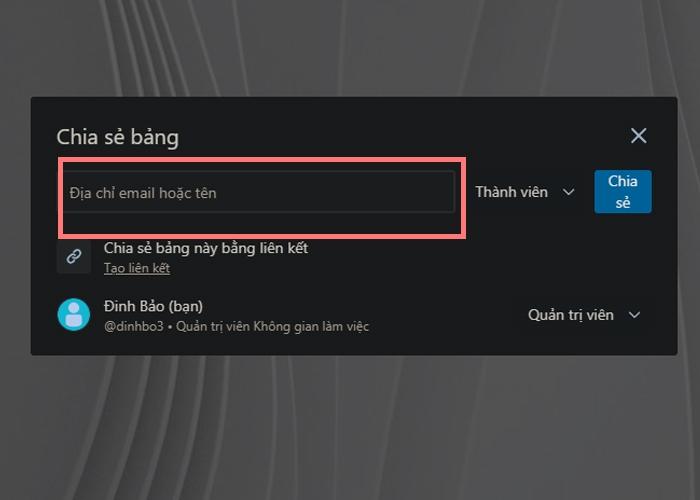
Hình 1.11- Tạo bảng trong Trello

**2. TẠO NHÓM TRONG TRELLO**

Khi bạn chưa thiết lập nhóm thì ngay bây giờ có thể thêm từng thành viên vào hội đồng quản trị của mình theo các bước sau:

**Bước 1:** Mở bảng từ trang chủ Trello.

**Bước 2:** Tìm người dùng bằng cách nhập địa chỉ email hoặc tên người dùng Trello.



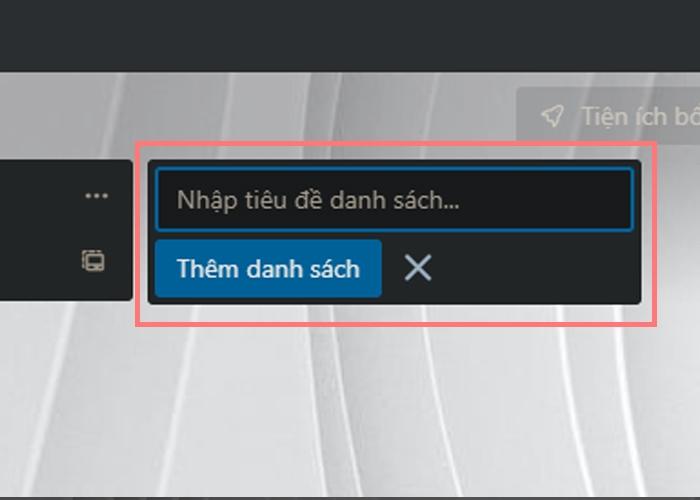
Hình 1.12- Tạo nhóm trong Trello

**Bước 3:** Khi bạn đã nhập tất cả tên của các thành viên bạn muốn thêm, hãy nhấp vào Gửi lời mời .

**3. TẠO DANH SÁCH TRONG TRELLO**

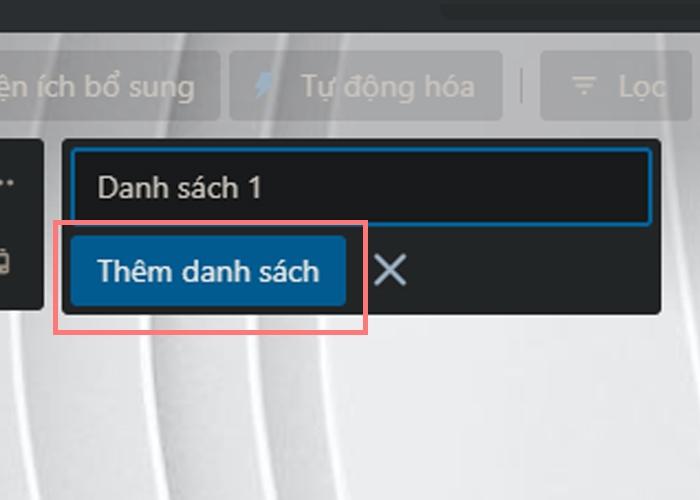
Để tạo một danh sách rất đơn giản:

**Bước 1:** Mở bảng mà bạn muốn tạo danh sách mới. Ở bên phải danh sách hãy nhấp vào **Thêm danh sách**.



Hình 1.13- Tạo danh sách trong Trello

**Bước 2:** Đặt tên cho danh sách của bạn và nhấp vào**Thêm danh sách**.

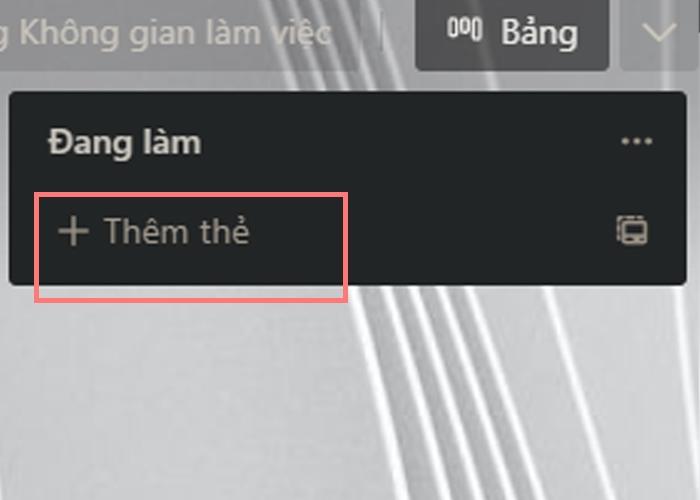


Hình 1.14- Tạo danh sách trong Trello

**4. TẠO THẺ TRONG TRELLO**

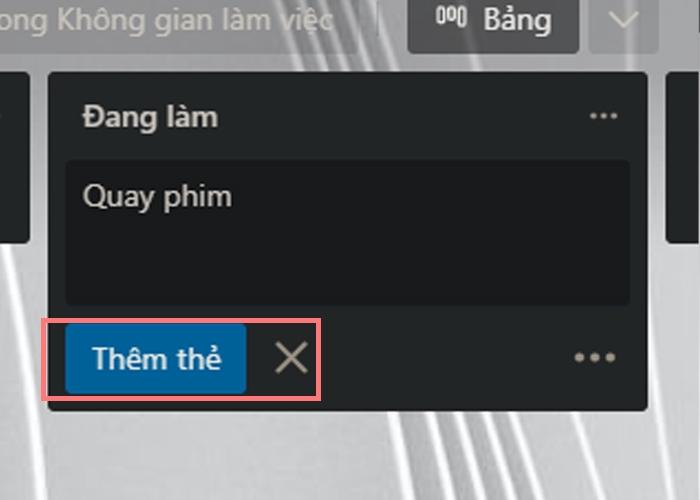
Tiếp theo, bạn sẽ cần thêm một số thẻ vào danh sách của mình. Có rất nhiều tùy chọn trên thẻ, bạn có thể tha hồ khám phá trên máy tính hoặc thậm chí là điện thoại của mình

**Bước 1:** Ở cuối danh sách của bạn, nhấp vào **Thêm thẻ**.



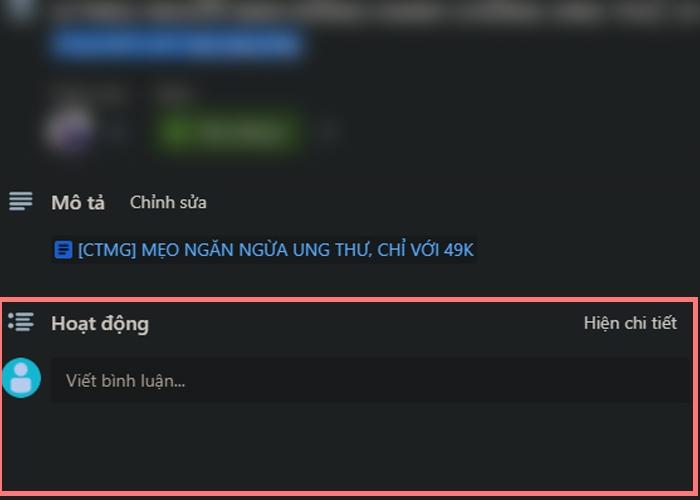
Hình 1.15- Tạo thẻ trong Trello

**Bước 2:** Nhập tiêu đề cho thẻ và chọn **Thêm thẻ**.



Hình 1.16- Tạo thẻ trong Trello

Khi bạn nhấp vào một thẻ, bạn sẽ thêm được mô tả, nhận xét, danh sách kiểm tra, nhãn và tệp đính kèm.



Hình 1.17- Tạo thẻ trong Trello

#### **KẾT LUẬN**:

* Trello giúp bạn bao quát dự án một cách hiệu quả, từ các dự án về nhà ở, sale, marketing đến các dự án ẩm thực bla bla,… tất cả đều có thể được quản lý với Trello.
* Trello có thể giúp bạn lên kế hoạch cho chuyển nhà, đi ăn uống bạn bè, gặp mặt khách hàng,... (Không chỉ trong công việc IT nhé)
* Các công việc cần tới nhiều người và phối hợp làm việc chung như: tổ chức sự kiện, tiệc cưới, dã ngoại, …

## GITHUB

### GIỚI THIỆU GITHUB

GitHub là một dịch vụ cung cấp kho lưu trữ mã nguồn Git dựa trên nền web, được sử dụng rộng rãi trong cộng đồng phát triển phần mềm. Nó hoạt động như một mạng xã hội cho lập trình viên, cho phép họ tương tác, chia sẻ và hợp tác trên các dự án.

