

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ



BÁO CÁO DỰ ÁN
Chơi cờ Caro online

Nhóm 11 : Gamer

Thành viên :

- Phạm Thị Thùy Tiên
- Chu Thị Thùy Linh
- Cao Thị Phương Thảo

Hà Nội - 2018

Mục lục

DANH MỤC BẢNG BIỂU	3
DANH MỤC HÌNH ẢNH	3
1. Mô tả dự án	4
2. Hướng dẫn cài đặt	4
3. Công nghệ áp dụng	4
3.1. AngularJS	4
3.1.1 Khái niệm	4
3.1.2 Thêm một số bộ điều khiển	4
3.1.3 Tạo component	5
3.1.4 Điều hướng, biểu mẫu và back-end	5
3.2. NodeJS	6
3.2.1 Khái niệm	6
3.2.2 Tại sao lại là NodeJS	6
3.3. Socket.io	7
3.3.1 Khái niệm	7
3.3.2 Độ tin cậy	7
3.3.3 Các câu lệnh tiêu biểu	8
4. Các chức năng chính	8
5. Các tác nhân	9
6. Đặc tả chức năng	9
7. Các màn hình chính	11
8. Database	12
9. Kết luận	13
10. Tài liệu tham khảo	14

DANH MỤC BẢNG BIỂU

Bảng 1 Các tác nhân.....	9
Bảng 2 Đặc tả chức năng	11
Bảng 3 Các màn hình chính	11

DANH MỤC HÌNH ẢNH

Hình 1 Các câu lệnh tiêu biểu trong Socket.IO	8
Hình 2 Màn hình đăng ký, đăng nhập.....	11
Hình 3 Màn hình danh sách phòng	12
Hình 4 Màn hình chơi	12
Hình 5 Cơ sở dữ liệu	13

1. Mô tả dự án

Cờ Caro là một trong những trò chơi phổ biến và được ưa chuộng hiện nay tại Việt Nam. Chơi cờ giúp chúng ta tăng sự tập trung, trí tưởng tượng theo logic, giúp người chơi tăng khả năng đánh giá và phân tích tình huống. Hướng đến những ích lợi này, sản phẩm Cờ Caro Online của nhóm.... được xây dựng để tất cả mọi người có thể tham gia chơi ở bất kỳ đâu mà không cần gặp mặt mới có thể chơi. Ứng dụng hướng tới sự tiện ích và linh hoạt cho người dùng.

2. Hướng dẫn cài đặt

Bước 1: Cài đặt NodeJs tại <https://nodejs.org/en/>

Bước 2: Clone project tại: <https://github.com/ThuyTienUet/Caro-Online.git>

Bước 3: Mở Command Line và dẫn vào thư mục project vừa tải về rồi chạy lệnh 'npm install'

Bước 4: Chạy lệnh 'npm start' và xem web tại <http://localhost:6001/>

Link app: <http://fit5.fit-uet.tk:6001>

3. Công nghệ áp dụng

3.1. AngularJS

3.1.1 Khái niệm

AngularJS (thường được gọi là Angular.js hoặc AngularJS 1.X) là một web framework JavaScript được phát triển và tài trợ bởi Google và cộng đồng để giải quyết các vấn đề gặp phải trong việc phát triển ứng dụng đơn trang. Ngoài ra, AngularJs còn có các thành phần bổ sung cho Cordova, framework sử dụng để viết các ứng dụng di động. Nó nhằm mục đích để đơn giản hóa cả phát triển và thử nghiệm của các ứng dụng bằng cách cung cấp một framework với kiến trúc cho client side đó là MVC và MVVM.

3.1.2 Thêm một số bộ điều khiển

- Sự ràng buộc dữ liệu

Liên kết dữ liệu là cách tự động cập nhật chế độ xem bất cứ khi nào mô hình thay đổi, cũng như cập nhật mô hình bất cứ khi nào chế độ xem thay đổi. Điều này thật tuyệt vời vì nó loại bỏ thao tác DOM khỏi danh sách những điều bạn phải lo lắng.

