

TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN



NGUYỄN ĐỨC THẮNG

**XÂY DỰNG HỆ THỐNG LIVESTREAM TƯƠNG TÁC
VỚI NGƯỜI CHƠI & TÍCH HỢP QUẢNG CÁO
DOANH NGHIỆP**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

HÀ NỘI, NĂM 2019

TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN



*Dán ảnh sinh viên
cỡ 4x6. Đây là
một yêu cầu của
riêng Khoa để lưu
trữ gương mặt
của sinh viên.*

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC
Ngành Công nghệ thông tin

**XÂY DỰNG HỆ THỐNG LIVESTREAM
TƯƠNG TÁC VỚI NGƯỜI CHƠI & TÍCH HỢP
QUẢNG CÁO DOANH NGHIỆP**

Sinh viên thực hiện:	NGUYỄN ĐỨC THẮNG
	Lớp K57TH3 – 1551060573
Giảng viên hướng dẫn:	Th.S KIỀU TUẤN DŨNG

Hà Nội, năm 2019

LỜI CẢM ƠN

Được sự hướng dẫn của Ths Kiều Tuấn Dũng và sự đồng ý của Khoa Công nghệ thông tin trường Đại Học Thủy Lợi, em đã được nhận đề tài **“Xây dựng hệ thống livestream tương tác với người chơi & tích hợp quảng cáo doanh nghiệp”** cho đồ án tốt nghiệp kết thúc 4,5 năm quá trình học tại trường mình.

Để có được điều này, trước hết cho phép em được gửi lời cảm ơn chân thành đến toàn thể các thầy cô giáo trong trường Đại học Thủy Lợi nói chung và đặc biệt là các thầy cô giáo trong khoa Công nghệ thông tin nói riêng, những người đã tận tình dạy dỗ, truyền đạt cho em những kiến thức, kỹ năng học tập trong suốt thời gian học tập tại trường và tạo mọi điều kiện tốt nhất để em hoàn thành các môn học tại trường cũng như là đồ án tốt nghiệp này. Đặc biệt hơn hết, em xin gửi lời cảm ơn chân thành nhất tới thầy giáo - Ths Kiều Tuấn Dũng - giảng viên khoa Công nghệ thông tin, người đã nhiệt tình chỉ bảo giúp đỡ em rất nhiều trong suốt quá trình hoàn thành đồ án từ lúc phân tích tới thiết kế và hoàn thành đồ án.

Mặc dù nhiều cố gắng để đạt được kết quả tốt cho đề tài, nhằm phản ánh thực lực trong quá trình học tập và rèn luyện tại trường Đại Học Thủy Lợi, song do những hạn chế về kiến thức và kinh nghiệm nên không thể tránh khỏi những thiếu sót. Em rất mong sự góp ý của các Giảng viên để em ngày một cải thiện khả năng hơn trong học tập cũng như trong công việc sau này.

Một lần nữa em xin cảm ơn chân thành tới Ths Kiều Tuấn Dũng cùng các thầy cô giáo trong Khoa đã giúp em trong suốt quá trình hoàn thành đồ án tốt nghiệp này.

Em xin chân thành cảm ơn!

Hà Nội, Tháng 10 năm 2019

Sinh viên

Nguyễn Đức Thắng

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU CHUNG	1
1.1. Giới thiệu đề tài	1
1.2. Mục tiêu đề tài	1
1.3. Cơ sở đề tài	1
<i>1.3.1. Giới thiệu về PubNub</i>	2
<i>1.3.2. Cách thức hoạt động của PubNub</i>	2
<i>1.3.3. Kỹ thuật livestream sử dụng PubNub SDK</i>	3
<i>1.3.4. Bảng giá và thống kê</i>	10
CHƯƠNG 2: KIẾN THỨC CHUNG	12
2.1. WebRTC	12
<i>2.1.1. Firewall</i>	12
<i>2.1.2. NAT</i>	13
<i>2.1.4. TURN (Traversal Using NAT Relay)</i>	14
<i>2.1.5. ICE (Interactive Communication Establishment)</i>	15
2.2. NodeJS	15
<i>2.2.1. Ưu điểm</i>	16
<i>2.2.2. Nhược điểm</i>	16
2.3. MongoDB	17
<i>2.3.1. NoSQL</i>	17
<i>2.3.2. MongoDB là gì?</i>	17
<i>2.3.3. Ưu điểm</i>	18
<i>2.3.4. Nhược điểm</i>	19

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM.....	20
3.1. Định nghĩa lại bài toán	20
<i>3.1.1. Xác định vai trò người dùng</i>	<i>20</i>
<i>3.1.2. Các yêu cầu ứng dụng đối với người dùng</i>	<i>20</i>
3.2. Phân tích đặc tả yêu cầu và thiết kế database	20
<i>3.2.1. Đặc tả yêu cầu</i>	<i>20</i>
<i>3.2.2. Phân tích yêu cầu</i>	<i>25</i>
3.3. Thiết kế database.....	47
<i>3.3.1. Bảng users (người dùng)</i>	<i>47</i>
<i>3.3.2. Bảng gifts (quà tặng)</i>	<i>48</i>
<i>3.3.3. Bảng user_requests (yêu cầu của agent)</i>	<i>49</i>
<i>3.3.4. Bảng questions (câu hỏi)</i>	<i>50</i>
<i>3.3.5. Bảng answers (câu trả lời).....</i>	<i>50</i>
<i>3.3.6. Bảng user_histories (lịch sử chơi)</i>	<i>51</i>
CHƯƠNG 4: GIỚI THIỆU GIAO DIỆN ỨNG DỤNG	52
<i>4.1. Màn hình đăng nhập , đăng ký, quên mật khẩu</i>	<i>52</i>
<i>4.2. Màn hình trang chủ</i>	<i>53</i>
<i>4.3. Màn hình quản lý quà tặng</i>	<i>54</i>
<i>4.4. Màn hình quản lý câu hỏi</i>	<i>55</i>
<i>4.5. Màn hình quản lý yêu cầu.....</i>	<i>55</i>
<i>4.6. Màn hình quản lý người dùng</i>	<i>56</i>
<i>4.7. Màn hình cấu hình khung giờ</i>	<i>56</i>
<i>4.8. Màn hình xem danh sách lịch sử chơi</i>	<i>56</i>
<i>4.9. Màn hình thay đổi thông tin tài khoản.....</i>	<i>57</i>

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	58
5.1 Kết luận	58
5.2 Hướng phát triển đề tài	58
TÀI LIỆU THAM KHẢO	59

CHƯƠNG 1: GIỚI THIỆU CHUNG

1.1. Giới thiệu đề tài

Livestream ngày nay là công cụ được rất nhiều người đơn vị doanh nghiệp, hộ kinh doanh, cá nhân ưa chuộng. Bằng việc sử dụng tính năng này, chúng ta có thể trực tiếp giới thiệu với sản phẩm tới những người đang xem.

Chạy quảng cáo khi livestream là gì?

Livestream là chức năng phát trực tiếp các video giúp cho mọi người tại nơi khác đều có thể theo dõi được những chuyện xảy ra xung quanh chúng ta. Tính năng này được rất nhiều người sử dụng. Và nếu đã sử dụng Facebook chúng ta sẽ biết đến livestream tương tác tặng quà như “Conffeti Việt Nam” đang rộ lên phong trào xem – trả lời câu hỏi – được phần thưởng. Vì vậy đề án này sẽ xây dựng bộ sản phẩm dành cho website tương tác trực tiếp trên website nhằm phong phú thêm nền tảng. Bên cạnh đó, xây dựng sản phẩm này nhằm đáp ứng cung cấp dịch vụ cho các công ty, doanh nghiệp, cá nhân có nhu cầu quảng cáo sản phẩm cũng như câu hỏi liên quan đến sản phẩm của công ty, doanh nghiệp, cá nhân đó.

Lợi ích chạy quảng cáo khi livestream

Dễ dàng giới thiệu sản phẩm

Để quảng bá sản phẩm rộng rãi, nhiều người thường chỉnh sửa hình ảnh dẫn đến sự thiếu tin tưởng của khách hàng thì việc livestream sẽ giúp khách hàng tin tưởng hơn. Đồng thời, chúng ta vừa có thể giới thiệu sản phẩm, vừa có thể tương tác với khách hàng.

Khả năng tiếp cận cao

Như các chúng ta đã biết, khi bất cứ ai tiến hành livestream thì đều có thể chia sẻ đường dẫn đến các cộng đồng khác nhằm tăng khả năng tiếp cận đến những người xung quanh. Giúp sản phẩm của chúng ta được nhiều người biết đến hơn.

1.2. Mục tiêu đề tài

- Livestream bằng công nghệ Web RTC
- Socket realtime phân chat trên kênh
- Tự động trao thưởng cho từng user khi kết thúc tương tác

1.3. Cơ sở đề tài

Vì lý do về thời gian và chi phí cài đặt ICE Server vì vậy đề án này sẽ thực hiện kỹ thuật livestream thông qua PubNub có sẵn giao thức kết nối và ICE Server miễn phí (có giới hạn).

1.3.1. Giới thiệu về PubNub

PubNub cung cấp dịch vụ cơ sở hạ tầng thời gian thực và cung cấp bảo mật cấp doanh nghiệp, độ tin cậy được hỗ trợ 99,999% SLA và khả năng mở rộng toàn cầu để hỗ trợ triển khai realtime lớn nhất, tất cả thông qua API đơn giản và 70+ SDK. Ngoài ra, các nhà phát triển có thể tận dụng các tính năng giá trị gia tăng cho các lĩnh vực ứng dụng như trò chuyện, cộng tác, kiểm soát thiết bị IoT và cập nhật theo thời gian thực để tăng tốc thời gian xây dựng và phát hành.

1.3.2. Cách thức hoạt động của PubNub

Người dùng ngày nay không chỉ mong đợi trải nghiệm luôn bật từ các ứng dụng của họ, mà còn là một trải nghiệm nhất quán trên tất cả các thiết bị của họ. Các ứng dụng phổ biến nhất hiện nay - ứng dụng trò chuyện và xã hội, cộng tác tài liệu, trò chơi nhiều người chơi, IoT, chia sẻ đi xe, nền kinh tế theo yêu cầu, v.v. - được cung cấp bởi dữ liệu thời gian thực. Nhiều ứng dụng đang trở thành thời gian thực và nhiều người dùng hơn đang kết nối; Các công ty nghiên cứu hàng đầu như Gartner và McKinsey dự đoán một bước nhảy vọt từ 6 tỷ thiết bị được kết nối chúng ta có trên toàn thế giới hiện nay, lên 20-30 tỷ vào năm 2020. Và đây là những con số bảo thủ so với ước tính của các nhà lãnh đạo ngành khác như Cisco và Intel, người dự án gấp đôi số thiết bị

Kiến trúc yêu cầu / phản hồi ba tầng truyền thống không đáp ứng được nhu cầu về khả năng mở rộng, độ trễ và độ tin cậy của các ứng dụng truyền dữ liệu hiện đại. Kiến trúc ứng dụng thời gian thực kéo dài đòi hỏi những cách suy nghĩ mới về mạng.