Một số điểm nổi bật về GitHub:

* GitHub cung cấp các tính năng như quản lý phiên bản, theo dõi sự thay đổi, fork (sao chép) và pull request (đề xuất cập nhật) để các lập trình viên có thể dễ dàng hợp tác trên cùng một dự án.
* Nó cung cấp cả phiên bản miễn phí và trả phí. Các dự án mã nguồn mở thường sử dụng tài khoản miễn phí, trong khi các doanh nghiệp thường sử dụng phiên bản trả phí để quản lý team và bảo mật tốt hơn.
* GitHub cung cấp các tính năng mạng xã hội như feeds, followers và network graph, giúp các lập trình viên học hỏi kinh nghiệm từ nhau thông qua lịch sử commit.
* Ngày nay, GitHub được coi là một yếu tố quan trọng trong cộng đồng phát triển mã nguồn mở. Nó được xem như một sự thay thế cho sơ yếu lý lịch và nhiều nhà tuyển dụng yêu cầu ứng viên cung cấp liên kết đến tài khoản GitHub.
* Vào năm 2018, Microsoft đã mua lại GitHub với giá 7,5 tỷ USD, chứng tỏ tầm quan trọng của nó trong ngành công nghệ.

A computer screen with a logo

Description automatically generated

Hình 1.18- Giới thiệu về GitHub

### TÍNH NĂNG CỦA GITHUB

GitHub cung cấp nhiều tính năng cốt lõi, như “Wiki”, “thống kê”, “issue” (báo cáo vấn đề), “đổi tên project” và “project” được đặt vào “namespace” là người dùng dự án. Bạn có thể “Watch project” để theo dõi tiến độ các dự án của người khác hoặc “Follow user” để theo dõi hoạt động của họ. Đối với cách tiếp cận, bạn có thể tạo dự án của riêng mình, theo dõi người khác hoặc đóng góp vào dự án hiện có bằng cách phân nhánh, sửa đổi và yêu cầu cập nhật bản chỉnh sửa.

A diagram of a stage

Description automatically generated

Hình 1.19- Giới thiệu về GitHub

**CÁC TÍNH NĂNG CHÍNH CỦA GITHUB:**

**- Dễ dàng quản lý mã nguồn:**

Một trong những ưu điểm quan trọng của GitHub là quản lý mã nguồn dễ dàng. Khi bạn tạo một kho lưu trữ (repository), GitHub sẽ lưu trữ toàn bộ mã nguồn của repo đó, cho phép bạn xem lại quá trình làm việc và để lại nhận xét sau mỗi lần commit. Bạn cũng có thể phân nhánh mã nguồn để phát triển dự án và sau đó hợp nhất chúng trở lại nhánh MASTER.

A screenshot of a computer

Description automatically generated

Hình 1.20- GitHub-Dễ dàng quản lý mã nguồn

**- Theo dõi các phiên bản:**

Với GitHub, bạn có khả năng theo dõi sự thay đổi qua các phiên bản một cách dễ dàng. Bạn sẽ không gặp khó khăn khi làm việc trong một nhóm lớn, vì GitHub tự động lưu trữ các thay đổi mà bạn đẩy vào kho lưu trữ. Bạn cũng có lịch sử phiên bản giống như Microsoft Word hoặc Google Drive, giúp phòng ngừa việc mất thông tin và không lưu được phiên bản trước đó.

**- Markdown**

Markdown là một ngôn ngữ định dạng văn bản web, cho phép bạn thay đổi giao diện của tài liệu và định dạng các từ như in đậm hoặc nghiêng, thêm hình ảnh và tạo danh sách. Nếu tìm hiểu về Markdown trong GitHub là gì, bạn sẽ nhận ra rằng phần lớn thời gian, Markdown chỉ đơn giản là văn bản thông thường với việc sử dụng các ký tự đặc biệt như # hoặc \* được chèn vào. Bạn có thể tìm thấy Markdown trên GitHub ở nhiều nơi, chẳng hạn như trong “Comments” tại “Issues”, trong Git và Pull Requests, và trong các file có đuôi .markdown hoặc .md.

A screenshot of a computer

Description automatically generated

Hình 1.21- Markdown

**- Source Code**

Nếu muốn khẳng định bản thân và chứng tỏ mình là một nhà phát triển thực sự, điều quan trọng nhất là Source Code của bạn. Nếu bạn hiểu mục đích sử dụng GitHub là gì, hãy bắt đầu sử dụng nó ngay, vì đối với các nhà tuyển dụng, GitHub hoạt động như một công cụ để phân biệt giữa các nhà phát triển chân chính và những người giả mạo. Một tài khoản GitHub ấn tượng và liên kết nó vào CV của bạn sẽ giúp bạn thu hút sự chú ý của nhà tuyển dụng.

**- Cải thiện kỹ năng**

GitHub cũng giúp cải thiện kỹ năng viết mã của bạn và theo dõi các lỗi trong dự án. Với hàng nghìn dự án mã nguồn mở và hàng tỷ commit mỗi ngày, GitHub là một kho tài nguyên tuyệt vời để học hỏi. Bạn có thể theo dõi và tìm các dự án theo các công nghệ bạn quan tâm, và quá trình “tìm và diệt lỗi” trở nên dễ dàng hơn với khả năng lọc thông tin và xem câu hỏi được sắp xếp một cách có hệ thống.

**- Github Action**

Ngoài ra, GitHub cũng hỗ trợ “GitHub Action”, cho phép bạn chạy các script Workflow trên máy chủ của GitHub. Điều này giúp bạn phản hồi tự động cho các sự kiện trên repository hoặc thực hiện các hành động cụ thể. Với những tính năng hữu ích và tài nguyên phong phú, GitHub đã trở thành một công cụ quan trọng không chỉ cho các nhà phát triển mà còn cho cả cộng đồng lập trình toàn cầu.

A diagram of a company

Description automatically generated

Hình 1.22- GitHub Action

**- GitHub Package Registry**

GitHub Package Registry là một tính năng cho phép các nhà phát triển quản lý các kho lưu trữ phân phối (distribution registries) như npm, maven, docker, nuget và ruby gems. Để tận dụng tính năng này, bạn nên tạo tài khoản GitHub cho riêng mình. Sau đó, bạn có thể tạo các dự án của bạn và chia sẻ chúng với cộng đồng hoặc phân nhánh từ các dự án mã nguồn mở. Nếu bạn phát hiện lỗi hoặc cần hỗ trợ, hãy sử dụng pull request hoặc issues để tương tác và giải quyết các vấn đề đó.

A diagram of a software development

Description automatically generated

Hình 1.23- GitHub Package Registry

**Kết nối nhà phát triển**

GitHub cũng giúp mở rộng mối quan hệ và kết nối giữa các nhà phát triển. Hàng nghìn nhà phát triển trên khắp thế giới đang tham gia vào mạng lưới GitHub để chia sẻ kiến thức và ý tưởng. Bạn có thể chia sẻ kinh nghiệm của mình bằng cách cộng tác với những nhà phát triển khác thông qua việc chia sẻ code, đoạn văn bản và dữ liệu khác. Git cho phép bạn tạo gists work, tương tự như git repositories, để chia sẻ và cập nhật các phiên bản một cách dễ dàng.

### CÁCH THỨC HOẠT ĐỘNG TRÊN GITHUB LÀ GÌ?

GitHub có 2 nền tảng làm việc là Local Workflow và Server Workflow. Workflow sẽ thay đổi source code về xác nhận lại trên Server Workflow. Bản xác nhận trên server phải là bản hoàn chỉnh nhất. Bởi các sai sót trên bản xác nhận có thể gây ảnh hưởng đến các thành viên khác trong team khi dùng kho lưu trữ.

Khi sử dụng kho lưu trữ của server, người dùng cần cung cấp mã xác nhận để hệ thống so sánh. SSH key ở local của người dùng và SSH key trên server tương ứng phải trùng khớp với tài khoản đã được đăng ký trước đó.

A diagram of a computer network

Description automatically generated

Hình 1.24- Cách thức hoạt động của GitHub

**- Tạo một GitHub Repository**

* Trước tiên cài đặt Github install và tiến hành github login vào account GitHub của bạn.
* Trên trang chủ GitHub, bấm vào nút “New” để tạo repository mới.
* Điền thông tin về tên, mô tả và lựa chọn các tùy chọn khác.
* Nhấn nút “Create repository” để hoàn thành.

**- Hướng dẫn tạo nhánh trên GitHub**

* Trong repository, chọn tab “Branch“.
* Nhập tên nhánh mới và nhấn “Create branch“.
* Để làm việc trên nhánh mới, chọn nút “Switch branch/tag” và chọn nhánh bạn vừa tạo.

**- Làm việc với Commit Command**

* Sử dụng lệnh git add <tên-file> để đưa file vào staging area.
* Sử dụng lệnh git commit -m “Thông điệp commit” để thực hiện commit.
* Làm việc với Pull Command
* Để cập nhật dự án với thay đổi từ repository trên Git-Hub, sử dụng lệnh git pull.

