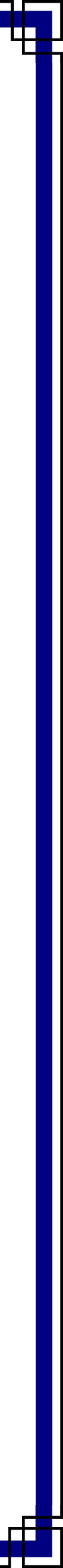
 **BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO**

**CÔNG CỤ VÀ MÔI TRƯỜNG PHÁT TRIỂN PHẦN MỀM**

**ĐỀ TÀI: WEBSITE GAME ĐẶT BOOM**

Công cụ hỗ trợ: Github & Jira

Sinh viên thực hiện:

**Đặng Hoàng Xuân Thái - 1911060989**

**Ngô Minh Tâm - 1911066348**

**Nguyễn Tấn Thái - 1911061864**

**Giảng viên hướng dẫn:** **Ths. Nguyễn Chí Toàn**

*TP. Hồ Chí Minh, 2023*

**LỜI CẢM ƠN**

Truớc tiên với tình cảm sâu sắc và chân thành nhất, cho phép chúng em đuợc bày tỏ lòng biết ơn đến tất cả thầy cô và nhà trường 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 Thông Tin 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 ThS.Nguyễn Chí Toàn - 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 hiện trong khoảng thời gian không nhiều buớc 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 kiến thức của mình.

Chúng em xin chân thành cảm ơn!

**Sinh viên thực hiện**

Đặng Hoàng Xuân Thái

Ngô Minh Tâm

Nguyễn Tấn Thái

**MỤC LỤC**

**[Chương 1. TỔNG QUAN 4](#_Toc132183181)**

**[1.1 Đề tài nhóm – Phân công công việc 4](#_Toc132183182)**

**[Chương 2. TÌM HIỂU VỀ JIRA 5](#_Toc132183183)**

**[2.1 Jira là gì? 5](#_Toc132183184)**

**[2.2 Các tính năng chính và thành phần cơ bản của Jira 6](#_Toc132183185)**

**[2.3 Mô hình dự án trong Jira 7](#_Toc132183186)**

**[2.4 Lợi ích của Jira 12](#_Toc132183187)**

**[Chương 3: GITHUB 15](#_Toc132183188)**

**[3.1 GitHub là gì? 15](#_Toc132183189)**

**[3.2 Các tính năng chính của GitHub? 16](#_Toc132183190)**

**[3.3 Một số lợi ích của GitHub 16](#_Toc132183191)**

**[3.4 Các thuật ngữ cơ bản của GitHub 17](#_Toc132183192)**

**[Chương 4: KẾT QUẢ THỰC NGHIỆM 19](#_Toc132183193)**

**[4.1 Kết quả đạt được 19](#_Toc132183194)**

**[Chương 5: TỔNG KẾT 21](#_Toc132183195)**

**[5.1 Kết quả đạt được 21](#_Toc132183196)**

**[5.2 Đánh giá 21](#_Toc132183197)**

**[TÀI LIỆU THAM KHẢO 22](#_Toc132183198)**

# **Chương 1. TỔNG QUAN**

* 1. **Đề tài nhóm – Phân công công việc**
     1. **Lý do chọn đề tài:**
* Dựa trên thực tế môn học cũng như những kiến thức tiếp thu trên lớp học. Nhóm em đã quyết định chọn đề tài làm web đặt boom và sử dụng công cụ Jira và Github để quản lý dự án này.
* Jira có khả năng cung cấp nhiều tính năng đa dạng như quản lý, theo dõi tiến độ đồ án, tasks, bugs, cải tiến, tính năng mới hoặc bất kỳ vấn đề gì xảy ra.
* Github dùng để kiểm soát phiên bản, theo dõi các thay đổi trong tập tin, nó dùng để thay đổi trong các tập tin bất kỳ. Nhanh chóng và thuận tiện.
* Vì những công dụng hữu ích của 2 công cụ nhóm em đã quyết định sử dụng 2 công cụ này để hoàn thiện đồ án của nhóm.
  + 1. **Phân công công việc:**

|  |  |  |
| --- | --- | --- |
| **Sinh viên** | **MSSV** | **Công việc** |
| Đặng Hoàng Xuân Thái | 1911060989 | Word, thuyết trình, tìm tài liệu |
| Ngô Minh Tâm | 1911066348 | Sử dụng github, tìm tài liệu |
| Nguyễn Tấn Thái | 1911061864 | Sử dụng Jira, tìm tài liệu |

# **Chương 2. TÌM HIỂU VỀ JIRA**

## **2.1 Jira là gì?**

****

Hình 2.1 logo Jira

• Jira là một ứng dụng theo dõi và quản lý lỗi / vấn đề trong dự án, được phát triển bởi công ty phần mềm Atlassian của Australia. Cách thức hoạt động của JIRA dựa vào trọng tâm là kết quả công việc, có thể sử dụng ngay và linh hoạt khi sử dụng.

• Cái tên Jira là sự cắt ngắn của Gojira (Godzilla trong tiếng Nhật). Tên này có nguồn gốc từ biệt hiệu mà các nhà phát triển Atlassian dùng để chỉ Bugzilla.

