|  |  |
| --- | --- |
| logomobile.png | BỘ GIÁO DỤC VÀ ĐÀO TẠO  TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM |

**ĐỒ ÁN MÔN HỌC**

**WEBSITE ĐÁNH CỜ TƯỚNG ONLINE**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên Ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn: **ThS.NGUYỄN CHÍ TOÀN**

Sinh viên thực hiện:

HOÀNG NGHĨA TUẤN MSSV: 1711062296

TẠ LÊ DUY KHANG MSSV: 1711061957

NGUYỄN PHƯƠNG HOÀNG MSSV:

LÊ QUỐC AN MSSV: 1711062249

NGUYỄN ĐOÀN GIA HƯNG MSSV: 1711062332

TRẦN QUAN DUY MSSV: 1711061903

TRƯƠNG LÂM HỮU LỘC MSSV: 1711062275

NGÔ TUẤN VŨ MSSV: 1711062086

TP.Hồ Chí Minh, 2021

|  |  |
| --- | --- |
| logomobile.png | BỘ GIÁO DỤC VÀ ĐÀO TẠO  TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM |

**ĐỒ ÁN MÔN HỌC**

**WEBSITE ĐÁNH CỜ TƯỚNG ONLINE**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên Ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn: **THS.NGUYỄN CHÍ TOÀN**

Sinh viên thực hiện:

HOÀNG NGHĨA TUẤN MSSV: 1711062296

TẠ LÊ DUY KHANG MSSV: 1711061957

NGUYỄN PHƯƠNG HOÀNG MSSV:

LÊ QUỐC AN MSSV: 1711062249

NGUYỄN ĐOÀN GIA HƯNG MSSV: 1711062332

TRẦN QUAN DUY MSSV: 1711061903

TRƯƠNG LÂM HỮU LỘC MSSV: 1711062275

NGÔ TUẤN VŨ MSSV: 1711062086

TP.Hồ Chí Minh, 2021

sMỤC LỤC

**[CHƯƠNG 1: GIỚI THIỆU 5](#_Toc3148)**

[1.1 WIREFRAME 5](#_Toc6072)

[1.2 JIRA 6](#_Toc18570)

[1.3 GIT.............................................................................................................................8](#_Toc18570)

[1.3.1 Định nghĩa Git là gì? 8](#_Toc3331)

[1.3.2 Những tính năng ưu việt của Git là gì so với SVN? 8](#_Toc2624)

[1.4 GITHUB 10](#_Toc1088)

[1.4.1 Github là gì?......................................................................................................10](#_Toc3085)

[1.4.2 Tính năng của Github 11](#_Toc29290)

[1.4.3 Một vài khái niệm của Git bạn cần nắm 11](#_Toc26738)

[1.4.4 Lợi ích của Github đối với lập trình viên 12](#_Toc25520)

[1.4.5 Github Package Registry 12](#_Toc4071)

**[CHƯƠNG 2: KẾT QUẢ 13](#_Toc20381)**

[2.1 Wireframe 13](#_Toc10458)

[2.2 Github 13](#_Toc25523)

[2.3 Jira 15](#_Toc28779)

[2.4 Website cờ tướng online 16](#_Toc8109)

LỜI CẢM ƠN

Trong thời gian làm đồ án môn công cụ và phát triển phần mềm, nhóm em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô bạn bè.

Nhóm em xin gửi lời cảm ơn chân thành đến Ths.Nguyễn Chí Toàn, giảng viên Bộ môn Công Cụ và Phát Triền Phần Mềm là người đã tận tình hướng dẫn, chỉ bảo nhóm em trong suốt quá trình làm đồ án.

Nhóm em cũng xin chân thành cảm ơn thầy đã dạy dỗ cho nhóm em kiến thức về các công cụ hỗ trợ cũng như giúp chúng em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ chúng em trong suốt quá trình học tập.

Cuối cùng, nhóm em xin chân thành cảm ơn Ths.Nguyễn Chí Toàn đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên nhóm em trong suốt quá trình học tập và hoàn thành tốt đồ án môn học.

|  |  |
| --- | --- |
|  |  |
|  | **Sinh Viên Thực Hiện** |

Hoàng Nghĩa Tuấn

Tạ Lê Duy Khang

Nguyễn Phương Hoàng

Nguyễn Đoàn Gia Hưng

Trần Quan Duy

Trương Lâm Hữu Lộc

Lê Quốc An

Ngô Tuấn Vũ

CHƯƠNG 1: GIỚI THIỆU

* 1. **WIREFRAME**

**[Wireframe](https://wireframe.cc/)** **(khung xương/cấu trúc dây) là một công cụ trực quan để**[thiết kế web](https://topdev.vn/blog/thiet-ke-giao-dien-web-bang-file-psd/)**ở cấp độ cấu trúc.** Một wireframe thường được sử dụng để bố trí nội dung và chức năng trên một trang. Wireframe còn được sử dụng trong quá trình thiết lập cấu trúc cơ bản của trang web trước khi thiết kế trực quan. Tất nhiên, quá trình ấy cần được tạo lập, bổ sung và hoàn thiện nội dung.



## **Ưu – nhược điểm của Wireframe**

### *Ưu điểm*

Wireframe cung cấp một hình ảnh tổng thể trong thời gian sớm nhất, được sử dụng để xem xét đàm phán với khách hàng. Nhìn nhận từ góc độ thực tế, các wireframe đảm bảo đầy đủ nội dung và chức năng của trang trên cơ sở định vị chính xác dựa trên nhu cầu của người dùng và doanh nghiệp. Đây cũng chính là lợi thế lớn nhất mà cấu trúc wireframe mang lại cho người dùng.

Khi phát triển dự án, chúng có vai trò thiết thực hỗ trợ sử dụng giữa các thành viên của một nhóm, giúp họ làm việc để thống nhất về tầm nhìn và phạm vi hoạt động của dự án.

### *Nhược điểm*

Vì các wireframe không bao gồm bất kỳ thiết kế hoặc yêu cầu kỹ thuật nào, nên không phải lúc nào khách hàng cũng dễ dàng nắm bắt được khái niệm/thuật ngữ này. Nhà thiết kế web cũng sẽ phải dịch các wireframe thành một thiết kế. Ngoài ra, cần có sự phối hợp giữa người thiết kế và copywriter khi làm việc với phần nội dung website cho phù hợp.

JIRA



Jira là một ứng dụng theo dõi và quản lý lỗi, vấn đề và dự án, được phát triển để làm quy trình này trở nên dễ dàng hơn cho mọi tổ chức. JIRA đã được thiết kế với trọng tâm vào kết quả công việc, có thể sử dụng ngay và linh hoạt khi sử dụng.