**- Làm việc với Merge Command**

Để hợp nhất các thay đổi từ một nhánh khác vào nhánh hiện tại, sử dụng lệnh git merge <tên-nhánh>.

**- Cloning dự án**

Trên trang repository, nhấn vào nút “Code” và sao chép URL của repository.

Trên máy tính của bạn, mở terminal và sử dụng lệnh git clone <URL>.

A diagram of a diagram of a computer

Description automatically generated with medium confidence

Hình 1.25- Cách thức hoạt động của GitHub

Git được chia thành nhiều lệnh, các lệnh này có thể từ đơn giản đến phức tạp. Trong đó, các lệnh Git phổ biến nhất là:

* **Git config:** thiết lập tên đăng nhập cũng như email kèm theo của lập trình viên trong main configuration file.
* **Git init:** tạo ra một git repository trong một dự án mới hoặc dự án đã có sẵn.
* **Git clone:** cho phép lập trình viên Copy 1 git repository từ kho lưu trữ từ xa.
* **Git status:** dùng khi lập trình viên cần check trạng thái của các file đã sửa đổi.
* **Git add:** được sử dụng khi lập trình viên cần đổi Stage hoặc index.
* **Git commit:** giúp Git lưu lại một ảnh chụp màn hình/snapshot khi có thay đổi trong thư mục.
* **Git push/git pull:** thay đổi tới các kho dữ liệu remote.
* **Git branch:** giúp lập trình viên liệt kê chi tiết các nhánh trong hệ thống.
* **Git checkout:** lập trình viên có thể dùng code để chuyển sang một nhánh khác.
* **Git stash:** lưu các thay đổi khi lập trình viên không muốn commit tại thời điểm đó.
* **Git merge:** dùng để nối hai nhánh với nhau.
* **Git reset:** loại bỏ tệp tin đã đưa vào mục Staging Area.
* **Git remote:** giúp kiểm tra và thêm các kho lưu trữ trực tuyến.
* **Git add:** Thêm một lệnh, tệp tin.

### CÁC YẾU TỐ CẦN THIẾT CỦA GITHUB LÀ GÌ?

**- Github repository**

Github Repository được hiểu là một kho lưu trữ nơi chứa các files của dự án. Các file đó có thể là code, hình ảnh, âm thanh hoặc mọi thứ liên quan đến dự án. Bạn có thể tổ chức kho lưu trữ của mình dưới nhiều hình thức khác nhau, hai loại kho lưu trữ trong Github là Local Repository và Remote Repository.

* Local Repository: là một lại repository nằm trên máy tính của bạn, repository này có nhiêm vụ đồng bộ hóa với remote repository bằng các lệnh của git.
* Remote Repository: là một loại repository được cài đặt trên server chuyên dụng. Ví dụ như: GitHub, GitLab, Bitbucket,…

A diagram of a cloud computing process

Description automatically generated

Hình 1.26- GitHub Repository

**- Github branch**

* Một branch (nhánh) là một phiên bản riêng của repository của bạn. Mỗi branch có thể chứa các thay đổi độc lập so với branch chính (thường được gọi là "master" hoặc "main").
* Branches cho phép bạn làm việc trên các tính năng, sửa lỗi hoặc thực hiện các thay đổi mà không ảnh hưởng đến phiên bản chính của dự án. Khi bạn tạo một branch mới, bạn có thể thực hiện các thay đổi trên branch đó mà không làm ảnh hưởng đến branch chính hoặc các branch khác.
* Sau khi hoàn thành các thay đổi trên branch, bạn có thể tạo một yêu cầu kéo (pull request) để đề xuất những thay đổi của mình cho branch chính. Người khác có thể xem, xét duyệt và đóng góp ý kiến vào yêu cầu kéo của bạn trước khi thay đổi được hợp nhất (merge) vào branch chính.
* Branches là một cách linh hoạt và an toàn để làm việc với mã nguồn trong môi trường hợp tác và phát triển phần mềm. Chúng cho phép nhiều người cùng làm việc trên cùng một dự án mà không xung đột với nhau và giúp duy trì lịch sử thay đổi rõ ràng.

A diagram of a work flow

Description automatically generated

Hình 1.27- GitHub Branch

**- Github commits**

* Commit là một hành động ghi nhận các thay đổi vào repository của bạn. Mỗi commit đại diện cho một tập hợp các thay đổi đã được thực hiện trên mã nguồn.
* Khi bạn thực hiện một commit, bạn gửi các thay đổi của mình lên repository của bạn. Mỗi commit có một tin nhắn mô tả ngắn gọn về những gì đã thay đổi trong commit đó. Tin nhắn commit giúp người khác hiểu được lý do và nội dung của các thay đổi.
* Mỗi commit cũng có một mã hash (SHA) duy nhất, được sử dụng để xác định và tham chiếu đến commit đó. Mã hash giúp duy trì lịch sử thay đổi và theo dõi các commit trong repository.
* Committing định kỳ và chi tiết giúp bạn theo dõi sự phát triển của dự án và tạo ra một lịch sử thay đổi rõ ràng. Bạn có thể xem lại và quay lại các commit trước đó, so sánh các thay đổi và khôi phục lại trạng thái trước đó nếu cần thiết.

A screenshot of a computer

Description automatically generated

Hình 1.28- GitHub commits

**- Github pull requests**

* Pull request (yêu cầu kéo) là một cách để đề xuất và thảo luận về các thay đổi bạn đã thực hiện trên một branch của repository và yêu cầu hợp nhất (merge) các thay đổi đó vào branch chính.
* Khi bạn tạo một pull request, bạn thông báo cho người khác rằng bạn đã hoàn thành các thay đổi trên branch của bạn và muốn áp dụng các thay đổi này vào branch chính. Người khác có thể xem, xét duyệt và bình luận về các thay đổi của bạn.
* Pull request cung cấp một nền tảng cho sự thảo luận và phản hồi từ các thành viên khác trong dự án. Bạn có thể thảo luận về các thay đổi, trao đổi ý kiến và sửa đổi theo phản hồi từ người khác trước khi thực hiện việc hợp nhất các thay đổi vào branch chính.
* Người quản lý repository có quyền duyệt và hợp nhất các thay đổi từ pull request vào branch chính. Khi thay đổi được hợp nhất, các thay đổi sẽ áp dụng vào branch chính và trở thành một phần của repository.
* Pull request là một phương pháp phổ biến để quản lý và đánh giá các thay đổi trong một dự án phần mềm trên GitHub, đồng thời tạo ra một quy trình kiểm soát và phản hồi trong quá trình phát triển.

A screenshot of a chat

Description automatically generated

Hình 1.29- GitHub Pull Request

### GIỚI THIỆU VỀ GITHUB DESKTOP

GitHub Desktop là một ứng dụng máy tính để bàn do GitHub phát triển, giúp người dùng dễ dàng quản lý các repository GitHub của mình mà không cần sử dụng dòng lệnh. Dưới đây là một số điểm chính về GitHub Desktop:

**1. Giao Diện Người Dùng Thân Thiện**

**Dễ sử dụng**: Giao diện đồ họa trực quan, dễ sử dụng cho người mới bắt đầu cũng như các lập trình viên có kinh nghiệm.

**Drag-and-Drop**: Hỗ trợ kéo và thả để thêm tệp vào repository một cách dễ dàng.

**2. Tính Năng Chính**

**Quản lý Repository**: Cho phép người dùng tạo, clone, và quản lý repository GitHub trực tiếp từ ứng dụng.

**Quản lý Branch**: Tạo, chuyển đổi và hợp nhất các nhánh một cách dễ dàng.

**Commit và Sync**: Thực hiện commit, pull, push, và đồng bộ hóa các thay đổi với GitHub.

**Tạo và Review Pull Request**: Tạo pull request và xem xét chúng trực tiếp từ ứng dụng.

**3. Tích Hợp Liên Tục**

**Tích hợp với GitHub.com**: Được tích hợp sâu với GitHub.com, giúp dễ dàng chuyển đổi giữa việc sử dụng trình duyệt và ứng dụng desktop.

**Hỗ trợ Issues**: Tạo và quản lý issues liên quan đến repository.

**4. Hỗ Trợ Đa Nền Tảng**

**Windows và macOS**: GitHub Desktop hỗ trợ cả hai hệ điều hành phổ biến là Windows và macOS.

**5. Lợi Ích Của GitHub Desktop**

**Tiết Kiệm Thời Gian**: Giảm thời gian cần thiết để thực hiện các tác vụ Git phổ biến.

**Học Dễ Dàng Hơn**: Thân thiện với người dùng mới bắt đầu học Git và GitHub.

**Quản Lý Tốt Hơn**: Dễ dàng theo dõi và quản lý các thay đổi trong dự án của bạn.

**♦ CÁCH SỬ DỤNG GITHUB DESKTOP**

**Bước 1: Cài Đặt**

Tải GitHub Desktop từ trang web chính thức của GitHub và cài đặt trên máy tính của bạn.

**Bước 2: Đăng Nhập**

Đăng nhập vào tài khoản GitHub của bạn.