• Người sử dụng phần mềm có thể là một người chuyên quản lý dự án và làm việc với đội nhóm, các nhóm DevOps, bộ phận marketing, bộ phận hỗ trợ hoặc làm việc một mình. Jira chính là phần mềm quản lý dự án không thể bỏ qua.

• Jira có tích hợp với các nhóm như:

o Nhóm thiết thế: AdobeXD, Invision, Figma, Glyffy, Draw.io, Balsamiq, Lucidchart, I look. o Nhóm CNTT: Opsgenie, Jenkins, Dyratrace, Github, Slack, Zendesk. Scriptrunner.

o Nhóm kinh doanh: Trello, Excel, Gmail, Microsoft Outlook, Microsoft Team, Google trang tính, Confluence.

o Nhóm kĩ thuật phần mềm: Bitbucket Pipelines, Optimizely, LaunchDarkly, CircleCI, Zephyr.

• Jira được cung cấp trong 4 gói:

o Jira Work Management: nhằm mục đích quản lý dự án chung.

o Jira Software: bao gồm phần mềm cơ sở, các tính năng quản lý dự án.

o Jira Service Management: được thiết kế để sử dụng cho các hoạt động CNTT hoặc dịch vụ kinh doanh.

o Jira Align: dành cho quản lý chiến lược sản phẩm và danh mục đầu tự.

• Bảo mật và quyền riêng tự theo tiêu chuẩn ISO 27001, ISO 27018, CSA, SOC, PCI DSS, GDPR, FedRAMP.

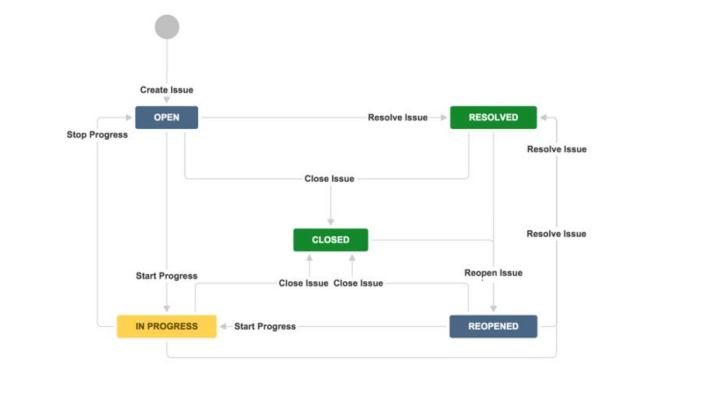
## **2.2 Các tính năng chính và thành phần cơ bản của Jira**

**2.2.1 Tính năng chính của Jira**

* Quản lý, theo dõi tiến độ dự án
* Quản lý lỗi, tính năng, công việc, những cải tiến hoặc bất kỳ vấn đề gì
* Tìm kiếm nhanh chóng với bộ lọc Jira Query Language
* Xây dựng quy trình làm việc tương thích với yêu cầu của từng dự án
* Cung cấp nhiều loại báo cáo thống kê với rất nhiều loại biểu đồ khác nhau phù hợp với nhiều loại hình dự án, nhiều đối tượng người dùng
* Dễ dàng tích hợp với các hệ thống khác (như Email, Excel, RSS…)
* Có thể chạy trên hầu hết các nền tảng phần cứng, hệ điều hành và cơ sở dữ liệu.

**2.2.2 Các thành phần trong Jira**

* Roles: Xác lập các role của dự án, Mục này xác nhận ai tham gia vào dự án, những người add vào role thì mới có thể tạo Resource Allocation và project team sau này. Nhiều người có thể vào 1 role.
* Issue: là các tasks, các bugs, các features hay bất kỳ các type khác của project work
* Project: Chức năng này dùng để phân quyền approve worklog cho thành viên của dự án. Ai là team lead của group nào thì sẽ được approve worklog cho member của group đó. Người quản lý dự án được quyền approve cho toàn bộ thành viên dự án.
* Component là sản phẩm của dự án. Ở đây sẽ nhập tất cả sản phẩm của dự án lấy từ file kế hoạch doanh số. Nếu dự án làm theo Scrum thì sẽ là Product của Sprint tương ứng.
* Workflow: Là một quản trị JIRA, ta có thể cấu hình nên quy trình làm việc.
* Priority: Là mức độ ưu tiên của một defect. Có 4 mức, chọn theo datalist
* Status: Đại diện cho các vị trí của vấn đề trong workflow
* Resolution