Tính năng chính của Jira:

* Quản lý, theo dõi tiến độ của 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

*Ưu điểm của Jira:*

* Jira có chức năng phân quyền cực kỳ chi tiết, không chỉ phân quyền trong dự án chung, mà còn phân quyền đối với từng nhiệm vụ, giúp team công nghệ bảo vệ thông tin độc quyền của mình.
* Dễ dàng tích hợp với các hệ thống khác (như Email, Excel, RSS,…)
* Hệ thống module và bộ công cụ phát triển bổ trợ cho phép tùy biến, mở rộng và tích hợp Jira vào trong hệ thống hiện tại
* Jira được phát triển sử dụng chuẩn HTML và được thử nghiệm với tất cả các trình duyệt phổ biến hiện nay.
* Có thể chạy trên hầu hết các nền tảng phần cứng, hệ điều hành và cơ sở dữ liệu
* Mỗi màn hình trong Jira có một phiên bản có thể in đảm bảo việc luân chuyển bản cứng một cách dễ dàng
* Có thể tích hợp trực tiếp với code trên môi trường phát triển, là một công cụ hoàn toàn phù hợp với các developer

*Nhược điểm của Jira:*

* Chi phí cao, sau 7 ngày dùng thử thì doanh nghiệp càng có quy mô lớn thì càng tốn nhiều chi phí: $10 mỗi tháng dành cho tối đa 10 tài khoản; từ 11-100 tài khoản là $7/tài khoản/tháng
* Tốn nhiều thời gian và công sức để setup nên chỉ phát huy tối ưu hiệu quả với dự án lớn, không phù hợp với dự án vừa và nhỏ (dưới 3 tháng)
* Ngôn ngữ tiếng Anh với nhiều thuật ngữ khó sử dụng
* Quy trình làm việc phức tạp đòi hỏi phải tìm hiểu kỹ lưỡng

# **1.3 GIT**



1.3.1 Định nghĩa Git là gì?

Git là một hệ thống quản lý phiên bản phân tán (distributed version control system). Nhờ Git, việc quản lý code và làm việc nhóm của developer trở nên đơn giản, thuận tiện hơn.

1.3.2 Những tính năng ưu việt của Git là gì so với SVN?

Nhờ tiếp cận theo hướng phân tán, Git mang đến những lợi ích vô cùng to lớn như hỗ trợ rất tốt cho teamwork, phân chia task, tổng hợp code trở nên dễ dàng hơn nhiều, cụ thể:

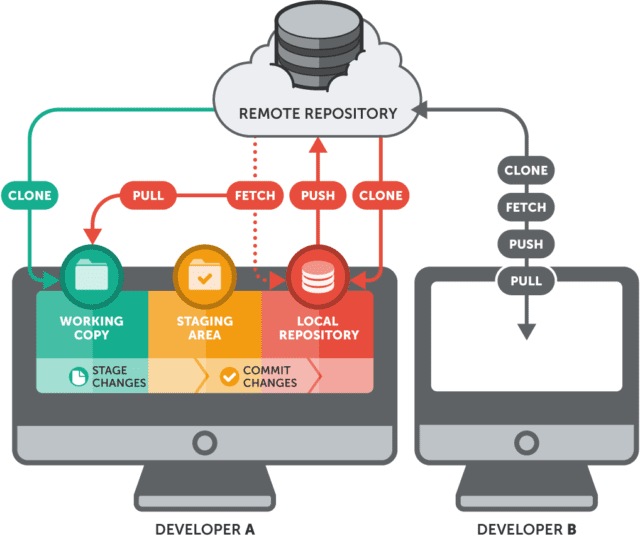
**a. Sắp xếp công việc tốt hơn**

Nghĩa là, bạn có thể tập trung giải quyết từng task mà không phải bận tâm lo lắng cho những task liên quan.

Nếu không dùng Git, khả năng cao là mọi người sẽ làm việc giẫm chân nhau, những task sắp hoàn thành sẽ bị trì hoãn. Ngoài ra, tất cả mọi task lớn nhỏ sẽ buộc phải hoàn thành hết trước khi deploy, bởi vì chỉ cần 1 task vẫn đang dang dở, cả phần mềm có thể bị sập.

**b. Linh hoạt hơn khi phải làm cùng lúc nhiều task**

Bởi vì bạn có thể cấu trúc công việc dễ dàng hơn nên việc làm nhiều task cùng lúc vô cùng dễ dàng. Ví dụ, cùng một lúc, chúng ta thường có một team làm tính năng mới, một vài team khác nâng cấp các tính năng hiện có, và một người fix bug.



**c. Tự tin hơn khi thử nghiệm những ý tưởng mới**

Bạn có thể tách biệt việc thử nghiệm với dự án chính, điều này giúp nâng cao chất lượng code cũng như tính sáng tạo.

Nhìn chung, hiện nay Git được coi là tiêu chuẩn bất thành văn trong ngành. Nếu chưa biết về Git, bạn nên dành thời gian để bắt đầu tìm hiểu ngay. Vì, sớm hay muộn, bạn cũng sẽ thuộc về một team phải dựa dẫm vào nó.

**d. Git cho phép chúng ta làm việc offline**

Theo anh Thành Nhân, ở thời điểm hiện tại thì Git ưu việt hơn hẳn so với các hệ thống quản lý phiên bản tập trung như SVN.

Ví dụ, **Git cho phép chúng ta làm việc offline trong một khoảng thời gian**. Bạn chỉ cần internet cho nhu cầu hợp tác nhóm, hoặc lưu lịch sử commit code lên remote repos. Ngược lại, với SVN, mỗi khi sử dụng đều cần có kết nối đến máy chủ SVN.

**e. Cách lưu trữ thông tin**

Anh Jonathan chia sẻ rằng nếu so sánh với [SVN](https://www.collab.net/products/subversion" \t "_blank) và [TFS](https://www.visualstudio.com/es/tfs/" \t "_blank), rõ ràng là Git hơn hẳn. Sự khác biệt cốt lõi trong cách quản lý storage và các nhánh của Git khiến cho việc merging cũng hoàn toàn khác.

Bên cạnh đó, cũng nhờ cách Git lưu trữ thông tin mà bạn có thể thực hiện vô vàn những điều thú vị để viết lại lịch sử commit.

Bổ dung thêm về tính năng này, anh Thành Nhân đưa ra một ví dụ khác. **Khi tách nhánh, Git chỉ sử dụng 41 bytes cho một nhánh mới**, giúp tiết kiệm không gian lưu trữ mà vẫn đảm bảo tốt nhu cầu công việc. Còn SVN, theo tôi biết, sẽ copy toàn bộ source code thành một bản mới khi tách nhánh.

1.4 GITHUB



1.4.1 Github là gì?

[GitHub](https://github.com/) là một dịch vụ nổi tiếng cung cấp kho lưu trữ mã nguồn [Git](https://topdev.vn/blog/git-la-gi/) cho các dự án phần mềm. **Github có đầy đủ những tính năng của Git**, ngoài ra nó còn bổ sung những tính năng về social để các developer tương tác với nhau.

Vài thông tin về GIT:

* Là công cụ giúp **quản lý source code** tổ chức theo **dạng dữ liệu phân tán**.
* Giúp **đồng bộ source code** của team lên 1 server.
* Hỗ trợ các thao tác **kiểm tra source code** trong quá trình làm việc (diff, check modifications, show history, merge source, …)

1.4.2 Tính năng của Github

**GitHub** được coi là một mạng xã hội dành cho lập trình viên lớn nhất và dễ dùng nhất với các tính năng cốt lõi như:

1. Wiki, issue, thống kê, đổi tên project, project được đặt vào namespace là user.
2. Watch project: theo dõi hoạt động của project của người khác. Xem quá trình người ta phát triển phầm mềm thế nào, project phát triển ra sao.
3. Follow user: theo dõi hoạt động của người khác.