PubNub là một mạng lập trình để phát triển các ứng dụng thời gian thực; một sự phát triển từ kiến trúc ba tầng, được xây dựng có mục đích để xử lý tất cả sự phức tạp của các luồng dữ liệu. PubNub hoạt động ở rìa mạng để xử lý và áp dụng logic cho dữ liệu thời gian thực, do đó giảm thiểu độ trễ xuống 100 mili giây hoặc ít hơn trên toàn thế giới và đảm bảo độ tin cậy và khả năng mở rộng.

PubNub có thể được sử dụng để nhanh chóng đẩy các tin nhắn nhỏ đến một hoặc nhiều thiết bị (điện thoại thông minh, máy tính bảng, máy tính để bàn, vi điều khiển, v.v.) - về cơ bản, chỉ về bất kỳ thiết bị nào có thể kết nối TCP / IP với internet - cũng như quay lại, cho giao tiếp hai chiều giữa các thiết bị. Các tin nhắn này có thể được sử dụng để liên

lạc với con người (như trò chuyện trực tuyến), điều khiển từ máy sang máy, IoT, định vị địa lý, nhà thông minh, dữ liệu tài chính, trò chơi nhiều người chơi, và nhiều hơn nữa.

PubNub biết rằng bảo mật, quyền sở hữu dữ liệu và kiểm soát truy cập rất quan trọng đối với bất kỳ ứng dụng thời gian thực nào - đó là lý do tại sao chúng tôi đã xây dựng toàn bộ Mạng luồng dữ liệu xung quanh các nguyên lý cốt lõi này.

PubNub DSN trao quyền cho các nhà phát triển cung cấp trải nghiệm 'luôn luôn, thời gian thực' vượt trội cho các ứng dụng của họ để các Kỹ sư, Kiến trúc sư, Nhà phát triển, Quản lý Sản phẩm và những người khác tập trung vào giải quyết các vấn đề kinh doanh thực tế cho công ty của họ và cho khách hàng của họ, thay vì lo lắng về mạng, cơ sở hạ tầng và các tài nguyên hỗ trợ cần thiết để duy trì Mạng luồng dữ liệu của riêng họ.

1.3.3. Kỹ thuật livestream sử dụng PubNub SDK

Bước 1: Tập tin HTML

```
<div class="row" style="margin: 5%">
  <div class="col-sm-6 col-md-6">
    <label id="labelQuestion"></label> <br>
    <label id="labelScore"></label> <br>
    <label id="labelTime"></label>
  </div>
  <div class="col-sm-6 col-md-6">
    <div id="stream-info" hidden="true">
      <label>Số người đang xem</label>
      
      <span id="here-now">0</span>
    </div>
  </div>
</div>

<div class="row">
  <button id="btnStream" class="btn btn-primary" onclick=" errWrap(stream);" hidden>Stream</button>
  <button id="btnWatch" class="btn btn-danger" onclick=" errWrap(watch);">Xem</button>
  <button class="btn btn-danger" id="end" onclick="end()" hidden="true">End Live</button>
  <!-- <div id="logs" class="ptext" style="background-color:white"></div> -->

  <button id="btnStart" class="btn btn-primary" data-name="start" hidden>Bắt đầu</button>
</div>
```

Hình 1: Tập tin HTML

Tại bước này, chúng ta tạo ra các nút “Stream” và nút “Watch” dùng để phát sóng hoặc xem livestream. Bên cạnh đó có nút “End Live” để kết thúc luồng phát sóng.

Số người xem sẽ được hiển thị tại thẻ div “stream-info”

Các hàm cần thiết:

- errWrap(stream)
- errWrap(watch)
- end()

Bước 2: Tập tin JavaScript

Có ba thư viện mà chúng ta sẽ cần bao gồm để thực hiện các hoạt động WebRTC dễ dàng hơn nhiều:

Bao gồm jQuery để làm cho các phần tử DOM sửa đổi trở nên dễ dàng.

Bao gồm SDK JavaScript PubNub để tạo điều kiện cho tín hiệu WebRTC.

Bao gồm PubNub WebRTC SDK và các thư viện SDK Wrapper giúp việc thực hiện các cuộc gọi điện thoại đơn giản như gọi dial(number) chức năng.

Bước 3: Phát sóng luồng

Để bắt đầu truyền phát luồng, chúng ta sẽ cần một khóa xuất bản và đăng ký. Để nhận khóa pub / sub của chúng ta, trước tiên cần đăng ký tài khoản PubNub . Khi đăng ký, chúng ta có thể tìm thấy các khóa PubNub duy nhất của mình trong Bảng điều khiển dành cho nhà phát triển PubNub . Tầng Sandbox miễn phí sẽ cung cấp cho chúng ta tất cả bằng thông chúng ta cần để xây dựng và kiểm tra Ứng dụng WebRTC của chúng ta.

a. Chuẩn bị phát sóng

Trước tiên, hãy xác định vị trí chủ sở hữu video cho luồng cục bộ, chủ sở hữu mã nhúng và chủ sở hữu hiện tại để cập nhật hiện diện. Ngoài ra, hãy tạo một biến toàn cục sẽ giữ tên luồng hiện tại của mình.

b. Phát sóng & Cấu hình

Bây giờ, đã đến lúc thực hiện stream chức năng của chúng tôi sẽ thiết lập kết nối WebRTC để phát sóng.

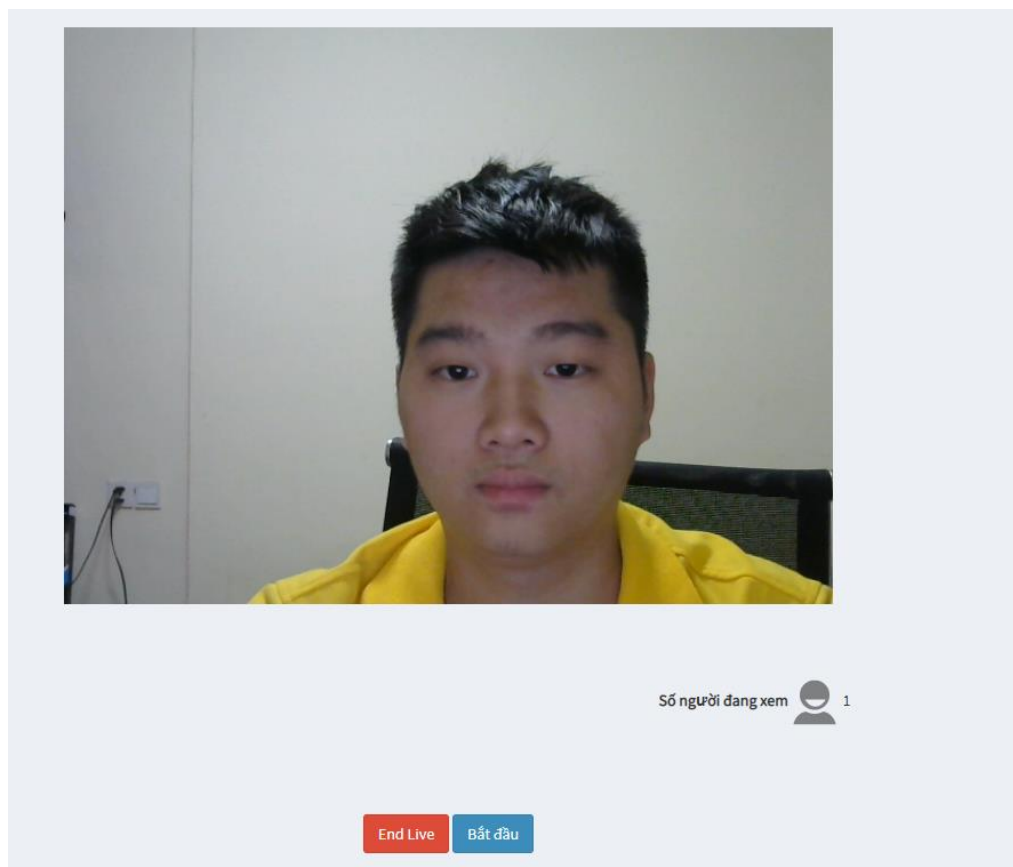
Bây giờ chúng ta đang truyền phát video của mình trên kênh streamName. Nếu nhớ khởi tạo phone đối tượng trong quá khứ, bạn sẽ nhận thấy rằng đã thêm một số cấu hình mới oneway và broadcast. Các oneway cấu hình đơn giản đặt SDK WebRTC được đắp cho khả năng một chiều streaming.

Sau đó, `broadcast` được đặt thành true để chỉ ra rằng `phone` sẽ là người phát sóng, không nhận được luồng. Chỉ báo `oneway` sẽ cho phép người dùng xem luồng của chúng ta mà không cần phải cấp quyền mic và video.

Các `ctrl.ready` chức năng phải mất một callback để thực hiện khi nó đã sẵn sàng để phát trực tuyến. Đầu tiên, chỉ cần thay đổi đầu vào thành màu xanh lá cây để biểu thị thành công và ấn nút gửi. Sau đó, sẽ sử dụng `ctrl.addLocalStream(video_out)` để đặt luồng cục bộ trong `div videoFor`. Cuối cùng, một cuộc gọi để `ctrl.stream` đăng ký vào một kênh truyền phát để bắt đầu phát sóng.

Điều cuối cùng chúng ta làm trong `stream` chức năng của mình là tạo một bộ `ctrl.streamPresence(function(m){...})` xử lý. Điều này đăng ký vào các sự kiện hiện diện trên luồng của chúng ta, sau đó trao tất cả các sự kiện hiện diện cho chức năng gọi lại. Chỉ đơn giản là kéo ra khỏi phòng hiện tại và hiển thị nó trong `div here_now`.

Để tham khảo thêm, hãy xem [tài liệu về PubNub Presence](#).



Hình 2: Mô tả phát sóng

Bước 4: Xem phát sóng

Bây giờ chúng ta có một luồng người dùng (người phát sóng), chúng ta cần cho phép những người dùng khác tham gia và xem luồng đó. Các `watch` chức năng có giá trị từ đầu vào HTML và gia nhập dòng đó.

Thiết lập ban đầu trông rất giống nhau, nhưng chúng ta sẽ nhận thấy chỉ xác định `oneway` thời gian này. Người dùng này chỉ nhận được luồng, không phát sóng là tốt.

Khi bộ điều khiển đã sẵn sàng để kết nối, nó sử dụng `ctrl.isStreaming(number, function(isStrm){})` để đảm bảo rằng người dùng họ muốn xem hiện đang phát trực tuyến. Hàm này có một số và gọi lại. Nó kiểm tra nếu số đó đang phát trực tuyến và chuyển một giá trị boolean `isStrm` cho hàm gọi lại. Nếu người dùng đang phát trực tuyến, chúng ta sử dụng `ctrl.joinStream` để tham gia vào luồng. Không cần cấp quyền truy cập vào mic và video vì chúng ta chỉ xem luồng.

Sau đó, chúng ta sử dụng `ctrl.streamPresence` lại để cập nhật lượng người dùng hiện đang xem luồng qua PubNub Presence.

Và cứ như thế, chúng ta đã xây dựng một ứng dụng video phát sóng trực tiếp thời gian thực với WebRTC.

