**Xây dựng ứng dụng điểm danh trên hệ sinh thái mini App Zalo cho khoa CNTT STU**

1. **Mục tiêu đặt ra**

Với việc sự phát triển mạnh mẽ của nền tảng Zalo cùng với đó là sự cần thiết về mặt bổ sung công nghệ cho trường Đại học Công nghệ Sài Gòn. Nhóm em quyết định xây dựng ứng dụng nhỏ về đề tài Xây dựng ứng dụng điểm danh trên hệ sinh thái mini App Zalo cho khoa CNTT STU trên nền tảng Zalo App.

Mục đích chính là tối ưu hiệu suất điểm danh, tích hợp công nghệ vào việc điểm danh sinh viên thay thế các phương pháp thủ công như: Ký tên, gọi tên, … Với việc điểm danh thông qua ứng dụng nhóm em mong muốn tạo ra một Mini Zalo App với một phương pháp điểm danh mới thay thế hoàn toàn phương pháp thủ công. Bên cạnh đó cũng dễ dàng quản lý và sử dụng cho cả giảng viên và sinh viên.

1. **Cách tiến hành**

Xây dựng một nhóm sinh viên khoảng tầm 7 – 9 bạn có kiến thức cũng như mong muốn chung đáp ứng đủ về kỹ thuật, tư duy, năng lực. Các bạn trong nhóm sẽ cùng nhau thảo luận đề ra phương án giải quyết vấn đề và cùng với nhau đi giải quyết vấn đề cùng với nhau đi đến cùng mục tiêu chung là tạo ra Zalo Mini App cho Khoa CNTT STU.

Danh sách thành viên:

1. Nguyễn Trọng Phụng – DH52111563. Chức vụ: Nhóm trưởng
2. Văn Đình Thuật – DH52111857. Chức vụ: Thành viên
3. Phạm Quốc Lân – DH52111201. Chức vụ: Thành viên
4. Lê Trần Trọng Phúc – DH52111529. Chức vụ: Thành viên
5. Đặng Thành Hải – DH52108549. Chức vụ: Thành viên
6. Lê Anh Tài – DH52111681. Chức vụ: Thành viên
7. Nguyễn Thanh Trường – DH52108640. Chức vụ: Thành viên
8. **Kế hoạch triển khai**
9. Mô hình phát triển

Vì đây là 1 dự án được xây dựng trong thời gian ngắn cũng như cần độ chính xác cao nên chúng ta chọn mô hình phát triển Scrum. Bên cạnh đó sử dụng các ứng dụng hỗ trợ giải quyết vấn đề làm việc nhóm như Bitbucket, Trello, ...

1. Tổng quan về Scrum

Scrum là một mô hình phát triển phần mềm theo phương pháp Agile, nổi bật với sự linh hoạt, khả năng thích ứng cao và tập trung vào giá trị người dùng. Được thiết kế để quản lý các dự án phức tạp, Scrum giúp các nhóm phát triển làm việc hiệu quả thông qua chu kỳ ngắn gọi là Sprint – nơi sản phẩm được xây dựng, kiểm thử và cải tiến liên tục. Với sự kết hợp giữa các vai trò cụ thể (Product Owner, Scrum Master, Development Team), các sự kiện chuẩn hóa và quy trình minh bạch, Scrum tạo điều kiện cho nhóm làm việc tự tổ chức, phản hồi nhanh với thay đổi và tối ưu hóa chất lượng sản phẩm qua từng giai đoạn. Đây là mô hình lý tưởng cho những môi trường phát triển có yêu cầu thay đổi thường xuyên và đề cao sự hợp tác.

Đặc điểm của Scrum:

* Các công việc được thực hành liên tục và đòi hỏi sự tập trung cao để hoàn thành đúng tiến độ.
* Bên cạnh đó yếu tố làm việc nhóm khá quan trọng vì với mô hình này việc làm việc giữa các cá nhân với nhau là diễn ra liên tục
* Scrum không phải là một phương pháp đầy đủ, mà nó cung cấp cấu trúc khung để nhóm tự tổ chức phát triển phần mềm.
* Phù hợp với các dự án đòi hỏi về mặt thời gian.