Có 2 cách tiếp cận GitHub: Tạo project của riêng mình Contribute cho project có sẵn: fork project có sẵn của người khác, sửa đổi, sau đó đề nghị họ cập nhật sửa đổi của mình (tạo pull request).

1.4.3 Một vài khái niệm của Git bạn cần nắm

* **git**: là prefix của các lệnh được sử dụng dưới CLI
* **branch**: được hiểu như là nhánh, thể hiện sự phân chia các version khi 2 version đó có sự sai khác nhất định và 2 version đều có sự khác nhau.
* **commit**: là một điểm trên cây công việc (Work Tree ) hay gọi là cây phát triển công việc
* **clone**: được gọi là nhân bản, hay thực hiện nhân bản. Sử dụng để clone các project, repository trên các hệ thống chạy trên cơ sở là git, ví dụ như: bitbucket, github, gitlab, cor(1 sản phẩm mã nguồn mở cho phép người dùng tự tạo git server cho riêng mình trên vps, server),… Việc clone này sẽ sao chép repository tại commit mình mong muốn, dùng để tiếp tục phát triển. Thao tác này sẽ tải toàn bộ mã nguồn, dữ liệu về máy tính của bạn.
* **folk**: Folk là thao tác thực hiện sao chép repository của chủ sở hữu khác về git account của mình. sử dụng và đối xử như 1 repository do mình tạo ra.
* **repository**: Kho quản lý dữ liệu, là nơi lưu trữ các dữ liệu, mã nguồn của project.
* **tag**: sử dụng để đánh dấu một commit khi bạn có quá nhiều commit tới mức không thể kiểm soát được.
* **remote**: sử dụng để điều khiển các nhánh từ một repository trên git server, đối xử với các nhánh trên remote tương tự như đối xử với các nhánh trên local
* **diff:** So sánh sự sai khác giữa phiên bản hiện tại với phiên bản muốn so sánh, nó sẽ thể hiện các sự khác nhau
* **.gitignore:** file mặc định của git sử dụng để loại bỏ (ignore) các thư mục, file mà mình không muốn push lên git server