Hình 2.2.2 Resolusion

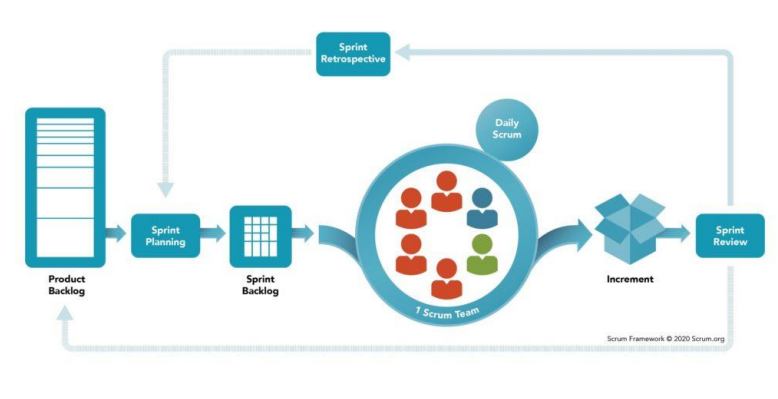
## **2.3 Mô hình dự án trong Jira**

Sau khi đăng nhập, ta sẽ bắt đầu thiết lập dự án. Dự án này có thể là bất cứ cái gì từ một item riêng lẻ cho tới một bộ tổng hợp các vấn đề mà team của chúng ta cần cộng tác với nhau. Mọi thứ trong JIRA là hoàn toàn tùy biến, phụ thuộc vào độ đơn giản hay phức tạp của từng dự án. Trước khi đi vào những mô hình trong Jira, ta cần hiểu một khái niệm đó là Agile - một phương pháp phát triển phần mềm linh hoạt, là một hướng tiếp cận cụ thể cho việc quản lý dự án phần mềm. Nó gồm một quá trình làm việc tương tác và tích hợp để có thể đưa sản phẩm đến tay người dùng càng nhanh càng tốt. Các mô hình trong Jira đều tuân thủ theo phƣơng pháp này.

**2.3.1 Scrum và các khái niệm cơ bản của Scrum:**

Trong số các framework (khung) của phương pháp Agile, Scrum được sử dụng phổ biến nhất.

Đặc trưng của Scrum là tách quá trình phát triển thành từng chu kỳ hoặc giai đoạn, được gọi là “sprint” (chạy nước rút). Các thành viên trong nhóm sẽ chia công việc của mình thành những mục tiêu cần đạt mỗi lần chạy nước rút.



Hình 2.1.3 Mô hình Scrum theo phương pháp Agile

Scrum giúp loại bỏ những công đoạn phức tạp và chỉ tập trung vào những công đoạn cần thiết đáp ứng được nhu cầu của khách hàng đưa ra. Ba yếu tố nòng cốt tạo thành một mô hình quản lý tiến hành thực nghiệm gồm: **Sự mình bạch** (transparency), **thanh tra** (inspection) và **thích nghi** (adaptation).

**2.3.1.1 Scrum Team**

Scrum team chia làm 3 vai trò bao gồm những thành phần sau:

* **Product Owner:** đảm bảo việc quản lý những công việc còn tồn đọng (**Product backlog**) của việc phát triển phần mềm.
* **Development Team:** là những lập trình viên sẽ tham gia vào việc phát triển từng tính năng cụ thể.
* **Scrum master:** chịu trách nhiệm lên kế hoạch để phân công sắp xếp công việc

**2.3.1.2 Sprint**

Sprint là mộ phân đoạn lặp đi lặp lại trong quy trình phát triển phần mềm, có khung thời gian thường là 1 tháng (từ 1 – 4 tuần) mà theo đó sản phẩm sẽ được release phiên bản 13 mới. Khi một Sprint kết thúc thì Scrum Master cần phải chuyển trạng thái của nó sang Done. Khi bắt đầu một Sprint thì Scrum Master cần đưa ra mục tiêu của Sprint đó và mục tiêu này không được phép thay đổi cho tới khi Sprint hoàn thành. Tuy nhiên Product Owner vẫn có quyền huỷ một Sprint trước thời hạn kết thúc của nó.

**2.3.1.3 Sprint Planning**

Đây là bước đầu tiên cần phải thực hiện trước khi một Sprint bắt đầu. Development team họp với Product Owner để lên kế hoạch cho một sprint. Những công việc nào cần phải được hoàn thành trong Sprint này và làm sao để có thể hoàn thành những công việc này. Sau khi thống nhất được số lượng công việc, thời gian hoàn thành thì chúng ta có thể bắt đầu Sprint. Trong khi thực hiện một Sprint chúng ta sẽ phải có những buổi họp được gọi là Daily Sprint hay Daily Meeting.

**2.3.1.4 Daily Sprint**

Các buổi họp Daily Sprint thường kéo dài khoản 15 phút, trong buổi họp này tất cả các thành viên sẽ lần lượt báo cáo lại:

* Những gì họ đã làm được ngày hôm qua
* Những gì họ cần làm ngày hôm nay
* Những khó khăn mà họ gặp phải

Mỗi buổi họp này sẽ giúp việc dự kiến được kế hoạch đưa ra trong Sprint đang làm sẽ tiến triển ra sao và liệu có cần phải cập nhật lại bản kế hoạch đã đưa ra hay không. Tất nhiên cần nhớ rằng việc thay đổi kế hoạch này không bao gồm thay đổi mục tiêu đã đưa ra của Sprint.

**2.3.1.5 Sprint Review**

Là công việc được thực hiện bởi nhóm phát triển và product owner ở cuối mối Sprint nhằm đánh giá lại kết quả thực hiện được. Từ lúc Sprint mới hoàn thành và qua đó đưa ra những chỉnh sửa, thay đổi cần thiết ở Sprint sau.

**2.3.1.6 Sprint Restrospective**

Dưới sự trợ giúp của Scrum master, team phát triển sẽ tổng kết những kiến nghị và đánh giá từ bước Sprint Review ở trên để đưa ra những cải tiến nhằm nâng cao hiệu quả làm việc cũng như sản phẩm.