1. Vai trò ( Scrum Roles )

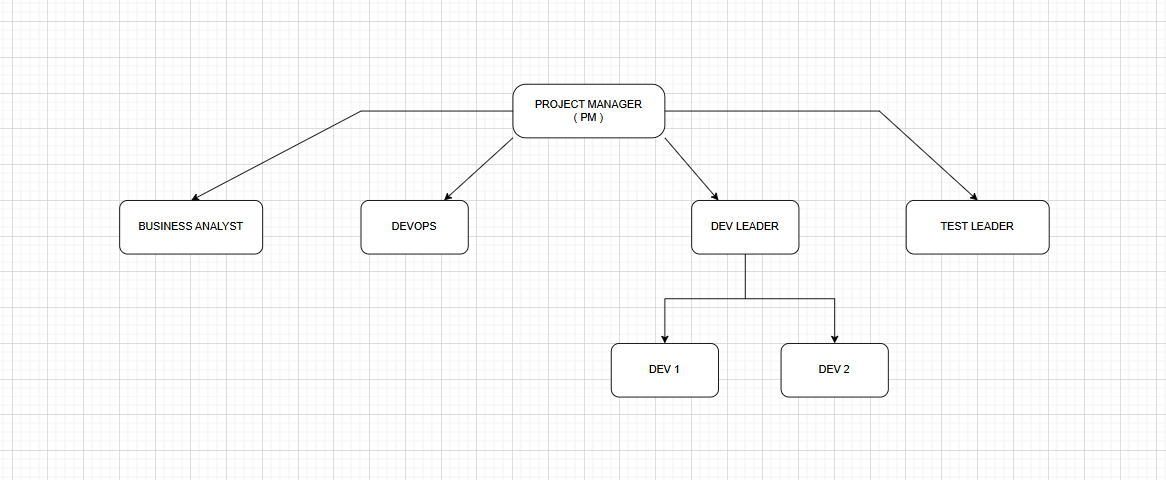
|  |  |
| --- | --- |
| Vai trò | Mô tả |
| Product Owner | |  | | --- | |  |  |  | | --- | | Đại diện cho khách hàng, chịu trách nhiệm về Product Backlog, xác định các tính năng cần phát triển và thứ tự ưu tiên. | |
| Scrum Master | Là người hỗ trợ nhóm phát triển, loại bỏ rào cản, giúp nhóm hiểu và áp dụng Scrum đúng cách. |
| Development Team | Nhóm kỹ thuật, thường từ 3-9 người, thực hiện phát triển, kiểm thử và bàn giao phần mềm. |

1. Sự kiện

|  |  |  |
| --- | --- | --- |
| Sự kiện | Thời điểm | Mục đích |
| Sprint Planning | Đầu mỗi Sprint | Lập kế hoạch cho Sprint: Chọn các hạng mục từ Product Backlog để đưa vào Sprint Backlog |
| Daily Scrum | Mỗi ngày | Cuộc họp ngắn ( 10-15p ), mỗi thành viên trả lời: Hôm qua làm gì? Hôm nay làm gì? Có vướng mắc gì không? |
| Sprint | 1 tuần | Là chu kỳ phát triển phần mềm. Mỗi sprint cho ra một bản phần mềm hoàn chỉnh có thể chạy được. |
| Sprint Review | Cuối mỗi Sprint | Nhóm dev trình bày sản phẩm đã hoàn thành với các bên liên quan |
| Sprint Restrospective | Sau Sprint Review | Nhóm thảo luận những gì đã làm tốt, chưa tốt và cách cải tiến cho Sprint tiếp theo |