Mã code (JavaScript)

```
var video_out = document.getElementById("vid-box")
var embed_code = document.getElementById("embed-code")
var embed_demo = document.getElementById("embed-demo")
var here_now = document.getElementById('here-now')
var stream_info = document.getElementById('stream-info')
var end_stream = document.getElementById('end')

var streamName

function stream() {
  //Kiểm tra xem user đó có được livestream ko
  callAPI('userrequest/check', 'GET', 'stream=played', token, null, (res) => {
    if(!res.success)
    {
      alert(res.error)
    }
    else
    {

```

```

        $('#btnStart').show()
        $('#btnStream').hide()
        $('#btnWatch').hide()
        request = res.data
        streamName = "a" //Tên phòng
        var phone = window.phone = PHONE({
            number      : streamName, // listen on username line else random
            publish_key  : 'pub-c-561a7378-fa06-4c50-a331-5c0056d0163c', // Your Pub Key
            subscribe_key : 'sub-c-17b7db8a-3915-11e4-9868-02ee2ddab7fe', // Your Sub Key
            oneway        : true,
            broadcast     : true,
        })
        //phone.debug(function(m){ console.log(m) })
        var ctrl = window.ctrl = CONTROLLER(phone, get_xirsys_servers)
        ctrl.ready(function(){
            ctrl.addLocalStream(video_out)
            ctrl.stream()
            stream_info.hidden=false
            end_stream.hidden =false
            //addLog("Đang stream đến phòng " + streamName)
        })

        ctrl.receive(function(session){
            session.connected(function(session){
                //addLog(session.number + " đã tham gia")
            })
            session.ended(function(session) {
                //addLog(session.number + " đã rời đi")
                console.log(session)
            })
        })

        ctrl.streamPresence(function(m){
            here_now.innerHTML=m.occupancy
            //addLog(m.occupancy + " đang xem")
        })
    }
})
return false
}

function watch(form){

```

```

var num = "a"
var phone = window.phone = PHONE({
    number      : "Viewer" + Math.floor(Math.random()*100), // listen on us
    urname line else random
    publish_key  : 'pub-c-6265d3da-b817-441c-8722-
d5aa3b0c94b2', // Your Pub Key
    subscribe_key : 'sub-c-99870162-9da7-11e9-9aea-
96d77c97c6d4', // Your Sub Key
    oneway       : true
})

var ctrl = window.ctrl = CONTROLLER(phone, get_xirsys_servers)
ctrl.ready(function(){
    ctrl.isStreaming(num, function(isOn){
        if (isOn)
        {
            ctrl.joinStream(num)
            $('#btnWatch').hide()
        }
        else alert("Người dùng không livestream")
    })
})

ctrl.receive(function(session){

    session.connected(function(session){
        video_out.appendChild(session.video)
        //addLog(session.number + " đã tham gia")
        stream_info.hidden=false
    })

    session.ended(function(session) {
        //addLog(session.number + " đã rời đi")
    })
})

ctrl.streamPresence(function(m){
    here_now.innerHTML=m.occupancy
    //addLog(m.occupancy + " số người đang xem")
})

return false
}

```

```

function getVideo(number){
    return $('*[data-number="'+number+'"]')
}

function addLog(log){
    $('#logs').append("<p>" + log + "</p>")
}

function end(){

    if (!window.phone) return
    ctrl.hangup()
    $('#btnStream').show()
    $('#btnStart').hide()
    $('#btnWatch').hide()
    stream_info.hidden = true
    video_out.innerHTML = ""
    // phone.pubnub.unsubscribe() // unsubscribe all?
}

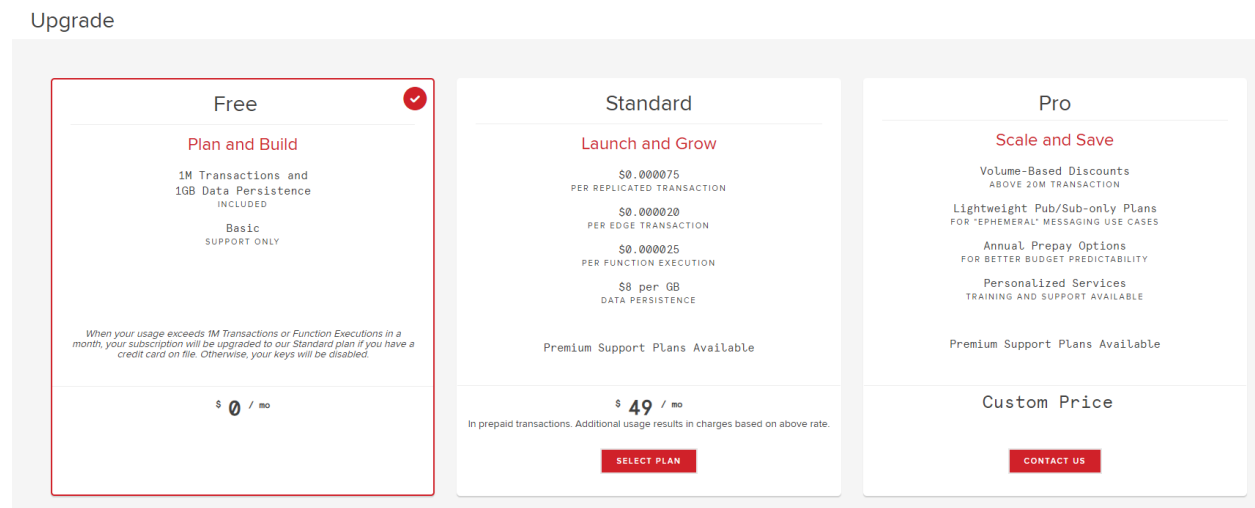
function get_xirsys_servers() {
    var servers

    $.ajax({
        type: 'POST',
        url: 'https://service.xirsys.com/ice',
        data: {
            room: 'default',
            application: 'default',
            domain: 'kevingleason.me',
            ident: 'gleasonk',
            secret: 'b9066b5e-1f75-11e5-866a-c400956a1e19',
            secure: 1,
        },
        success: function(res) {
            console.log(res)
            res = JSON.parse(res)
            if (!res.e) servers = res.d.iceServers
        },
        async: false
    })
    return servers
}

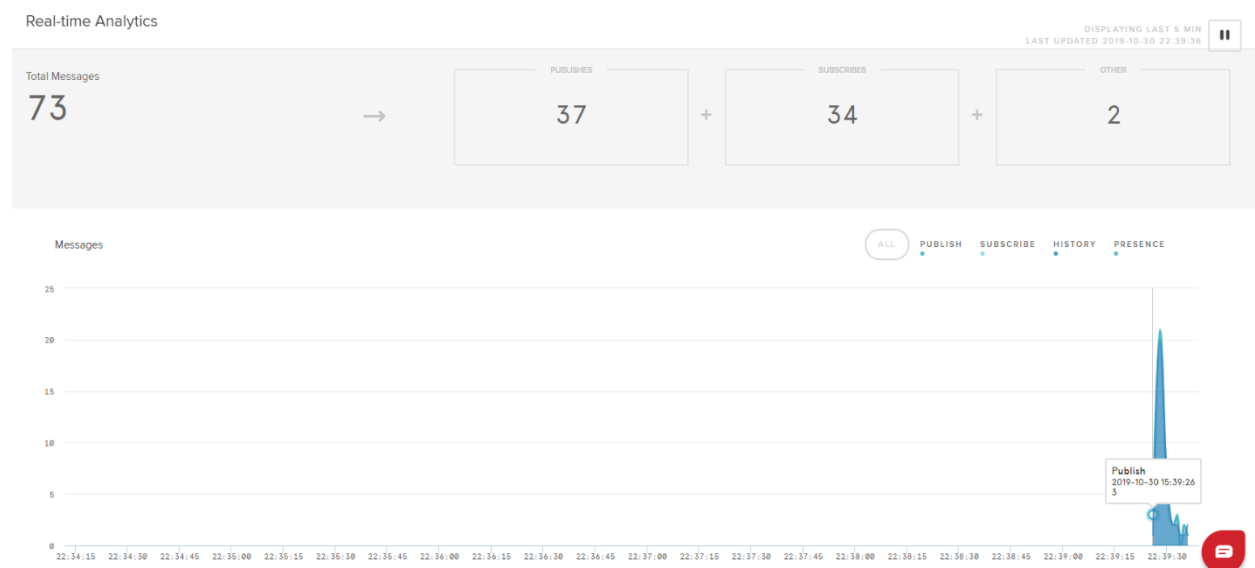
```

```
function errWrap(fxn){
  try {
    return fxn()
  } catch(err) {
    alert("WebRTC is currently only supported by Chrome, Opera, and Firefox")
    return false
  }
}
```

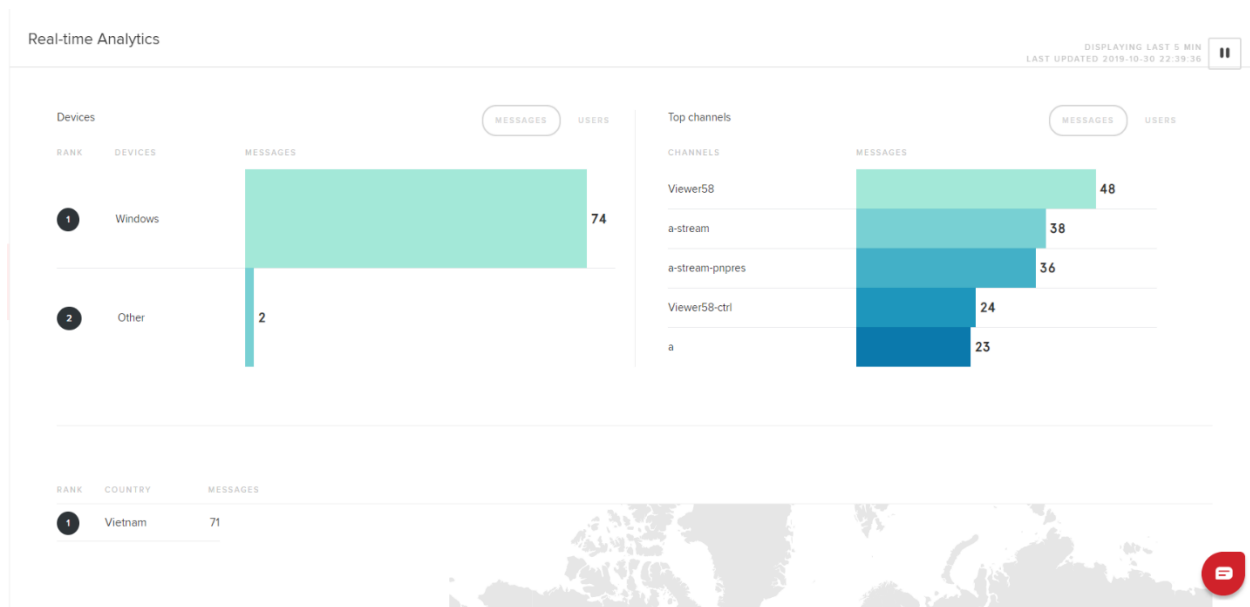
1.3.4. Bảng giá và thống kê



Hình 3: Bảng giá dịch vụ của PubNub



Hình 4: Biểu đồ thống kê với lượng người xem live là 3 trong khoảng 1p



Hình 5: Thống kê với lượng người xem live là 3 trong khoảng 1p

CHƯƠNG 2: KIẾN THỨC CHUNG

Các công nghệ được sử dụng trong phát triển website:

- **WebRTC**
- **NodeJS**
- **MongoDB**
- **Jquery**

2.1. WebRTC

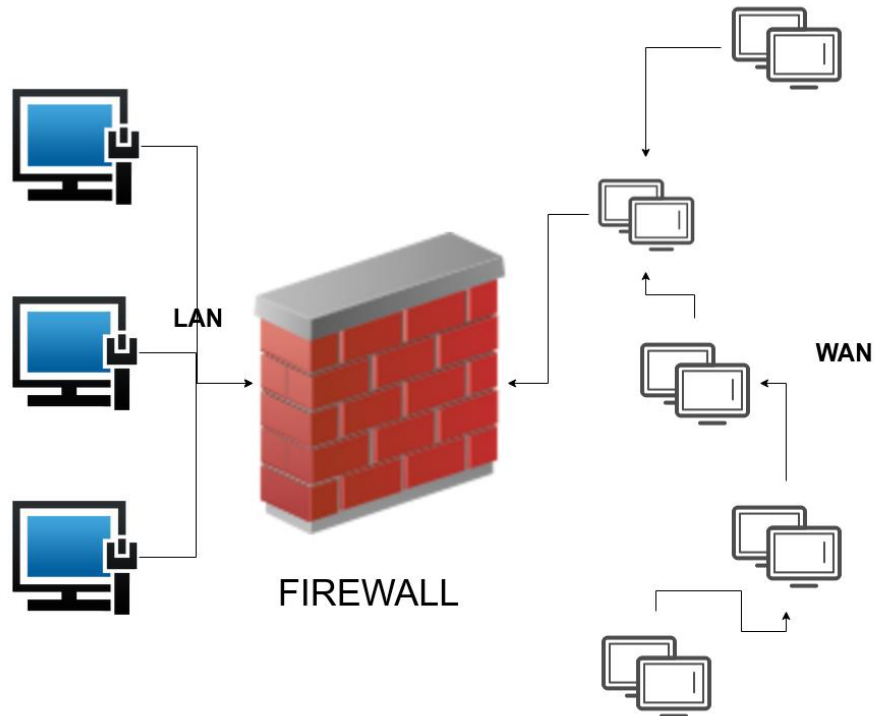
WebRTC là Web Real-Time Communication: "và là một web API được phát triển bởi World Wide Web Consortium (W3C), khả năng hỗ trợ trình duyệt (browser) giao tiếp với nhau thông qua VideoCall, VoiceCall hay transfer data "Peer-to-Peer" (P2P) mà không cần browser phải cài thêm plugins hay phần mềm hỗ trợ nào từ bên ngoài. Là tập hợp các tiêu chuẩn và giao thức cho phép các trình duyệt Web thực hiện trực tiếp các tính năng truyền thông đa phương tiện thời gian thực như gọi điện, tin nhắn hình, truyền dữ liệu bằng các API JavaScript." trích từ **Wikipedia**.

WebRTC truyền dữ liệu "Peer-to-Peer" ví dụ dễ hiểu là nếu A muốn gửi thư với B thì bước đầu tiên A và B phải biết địa chỉ nhà của nhau để mà gửi thư. Vậy việc đầu tiên cần làm là phải tạo kết nối "Peer-to-Peer"

Thông thường thì nếu gia đình của A và B đều thoải mái, cho con cháu giao lưu tứ tung thì vô tư đi A cứ gửi thư cho B và B reply cho A. Đơn giản thế mà trôi đi đơn giản như vậy. Nhưng đối với một số gia đình hơi khó khăn tý thì việc tạo kết nối "Peer-to-Peer" giữa A và B sẽ bị cản trở. Nên giống như các hệ thống VoIP, **WebRTC** cũng bị cản trở khi tạo kết nối peer-to-peer bởi tường lửa và NAT. Vậy Firewall(tường lửa) là gì và NAT là gì?

2.1.1. Firewall

Là biện pháp bảo vệ an toàn cho máy tính của bạn, nó giống như một bức tường ngăn cách máy tính của bạn hoặc mạng cục bộ (local network) với mạng khác như Internet.



King

Hình 6: Firewall

2.1.2. NAT

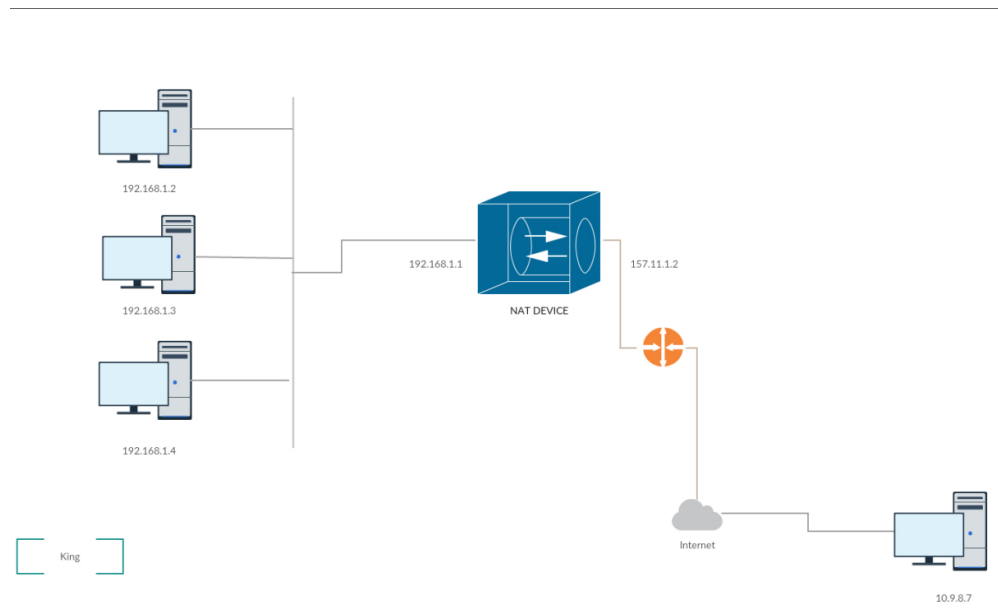
Như chúng ta đã biết thì có 2 loại IP là IP public và IP private, các máy trong mạng LAN được đặt IP Private vì máy IP private không tồn tại ngoài Internet (Vd: 192.168.1.1, 192.168.1.2 etc) . Và IP public thì tất nhiên tồn tại trên Internet rồi vì chúng ta đã đăng ký với ISP(nhà cung cấp dịch vụ Internet). Thông thường NAT thường thay đổi địa chỉ thường là địa chỉ riêng (IP Private) của một kết nối mạng thành địa chỉ công cộng (IP Public).

Vậy khi A gửi B một gói tin với thông tin tóm lược như : IP nguồn (IP của A) : 192.168.1.1 , IP đích: (113.xxx.y.z)

Mà như chúng ta biết là IP private không tồn tại ngoài Internet nên chúng ta cần NAT để thay thế IP private của A thành IP public.

Hiện tại hiểu đơn giản là "NAT sử dụng IP của chính nó làm IP công cộng cho mỗi máy con (client) với IP riêng. Khi một máy con thực hiện kết nối hoặc gửi dữ liệu tới một máy tính nào đó trên internet, dữ liệu sẽ được gửi tới NAT, sau đó NAT sẽ thay thế địa chỉ IP gốc của máy con đó rồi gửi gói dữ liệu đi với địa chỉ IP của NAT. Máy tính từ xa hoặc máy tính nào đó trên internet khi nhận được tín hiệu sẽ gửi gói tin trở về cho NAT computer bởi vì chúng nghĩ rằng NAT computer là máy đã gửi những gói dữ liệu đi. NAT ghi lại bảng

thông tin của những máy tính đã gửi những gói tin đi ra ngoài trên mỗi cổng dịch vụ và gửi những gói tin nhận được về đúng máy tính đó (client).



Hình 7: NAT

2.1.3. STUN (Simple Traversal Of UDP Through NAT)

Tiếp theo là STUN nhé, mấy cái khái niệm này rất quan trọng, nắm chắc thì khi implement rất dễ dàng. STUN thì là khi một máy chủ nào xài NAT (behind NAT) thì STUN server sẽ giúp cho client đó biết được địa chỉ IP và Port mà thiết bị NAT sử dụng. Và từ đó giúp cho các peer có thể lấy được địa chỉ của peer khác (IP nào, cổng mấy, NAT loại gì) để mà vượt.

Nhưng STUN có một nhược điểm là nó không support Symmetric NAT (NAT có nhiều loại), nhưng đừng lo "Không làm được thì TURN, TURN biến hình"

2.1.4. TURN (Traversal Using NAT Relay)

Cũng giống như STUN tuy nhiên TURN hỗ trợ cả giao thức TCP làm giao thức truyền tải. TURN bổ xung cho hạn chế của STUN là hỗ trợ Symmetric NAT. Dữ liệu thay vì được gửi trực tiếp tới các peer thì các peer sẽ gửi dữ liệu tới các TURN server và TURN

server sẽ đóng vai trò trung gian vận chuyển gói tin. Điều này nâng cao giúp chất lượng dịch vụ của ứng dụng mà còn đảm bảo an toàn thông tin khi truyền dẫn.

Nhưng cái gì cũng có hai mặt. Nhược điểm của TURN là chi phí sử dụng lớn, vì sẽ có một lưu lượng băng thông lớn được sử dụng. Nhất là với chất lượng full HD hay video HD nữa.

2.1.5. ICE (*Interactive Communication Establishment*)

ICE là một giao thức được dùng để thiết lập phiên media dựa trên UDP đi qua NAT một cách nhanh nhất.

ICE sẽ tìm đường tốt nhất để kết nối giữa các peer, nó thử tất cả khả năng có thể kết nối một cách song song và lựa chọn con đường hiệu quả nhất.

Đầu tiên nó sẽ cố gắng tạo ra một kết nối bằng cách sử dụng địa chỉ thu được từ hệ điều hành và card mạng của thiết bị, nếu không thành công (có thể thiết bị đăng sau NAT) thì ICE sẽ lấy địa chỉ bên ngoài của thiết bị bằng cách sử dụng máy chủ STUN, nếu không thành công nữa thì nó sẽ chuyển lưu lượng mạng qua một máy chủ chuyển tiếp là TURN.

Nói dễ hiểu hơn là ICE xài STUN xong không được thì đi xài TURN.

2.2. NodeJS

Node.js là một nền tảng chạy trên môi trường V8 JavaScript runtime - một trình thông dịch JavaScript cực nhanh chạy trên trình duyệt Chrome. Bình thường thì bạn cũng có thể tải bộ V8 và nhúng nó vào bất cứ thứ gì; Node.js làm điều đó đối với các web server. JavaScript suy cho cùng cũng chỉ là một ngôn ngữ - vậy thì không có lý do gì để nói nó không thể sử dụng trên môi trường server tốt như là trong trình duyệt của người dùng được.