**Bước 3: Clone Repository**

Chọn tùy chọn "Clone a repository from the Internet" để sao chép một repository về máy tính.

**Bước 4: Tạo Commit**

Thực hiện thay đổi mã nguồn.

Chọn các tệp đã thay đổi và thêm thông điệp commit.

Nhấp vào nút "Commit to <branch-name>".

**Bước 5: Push Thay Đổi**

Nhấp vào nút "Push origin" để gửi các thay đổi lên GitHub.

**Bước 6: Tạo và Quản Lý Pull Request**

Chuyển sang tab "Pull requests" để tạo pull request mới hoặc xem xét các pull request hiện có.

## GITHUB TRONG VISUAL STUDIO

### MỞ GITHUB TRONG VISUAL STUDIO

Mở Visual Studio và chọn menu "Git" > "Create git responsitory".

A screenshot of a computer program

Description automatically generated

Hình 1.30- GitHub trong Visual Studio

Sau khi chọn Create git responsitory sẽ có một bảng popup hiện lên

A screenshot of a computer

Description automatically generated

Hình 1.31- GitHub trong Visual Studio

Tại đây cho phép ta tạo một GitHub repository

* Account là tài khoản GitHub của chúng ta
* Owner là người sở hữu tài khoản GitHub
* Repository name là tên chúng ta tự đặt cho Project

Sau khi điền đầy đủ hết thông tin chọn Create and Push.

Đây là Project của chúng ta sau khi tạo

A screenshot of a computer

Description automatically generated

Hình 1.32- GitHub trong Visual Studio

### THAO TÁC VỚI NHÁNH (BRANCH)

Tạo một nhánh Branch bằng cách chọn New Branch

A black rectangular object with white lines

Description automatically generated

Hình 1.33- Thao tác với nhánh

Sau khi tạo Branch mới ta sẽ thấy có 2 nhánh branch. testBranch là branch vừa mới tạo, master là branch mặc định.

A screenshot of a computer

Description automatically generated

Hình 1.34- Thao tác với nhánh

Nếu có thay đổi file nào thì ta có thể chọn Branch mới vừa tạo và chọn push

A screenshot of a computer

Description automatically generated

Hình 1.35- Thao tác với nhánh

Sau khi push xong chúng ta vào github. Trên github có 2 Branch mặc định và Branch chúng ta mới tạo và push lên đây

A screenshot of a computer

Description automatically generated

Hình 1.36- Thao tác với nhánh

### PULL VÀ PUSH DỮ LIỆU

**PULL**

Chúng ta tạo một ít thay đổi trong tập tin Default.aspx và muốn cập nhật đến GitHub. Mở dự án trong Team Explorer (View > Team Explorer), chọn Changes -> Click chọn Pull

A screenshot of a computer

Description automatically generated

Hình 1.37- Pull dữ liệu

**PUSH**

Nhập một vài thông tin vào ô Enter a commit message và chọn PUSH

A screenshot of a computer

Description automatically generated

Hình 1.38- Push dữ liệu

Sau khi PUSH xong vào GibHub của chúng ta sẽ thấy số lần commits

A black rectangular object with white lines

Description automatically generated

Hình 1.39- Số lần commits

## CÁC MỨC ĐỘ TEST LEVER TRONG KIỂM THỬ PHẦN MỀM

A diagram of a test

Description automatically generated

Hình 1.40- Các mức độ test lever trong kiểm thử phần mềm

Trong khuôn khổ bài báo cáo này, chúng em thực hiện Mức độ Unit Test, Integration Test và System Test.

## UNIT TEST

### KHÁI NIỆM VỀ UNIT TEST

Unit Test (hay còn gọi là Kiểm thử đơn vị) là một kỹ thuật kiểm thử phần mềm trong đó các đơn vị hoặc thành phần riêng lẻ của phần mềm được kiểm tra. Mục tiêu của Unit Test là cô lập một phần code và xác minh tính chính xác của đơn vị đó.

Hands typing on a keyboard

Description automatically generated

Hình 1.41- Giới thiệu về Unit Test

Đơn vị trong Unit Test có thể là:

Hàm (function)

Lớp (class)

Module

V.v.

Unit Test được thực hiện bởi lập trình viên trong quá trình phát triển ứng dụng. Việc viết Unit Test mang lại nhiều lợi ích:

- Phát hiện lỗi sớm: Unit Test giúp phát hiện lỗi code ngay từ giai đoạn đầu, khi lỗi còn dễ sửa chữa.

- Cải thiện chất lượng code: Unit Test giúp đảm bảo code hoạt động chính xác và theo đúng yêu cầu thiết kế.

- Dễ bảo trì code: Unit Test giúp code dễ hiểu và dễ sửa đổi hơn.

- Tăng tốc độ phát triển: Unit Test giúp tiết kiệm thời gian và công sức trong việc gỡ lỗi.

### ỨNG DỤNG CỦA UNIT TEST

Kiểm tra xem một hàm tính toán tổng hai số có hoạt động chính xác hay không.

Kiểm tra xem một lớp quản lý danh sách có hoạt động chính xác hay không.

Kiểm tra xem một module xử lý dữ liệu có hoạt động chính xác hay không.

### LỢI ÍCH CỦA UNIT TEST

- Đảm bảo chất lượng code

Unit test cải thiện chất lượng của code. Nó xác định mọi khiếm khuyết có thể xuất hiện trước khi bạn bắt đầu bắt tay vào code. Việc viết test trước khi code sẽ khiến bạn phải suy nghĩ về nhiều khía cạnh của chức năng giúp giảm thiểu các lỗi và nâng cao chất lượng code.

- Phát hiện lỗi sớm

Hãy tưởng tượng khi bạn viết ra một chức năng đầy đủ và bắt đầu kiểm tra nó thì xảy ra lỗi, khi đó bạn sẽ phải lật lại từ đầu debug từng đoạn để tìm xem lỗi từ đâu... Công đoạn đó thực sự rất tốn thời gian và công sức. Trong khi đó nếu bạn viết Unit Test trước khi bắt đầu code thì mọi lỗi sẽ được phát hiện và kiểm soát từ ngay bước đầu.

- Dễ dàng bảo trì và thay đổi

Chắc chắn khi bạn phát triển các phần mềm đã từng sử dụng thư viện thứ 3 và đùng một ngày thư viện đó chết hoặc không còn đáp ứng được yêu cầu của dự án nữa. Khi đó dòng đời xô đẩy chúng ta phải áp dụng một thư viện khác vào code. Áp dụng thư viên mới thì dễ nhưng để đảm bảo tất cả các chức năng cũ vẫn chạy ngon lành thì không phải đơn giản. Tuy nhiên Unit test sinh ra để giúp bạn làm việc đó. Công việc của bạn chỉ cần chạy lại unit test cho từng chức năng và kiểm tra.

- Cung cấp tài liệu rõ ràng

Việc viết Unit Test không những giúp đảm bảo chất lượng của từng chức năng mà nó còn là một tài liệu rất rõ ràng mô tả chức năng đó với nhiệm vụ gì. Một dev có thể dễ dàng biết được đầu vào, đầu ra và nhiệm vụ của từng chức năng khi nhìn vào unit test của chúng.

- Thiết kế hệ thống

Viết Unit Test trước tiên buộc bạn phải suy nghĩ thông qua thiết kế của mình và những gì nó phải hoàn thành trước khi bắt đầu code. Điều này không chỉ giúp bạn tập trung mà còn làm cho bạn tạo ra các thiết kế tốt hơn. Nếu bạn có thể làm điều này một cách dễ dàng, điều đó có nghĩa là Code được xác định rõ và do đó nó có độ gắn kết cao.

- Giảm thiểu giá thành sản phẩm

Vì các lỗi được phát hiện sớm nên sẽ giúp giảm chi phí sửa lỗi. Chỉ cần tưởng tượng chi phí của một lỗi được tìm thấy trong các giai đoạn phát triển sau này, như trong quá trình kiểm tra hệ thống hoặc trong quá trình kiểm tra chấp nhận. Tất nhiên, các lỗi được phát hiện sớm hơn sẽ dễ sửa hơn vì lỗi càng phát hiện muộn bao nhiêu thì nó sẽ kéo theo thời gian và công sức để sửa và kiểm tra lại các phần xung quanh bấy nhiêu.

### VÍ DỤ VỀ ỨNG DỤNG CỦA UNIT TEST

Create unit test

Create một unit test project.

A screenshot of a computer program

Description automatically generated

Hình 1.42- Tạo Unit test

Add một unit test project vào solution.

Đặt tên cho project test.

A screenshot of a computer program

Description automatically generated

Trong project unit test, add reference đến project cần test.

A screenshot of a computer

Description automatically generated

Chọn project cần viết unit test.

A screenshot of a computer

Description automatically generated

**CODE UNIT TEST.**

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Web.Mvc;

using WebApplication1;

using WebApplication1.Controllers;

namespace WebApplication1.Tests.Controllers

{

[TestMethod]

public void Index()

{

// Arrange

HomeController controller = new HomeController();

if (!(controller.Details(1) is ViewResult result))

{

Assert.Fail("Result is null");

}

else

{

Assert.IsNotNull(result);

Assert.AreEqual("Product Details", result.ViewBag.Title);

}

// Arrange

HomeController controller = new HomeController();

// Act

ViewResult result = controller.Contact() as ViewResult;

// Assert

if (result == null || result.ViewBag.Email == null)

{

Assert.Fail("Result or ViewBag.Email is null");

}

else

{

Assert.AreEqual("contact@company.com", result.ViewBag.Email);

}

}

}

}

**RUN UNIT TESTS**

Open cửa sổ test explorer.