1.4.4 Lợi ích của Github đối với lập trình viên

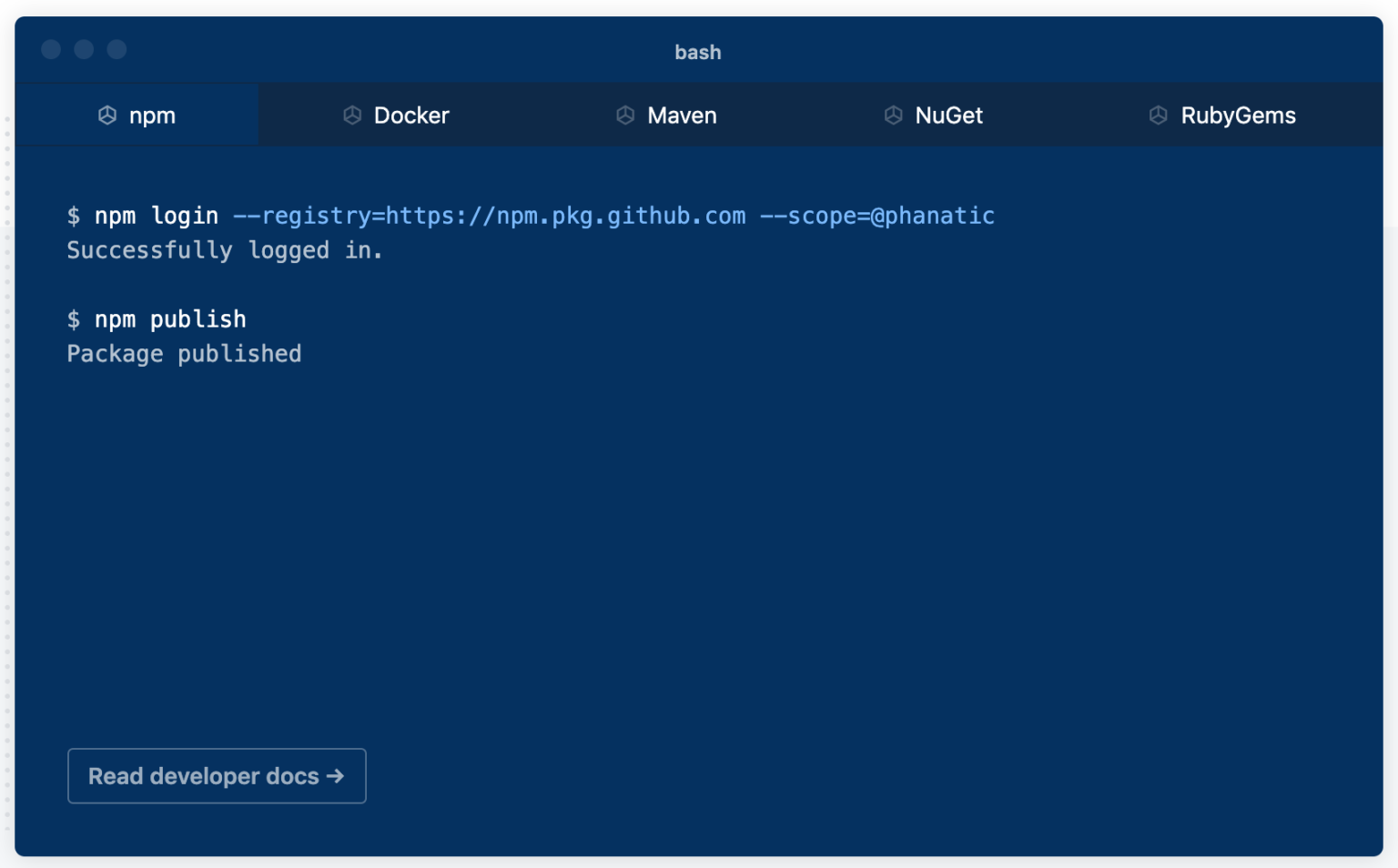
### Quản lý source code dễ dàng

### Tracking sự thay đổi qua các version

### Markdown

1.4.5 Github Package Registry

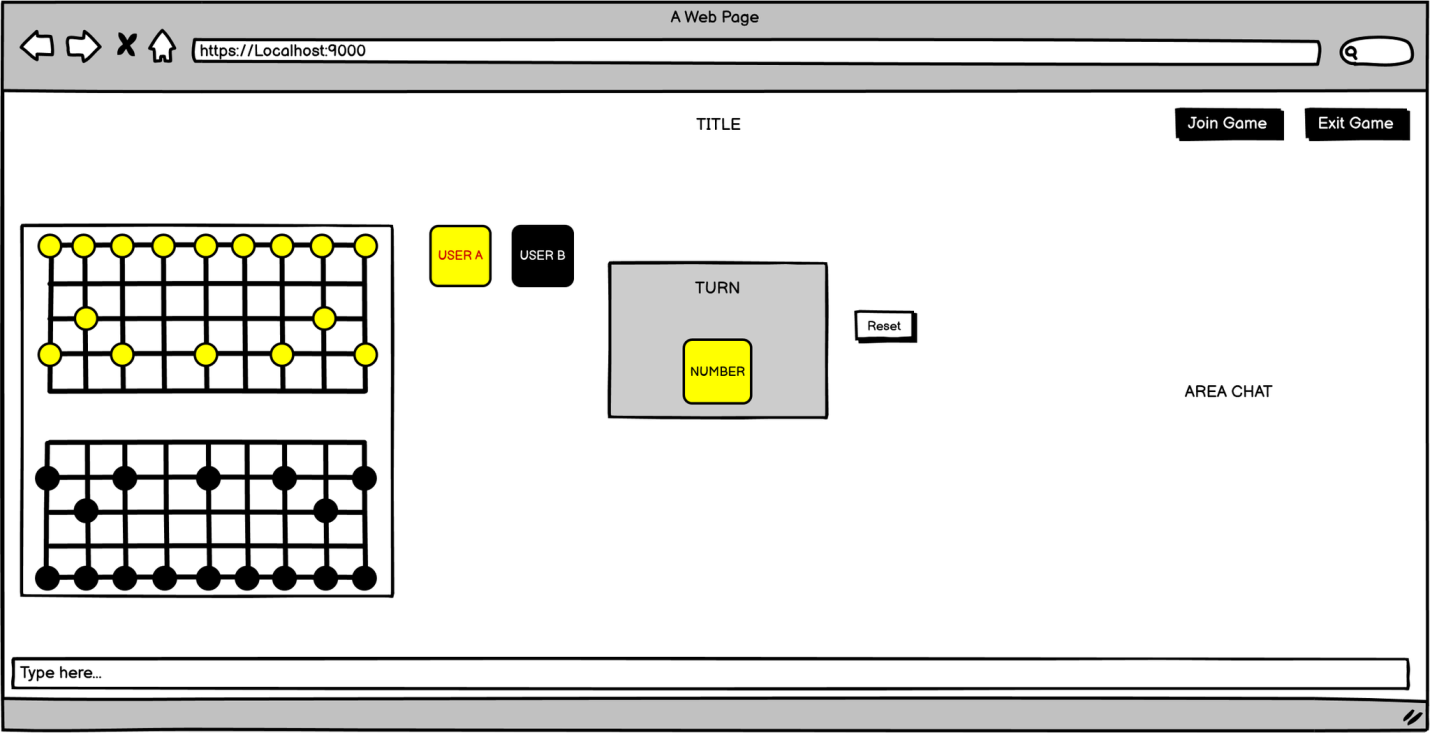
Cái package registry này cho phép lập trình viên duy trì distribution registries của họ, bao gồm npm, docker, maven, nuget và Ruby gems.



CHƯƠNG 2: KẾT QUẢ

2.1 Wireframe

Nhóm sử dụng phần mềm Balsamiq để tiến hành thiết kế wireframe cho website.