Trong một môi trường server điển hình LAMP (Linux-Apache-MySQL-PHP), bạn có một web server là Apache hoặc NGINX nằm dưới, cùng với PHP chạy trên nó. Mỗi một kết nối tới server sẽ sinh ra một thread mới, và điều này khiến ứng dụng nhanh chóng trở nên chậm chạp hoặc quá tải - cách duy nhất để hỗ trợ nhiều người dùng hơn là bằng cách bổ sung thêm nhiều máy chủ. Đơn giản là nó không có khả năng mở rộng tốt. Nhưng với Node.js

thì điều này không phải là vấn đề. Không có một máy chủ Apache lắng nghe các kết nối tới và trả về mã trạng thái HTTP - bạn sẽ phải tự quản lý kiến trúc lõi của máy chủ đó. May mắn thay, có một số module giúp thực hiện điều này được dễ dàng hơn, nhưng công việc này vẫn gây cho một chút khó khăn khi mới bắt đầu. Tuy nhiên, kết quả thu được là một ứng dụng web có tốc độ thực thi cao.

Một khái niệm cốt lõi của Node.js đó là các function bất đồng bộ (asynchronous functions) - vì vậy về cơ bản thì mọi thứ chạy trên nền tảng này. Với hầu hết các ngôn ngữ kịch bản máy chủ, chương trình phải đợi mỗi function thực thi xong trước khi có thể tiếp tục chạy tiếp. Với Node.js, việc xác định các function sẽ chạy để hoàn thành một tác vụ nào đó, trong khi phần còn lại của ứng dụng vẫn chạy đồng thời. Nó là một chủ đề phức tạp, nhưng đó là một trong những đặc trưng tiêu biểu của Node.js, vì vậy việc nắm vững nó là điều hết sức quan trọng.

2.2.1. Ưu điểm

Đầu tiên là ưu điểm về tốc độ thực thi và khả năng mở rộng. Node.js có tốc độ rất nhanh. Đó là một yêu cầu khá quan trọng khi bạn là một startup đang cố gắng tạo ra một sản phẩm lớn và muốn đảm bảo có thể mở rộng nhanh chóng, đáp ứng được một lượng lớn người dùng khi trang web của bạn phát triển lên.

Node.js có thể xử lý hàng ngàn kết nối đồng thời trong khi PHP sẽ chỉ có nước sục sôi. Bên cạnh các lợi ích về tốc độ thực thi và khả năng mở rộng, có thể bạn cũng đã biết một chút về JavaScript, vì vậy tại sao lại phải phiên toái để học thêm về một ngôn ngữ lập trình hoàn toàn mới như PHP? Và sau đó bạn sẽ có một sự phấn khích khi học về một cái gì đó mới mẻ và gần như chưa được khám phá. Bạn còn nhớ cái cảm giác khi mà một cái gì đó mới xuất hiện và sau đó trở thành phổ biến khắp mọi nơi mà bạn hối tiếc đã không học về nó sớm hơn, và mãi mãi chỉ là người đến sau? Đừng phạm phải sai lầm như vậy lần này nữa. Node.js đang ngày càng trở nên lớn mạnh hơn.

2.2.2. Nhược điểm

Giống như hầu hết các công nghệ mới, việc triển khai Node.js trên host không phải là điều dễ dàng. Nếu chúng ta có một web hosting xài chung, chúng ta không thể đơn giản tải lên một ứng dụng Node.js và mong chờ nó hoạt động tốt. VPS và dedicated server là một sự lựa chọn tốt hơn – chúng ta có thể cài đặt Node.js trên chúng. Thậm chí dễ hơn là sử dụng một dịch vụ có khả năng mở rộng như là Heroku, và chúng ta có thể hoàn toàn an tâm để phát triển trang web của mình trên đó – chúng ta chỉ cần trả tiền khi cần thêm nhiều tài nguyên hơn.

Mặt khác, chúng ta rất dễ cài đặt Node.js chạy cục bộ trên máy tính của chúng ta sử dụng các hệ điều hành như Windows, Mac hoặc Linux và bắt đầu phát triển ứng dụng ngay lập tức - chỉ việc tải phiên bản Node.js tương ứng. Một điều quan trọng nên chú ý là Node.js không chỉ đơn giản là một sự thay thế cho Apache - các ứng dụng web đang tồn tại sẽ không

có khả năng tương thích, và chúng ta sẽ làm việc hiệu quả với những ứng dụng phát triển từ đầu (*mặc dù có rất nhiều framework ngoài kia để giúp đỡ chúng ta với nhiều đặc trưng phổ biến*).

Một nhược điểm lớn khác của Node.js đó là nó vẫn đang trong giai đoạn phát triển ban đầu, điều này có nghĩa là một số đặc trưng sẽ thay đổi trong quá trình phát triển tiếp theo. Trong thực tế, nếu chúng ta đọc các tài liệu đi kèm, thì nó bao gồm một chỉ số ổn định (stability index), chỉ số này cho thấy mức độ rủi ro khi chúng ta sử dụng các đặc trưng hiện có.

2.3. MongoDB

2.3.1. NoSQL

NoSQL là 1 dạng CSDL mã nguồn mở và được viết tắt bởi: None-Relational SQL hay có nơi thường gọi là Not-Only SQL.

NoSQL được phát triển trên Javascript Framework với kiểu dữ liệu là JSON và dạng dữ liệu theo kiểu key và value.

NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu sót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS (Relational Database Management System - Hệ quản trị cơ sở dữ liệu quan hệ) về tốc độ, tính năng, khả năng mở rộng,...

Với NoSQL bạn có thể mở rộng dữ liệu mà không lo tới những việc như tạo khóa ngoại, khóa chính, kiểm tra ràng buộc .v.v ...

NoSQL bỏ qua tính toàn vẹn của dữ liệu và transaction để đổi lấy hiệu suất nhanh và khả năng mở rộng.

NoSQL được sử dụng ở rất nhiều công ty, tập đoàn lớn, ví dụ như FaceBook sử dụng Cassandra do FaceBook phát triển, Google phát triển và sử dụng BigTable,...

2.3.2. MongoDB là gì?

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.

MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là **collection** thay vì bảng. So với RDBMS thì trong MongoDB **collection** ứng

với **table**, còn **document** sẽ ứng với **row** , MongoDB sẽ dùng các document thay cho row trong RDBMS. Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định. Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB.

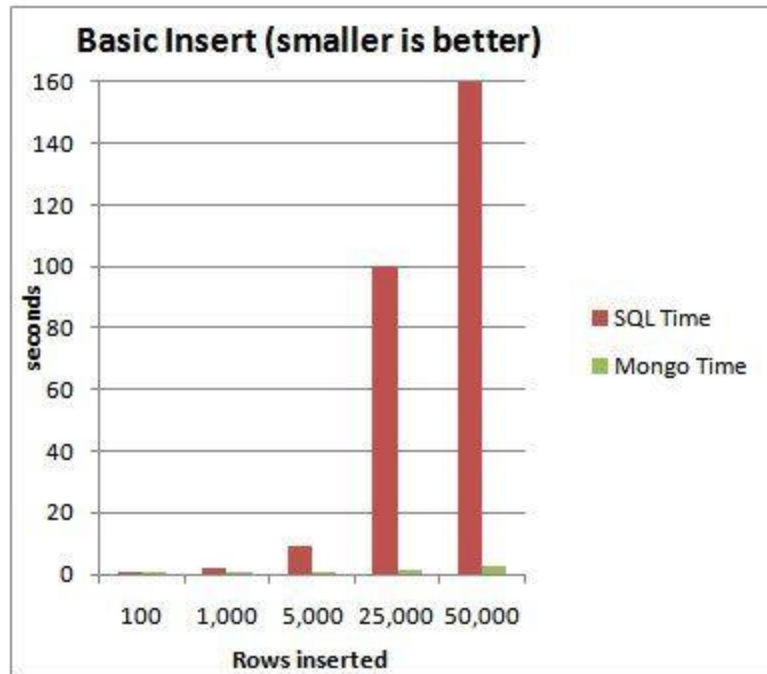
2.3.3. Ưu điểm

Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau, linh hoạt trong việc lưu trữ dữ liệu, nên bạn muốn gì thì cứ insert vào thoải mái. Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS. MongoDB rất dễ mở rộng (Horizontal Scalability). Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster:

Trường dữ liệu “_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.

Khi có một truy vấn dữ liệu, bản ghi được cached lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.

Hiệu năng cao: Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL.



Hình 8: So sánh tốc độ insert giữa MySQL và MongoDB

2.3.4. Nhược điểm

Một ưu điểm của MongoDB cũng chính là nhược điểm của nó. MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với mongoDB thì phải hết sức cẩn thận.

Tồn bộ nhớ do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên dễ bị dư thừa dữ liệu.

Khi insert/update/remove bản ghi, MongoDB sẽ chưa cập nhật ngay xuống ổ cứng, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng điều này sẽ là nhược điểm vì sẽ có nguy cơ bị mất dữ liệu khi xảy ra các tình huống như mất điện...

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM

3.1. Định nghĩa lại bài toán

3.1.1. Xác định vai trò người dùng

Người dùng được chia làm 3 loại: Normal, Agent, Admin.

- Người dùng **Normal**: Có quyền lựa chọn các kênh đang phát sóng để tham gia tương tác trả lời câu hỏi và nhận phần thưởng.
- Người dùng **Agent**: Có quyền lên sóng bắt đầu cho phép người dùng **Normal** tương tác với kênh của mình.
- Người dùng **Admin**: Có quyền quản lý cao nhất về mọi mặt trên hệ thống.

3.1.2. Các yêu cầu ứng dụng đối với người dùng

- Quản lý câu hỏi, quản lý yêu cầu, quản lý, quản lý quà tặng: Đăng nhập tài khoản **Agent** để thao tác thêm, sửa, xóa.
- Duyệt yêu cầu của **Agent**, quản lý người dùng: Đăng nhập tài khoản **Admin** để thao tác duyệt, khóa tài khoản.
- Người dùng Normal chỉ sử dụng tính năng chat và thao tác những tính năng cơ bản không có quyền thao tác quản lý.