1. Phân công vai trò

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ và tên, vai trò** | **Nội dung công việc** |
| 1 | Nguyễn Trọng Phụng  (PM, BA) | - Quản lý tiến độ dự án, phân công công việc cho các thành viên  - Làm việc với khách hàng để thu thập, phân tích yêu cầu  - Theo dõi, đánh giá kết quả công việc, đảm bảo đúng deadline |
| 2 | Văn Đình Thuật  (DevOps) | - Thiết lập và quản lý môi trường triển khai (CI/CD) - Cấu hình server, cơ sở hạ tầng dự án - Đảm bảo an toàn và hiệu năng hệ thống - Hỗ trợ kỹ thuật triển khai và quản lý mã nguồn |
| 3 | Phạm Quốc Lân  (Leader Dev) | - Thiết kế kiến trúc hệ thống, phân chia module - Hướng dẫn, hỗ trợ các developer trong nhóm - Code các chức năng quan trọng, tối ưu hiệu suất - Review code và đảm bảo chất lượng mã nguồn |
| 4 | Đặng Thành Hải  (Developer) | - Phát triển các tính năng frontend/backend theo phân công từ Leader Dev - Viết mã nguồn đúng chuẩn, dễ bảo trì và mở rộng - Tích hợp API, xử lý dữ liệu và tối ưu hiệu suất - Thực hiện unit test, phối hợp tester để fix lỗi và đảm bảo chất lượng sản phẩm |
| 5 | Nguyễn Thành Trường  (Developer) |
| 6 | Lê Trần Trọng Phúc  (Leader Tester) | - Xây dựng kế hoạch kiểm thử (test plan) - Phân công công việc cho các tester (nếu có) - Viết testcase, thực hiện test chức năng, UI, hiệu năng - Báo cáo lỗi và phối hợp với dev để khắc phục |
| 7 | Lê Anh Tài  (UI/UX Designer) | - Thiết kế giao diện người dùng (Figma, Adobe XD, v.v.) - Đảm bảo tính thẩm mỹ, trải nghiệm người dùng tốt - Phối hợp với BA và Dev để nắm yêu cầu và truyền tải vào thiết kế - Bảo đảm thống nhất UI toàn hệ thống |



1. **Các quy định**
2. Convention
3. Tên Ứng Dụng/Dự Án

Dùng danh từ, chữ hoa đầu từ (PascalCase), trừ trường hợp đặc biệt (iCloud).

Không dùng khoảng trắng, dấu tiếng Việt, hoặc ký tự đặc biệt.

Ví dụ: StudentManagement, HelloWorld, PetCareSystem

1. Tên Package/Thư Mục

Dùng chữ thường, danh từ, cố gắng sử dụng từ đơn, các thư mục con phân cách bởi dấu chấm.

Không dùng khoảng trắng, dấu tiếng Việt, hoặc ký tự đặc biệt.

Ví dụ: com.apple.quicktime, util, data

1. Tên Class và Interface

Dùng danh từ, chữ hoa từng đầu từ (PascalCase).

Viết đầy đủ từ, tránh viết tắt.

Ví dụ: Dog, Cat, File, Student, Person

1. Tên Biến

Dùng camelCase, bắt đầu bằng chữ thường.

Đặt tên có ý nghĩa, không viết tắt trừ các biến trung gian (i, j, t, tmp).

Ví dụ: luongCoBan, dienTich, chuVi, basicSalary, yearOfBirth

1. Tên Hằng Số

Dùng UPPER\_CASE, các từ cách nhau bằng dấu gạch dưới \_.

Ví dụ: MAX\_SPEED, PI

1. Tên Hàm/Method

Dùng camelCase, bắt đầu bằng động từ mô tả hành động.

Ví dụ: tinhLuong(), computeArea(), getSalary(), showInfo(), sortByName()

1. Quy tắc Commit Git

Không được trực tiếp commit thẳng vào nhánh master. Mà cần được review code mới đc thêm vào

Các thành viên chỉ được quyền commit lên 1 nhánh riêng trước khi được merge vào nhánh chính

Cấu Trúc Commit Message

[Module] Mô tả ngắn gọn thay đổi

Ví dụ:

[Backend] Thêm API lấy danh sách sinh viên

[Frontend] Cập nhật giao diện điểm danh

[DevOps] Cấu hình CI/CD

Tiêu đề

Ngắn gọn, súc tích.

Mô tả rõ ràng thay đổi.

Sử dụng động từ ở thì hiện tại (ví dụ: "Add", "Fix", "Update").

Nội dung (Body)

Mô tả chi tiết hơn về thay đổi.