**2.3.1.7 Các công cụ Scrum**

Scrum sử dụng các công cụ rất đơn giản nhưng hiệu quả để trợ giúp công việc.

**Product backlog**

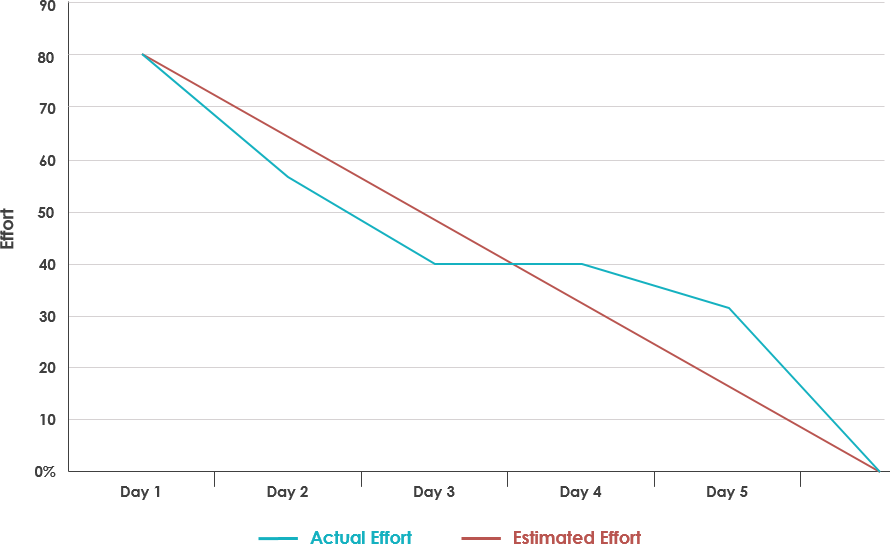
Đây là danh sách ưu tiên các tính năng (feature) hoặc đầu ra khác của dự án. Có thể hiểu như là danh sách yêu cầu (requirement) của dự án. Product Owner chịu trách nhiệm sắp xếp độ ưu tiên cho từng hạng mục (Product Backlog Item) trong Product Backlog dựa trên các giá trị do Product Owner định nghĩa (thường là giá trị thương mại – business value).

**Sprint backlog**

Đây là bản kế hoạch cho một Sprint là kết quả của buổi họp lập kế hoạch (Sprint Planning).

Với sự kết hợp của Product Owner, nhóm sẽ phân tích các yêu cầu theo độ ưu tiên từ cao xuống thấp để hiện thực hóa các hạng mục trong Product Backlog dưới dạng danh sách công việc (TODO list)

**Burndown Chart**



Hình 2.3.1.7 Burn Down Chart

Đây là biểu đồ hiển thị xu hướng của dự án dựa trên lượng thời gian cần thiết còn lại để hoàn tất công việc.

Burndown Chart có thể được dùng để theo dõi tiến độ của Sprint (được gọi là Sprint Burndown Chart) hoặc của cả dự án (Project Burndown Chart).

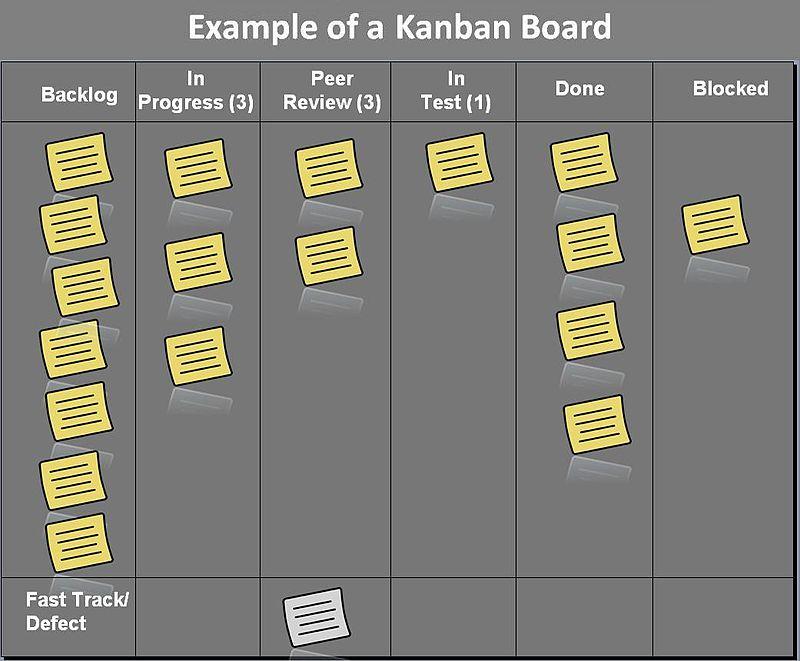
Biểu đồ burndown không phải là một thành tố tiêu chuẩn của Scrum theo định nghĩa mới, nhưng vẫn được sử dụng rộng rãi do tính hữu ích của nó.

**2.3.2 Kanban**

Phương pháp Kaban đòi hỏi sự giao tiếp và minh bạch trong công việc. Mục đích của nó là giúp thành viên trong nhóm có thể biết chính xác quá trình phát triển đang ở giai đoạn nào. Đồng thời, họ có thể nhìn thấy tình trạng của dự án bất cứ lúc nào.