3.2. Phân tích đặc tả yêu cầu và thiết kế database

3.2.1. Đặc tả yêu cầu

3.2.1.1. Đăng nhập

Mô tả	Đăng nhập và phân quyền trên hệ thống
Đầu vào	Tên đăng nhập và mật khẩu

Xử lý	Hệ thống nhận thông tin tài khoản, mật khẩu. Kiểm tra tài khoản có tồn tại trên hệ thống hay không? Nếu tồn tại cho vào hệ thống. Ngược lại, thông báo lỗi đăng nhập cho người dùng
Đầu ra	Thành công: Vào sử dụng hệ thống Thất bại: Thông báo lỗi

3.2.1.2. Thay đổi thông tin tài khoản

Mô tả	Tất cả người dùng trên hệ thống đều có quyền thay đổi thông tin tài khoản
Đầu vào	Id người dùng, họ tên
Xử lý	Hệ thống dựa vào id người dùng để cập nhật họ tên cho tài khoản người dùng
Đầu ra	Thành công: Thay đổi thông tin tài khoản thành công Thất bại: Thông báo lỗi

3.2.1.3. Quản lý người dùng

Mô tả	Admin có quyền khóa, kích hoạt người dùng trên hệ thống
Đầu vào	Id người dùng và trạng thái cập nhật (lock, active)

Xử lý	Hệ thống xác định người tác động lên người dùng là admin hay không? Nếu đúng cho phép thực hiện cập nhật trạng thái người dùng theo trạng thái truyền lên.
Đầu ra	Thành công: Cập nhật trạng thái người dùng (lock, active) Thất bại: Thông báo lỗi

3.2.1.4. Quản lý yêu cầu

Mô tả	<p>Admin chỉ có quyền xem thông tin yêu cầu, danh sách câu hỏi, duyệt và hủy duyệt yêu cầu.</p> <p>Agent có quyền thêm, sửa, xóa, yêu cầu của chính tài khoản đó.</p>
Đầu vào	Thông tin yêu cầu
Xử lý	<p>Hệ thống xác định quyền Admin sẽ cho phép xem thông tin yêu cầu, danh sách câu hỏi, duyệt và hủy duyệt yêu cầu.</p> <p>Hệ thống xác định quyền Agent sẽ cho phép thêm, sửa, xóa yêu cầu của chính tài khoản đó. Nếu yêu cầu đó đã hoặc đang được phát sẽ không được chỉnh sửa</p>
Đầu ra	<p>Thành công: Thông báo thành công các thao tác</p> <p>Thất bại: Thông báo lỗi</p>

3.2.1.5. Quản lý câu hỏi

Mô tả	Chỉ Agent có quyền thêm câu hỏi, câu trả lời vào trong các yêu cầu livestream của Agent đó
Đầu vào	Thông tin câu hỏi, câu trả lời và id yêu cầu
Xử lý	Hệ thống xác định quyền Agent tại hệ thống cho phép thêm câu hỏi và câu trả lời.
Đầu ra	Thành công: Thông báo thành công các thao tác Thất bại: Thông báo lỗi

3.2.1.6. Quản lý quà tặng

Mô tả	Chỉ Agent có quyền thêm, sửa, xóa quà tặng.
Đầu vào	Thông tin quà tặng
Xử lý	Hệ thống xác định quyền Agent tại hệ thống cho phép thêm, sửa, xóa quà tặng.
Đầu ra	Thành công: Thông báo thành công các thao tác Thất bại: Thông báo lỗi

3.2.1.7. Livestream

Mô tả	Agent livestream và bật câu hỏi để người dùng tương tác
-------	--

Đầu vào	Agent có yêu cầu được duyệt và camera thao tác trên web.
Xử lý	Hệ thống xét duyệt Agent có yêu cầu thuộc thời gian hiện tại hay không?. Nếu có duyệt camera của Agent bắt đầu cho phép Agent thao tác livestream trên kênh.
Đầu ra	Thành công: Thành công các thao tác Thất bại: Thông báo lỗi

3.2.1.8. Đăng xuất

Mô tả	Cho phép người dùng đăng xuất khỏi hệ thống
Đầu vào	Người dùng chọn nút đăng xuất
Xử lý	Xóa phiên đăng nhập và đăng xuất khỏi hệ thống
Đầu ra	Thành công: Thông báo thành công các thao tác và quay lại màn hình đăng nhập Thất bại: Thông báo lỗi

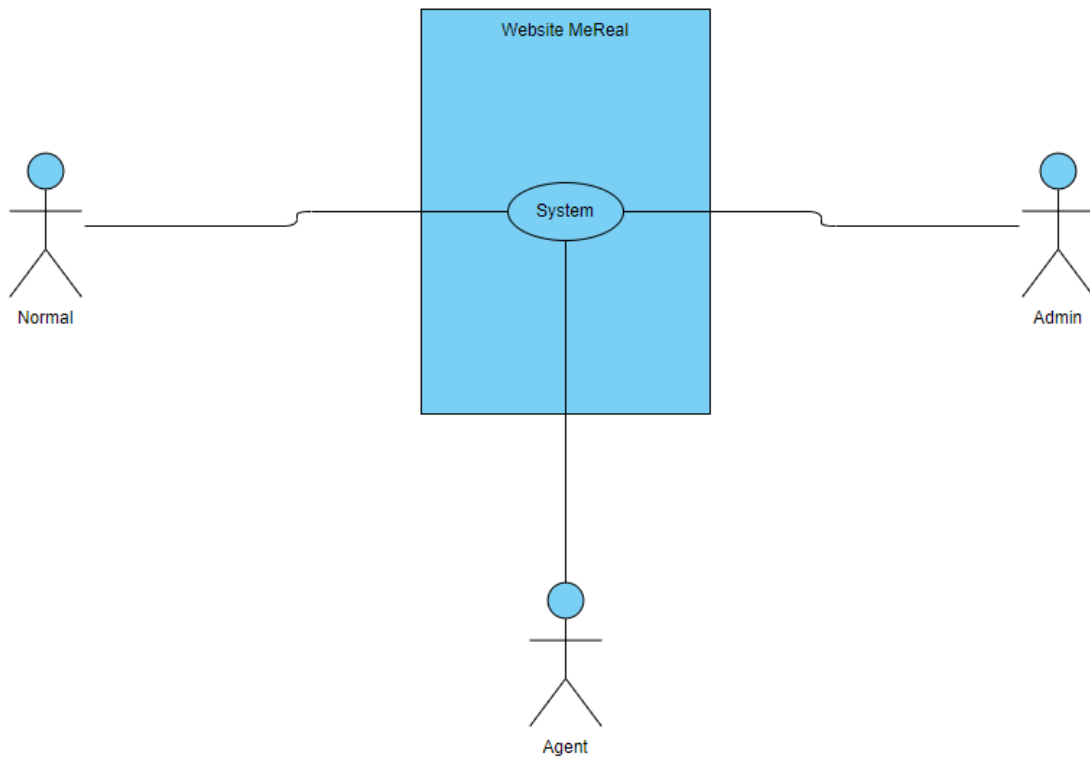
3.2.1.9. Quên mật khẩu/Đổi mật khẩu

Mô tả	Cho phép người dùng quên mật khẩu và đổi mật khẩu
Đầu vào	Email người dùng, tên đăng nhập, mật khẩu mới

Xử lý	<p>Quên mật khẩu: Hệ thống sẽ gửi email đường dẫn đổi mật khẩu theo email người dùng truyền lên.</p> <p>Đổi mật khẩu: Hệ thống sẽ đổi mật khẩu nếu mật khẩu cũ đúng</p>
Đầu ra	<p>Thành công: Thông báo thành công các thao tác và quay lại màn hình đăng nhập</p> <p>Thất bại: Thông báo lỗi</p>

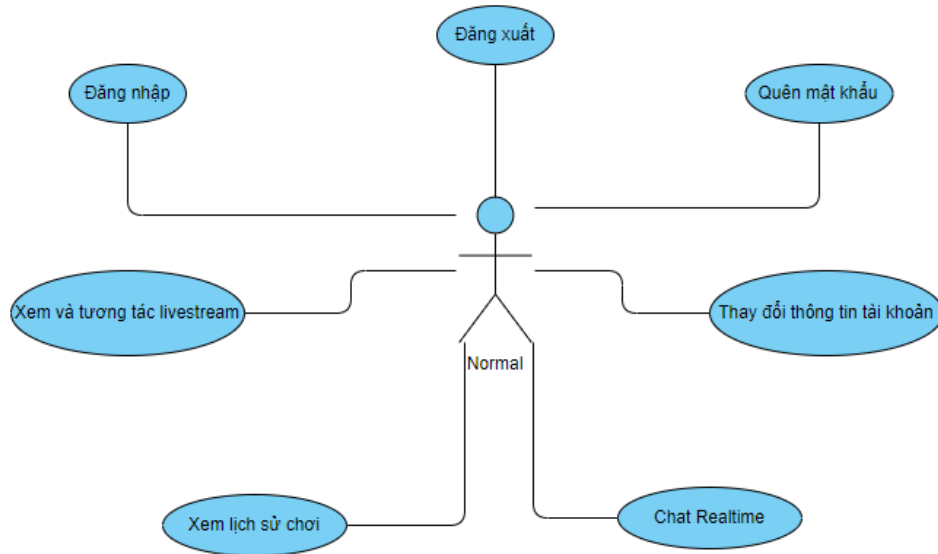
3.2.2. Phân tích yêu cầu

Usercase tổng quát



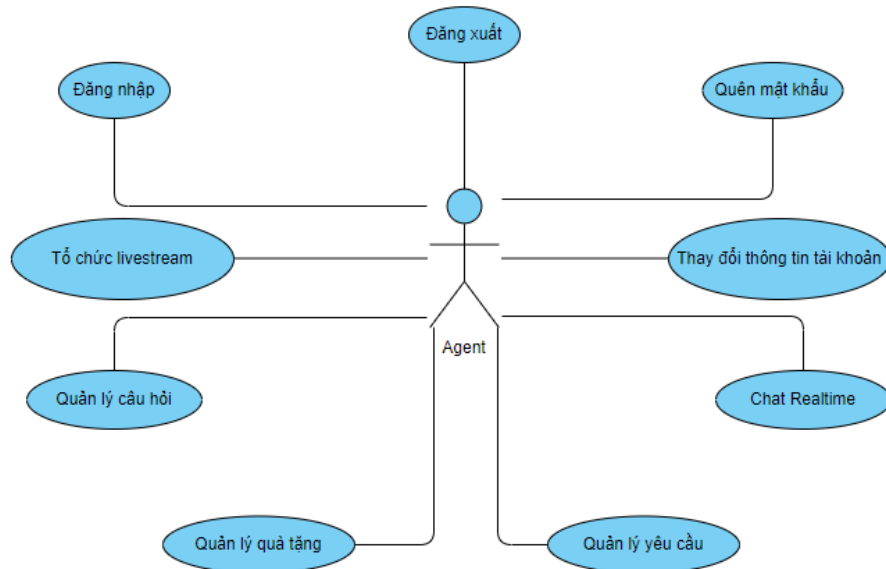
Hình 9: Usecase tổng quát

a. Phía người dùng “Normal”