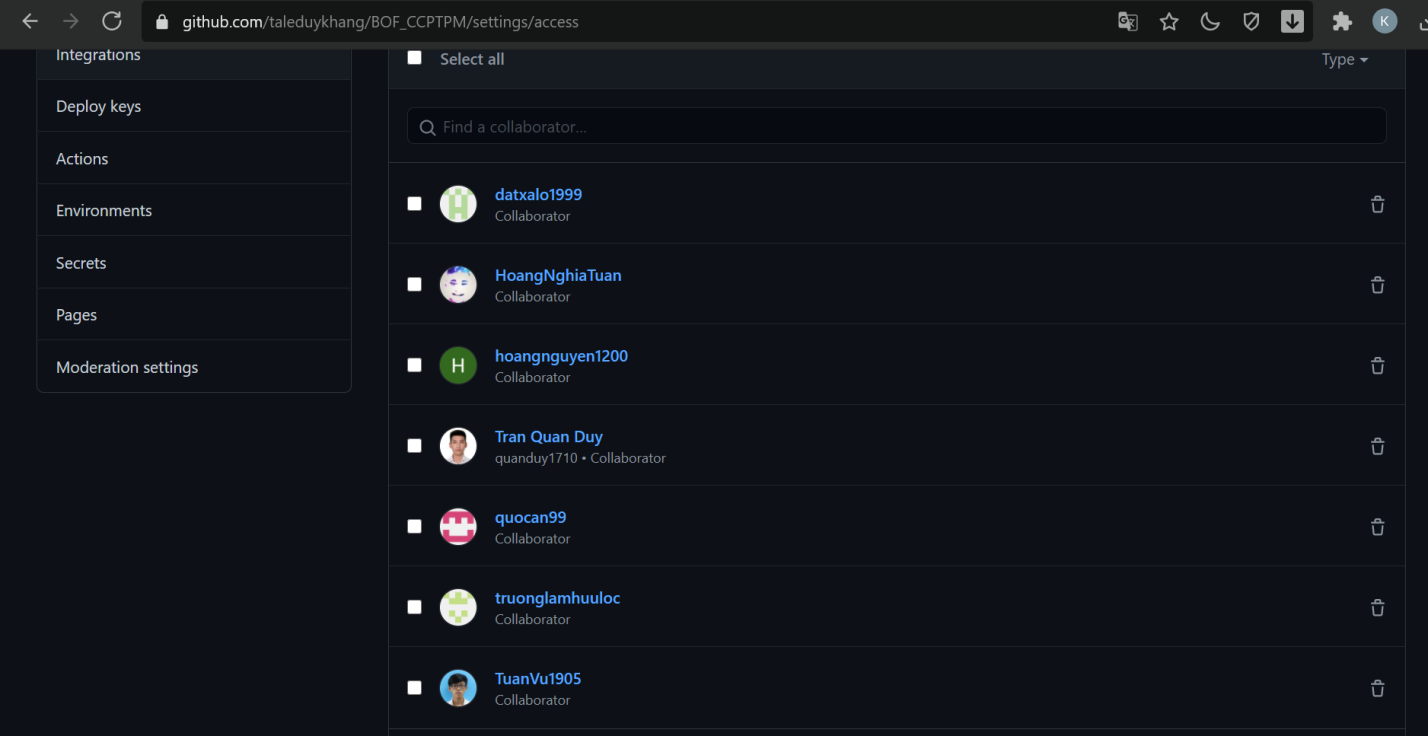


Hình 2.1 Wireframe website cờ tướng online

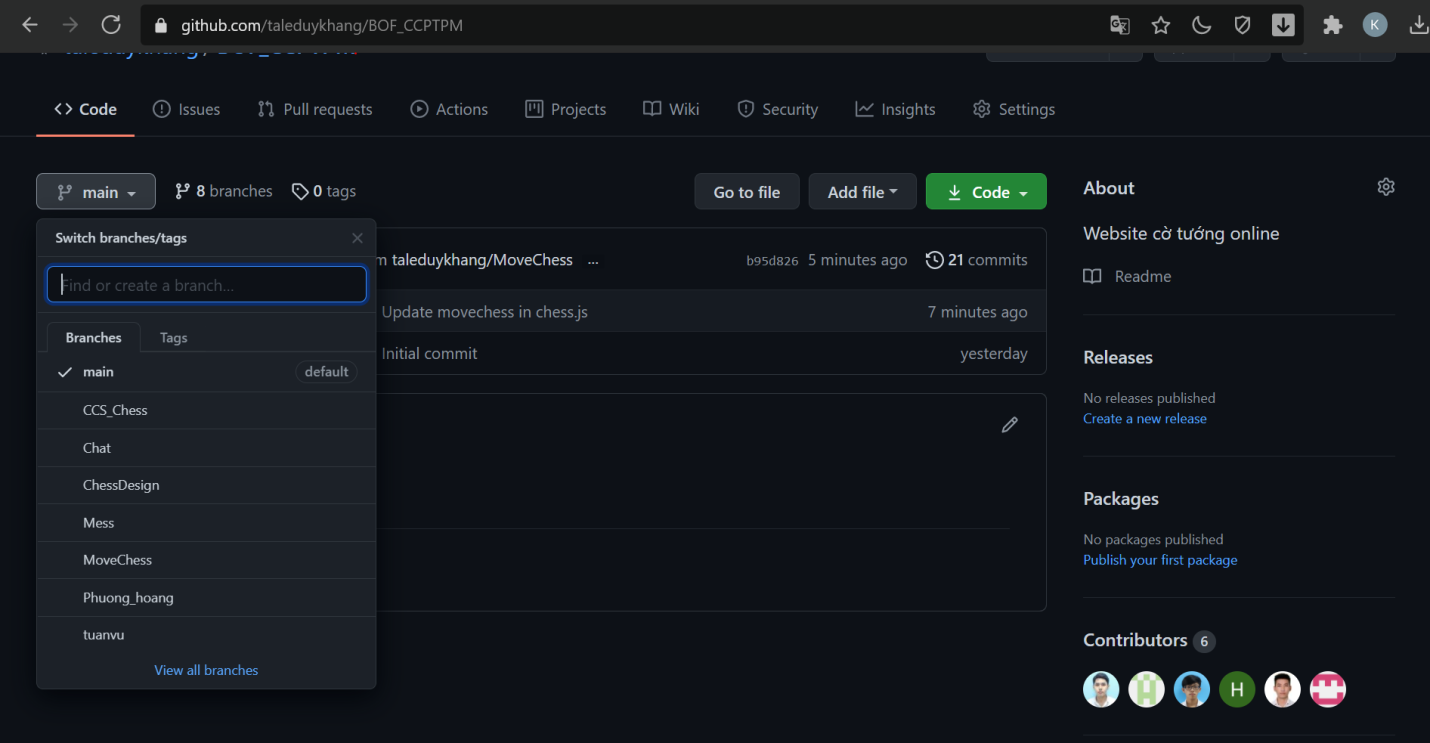
2.2 Github

Nhóm sử dụng phần mềm Tortoisegit để quản lý phiên bản của website.

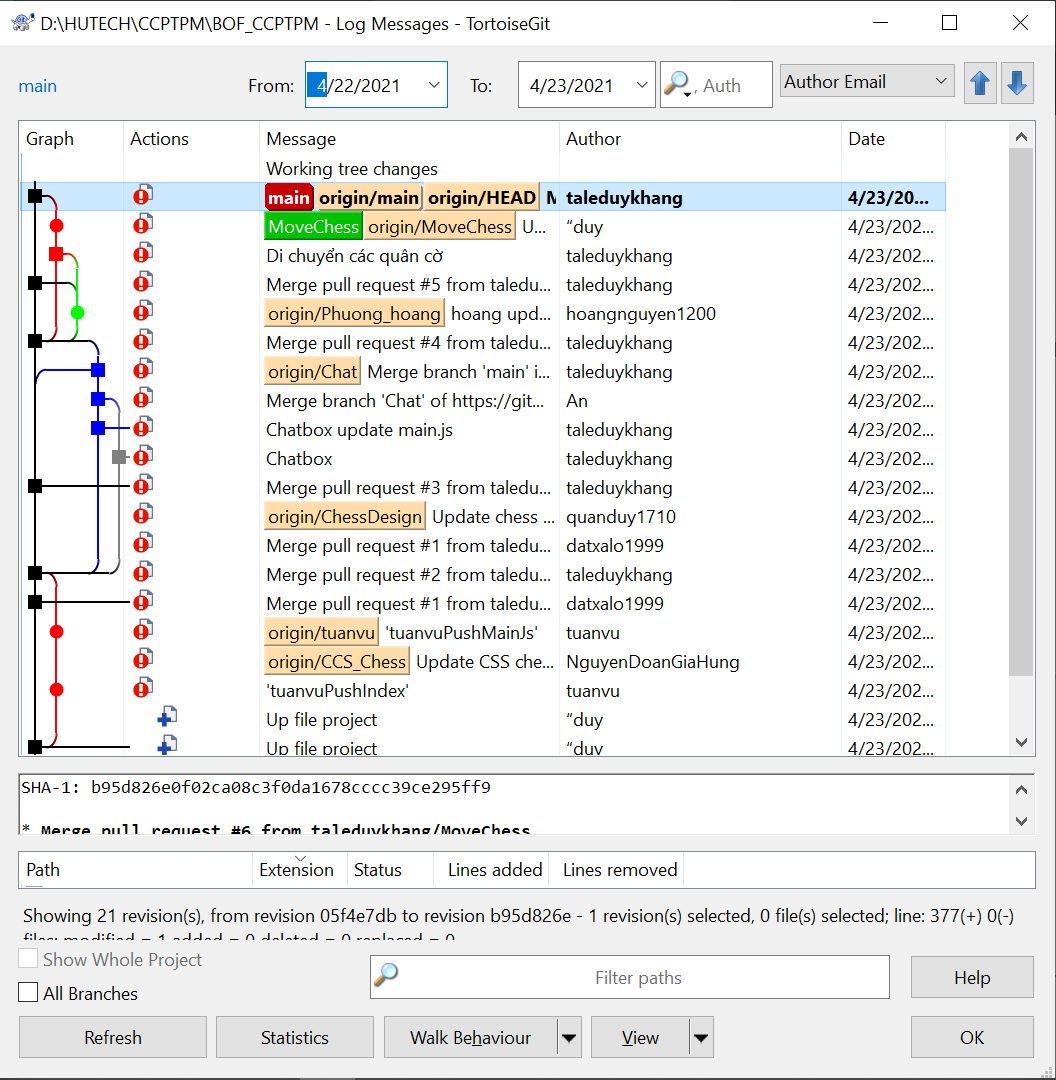
Link github của đồ án: https://github.com/taleduykhang/BOF\_CCPTPM.git