**2.3.2.1 Lợi ích của kanban**

* Lập kế hoạch linh hoạt
* Chu kì thời gian làm việc được rút ngắn
* Ít tắc nghẽn hơn
* Số liệu trực quan
* Chuyển giao liên tục



Hình 2.3.2.1 Ví dụ về Kanban

## **2.4 Lợi ích của Jira**

**2.4.1 Các tính năng**

JIRA có những tính năng mà bạn không thể tìm thấy ở bất kỳ hệ thống quản lý và tìm kiếm các vấn đề (issues) nào khác.

- Dễ dàng tạo ra và lưu lại những bộ lọc có thể cấu hình cao ((dynamic queries) xuyên suốt mọi vấn đề trong hệ thống.

- Chia sẻ bộ lọc với người sử dụng khác, hoặc đăng ký và nhận được các kết quả qua hệ thống điện tử định kỳ.

- Kết nối vấn đề động cho phép bạn liên kết các vấn đề của các dự án, như lặp lại hoặc vấn đề liên đới.

- Bảng phân tích đồ họa (Dashboard) cung cấp cho mỗi người dùng một khoảng không riêng để xem mọi thông tin liên quan đến họ ngay tức thì.

- Các trường thuộc tính, tích hợp Excel, lộ trình của dự án, lưu sự thay đổi, REST API.

**2.4.2 Chính sách bản quyền trả trước hợp lí**

- Chính sách phí bản quyền của JIRA đảm bảo tất cả chi phí của bạn được đều được đự tính đầy đủ trước khi mua.

- Chính sách phí bản quyền theo máy chủ, truy cập mã nguồn, 12 tháng nâng cấp miễn phí và dịch vụ hỗ trợ tối đa.

**2.4.3 Được kiểm chứng bởi khách hàng lớn trên thế giới**

Jira được sử dụng bởi hàng nghìn tổ chức trên 55 quốc gia gồm các công ty thuộc Fortune 1000, công ty cổ phần, các tổ chức giáo dục và công ty thuộc lĩnh vực công nghệ.

- Tất cả các tổ chức từ nhỏ tới lớn đều thu được lới ích từ việc sử dụng đơn giản của JIRA nhưng là một công cụ mạnh mẽ để kiểm soát các vấn đề (issues).

**2.4.4 Đa dạng về cách sử dụng**

* Giám đốc cần cập nhật dự án
* Lập trình viên cần tác nghiệp, cập nhật và sửa lỗi
* Help desk cần theo dõi khách hàng suốt quá trình hoàn thành.
* Jira không chỉ đưa ra một hệ thống cấp phép mở rộng và luồng công việc có thể tùy biến, mà còn đưa ra khả năng cho mỗi người dùng tự cấu hình bảng phân tích đồ họa (dashboard) theo nhu cầu – đặt mọi thông tin họ cần hoàn toàn ngay tức thời.

**2.4.5 Tích hợp hệ thống hiện đại**

* Jira được phát triển sử dụng chuẩn HTML và được thử nghiệm trên toàn bộ các trình duyệt phổ biến hiện nay.
* Danh sách các vấn đề có thể được xem dưới dạng trình duyệt
* Mỗi màn hình trong jira có một phiên bản có thể in đảm bảo việc luân chuyển bản một cách dễ dàng
* Jira kết hợp với một số hệ thống kiểm soát mã nguồn phổ biến.