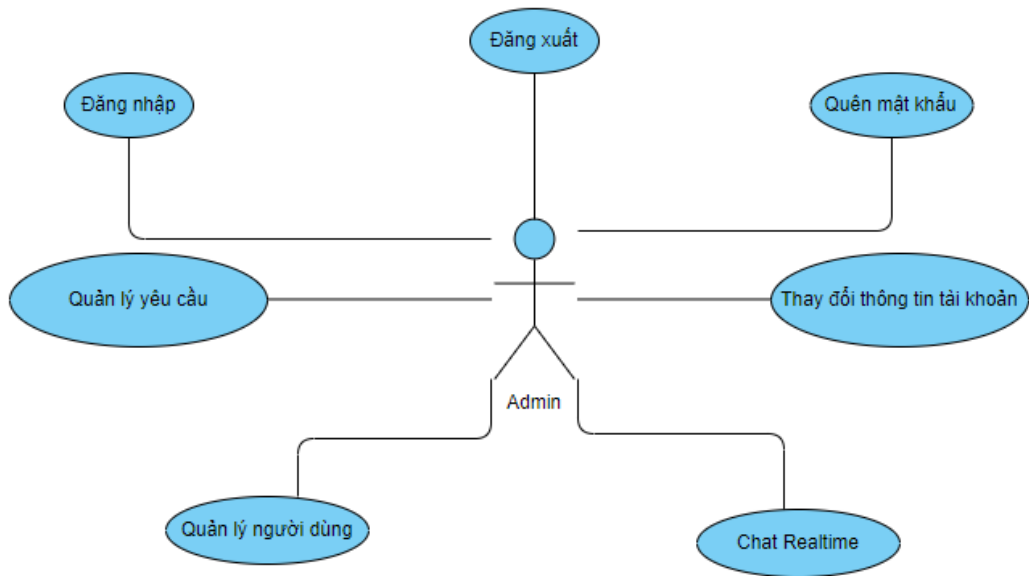
Hình 10: Usecase người dùng Normal

b. Phía người dùng “Agent”



Hình 11: Usecase người dùng Agent

c. Phía người dùng “Admin”

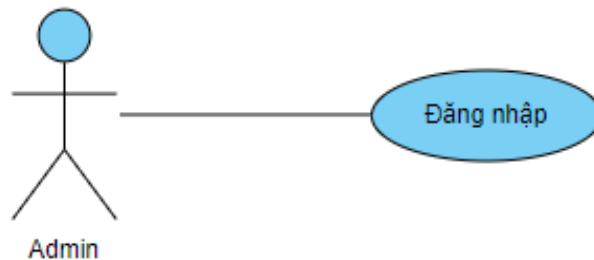


Hình 12: Usecase người dùng Admin

3.2.2.1. Chức năng của người dùng

a. Đăng nhập

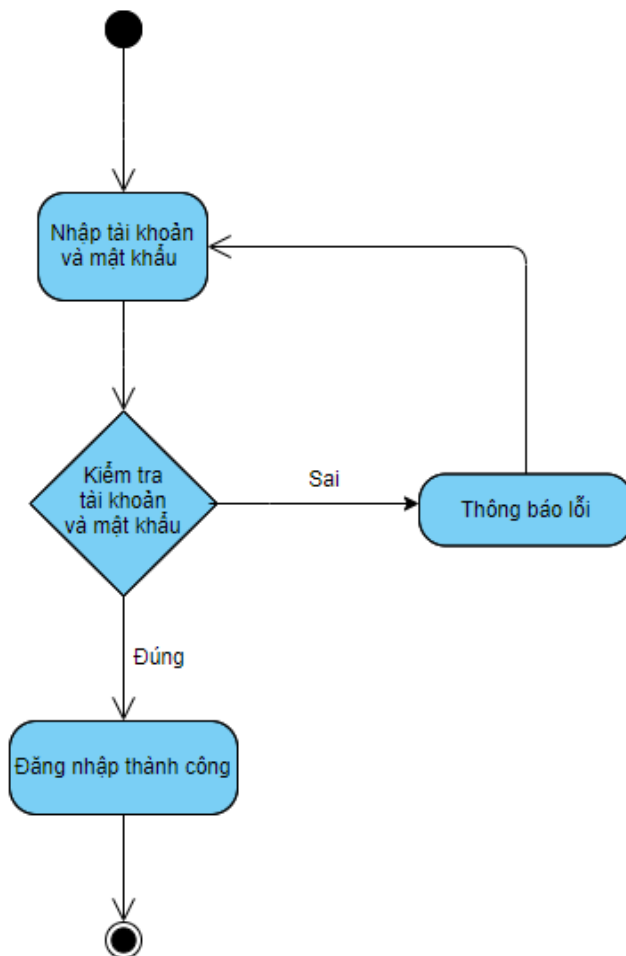
Mô tả: Chức năng cho phép người dùng đăng nhập vào hệ thống thông qua tài khoản và mật khẩu.



Hình 13: Usecase Đăng nhập

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhập tài khoản và mật khẩu.
- **Bước 2:** Kiểm tra tài khoản và mật khẩu trên hệ thống. Nếu đúng đến bước 3, nếu sai thông báo lỗi và quay lại bước 1
- **Bước 3:** Đăng nhập thành công. Hệ thống chuyển màn hình trang chủ
- **Bước 4:** Kết thúc.

Biểu đồ hoạt động



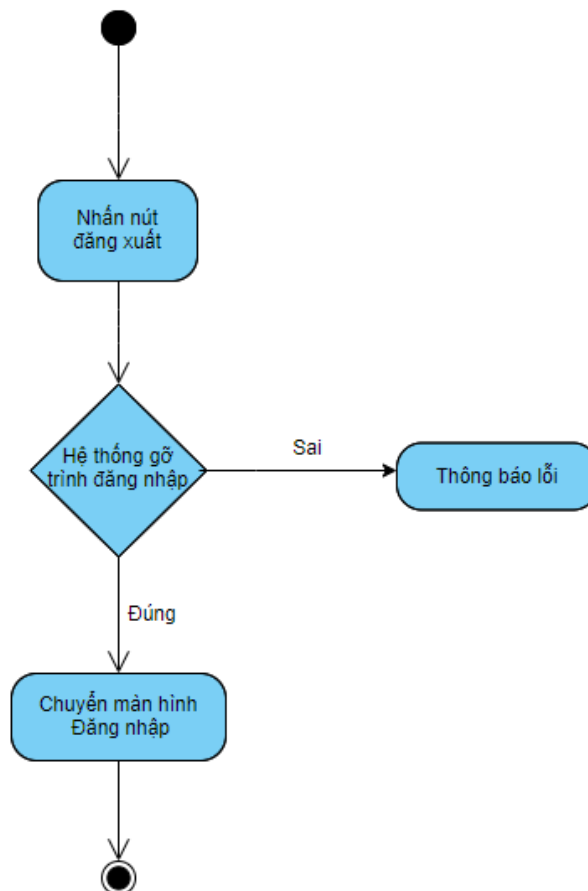
Hình 14: Biểu đồ hoạt động chức năng đăng nhập

b. Đăng xuất

Mô tả: Hệ thống cho phép người dùng đăng xuất khỏi hệ thống

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhấp chọn “Đăng xuất”
- **Bước 2:** Hệ thống gỡ đăng nhập của người dùng. Nếu trong quá trình gỡ bị lỗi sẽ hiển thị thông báo ngược lại sẽ thực hiện bước 3.
- **Bước 3:** Hệ thống chuyển màn hình Đăng nhập
- **Bước 4:** Kết thúc.

Biểu đồ hoạt động



Hình 15: Biểu đồ hoạt động chức năng đăng xuất

c. Quên mật khẩu

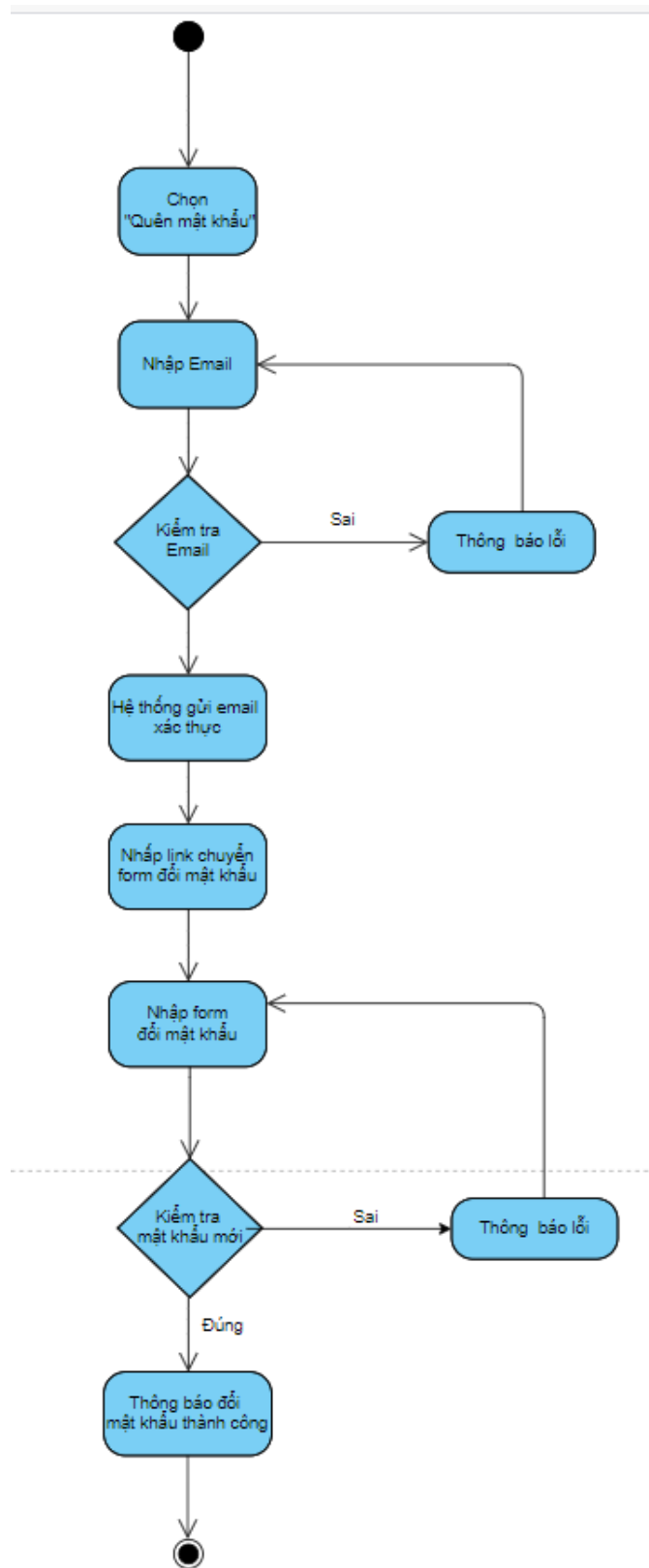


Hình 16: Usecase chức năng quên mật khẩu

Mô tả: Hệ thống cho phép người dùng quên mật khẩu qua email đã đăng ký. Hệ thống gửi một email tới và xác thực chuyển form đổi mật khẩu mới.