- **Các bộ điều khiển (Controllers)**

Bộ điều khiển là hành vi sau các thành phần DOM. AngularJS cho phép bạn thực hiện hành vi ở dạng dễ đọc mà không cần bản tóm tắt thông thường về việc cập nhật DOM, đăng ký lại hoặc xem các thay đổi mô hình.

- **JavaScript đơn giản**

Không giống như các framework khác, không cần phải kế thừa từ các phương thức độc quyền để bọc mô hình trong các phương thức truy cập. Các mô hình AngularJS là các đối tượng JavaScript cũ đơn giản. Điều này làm cho mã của bạn dễ dàng kiểm tra, bảo trì, tái sử dụng và một lần nữa không bị lỗi.

3.1.3 Tạo component

- **Directives**

Directives là một tính năng độc đáo và mạnh mẽ có sẵn trong AngularJS. Directives cho phép bạn phát minh ra cú pháp HTML mới, cụ thể cho ứng dụng của bạn.

- **Lĩnh kiện tái sử dụng**

Chúng tôi sử dụng các chỉ thị để tạo ra các thành phần có thể tái sử dụng. Một thành phần cho phép bạn ẩn cấu trúc DOM, CSS và hành vi phức tạp. Điều này cho phép bạn tập trung vào những gì ứng dụng làm hoặc cách ứng dụng trông riêng biệt.

- **Địa phương hóa**

Một phần quan trọng của các ứng dụng nghiêm trọng là nội địa hóa. Các bộ lọc nhận biết và chỉ thị cục bộ của AngularJS cung cấp cho bạn các block xây dựng để làm cho ứng dụng của bạn có sẵn ở tất cả các địa phương.

3.1.4 Điều hướng, biểu mẫu và back-end

- **Liên kết sâu**

Một liên kết sâu phản ánh nơi người dùng đang ở trong ứng dụng. Điều này rất hữu ích để người dùng có thể đánh dấu

và liên kết email đến các vị trí trong ứng dụng. Các ứng dụng khứ hồi có được điều này tự động, nhưng bản chất ứng dụng AJAX thì không. AngularJS kết hợp những lợi ích của liên kết sâu với hành vi giống như ứng dụng trên máy tính để bàn.

- **Xác thực mẫu**

Xác thực mẫu phía khách hàng là một phần quan trọng của trải nghiệm người dùng tuyệt vời. AngularJS cho phép bạn khai báo các quy tắc xác thực của biểu mẫu mà không phải viết mã JavaScript. Viết ít mã hơn, đi uống bia sớm hơn.

- **Giao tiếp máy chủ**

AngularJS cung cấp các dịch vụ tích hợp trên XHR cũng như các phụ trợ khác bằng thư viện của bên thứ ba. Hứa sẽ tiếp tục đơn giản hóa mã của bạn bằng cách xử lý trả lại dữ liệu không đồng bộ.

3.2. *NodeJS*

3.2.1 *Khái niệm*

Node.js là một hệ thống phần mềm được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là máy chủ web. Chương trình được viết bằng JavaScript, sử dụng kỹ thuật điều khiển theo sự kiện, nhập/xuất không đồng bộ để tối thiểu tổng chi phí và tối đại khả năng mở rộng

3.2.2 *Tại sao lại là NodeJS*

Node.js sử dụng lập trình không đồng bộ!

Một nhiệm vụ chung cho một máy chủ web có thể là mở một tệp trên máy chủ và trả lại nội dung cho máy khách.