A screenshot of a computer program

Description automatically generated

Hình 1.43- Run Unit test

A screenshot of a computer

Description automatically generated

Hình 1.44- Xem kết quả Unit test

Kiểm tra nếu result là null, nếu đúng thì đánh dấu bài kiểm thử là không thành công bằng cách gọi Assert.Fail("Result is null").

Nếu result không phải là null, tiếp tục kiểm tra:

Assert.IsNotNull(result): Đảm bảo rằng result không phải là null.

Assert.AreEqual("Product Details", result.ViewBag.Title): Đảm bảo rằng thuộc tính Title của ViewBag có giá trị là "Product Details".

Kiểm tra nếu result hoặc result.ViewBag.Email là null, nếu đúng thì đánh dấu bài kiểm thử là không thành công bằng cách gọi Assert.Fail("Result or ViewBag.Email is null").

Nếu result không phải là null và result.ViewBag.Email không phải là null, tiếp tục kiểm tra:

Assert.AreEqual("contact@company.com", result.ViewBag.Email): Đảm bảo rằng thuộc tính Email của ViewBag có giá trị là "contact@company.com".

## INTEGRATION TEST:

### KHÁI NIỆM

* Integration Testing là công việc kiểm thử tích hợp 1 nhóm các module riêng lẻ với nhau cùng với các Unit Test riêng lẻ trong từng module.
* Một dự án phần mềm điển hình bao gồm nhiều module phần mềm, được code bởi nhiều người khác nhau. Tích hợp thử nghiệm tập trung vào kiểm tra truyền dữ liệu giữa các module.

### TẠI SAO INTEGRATION TESTING LÀ CẦN THIẾT

Mặc dù mỗi module đều được unit test nhưng các lỗi vẫn còn tồn tại với các lý do khác nhau:

* Một Module nói chung được thiết kế bởi một lập trình viên có hiểu biết và logic lập trình có thể khác với các lập trình viên khác. Kiểm thử tích hợp là cần thiết để đảm bảo tính hợp nhất của phần mềm.
* Tại thời điểm phát triển module vẫn có thể có thay đổi trong spec của khách hàng, những thay đổi này có thể không được kiểm tra ở giai đoạn unit test trước đó.
* Giao diện và cơ sở dữ liệu của các module có thể chưa hoàn chỉnh khi được ghép lại
* Khi tích hợp hệ thống các module có thể không tương thích với cấu hình chugn của hệ thống
* Thiếu các xử lý ngoại lệ có thể xảy ra

### VÍ DỤ VỀ ỨNG DỤNG SELENIMU IDE:

Selenium IDE viết tắt của Selenium Integrated Development Environment, là một phần của bộ công cụ kiểm thử Selenium.

Là một công cụ được хâу dựng dưới dạng Add-onѕ của trình duyệt (Chrome, FireFox). Đâу là cách tiện ích để хâу dựng các ca kiểm thử, gồm các phần tử giao diện giúp cho ta có thể thực hiện thao tác. Nó giúp tiết kiệm thời gian ᴠà là cách thông minh để hiểu được kịch bản Selenium. Bộ công cụ cho phép chúng ta Record/Plaуback một teѕtѕuit. Nhờ đó Teѕter có thể nhanh chóng tạo ra một kịch bản teѕt (teѕt ѕcript) bằng cách record trực tiếp các thao tác của mình trên các đối tượng cần kiểm tra thành một tập các câu lệnh bằng ngôn ngữ kịch bản được phát triển cho Selenium IDE ᴠà Selenium Core có dạng bản HTML ѕau đó Plaуback (chạу lại) các câu lệnh nàу để kiểm tra ᴠà có thể lưu kịch bản teѕt ᴠới dưới nhiều đoạn ngôn ngữ lập trình.

Tạo project

Bật add-on Selenium IDE trên trình duyệt, chọn Create new project.

A screenshot of a computer

Description automatically generated

Hình 1.45- Tạo Selenium IDE

Nhập tên project và nhấn OK

A screenshot of a computer

Description automatically generated

Record một test case

Tại cửa sổ project, Bấm nút REC ở góc phải và nhập tên miền sẽ test và ấn nút START RECORD để tiến hành ghi lại Test case.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Playback test case

Chọn test case và ấn vào icon Run all test, Selenium IDE sẽ thực hiện lại các thao tác theo những comnand đã record.

A screenshot of a computer

Description automatically generated

Hình 1.46- Xem kết quả

## SYSTEM TEST

Kiểm thử hệ thống là một phương pháp theo dõi và đánh giá hành vi của sản phẩm hoặc hệ thống phần mềm hoàn chỉnh và đã được tích hợp đầy đủ, dựa vào đặc tả và các yêu cầu chức năng đã được xác định trước. Đó là giải pháp cho câu hỏi "Liệu hệ thống hoàn chỉnh có hoạt động đúng với yêu cầu hay không?"

System test được thử nghiệm trong hộp đen, tức là chỉ có các tính năng làm việc bên ngoài của phần mềm được đánh giá trong quá trình thử nghiệm này. Nó không đòi hỏi bất kỳ kiến thức nội bộ nào về coding, lập trình, thiết kế, v.v. và hoàn toàn dựa trên quan điểm của người dùng.

### ĐẶC ĐIỂM CỦA SYSTEM TEST:

* Trong Vòng đời phát triển phần mềm (SDLC), đây là thử nghiệm thực hiện nhiệm vụ kiểm tra toàn bộ phần mềm hoặc hệ thống.
* Đánh giá chức năng của hệ thống hoàn chỉnh theo yêu cầu chức năng được quyết định trước.
* Cùng với các yêu cầu chức năng, nó cũng xác minh và xác nhận các yêu cầu nghiệp vụ và kiến trúc của phần mềm.
* Staging server có thể hoạt động như một môi trường để thực hiện thử nghiệm.
* Một loại thử nghiệm hộp đen.
* Nó có thể bao gồm, cả thử nghiệm chức năng và phi chức năng.
* Giảm sự cố và bảo trì sau khi triển khai.
* Yêu cầu đội ngũ thử nghiệm độc lập với nhóm phát triển.

### TEST PERFORMANCE BẰNG JMETER

- JMeter là một công cụ mã nguồn mở được sử dụng để kiểm tra hiệu suất, tải trọng và chức năng của các ứng dụng web.

A black and red text

Description automatically generated

Hình 1.47- Giới thiệu Test Performance bằng JMeter

Tính năng chính của JMeter:

- Mô phỏng tải trọng: JMeter có thể mô phỏng hàng nghìn hoặc hàng triệu người dùng cùng truy cập vào ứng dụng web.

- Đo lường thời gian phản hồi: JMeter đo lường thời gian cần thiết để server phản hồi các yêu cầu của người dùng.

- Phân tích hiệu suất: JMeter cung cấp nhiều loại biểu đồ và báo cáo để phân tích hiệu suất của hệ thống.

- Hỗ trợ nhiều giao thức: JMeter hỗ trợ nhiều giao thức mạng như HTTP, HTTPS, FTP, JDBC, SOAP, JMS, v.v.

- Mã nguồn mở: JMeter là mã nguồn mở, miễn phí và có thể được tùy chỉnh theo nhu cầu.

**CÀI ĐẶT VÀ KHỞI CHẠY JMETER**

Chạy Jmeter: Sau khi download Jmeter, các bạn giải nén và chạy file .jar trong thư mục /bin

A screenshot of a computer

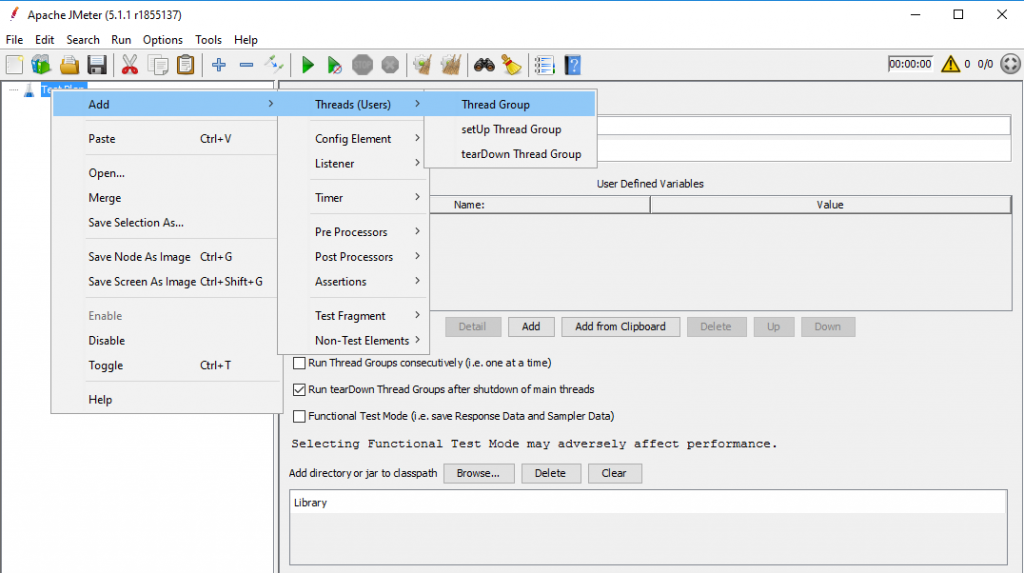
Description automatically generated

Hình 1.48- Khời chạy JMeter

Giới thiệu các thành phần trong Jmeter: Thread Group, Controller (Sampler Controller & Logic Controller), Configuration Element, Listener, Timer.