Giải thích lý do thay đổi.

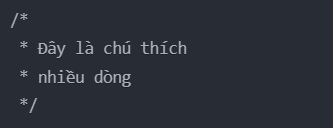
Liệt kê các thay đổi cụ thể.

1. Quy tắc comment

Chú Thích Một Dòng



Chú Thích Nhiều Dòng



Comment để làm rõ logic xử lý đoạn code phức tạp

Comment để lưu ý, cảnh báo những trường hợp có thể xảy ra trong code.

1. Quy tắc xuống hàng

Nếu một hàm có nhiều cấp lồng nhau, mỗi cấp nên được xuống dòng

Các đoạn code bằng cấp nên ở cùng một cột với nhau

1. Quy tắc đặt tên cho database:

Tên database nên mô tả nội dung của db.

Ví dụ:

sinh\_vien\_db (Cơ sở dữ liệu quản lý sinh viên)

dang\_ky\_hoc\_phan\_db (Cơ sở dữ liệu đăng ký học phần)

diem\_sinh\_vien\_db (Cơ sở dữ liệu điểm sinh viên)

1. Cấu trúc thư mục

/project-root

│── /src

│ │── /components # Chứa các component tái sử dụng

│ │── /pages # Chứa các trang chính của ứng dụng

│ │ │── /admin # Chứa các file liên quan đến trang Admin

│ │ │ │── dashboard.js

│ │ │ │── settings.js

│ │ │── /user # Chứa các file liên quan đến trang User

│ │ │ │── profile.js

│ │── /assets # Chứa các tài nguyên như hình ảnh, icon, fonts

│ │ │── /admin # Tài nguyên dành riêng cho Admin

│ │ │ │── images/

│ │ │ │── icons/

│ │ │── /user # Tài nguyên dành riêng cho User

│ │ │ │── images/

│ │ │ │── icons/

│ │ │── /global # Tài nguyên dùng chung

│ │ │ │── images/

│ │ │ │── icons/

│ │── /styles # Chứa file CSS/SCSS

│ │ │── /admin # Style riêng cho Admin

│ │ │ │── admin.css

│ │ │── /user # Style riêng cho User

│ │ │ │── user.css

│ │ │── global.css # Style chung

│ │ │── theme.css

│ │── /utils # Chứa các hàm tiện ích, helpers

│ │── main.js # File JavaScript chính

│── /public # Chứa các file tĩnh có thể truy cập trực tiếp

│ │── /admin # File tĩnh dành riêng cho Admin

│ │ │── css/

│ │ │── js/

│ │── /user # File tĩnh dành riêng cho User

│ │ │── css/

│ │ │── js/

│ │── /global # File tĩnh dùng chung

│ │ │── css/

│ │ │── js/

│ │── index.html

│── package.json # File cấu hình dự án

│── README.md # Hướng dẫn dự án

1. Nội quy
2. Quy tắc làm việc

* Mỗi thành viên có trách nhiệm cập nhật, theo dõi tiến độ dự án trên Trello
* Mỗi sáng hàng ngày sẽ có 1 cuộc họp ngắn từ 15-10p để báo cáo về tình hình công việc
* Tuân thủ deadline: Mọi thành viên cần bám sát tiến độ công việc được giao.
* Mỗi thành viên cần chịu trách nhiệm với phần công việc được giao. Nếu có tình huống khẩn cấp hay trục trặc cần báo cáo gấp cho PM
* Luôn cập nhật tiến độ trên hệ thống quản lý công việc Trello

1. Quy tắc giao tiếp

* Tôn trọng ý kiến đồng đội, tranh luận trên tinh thần xây dựng
* Không tự ý thay đổi code của người khác mà không thảo luận trước
* Nếu thực sự cần thiết hay có các ý kiến mang tính cá nhân, trao đổi trên Zalo hoặc Facebook cho PM

1. Chế tài vi phạm

* Vi phạm quy tắc code, nội quy bảo mật: Cảnh cáo lần 1.
* Vi phạm lần 2: Báo cáo lên PM và xem xét trừ điểm
* Vi phạm lần 3: Loại khỏi dự án