- Đây là cách PHP hoặc ASP xử lý yêu cầu tệp:
 - Gửi tác vụ đến hệ thống tệp của máy tính.
 - Chờ trong khi hệ thống tệp mở và đọc tệp.
 - Trả về nội dung cho khách hàng.
 - Sẵn sàng để xử lý các yêu cầu tiếp theo.
- Đây là cách Node.js xử lý yêu cầu tệp:

- Gửi tác vụ đến hệ thống tệp của máy tính.
- Sẵn sàng để xử lý các yêu cầu tiếp theo.
- Khi hệ thống tệp đã mở và đọc tệp, máy chủ sẽ trả lại nội dung cho máy khách.

Node.js loại bỏ sự chờ đợi và chỉ cần tiếp tục với yêu cầu tiếp theo.

Node.js chạy lập trình đơn luồng, không chặn, lập trình không đồng bộ, rất hiệu quả về bộ nhớ.

3.3. *Socket.io*

3.3.1 *Khái niệm*

Socket.IO là một thư viện JavaScript cho các ứng dụng web thời gian thực. Nó cho phép giao tiếp hai chiều giữa máy khách và máy chủ web. Nó có hai phần: một client-side thư viện mà chạy trong trình duyệt, và một server-side thư viện cho Node.js. Cả hai thành phần có một API gần như giống hệt nhau. Giống như Node.js, nó được điều khiển theo sự kiện .

Socket.IO chủ yếu sử dụng giao thức WebSocket với việc bỏ phiếu làm tùy chọn dự phòng, trong khi cung cấp cùng một giao diện. Mặc dù nó có thể được sử dụng đơn giản như một trình bao bọc cho WebSocket, nhưng nó cung cấp nhiều tính năng hơn, bao gồm phát đến nhiều ổ cắm, lưu trữ dữ liệu được liên kết với từng máy khách và I / O không đồng bộ.

Nó có thể được cài đặt với công cụ npm.

3.3.2 *Độ tin cậy*

Các kết nối được thiết lập ngay cả khi tồn tại:

- Proxy và cân bằng tải.
- Tường lửa cá nhân và phần mềm chống vi-rút.

Với mục đích này, nó dựa vào Engine.IO , lần đầu tiên thiết lập kết nối bỏ phiếu dài, sau đó cố gắng nâng cấp lên các phương tiện vận chuyển tốt hơn, được thử nghiệm trên một bên, như WebSocket.

3.3.3 Các câu lệnh tiêu biểu

```
io.on('connect', onConnect);

function onConnect(socket){

  // sending to the client
  socket.emit('hello', 'can you hear me?', 1, 2, 'abc');

  // sending to all clients except sender
  socket.broadcast.emit('broadcast', 'hello friends!');

  // sending to all clients in 'game' room except sender
  socket.to('game').emit('nice game', "let's play a game");

  // sending to all clients in 'game1' and/or in 'game2' room, except sender
  socket.to('game1').to('game2').emit('nice game', "let's play a game (too)");

  // sending to all clients in 'game' room, including sender
  io.in('game').emit('big-announcement', 'the game will start soon');

  // sending to all clients in namespace 'myNamespace', including sender
  io.of('myNamespace').emit('bigger-announcement', 'the tournament will start soon');

  // sending to a specific room in a specific namespace, including sender
  io.of('myNamespace').to('room').emit('event', 'message');

  // sending to individual socketid (private message)
  io.to(`${socketId}`).emit('hey', 'I just met you');

  // WARNING: `socket.to(socket.id).emit()` will NOT work, as it will send to everyone in the room
  // named `socket.id` but the sender. Please use the classic `socket.emit()` instead.

  // sending with acknowledgement
  socket.emit('question', 'do you think so?', function (answer) {});

  // sending without compression
  socket.compress(false).emit('uncompressed', "that's rough");

  // sending a message that might be dropped if the client is not ready to receive messages
  socket.volatile.emit('maybe', 'do you really need it?');

  // specifying whether the data to send has binary data
  socket.binary(false).emit('what', 'I have no binaries!');

  // sending to all clients on this node (when using multiple nodes)
  io.local.emit('hi', 'my lovely babies');

  // sending to all connected clients
  io.emit('an event sent to all connected clients');

};
```

Hình 1 Các câu lệnh tiêu biểu trong Socket.IO

4. Các chức năng chính

Một số chức năng chính của ứng dụng:

- Xem danh sách và trạng thái các phòng chơi.
- Tạo phòng mới hoặc chọn phòng tham gia chơi.
- Hỗ trợ 2 người chơi.
- Xếp hạng người chơi.