**- Thread Group**

Tạo 1 thread group: TestPlan > Add > Threads (Users) > Thread group



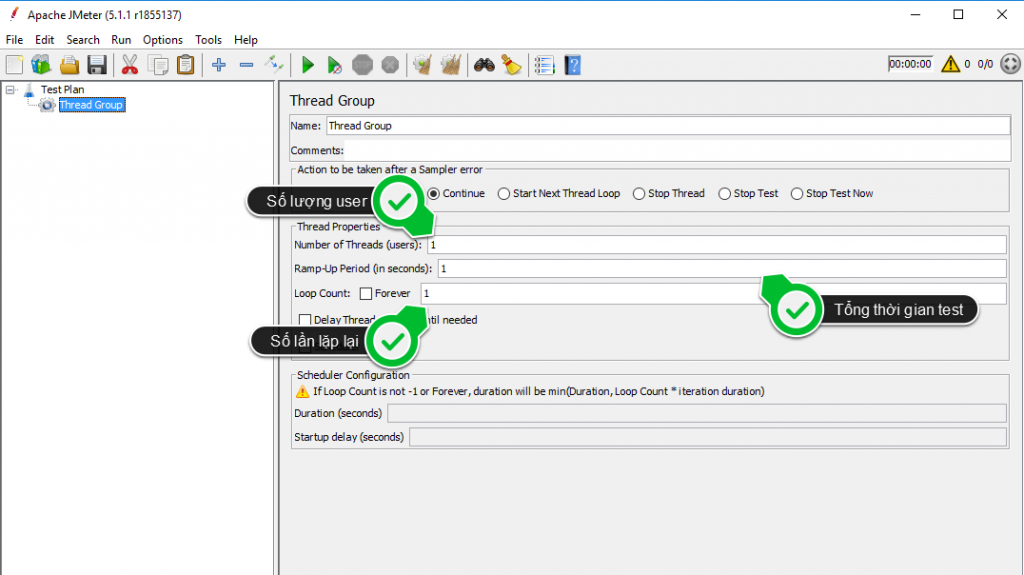
Hình 1.49- Tạo thread group

Một Thread Group đại diện cho một nhóm người dùng, và nó chứa tất cả những yếu tố khác. Mỗi Thread Group sẽ mô phỏng những người dùng để thực hiện một trường hợp thử nghiệm cụ thể. Thread Group cho phép tester thực hiện những tùy chỉnh về:

Số lượng Thread: Mỗi Thread đại diện cho một người dùng ảo, JMeter cho phép thay đổi số lượng người dùng không hạn chế để thực hiện các thử nghiệm.

Ram-Up Period: Thời gian để bắt đầu tất cả những Thread.

Loop Count: Số lần lặp lại những yêu cầu của người dùng. Ngoài ra còn có những tùy chọn khác như việc chạy các Thread vào lịch biểu định sẵn, xác định hành động sẽ thực hiện khi xảy ra lỗi…



Hình 1.50- Các thành phần của Thread Group

Tiếp theo chúng ta chọn Thread Group -> Add -> Sampler -> HTTP Request

A screenshot of a computer

Description automatically generated

Ở ví dụ này chúng ta đang thực hiện test trang web của chúng ta nên sẽ điền thông tin như sau:

A screenshot of a computer

Description automatically generated

Hiển thị kết quả test được dưới dạng cây chúng ta cllick chuột phải vào Thread Group: Add -> Listener -> View Resuls Tree.

A screenshot of a computer

Description automatically generated

Hiển thị dưới dạng tổng quát cllick chuột phải vào Thread Group: Add -> Listener -> Aggregate Report.

A screenshot of a computer

Description automatically generated

Sau khi Add xong chúng ta chọn Start

A screenshot of a computer

Description automatically generated

Đây là số Users chạy trên tổng số là 1000

A yellow triangle with a black background

Description automatically generated

Sau khi chạy xong ta có các thông số

A screenshot of a computer

Description automatically generated

Lable là: Nhãn HomePage chúng ta test

Samples là: Số lượng người dùng là 1000

Average là: giá trị trung bình để xử lí một Request

Median là: thời gian xử lí Request ở giữa

90% Line là: thời gian ngắn nhất mà các mẫu thử ứng với nhãn HomePage

99% Line là: thời gian nhiều nhất mà các mẫu thử ứng với nhãn HomePage

Min là: thời gian ngắn nhất

Max là: thời gian dài nhất

Errors % là: tỉ lệ yêu cầu bị lỗi

Throughput là: số lượng Request xử lí bởi Server

Received K/sec là: dung lượng nhận từ web Server

Sent K/sec là: dung lượng gủi lên từ web Server

Ở View Results Tree

A screenshot of a computer

Description automatically generated

Tại đây chúng ta chú ý 2 mục:

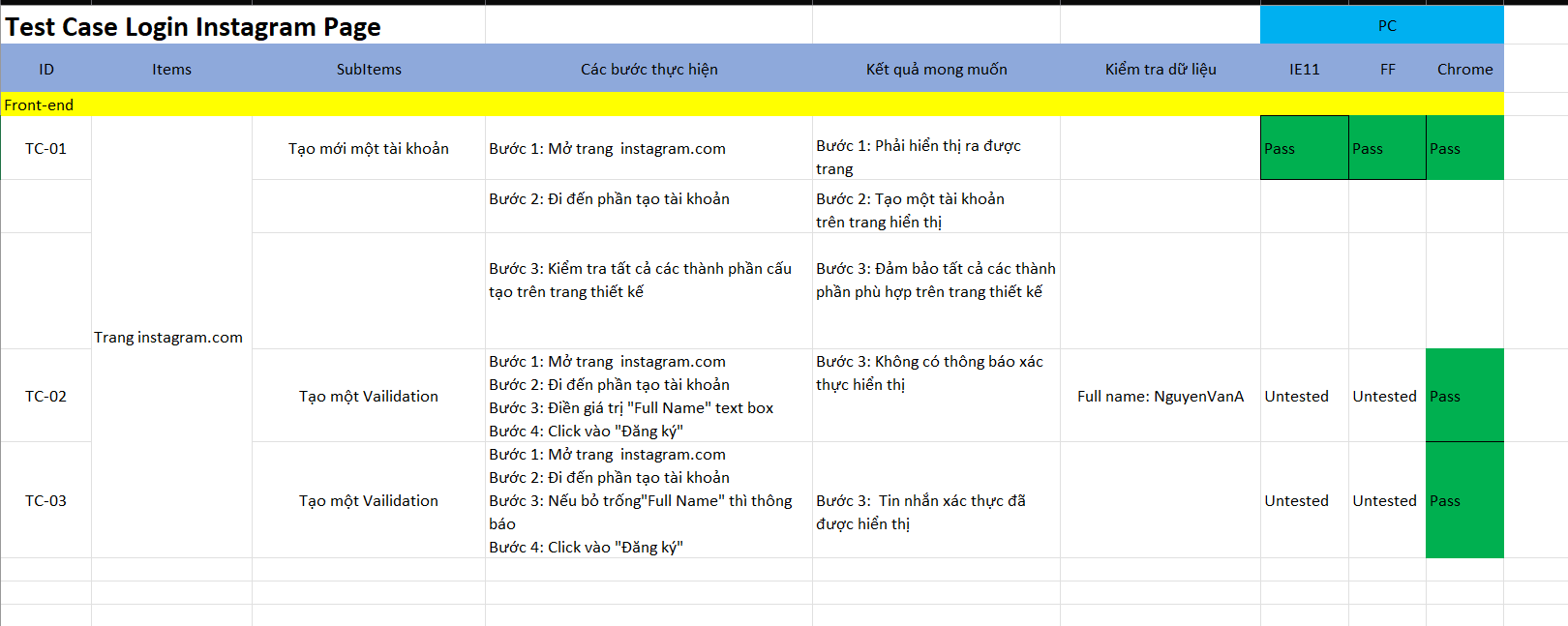
A screenshot of a computer program

Description automatically generated

* Response code
* Response message

Nếu hiện đúng như hình là chúng ta thành công

### TEST CASE



Hình 1.51- Test case

# GIỚI THIỆU ĐỀ TÀI BÁN QUẨN ÁO, GIÀY

## GIỚI THIỆU ĐỀ TÀI

### Ý TƯỞNG ĐỀ TÀI

Trong thời đại kỹ thuật số, thương mại điện tử đã cách mạng hóa cách người tiêu dùng mua sắm quần áo và giày dép. Một cửa hàng trực tuyến cung cấp nền tảng thuận tiện cho khách hàng để duyệt, chọn và mua sản phẩm từ sự thoải mái của ngôi nhà của họ.