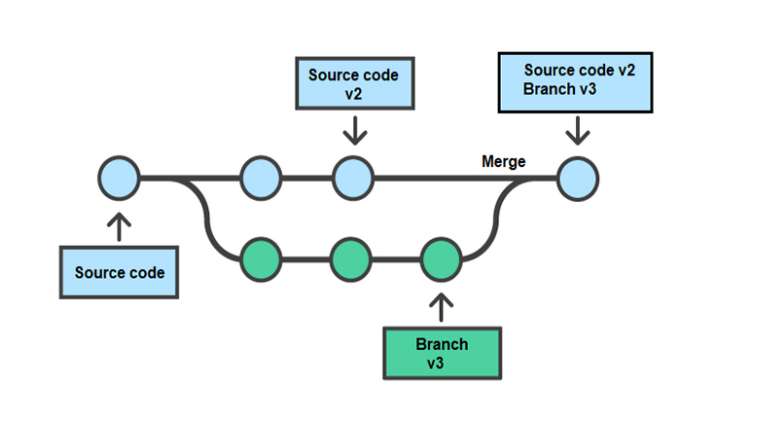
# **Chương 3: GITHUB**

## **3.1 GitHub là gì?**

*Hình 3.1 Logo GitHub*

* Đây là *một hệ thống quản lý dự án cũng như các phiên bản Code*. Nền tảng này được ví như mạng xã hội dành riêng cho các lập trình viên. Tại đó, coder được quyền clone lại mã nguồn từ một Repository.
* GitHub cũng đóng vai trò là một dịch vụ máy chủ Repository. Mục đích sử dụng để lưu trữ code, mỗi người có thể tạo tài khoản riêng và tạo ra kho chứa của mình. Nhờ thế, mỗi lần cần tái sử dụng, bạn nhanh chóng tìm thấy và vận dụng dễ dàng.
* Khi tìm hiểu GitHub là gì, bạn sẽ thấy hệ thống này có đầy đủ tính năng của Git. Bên cạnh đó là các đặc điểm bổ sung về Social giúp các nhà phát triển tương tác với nhau một cách thuận lợi.
* Người sử dụng có thể khai thác chức năng feeds, follower và network graph. Từ đó, mọi người học hỏi được kinh nghiệm từ lịch sử commit. Hiện nay, nền tảng này cung cấp hai hình thức là miễn phí và trả phí.
* Với phiên bản có phí, các doanh nghiệp công nghệ sử dụng để tăng khả năng quản lý đội nhóm. Đồng thời, Manager cũng phân quyền bảo mật dự án một cách dễ dàng khi hiểu GitHub là gì.
* Đối với phiên bản miễn phí, hầu hết người dùng đều sử dụng để lưu trữ source code.
* Có thể thấy rằng GitHub là nơi tuyệt vời để bạn tham khảo những nguồn code chất lượng. Từ đó, quá trình hoàn thành dự án nhanh chóng, các phần được tối ưu hơn.

## **3.2 Các tính năng chính của GitHub?**

****

*Hình 3.2 Mô tả tính năng của github*

* Có thể wiki, issue, thống kê, đổi tên project, project được đặt vào namespace là User
* GitHub cung cấp tính năng Watch Project cho người tham gia. Điều này nghĩa là theo dõi các hoạt động diễn ra trong dự án của người khác. Từ đó, bạn học hỏi được rất nhiều về cách phát triển phần mềm như thế nào.
* Tính năng Follow User cấp quyền theo dõi các hoạt động của người khác.
* Bạn có thể chọn một trong hai cách tiếp cận GitHub phổ biến nhất. Thứ nhất là hãy tạo Project của riêng mình và đóng góp vào dự án sẵn có. Tiếp đó là fork Project có sẵn của người khác, sửa đổi.

## **3.3 Một số lợi ích của GitHub**

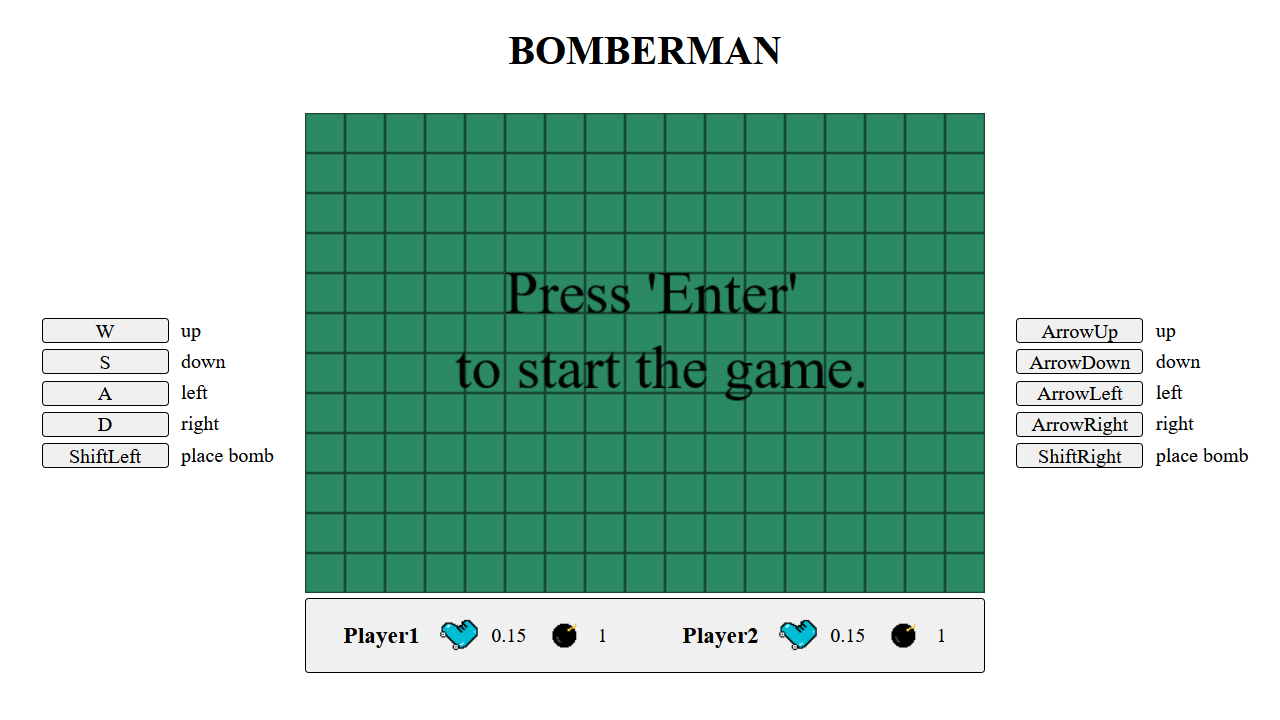
* Dễ dàng Markdown
* Có thể tracking về sự thay đổi qua nhiều version
* Quản lý source code cực kỳ dễ dàng