5. Các tác nhân

STT	Tác nhân	Hành động
1	Admin	1. Đăng nhập 2. Xem danh sách phòng 3. Tạo phòng 4. Chọn phòng 5. Xóa phòng 6. Chơi 7. Xóa người trong phòng 8. Chat 9. Xóa User 10. Xếp hạng người chơi
2	User	1. Đăng ký 2. Đăng nhập 3. Xem danh sách phòng 4. Tạo phòng 5. Chọn phòng 6. Chơi 7. Xóa người trong phòng 8. Chat

Bảng 1 Các tác nhân

6. Đặc tả chức năng

STT	Chức năng	Tác nhân	Đặc tả
1	Đăng ký	User	Người chơi cần có tài khoản để chơi
2	Đăng nhập	Admin	Người chơi muốn chơi thì cần phải đăng nhập
		User	

3	Xem danh sách phòng	Admin	Hiển thị danh sách và trạng thái các phòng chơi
		User	
4	Tạo phòng	Admin	Người chơi có thể tự tạo phòng chơi mới.
		User	
5	Chọn phòng	User	Người chơi chọn phòng
6	Chơi	Admin	<ul style="list-style-type: none"> - Chủ phòng sẽ là X, bạn chơi là O - Khi có 5 quân giống nhau liên tiếp thẳng hàng thì bên đó thắng. - Người thắng sẽ được cộng điểm để xếp hạng.
		User	
7	Xóa người trong phòng	Admin	Chủ phòng có quyền xóa những người trong phòng.
		User	
8	Chat	Admin	Người trong phòng có thể tham gia chat nói chuyện với nhau.
		User	
9	Xóa phòng	Admin	Admin có quyền xóa phòng, kể cả phòng đó còn người.
10	Xóa User	Admin	Admin có quyền xóa User

11	Xếp hạng người chơi	Admin	Hệ thống xếp hạng người chơi theo điểm tích lũy.
----	---------------------	-------	--------------------------------------------------