Hình 2.2 Danh sách các thành viên trong đồ án trên github



Hình 2.3 Các branch trong quá trình phát triển đồ án



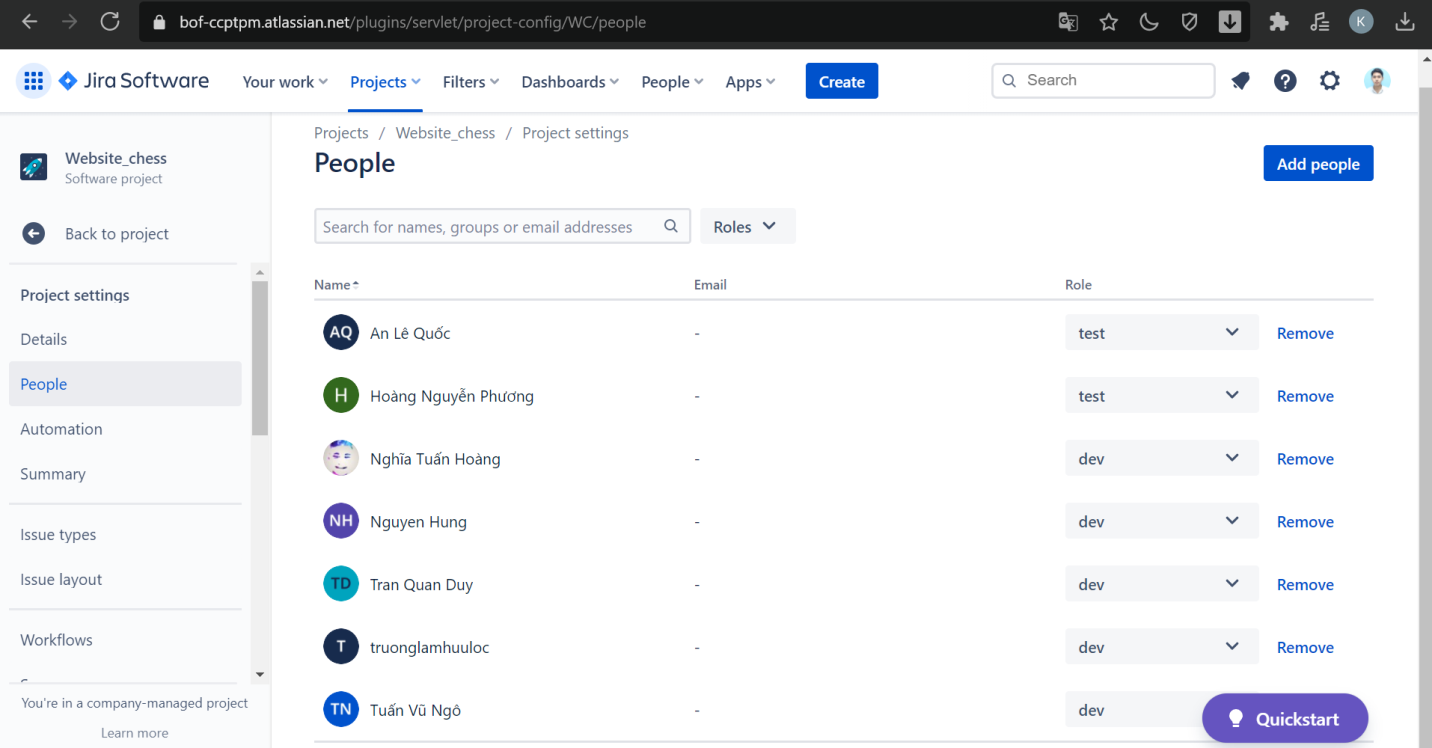
Hình 2.4 Lịch sử các hoạt động trên github

2.3 Jira

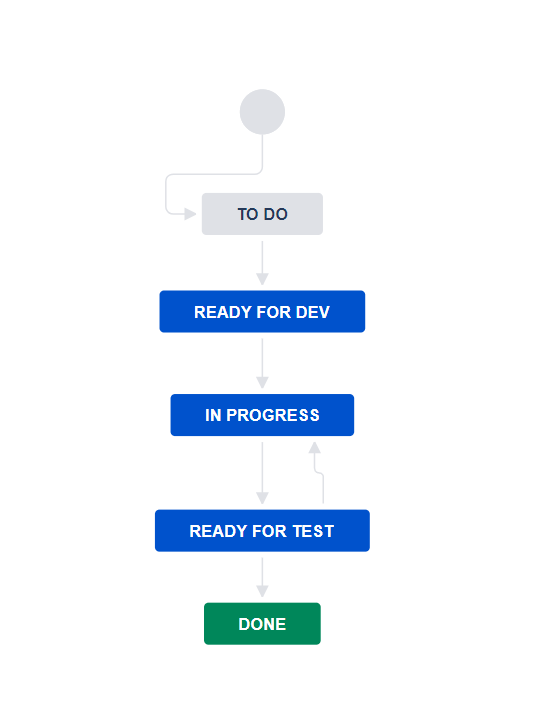
Nhóm sử dụng Jira để quản lý tiến độ của đồ án.