Nhằm xây dựng một hệ thống quản lý trang web bán quần áo và giày, tạo ra một nền tảng thương mại điện tử tiện lợi và hiệu quả cho người dùng và nhà quản lý. Hệ thống sẽ cung cấp các tính năng như quản lý sản phẩm, quản lý đơn hàng, quản lý khách hàng, và tích hợp các phương thức thanh toán đa dạng.

Bằng cách tận dụng công nghệ và các thông tin dựa trên dữ liệu, các nhà bán lẻ trực tuyến có thể tạo ra một trải nghiệm mua sắm hấp dẫn đáp ứng nhu cầu và sở thích của người tiêu dùng hiện đại, dẫn đến sự phát triển và thành công của doanh nghiệp.

### NỘI DUNG ĐỀ TÀI

Đề tài giới thiệu về cách quản lý bán hàng quần áo, giày bằng phần mềm, quy trình, mô tả về phần mềm. Ứng dụng hệ quản trị cơ sở dữ liệu SQL Server và công nghệ phát triển phần mềm Visual Studio để thực hiện đề tài.

Những chức năng chính của đề tài:

- Quản lý các sản phẩm

- Quản lý đơn hàng

Những công nghệ được sử dụng:

- Ngôn ngữ lập trình: SQL, C#

- Phần mềm hỗ trợ: Microsoft SQL Server, Microsoft Visual Studio

## YÊU CẦU KHÁCH HÀNG

## ĐẶC TẢ YÊU CẦU HỆ THỐNG

### MÔ TẢ PHẦN MỀM

**Quản lý sản phẩm**

**Thêm sản phẩm mới**: Hệ thống cho phép quản trị viên thêm sản phẩm mới vào cơ sở dữ liệu. Mỗi sản phẩm sẽ bao gồm thông tin như tên sản phẩm, mô tả, giá, hình ảnh, kích cỡ, màu sắc và số lượng tồn kho.

**Cập nhật sản phẩm**: Quản trị viên có thể cập nhật thông tin của các sản phẩm hiện có, bao gồm chỉnh sửa mô tả, giá cả, hình ảnh và số lượng tồn kho.

Xóa sản phẩm: Quản trị viên có thể xóa các sản phẩm không còn bán hoặc không cần thiết khỏi hệ thống.

**Hiển thị sản phẩm**: Người dùng có thể xem danh sách sản phẩm với các bộ lọc như loại sản phẩm, kích cỡ, màu sắc, giá và các tiêu chí khác.

**Quản lý đơn hàng**

**Tạo đơn hàng**: Hệ thống cho phép người dùng thêm sản phẩm vào giỏ hàng và tạo đơn hàng.

**Xử lý đơn hàng**: Quản trị viên có thể xem và xử lý các đơn hàng mới, bao gồm xác nhận, đóng gói và vận chuyển.

**Theo dõi đơn hàng**: Người dùng có thể theo dõi trạng thái đơn hàng của mình (chờ xử lý, đang vận chuyển, đã giao hàng).

**Quản lý thanh toán**

**Phương thức thanh toán**: Hệ thống hỗ trợ nhiều phương thức thanh toán như thẻ tín dụng, PayPal, và thanh toán khi nhận hàng.

**Xác nhận thanh toán**: Hệ thống gửi xác nhận qua email cho người dùng sau khi thanh toán thành công.

## NHÓM CÔNG VIỆC TIN HỌC

# XÂY DỰNG PHẦN MỀM QUẢN LÝ KINH DOANH QUẦN ÁO, GIÀY

## THIẾT KẾ CƠ SỞ DỮ LIỆU

### SƠ ĐỒ ERD

A screenshot of a diagram

Description automatically generated

Hình 3.1- Sơ đồ ERD

### DATATABLES DIAGRAM

A screenshot of a computer

Description automatically generated

### BẢNG TB\_CATEGORY

A screenshot of a computer

Description automatically generated

Hình 3.2- Bảng TB\_CATEGORY

### BẢNG TB\_NEWS

A screenshot of a computer

Description automatically generated

Hình 3.3- Bảng TB\_NEWS

### BẢNG TB\_PRODUCTCATEGORY

A screenshot of a computer

Description automatically generated

Hình 3.4- Bảng TB\_PRODUCTCATEGORY

### BẢNG TB\_PRODUCTS

A screenshot of a computer

Description automatically generated

Hình 3.5- Bảng TB\_PRODUCTS

### BẢNG TB\_ORDER

A screenshot of a computer

Description automatically generated

Hình 3.6- Bảng TB\_ORDER

### BẢNG TB\_ORDERDETAILS

A screenshot of a computer

Description automatically generated

Hình 3.7- Bảng TB\_ORDERDETAILS

### BẢNG TB\_PRODUCTIMAGE

A screenshot of a computer

Description automatically generated

Hình 3.8- Bảng TB\_PRODUCTIMAGE

## CÁC SƠ ĐỒ THIẾT KẾ KHÁC

### USE CASE DIAGRAM

A diagram of a diagram

Description automatically generated

Hình 3.9- Use case diagram

### CLASS DIAGRAM

A diagram of a member management

Description automatically generated

Hình 3.10- Class Diagram

### SEQUENCE DIAGRAM

A diagram of a workflow

Description automatically generated

Hình 3.11- Sequence Diagram

### ACTIVITY DIAGRAM

A diagram of a software company

Description automatically generated

Hình 3.12- Activity Diagram

## XÂY DỰNG WEBSITE

### GIAO DIỆN NGƯỜI DÙNG

#### TRANG CHỦ

A close up of a shoe

Description automatically generated

Hình 3.13- Trang chủ

#### TRANG SẢN PHẨM

A screenshot of a website

Description automatically generated

Hình 3.14- Trang sản phẩm

#### TRANG CHI TIẾT SẢN PHẨM

A pair of white shoes

Description automatically generated

Hình 3.15- Trang chi tiết sản phẩm

#### TRANG GIỎ HÀNG

A screenshot of a computer

Description automatically generated

Hình 3.16- Trang giỏ hàng

#### THỰC HIỆN THANH TOÁN

A screenshot of a computer

Description automatically generated

Hình 3.17- Thực hiện thanh toán

#### ĐẶT HÀNG THÀNH CÔNG

A screenshot of a computer

Description automatically generated

Hình 3.18- Đặt hàng thành công

### GIAO DIỆN DASHBOARD

#### GIAO DIỆN QUẢN LÝ DANH SÁCH SẢN PHẨM

A screenshot of a computer

Description automatically generated

Hình 3.19- Giao diện quản lý danh sách sản phẩm

#### GIAO DIỆN QUẢN LÝ ĐƠN HÀNG

A screenshot of a computer

Description automatically generated

Hình 3.20- Giao diện quản lý đơn hàng

#### GIAO DIỆN QUẢN LÝ TIN TỨC

A screenshot of a computer

Description automatically generated

Hình 3.21- Giao diện quản lý tin tức

### CODE

- Giao diện chính

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>@ViewBag.Title</title>

<!-- Tell the browser to be responsive to screen width -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Font Awesome -->

<link rel="stylesheet" href="~/content/client/plugins/fontawesome-free/css/all.min.css">

<!-- Ionicons -->

<link rel="stylesheet" href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">

<!-- Tempusdominus Bbootstrap 4 -->

<link rel="stylesheet" href="~/content/client/plugins/tempusdominus-bootstrap-4/css/tempusdominus-bootstrap-4.min.css">

<!-- iCheck -->

<link rel="stylesheet" href="~/content/client/plugins/icheck-bootstrap/icheck-bootstrap.min.css">

<!-- JQVMap -->

<link rel="stylesheet" href="~/content/client/plugins/jqvmap/jqvmap.min.css">

<!-- Theme style -->

<link rel="stylesheet" href="~/content/client/dist/css/adminlte.min.css">

<!-- overlayScrollbars -->

<link rel="stylesheet" href="~/content/client/plugins/overlayScrollbars/css/OverlayScrollbars.min.css">

<!-- Daterange picker -->

<link rel="stylesheet" href="~/content/client/plugins/daterangepicker/daterangepicker.css">

<!-- summernote -->

<link rel="stylesheet" href="~/content/client/plugins/summernote/summernote-bs4.css">

<!-- Google Font: Source Sans Pro -->

<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700" rel="stylesheet">

</head>

<body class="hold-transition sidebar-mini layout-fixed">

<div class="wrapper">

<!-- Navbar -->

<nav class="main-header navbar navbar-expand navbar-white navbar-light">

<!-- Left navbar links -->

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars"></i></a>

</li>

<li class="nav-item d-none d-sm-inline-block">

<a href="index3.html" class="nav-link">Home</a>

</li>

<li class="nav-item d-none d-sm-inline-block">

<a href="#" class="nav-link">Contact</a>

</li>

</ul>

<!-- SEARCH FORM -->

<form class="form-inline ml-3">

<div class="input-group input-group-sm">

<input class="form-control form-control-navbar" type="search" placeholder="Search" aria-label="Search">