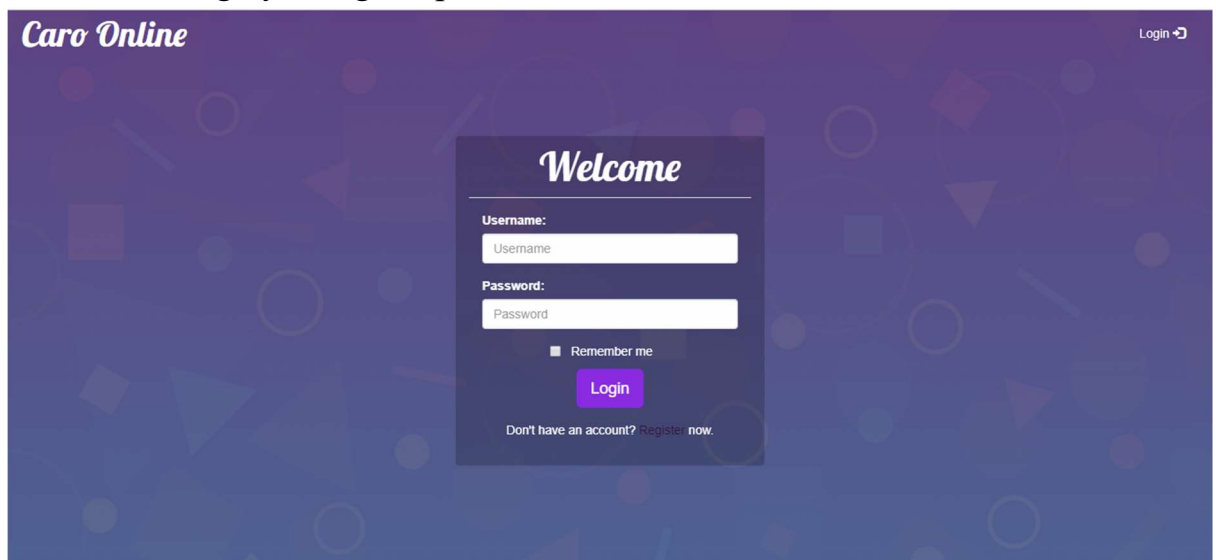
Bảng 2 Đặc tả chức năng

7. Các màn hình chính

STT	Màn hình	Chức năng
1	Đăng ký, Đăng nhập	Đăng ký, Đăng nhập
2	Danh sách phòng, bảng xếp hạng	<ul style="list-style-type: none"> - Hiện thị danh sách - Hiện thị bảng xếp hạng
3	Chơi	<ul style="list-style-type: none"> - Hiện thị bàn cờ - Hiện thị bảng điều khiển, trạng thái ván chơi.

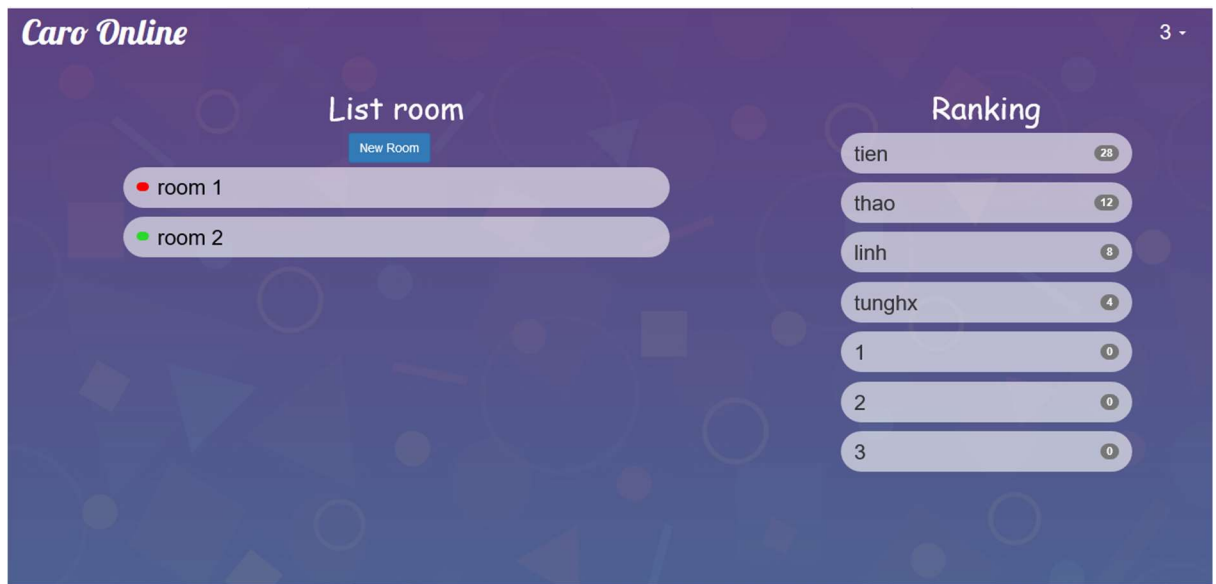
Bảng 3 Các màn hình chính

- Màn hình đăng ký, đăng nhập:



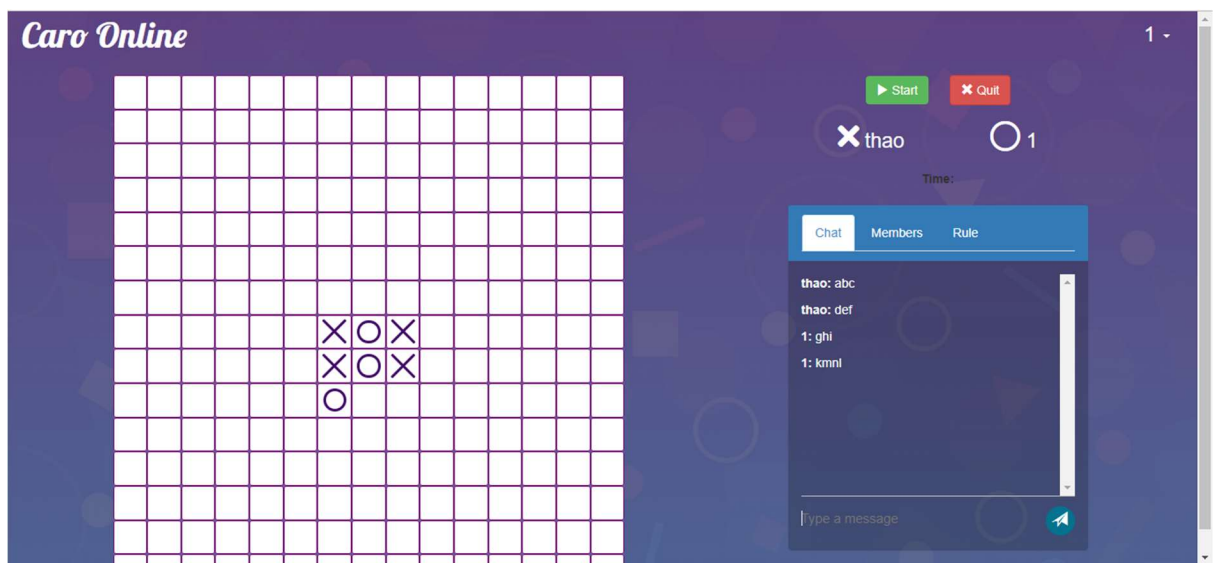
Hình 2 Màn hình đăng ký, đăng nhập

- Màn hình danh sách phòng, bảng xếp hạng:



Hình 3 Màn hình danh sách phòng

- Màn hình chơi:



H

Hình 4 Màn hình chơi

8. Database

Cơ sở dữ liệu bao gồm 2 bảng: users và rooms.

Bảng users chứa thông tin người chơi, bảng rooms chứa thông tin phòng chơi. Một room có thể chứa nhiều user nhưng một user tại một thời điểm chỉ có thể thuộc một room. Nếu trong room không có ai thì room sẽ bị xóa khỏi CSDL.



Hình 5 Cơ sở dữ liệu

9. Kết luận

Ứng dụng chơi cờ online đã hoàn thành các chức năng cốt lõi. Người dùng có thể truy cập trực tuyến vào mọi thời điểm và tiến hành chơi. Ngoài ra, người dùng nhận được kết quả ván chơi của mình ngay sau khi chơi xong, cộng dồn thành tích vào bảng xếp hạng. Ứng dụng cũng đã đáp ứng được các yêu cầu phi chức năng được đặt ra ban đầu.

Trong tương lai, ứng dụng sẽ tiếp tục được phát triển và hoàn thiện hơn về mặt chức năng và giao diện, giúp người dùng có những trải nghiệm tốt hơn. Một số chức năng được ưu tiên phát triển trong giai đoạn tới như: Lựa chọn kích thước bàn cờ, đồng hồ đếm thời gian,... và cải thiện giao diện ứng dụng.

10. Tài liệu tham khảo

- [1] AngularJS: <https://angularjs.org/>
- [2] NodeJS: <https://nodejs.org/en/>
- [3] Socket.IO: <https://socket.io/>