Link jira của đồ án: <https://bof-ccptpm.atlassian.net/secure/RapidBoard.>

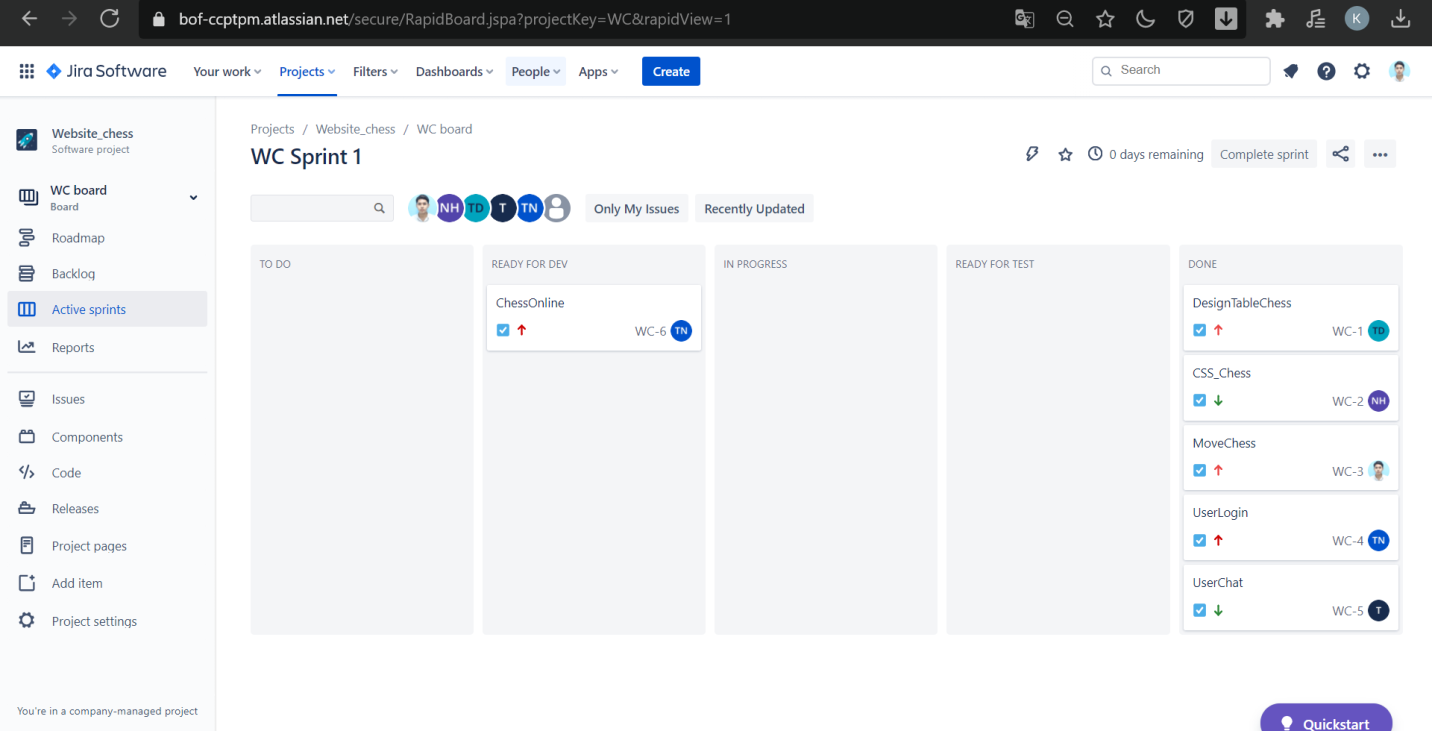
jspa?rapidView=1&projectKey=WC



Hình 2.5 Danh sách các thành viên và chức vụ trên Jira

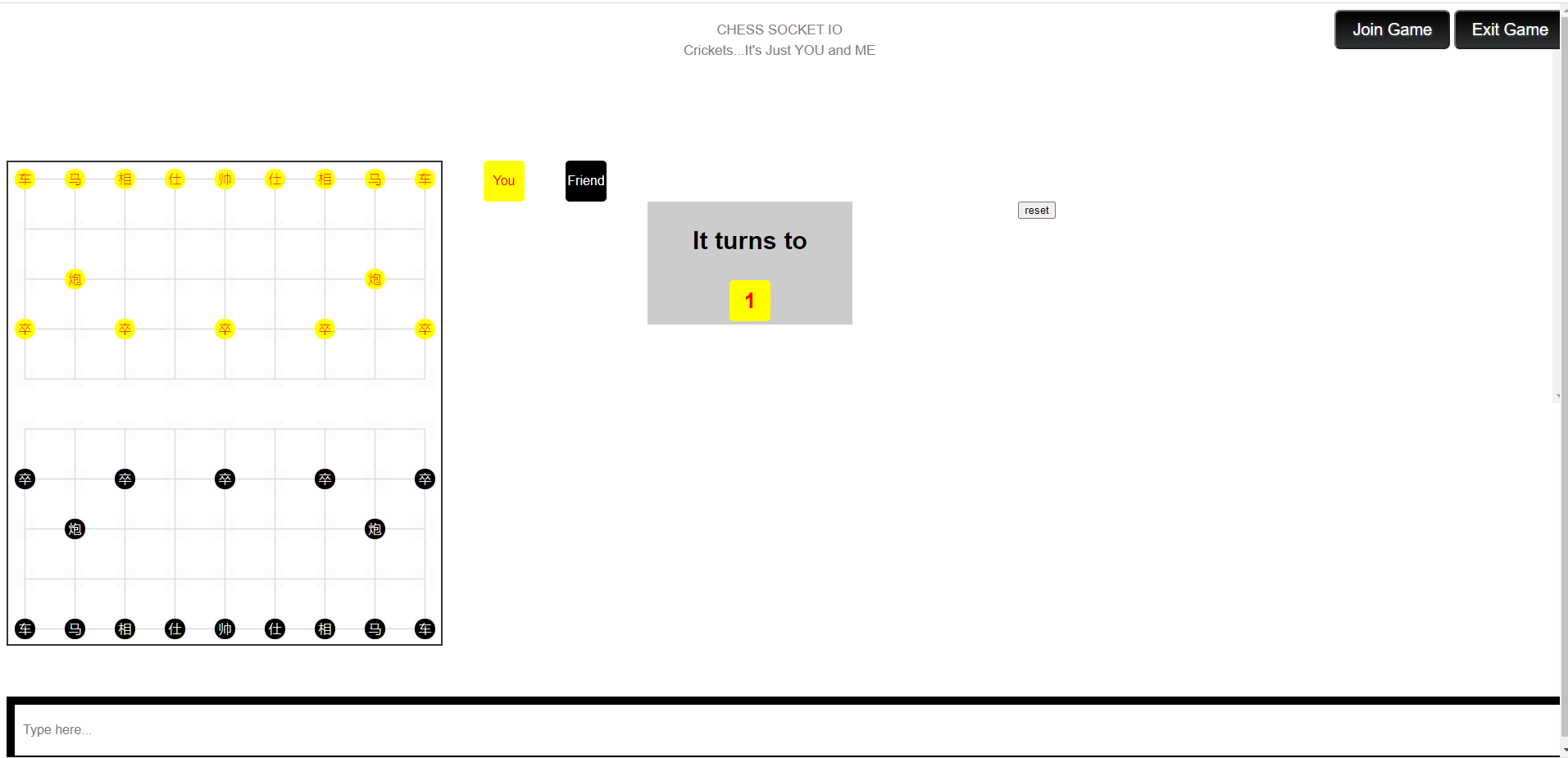


Hình 2.6 Workflow công việc của đồ án

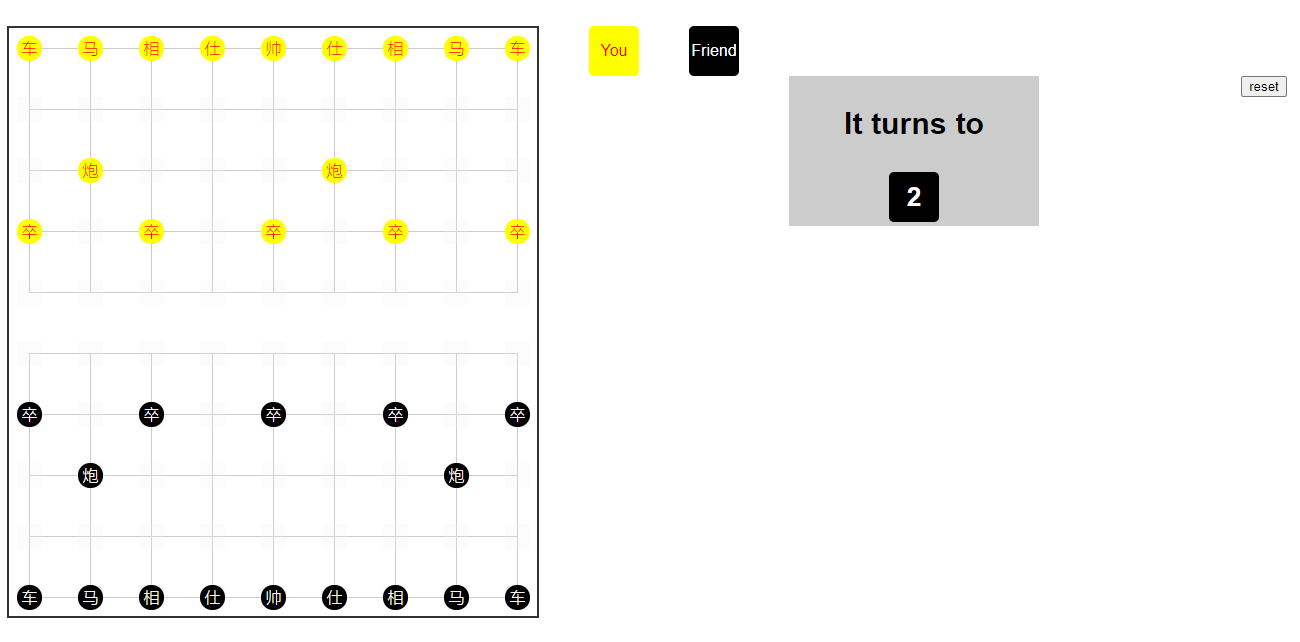