## **3.4 Các thuật ngữ cơ bản của GitHub**

|  |  |
| --- | --- |
| **Thuật ngữ** | **Giải thích** |
| **Git** | Đây là prefix của các lệnh được sử dụng dưới CLI. |
| **Branch** | Bạn coi đây là nhánh, thể hiện sự phân chia các Version |
| **Commit** | Đó là cách gọi tên của một điểm trên cây công việc Work Tree. |
| **Clone** | Trong Tiếng Việt, bạn có thể hiểu đây là nhân bản hoặc hành động thực hiện nhân bản. Bạn sử dụng với mục đích Clone dự án, Repository từ các hệ thống chạy trên cơ sở Git. Tiêu biểu như là Bitbucket, GitHub, GitLab,… Việc Clone sẽ giúp sao chép Repository tại Commit bạn mong muốn và dùng để tiếp tục phát triển. Các thao tác này sẽ tải toàn bộ mã nguồn, dữ liệu về máy tính của bạn. |
| **Folk** | Đây được hiểu là thao tác thực hiện sao chép Repository của chủ sở hữu khác về Git Account của mình. Sau đó, bạn sử dụng và đối xử như một Repository do mình tạo ra. |
| **Repository** | Quản lý dữ liệu, mã nguồn của Project. |
| **Tag** | Nhân tố này sử dụng để đánh dấu một Commit khi bạn có quá nhiều Commit tới độ không thể kiểm soát được. |
| **Remote** | Bạn sử dụng để điều khiển các nhánh từ một Repository trên Git Server. Cách đối xử với các nhánh trên Remote tương tự như với các nhánh xuất hiện trên Local. |
| **Diff** | Khái niệm muốn nói đến sự so sánh sai khác giữa phiên bản hiện tại và version cần đối chiếu. |

# **Chương 4: KẾT QUẢ THỰC NGHIỆM**

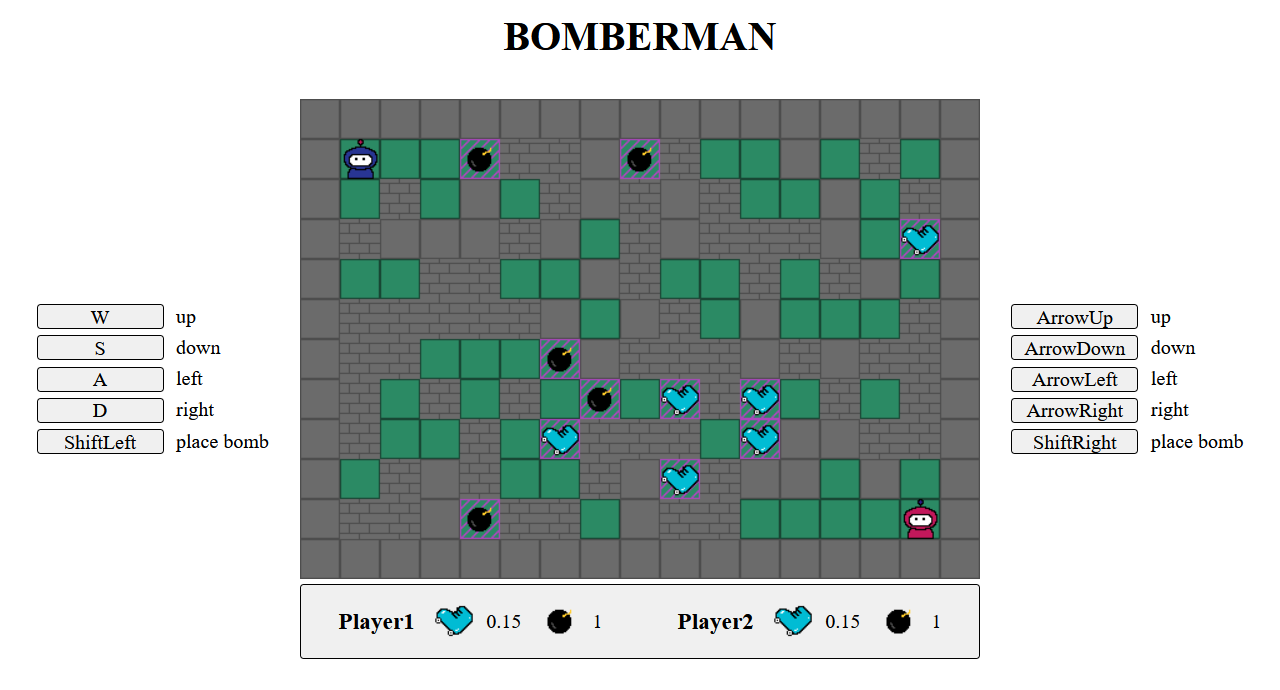
## **4.1 Kết quả đạt được**



Hình 4. 1. Giao diện màn hình chờ của game.