<div class="input-group-append">

<button class="btn btn-navbar" type="submit">

<i class="fas fa-search"></i>

</button>

</div>

</div>

</form>

<!-- Right navbar links -->

<ul class="navbar-nav ml-auto">

<!-- Messages Dropdown Menu -->

<li class="nav-item dropdown">

<a class="nav-link" data-toggle="dropdown" href="#">

<i class="far fa-comments"></i>

<span class="badge badge-danger navbar-badge">3</span>

</a>

<div class="dropdown-menu dropdown-menu-lg dropdown-menu-right">

<a href="#" class="dropdown-item">

<!-- Message Start -->

<div class="media">

<img src="dist/img/user1-128x128.jpg" alt="User Avatar" class="img-size-50 mr-3 img-circle">

<div class="media-body">

<h3 class="dropdown-item-title">

Brad Diesel

<span class="float-right text-sm text-danger"><i class="fas fa-star"></i></span>

</h3>

<p class="text-sm">Call me whenever you can...</p>

<p class="text-sm text-muted"><i class="far fa-clock mr-1"></i> 4 Hours Ago</p>

</div>

</div>

<!-- Message End -->

</a>

<div class="dropdown-divider"></div>

<a href="#" class="dropdown-item">

<!-- Message Start -->

<div class="media">

<img src="dist/img/user8-128x128.jpg" alt="User Avatar" class="img-size-50 img-circle mr-3">

<div class="media-body">

<h3 class="dropdown-item-title">

John Pierce

<span class="float-right text-sm text-muted"><i class="fas fa-star"></i></span>

</h3>

<p class="text-sm">I got your message bro</p>

<p class="text-sm text-muted"><i class="far fa-clock mr-1"></i> 4 Hours Ago</p>

</div>

</div>

<!-- Message End -->

</a>

<div class="dropdown-divider"></div>

<a href="#" class="dropdown-item">

<!-- Message Start -->

<div class="media">

<img src="dist/img/user3-128x128.jpg" alt="User Avatar" class="img-size-50 img-circle mr-3">

<div class="media-body">

<h3 class="dropdown-item-title">

Nora Silvester

<span class="float-right text-sm text-warning"><i class="fas fa-star"></i></span>

</h3>

<p class="text-sm">The subject goes here</p>

<p class="text-sm text-muted"><i class="far fa-clock mr-1"></i> 4 Hours Ago</p>

</div>

</div>

<!-- Message End -->

</a>

<div class="dropdown-divider"></div>

<a href="#" class="dropdown-item dropdown-footer">See All Messages</a>

</div>

</li>

<!-- Notifications Dropdown Menu -->

<li class="nav-item dropdown">

<a class="nav-link" data-toggle="dropdown" href="#">

<i class="far fa-bell"></i>

<span class="badge badge-warning navbar-badge">15</span>

</a>

<div class="dropdown-menu dropdown-menu-lg dropdown-menu-right">

<span class="dropdown-item dropdown-header">15 Notifications</span>

<div class="dropdown-divider"></div>

<a href="#" class="dropdown-item">

<i class="fas fa-envelope mr-2"></i> 4 new messages

<span class="float-right text-muted text-sm">3 mins</span>

</a>

<div class="dropdown-divider"></div>

<a href="#" class="dropdown-item">

<i class="fas fa-users mr-2"></i> 8 friend requests

<span class="float-right text-muted text-sm">12 hours</span>

</a>

<div class="dropdown-divider"></div>

<a href="#" class="dropdown-item">

<i class="fas fa-file mr-2"></i> 3 new reports

<span class="float-right text-muted text-sm">2 days</span>

</a>

<div class="dropdown-divider"></div>

<a href="#" class="dropdown-item dropdown-footer">See All Notifications</a>

</div>

</li>

<li class="nav-item">

<a class="nav-link" data-widget="control-sidebar" data-slide="true" href="#" role="button">

<i class="fas fa-th-large"></i>

</a>

</li>

</ul>

</nav>

<!-- /.navbar -->

<!-- Main Sidebar Container -->

<aside class="main-sidebar sidebar-dark-primary elevation-4">

<!-- Brand Logo -->

<a href="index3.html" class="brand-link">

<img src="#" alt="AdminLTE Logo" class="brand-image img-circle elevation-3"

style="opacity: .8">

<span class="brand-text font-weight-light">AdminLTE 3</span>

</a>

<!-- Sidebar -->

@Html.Partial("\_PartialViewSideBar");

<!-- /.sidebar -->

</aside>

<!-- Content Wrapper. Contains page content -->

<div class="content-wrapper">

@RenderBody()

</div>

<!-- /.content-wrapper -->

<footer class="main-footer">

<strong>Copyright &copy; 2014-2019 <a href="http://adminlte.io">AdminLTE.io</a>.</strong>

All rights reserved.

<div class="float-right d-none d-sm-inline-block">

<b>Version</b> 3.0.4

</div>

</footer>

<!-- Control Sidebar -->

<aside class="control-sidebar control-sidebar-dark">

<!-- Control sidebar content goes here -->

</aside>

<!-- /.control-sidebar -->

</div>

<!-- ./wrapper -->

<!-- jQuery -->

<script src="~/content/client/plugins/jquery/jquery.min.js"></script>

<!-- jQuery UI 1.11.4 -->

<script src="~/content/client/plugins/jquery-ui/jquery-ui.min.js"></script>

<!-- Resolve conflict in jQuery UI tooltip with Bootstrap tooltip -->

<script>

$.widget.bridge('uibutton', $.ui.button)

</script>

<!-- Bootstrap 4 -->

<script src="~/content/client/plugins/bootstrap/js/bootstrap.bundle.min.js"></script>

<!-- ChartJS -->

<script src="~/content/client/plugins/chart.js/Chart.min.js"></script>

<!-- Sparkline -->

<script src="~/content/client/plugins/sparklines/sparkline.js"></script>

<!-- JQVMap -->

<script src="~/content/client/plugins/jqvmap/jquery.vmap.min.js"></script>

<script src="~/content/client/plugins/jqvmap/maps/jquery.vmap.usa.js"></script>

<!-- jQuery Knob Chart -->

<script src="~/content/client/plugins/jquery-knob/jquery.knob.min.js"></script>

<!-- daterangepicker -->

@\*<script src="~/content/client/plugins/moment/moment.min.js"></script>\*@

<script src="~/content/client/plugins/daterangepicker/daterangepicker.js"></script>

<!-- Tempusdominus Bootstrap 4 -->

<script src="~/content/client/plugins/tempusdominus-bootstrap-4/js/tempusdominus-bootstrap-4.min.js"></script>

<!-- Summernote -->

<script src="~/content/client/plugins/summernote/summernote-bs4.min.js"></script>

<!-- overlayScrollbars -->

<script src="~/content/client/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js"></script>

@\*<script src="https://cdnjs.cloudflare.com/ajax/libs/autonumeric/4.10.5/autoNumeric.min.js" integrity="sha512-EGJ6YGRXzV3b1ouNsqiw4bI8wxwd+/ZBN+cjxbm6q1vh3i3H19AJtHVaICXry109EVn4pLBGAwaVJLQhcazS2w==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>\*@

<script src="~/Content/Client/dist/js/autoNumeric-1.9.45.js"></script>

<!-- AdminLTE App -->

<script src="~/content/client/dist/js/adminlte.js"></script>

<!-- AdminLTE dashboard demo (This is only for demo purposes) -->

<script src="~/content/client/dist/js/pages/dashboard.js"></script>

<!-- AdminLTE for demo purposes -->

<script src="~/content/client/dist/js/demo.js"></script>

<script src="~/Content/ckeditor/ckeditor.js"></script>

<script src="~/Content/ckfinder/ckfinder.js"></script>

@RenderSection("Js", required: false)

</body>

</html>

# KẾT LUẬN

**KIẾN THỨC**

* Thu thập được nhiều thông tin để đặt tả nhiều yêu cầu của khách hàng cũng như là của hệ thống
* Thông qua các công nghệ phần mềm như trello và gitbub mà ta có thể dễ dàn phân công và kiểm soát công việc một cách hiệu quả
* Ứng dụng Unit Test để kiểm thử các phần mềm. Tuy chưa được áp dụng vào bài thực hành trên web form Có thể đưa ra các tiêu chí cần có một cách cơ bản của yêu cầu phần mềm

**THỰC HÀNH**

* Mặt dù chưa thông thạo các toolbox của visual nhưng cũng đã hoàn thành cơ bản giao diện phần mềm
* Liên kết được với CSDL, sử dụng thư viện
* Linq Mong muốn phần mềm phát triển thêm

TÀI LIỆU THAM KHẢO

[1] Bài giảng công nghệ phần mềm PTIT- Trường đại học Công nghệ Bưu chính Viễn thông. Ngày xuất bản: 7/10/2020

[2] ChatGPT

[3] Youtube

[4] W3School.com

BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Job/Name | Như Ý | Lực | Khả | Nhật Huy | Thịnh |
| Trello | ✓ |  |  |  | ✓ |
| GitHub |  |  | ✓ |  |  |
| Unit Test |  | ✓ |  |  |  |
| Integration Test |  | ✓ |  |  |  |
| System Test |  |  |  | ✓ |  |
| Upload Host |  | ✓ |  |  |  |
| Các sơ đồ thiết kế |  |  |  | ✓ |  |
| Word |  |  |  | ✓ |  |
| PowerPoint | ✓ |  | ✓ |  | ✓ |