Hình 2.7 Sprint của dự án

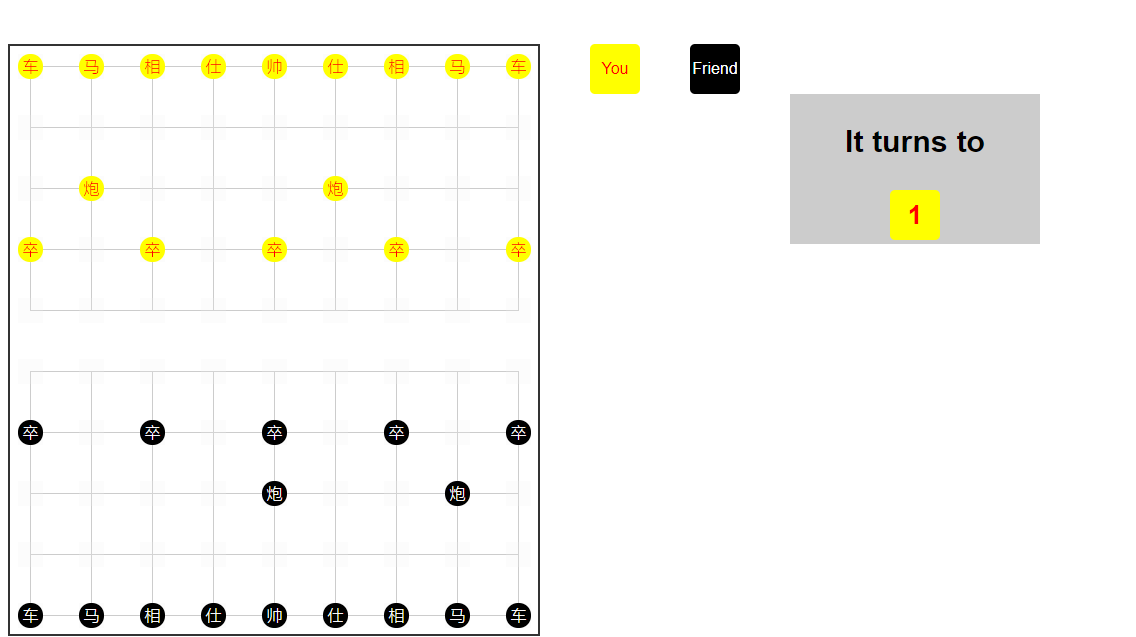
2.4 Website cờ tướng online



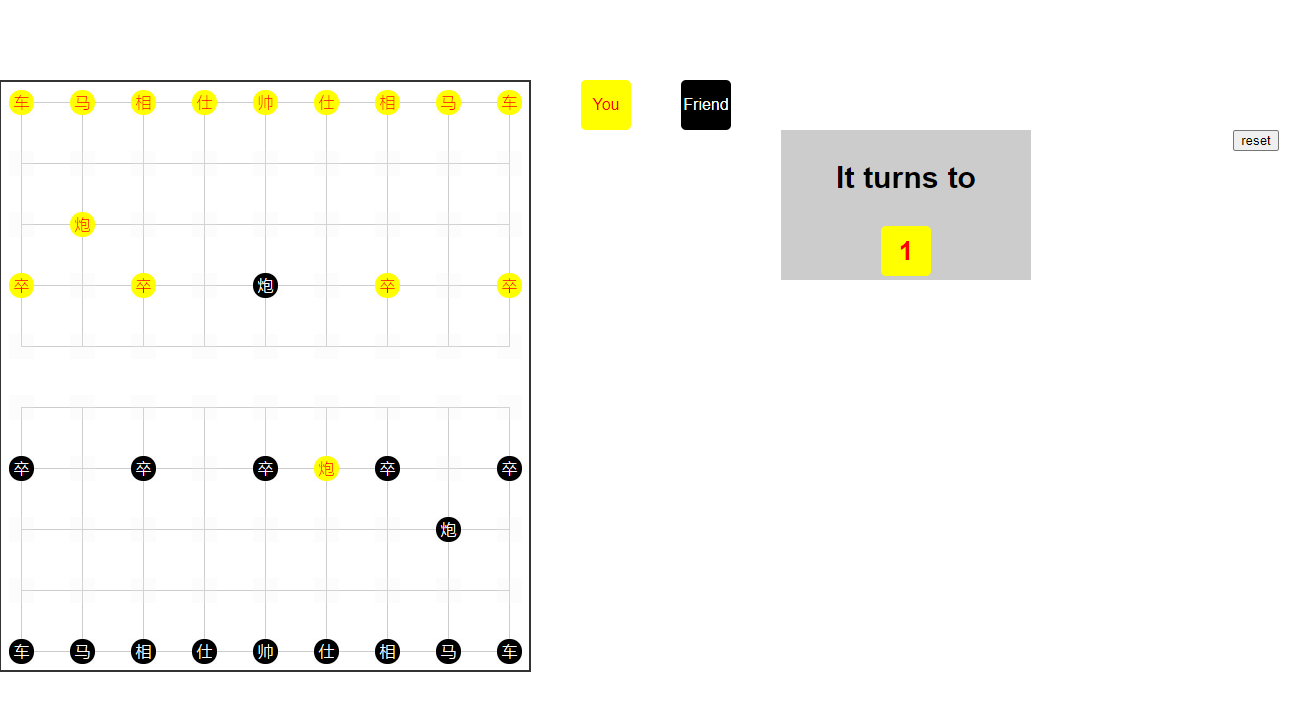
Hình 2.8 Giao diện của website khi bắt đầu vào phòng chơi cờ



Hình 2.8 Cờ vàng di chuyển quân pháo



Hình 2.9 Cờ đen di chuyển quân pháo



Hình 2.10 Con pháo của cờ đen ăn quân tốt của cờ vàng