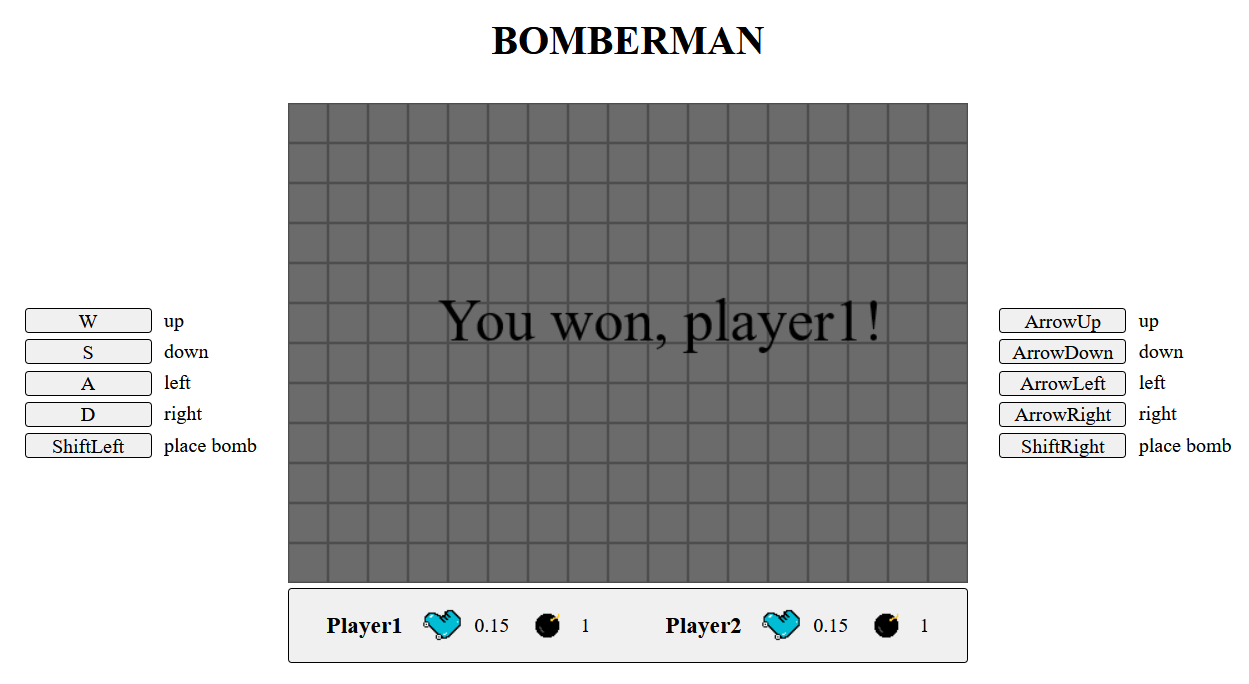
- Nhấn Enter để bắt đầu vào game

- Hiển thị các nút hướng dẫn để người dùng biết cách chơi



Hình 4. 2. Giao diện màn hình chính của game.

- Hiển thị 2 player, các vật phẩm, boom



Hình 4. 3. Thông báo.

- Khi bạn thắng thì sẽ có 1 thông báo hiện ra.

**4.1.1 Jira**

**4.1.1.1 Roadmap**

**Hình 4.1.1.1 Roadmap**

**4.1.1.2 Backlog**

**Hình 4.1.1.2**

**4.1.1.3 Board**

**Hình 4.1.1.3 Board**

**4.1.1.4 Manage workflow**

**Hình 4.1.1.4 Manage workflow**

**4.1.1.5 All Sprints**

**Hình 4.1.5 All Sprints**

**4.1.1.6 Access**

**Hình 4.1.1.6 Access**

**4.1.2 Github**

**4.1.2.1 Contributors**

**Hình 4.1.2.1 Contributors**

**4.1.2.2 Commits**

**Hình 4.1.2.2 Commits**

**4.1.2.3 Pull Request thành công**

**Hình 4.1.2.3 Pull Request thành công**

**4.1.2.4 Pull Request bị xung đột**

**Hình 4.1.2.4 Pull Request bị xung đột**

# **Chương 5: TỔNG KẾT**

## **5.1 Kết quả đạt được**

Qua việc nghiên cứu xây dựng đề tài này, một phần đã củng cố cho chúng em kiến thức về các công cụ quản lý source code cũng như công việc của từng thành viên trong nhóm một cách rõ ràng. Từ đó, chúng em có thể dễ dàng thực hiện các công việc được giao một cách nhanh chóng, chuyên nghiệp hơn để có thể áp dụng vào công việc sau này.

## **5.2 Đánh giá**

**5.2.1 Ưu điểm**

- Cơ bản.

- Thao tác dễ dàng nếu đã nắm rõ kiến thức.

- Chất lượng quản lý từng công việc rất tốt.

- Luôn thông báo rõ ràng cho từng thành viên thực hiện.

- Thời gian thực hiện công việc luôn cụ thể, rõ ràng.

**5.2.2 Nhược điểm**

- Sẽ rất khó thao tác nếu như chưa từng đọc, dùng qua.

- Khi thực hiện chắc chắn vẫn sẽ có lỗi.

- Sẽ bị xung đột nếu như sắp xếp sai thứ tự công việc.

# **TÀI LIỆU THAM KHẢO**

- https://viblo.asia/p/huong-dan-cach-su-dung-jira-don-gian-nhat-cho-developer- 6J3ZgOJqZmB

- https://fastdo.vn/jira-la-gi/

- https://codelearn.io/sharing/git-github-tu-co-ban-den-nang-cao-p1

- https://viblo.asia/p/nhung-lenh-git-co-ban-can-nho-V3m5W1OyZO7

- https://git-scm.com/doc

- https://www.youtube.com/playlist?list=PLp1Emx1rT4z-qrcEmliK7\_vyQGPLUGT\_l