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhấp chọn “Quên mật khẩu”
- **Bước 2:** Người dùng nhập email đã sử dụng để đăng ký tài khoản.
- **Bước 3:** Hệ thống kiểm tra email nếu đúng thực hiện bước 4. Nếu sai thông báo lỗi.
- **Bước 4:** Hệ thống gửi một email xác thực.
- **Bước 5:** Người dùng nhấp chọn link trong email hệ thống chuyển form đổi mật khẩu mới.
- **Bước 6:** Người dùng nhập mật khẩu mới tối thiểu 8 ký tự.
- **Bước 7:** Hệ thống kiểm tra tính hợp lệ của mật khẩu. Nếu đúng thực hiện bước 8. Ngược lại quay lại bước 6.
- **Bước 8:** Thông báo đổi mật khẩu thành công.
- **Bước 9:** Kết thúc

Biểu đồ hoạt động



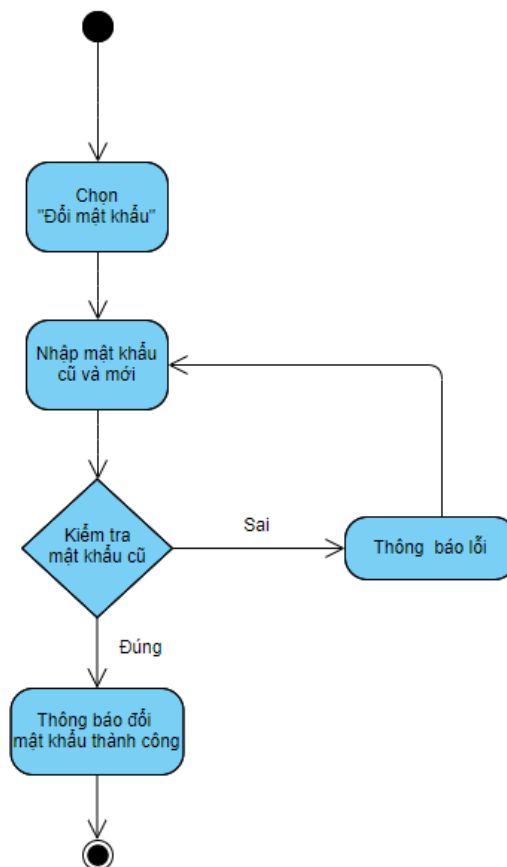
Hình 17: Biểu đồ hoạt động chức năng quên mật khẩu

d. Đổi mật khẩu

Mô tả: Hệ thống cho phép người dùng đổi mật khẩu.

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhập chọn “Đổi mật khẩu”
- **Bước 2:** Người dùng nhập mật khẩu cũ và mới.
- **Bước 3:** Hệ thống kiểm tra cũ nếu đúng thực hiện bước 4. Nếu sai thông báo lỗi.
- **Bước 4:** Thông báo đổi mật khẩu thành công.
- **Bước 5:** Kết thúc

Biểu đồ hoạt động



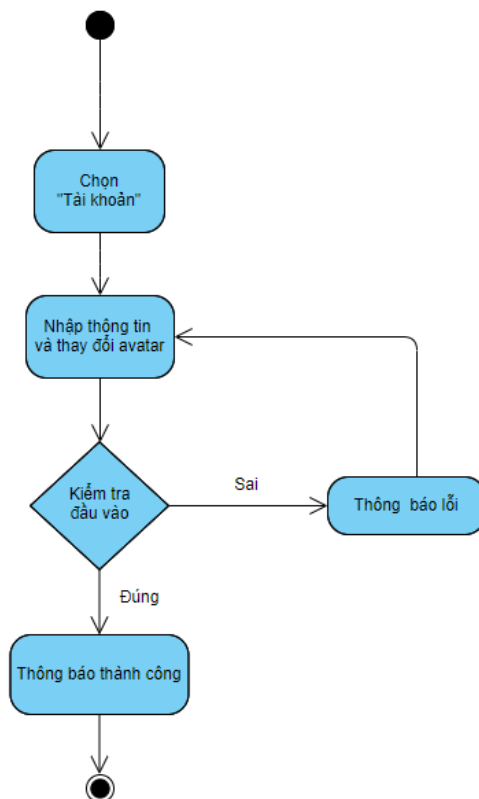
Hình 18: Biểu đồ hoạt động chức năng đổi mật khẩu

e. Thay đổi thông tin tài khoản

Mô tả: Hệ thống cho phép người dùng đổi mật khẩu.

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhấp chọn “Đổi mật khẩu”
- **Bước 2:** Người dùng thông tin và thay đổi ảnh đại diện (avatar)
- **Bước 3:** Hệ thống kiểm tra đầu vào nếu đúng thực hiện bước 4. Nếu sai thông báo lỗi.
- **Bước 4:** Thông báo thành công.
- **Bước 5:** Kết thúc

Biểu đồ hoạt động



Hình 19: Biểu đồ hoạt động chức năng đổi thông tin tài khoản

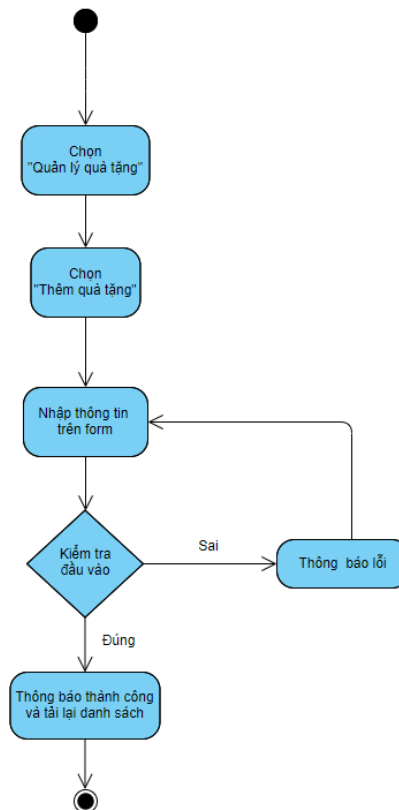
f. Quản lý quà tặng

Mô tả: Hệ thống cho phép người dùng “Agent” thêm, sửa, xóa quà tặng.

- Thêm quà tặng:

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhấp chọn “Quản lý quà tặng”.
- **Bước 2:** Người dùng nhấn nút “Thêm quà tặng”.
- **Bước 3:** Người dùng nhập thông tin tại form.
- **Bước 4:** Hệ thống kiểm tra đầu vào. Nếu đúng thực hiện bước 5, ngược lại quay lại bước 3.
- **Bước 5:** Thông báo thành công và tải lại danh sách quà tặng
- **Bước 6:** Kết thúc

Biểu đồ hoạt động

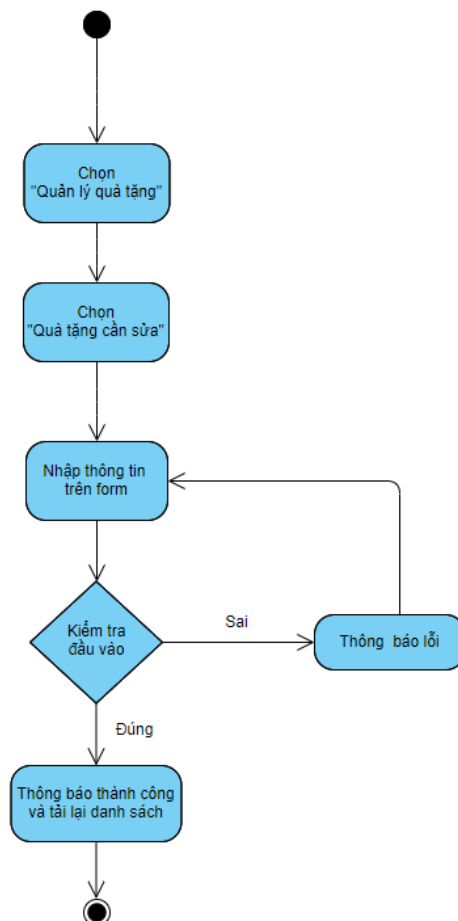


Hình 20: Biểu đồ hoạt động chức năng thêm quà tặng

- Sửa quà tặng

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhấp chọn “Quản lý quà tặng”.
- **Bước 2:** Người dùng chọn quà tặng và nhấn nút “Sửa quà tặng”.
- **Bước 3:** Người dùng nhập thông tin tại form.
- **Bước 4:** Hệ thống kiểm tra đầu vào. Nếu đúng thực hiện bước 5, ngược lại quay lại bước 3.
- **Bước 5:** Thông báo thành công và tải lại danh sách quà tặng
- **Bước 6:** Kết thúc

Biểu đồ hoạt động

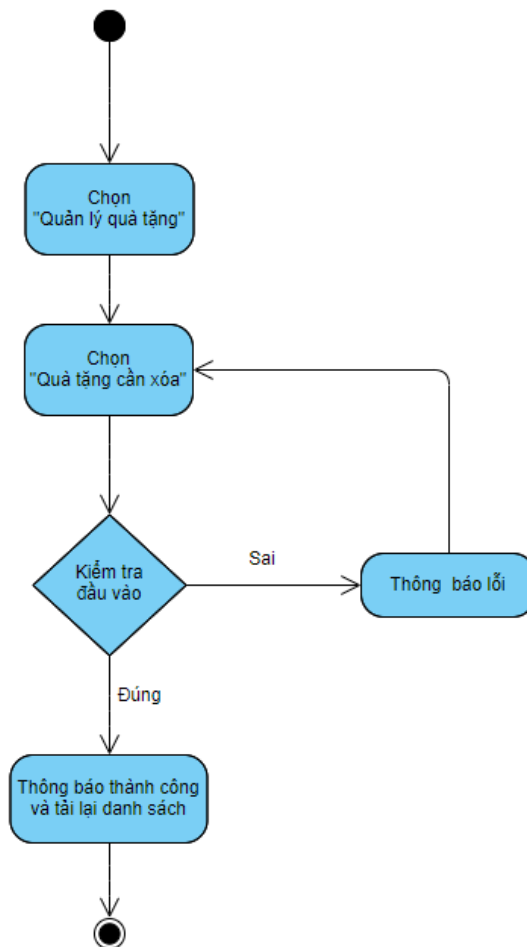


Hình 21: Biểu đồ hoạt động chức năng sửa quà tặng

- Xóa quà tặng

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhập chọn “Quản lý quà tặng”.
- **Bước 2:** Người dùng chọn quà tặng và nhấn nút “Xóa quà tặng”.
- **Bước 3:** Hệ thống kiểm tra đầu vào. Nếu đúng thực hiện bước 4, ngược lại quay lại bước 2.
- **Bước 4:** Thông báo thành công và tải lại danh sách quà tặng
- **Bước 5:** Kết thúc

Biểu đồ hoạt động



Hình 22: Biểu đồ hoạt động chức năng xóa quà tặng

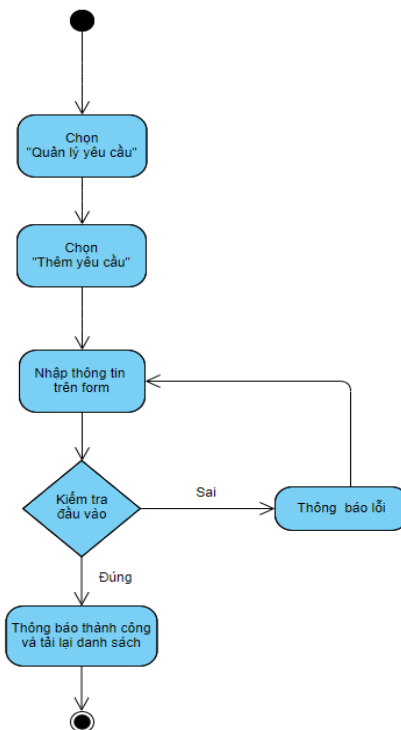
g. Quản lý yêu cầu

Mô tả: Hệ thống cho phép người dùng “**Agent**” thêm, sửa, xóa yêu cầu. “**Admin**” có quyền hủy/duyet yêu cầu của “**Agent**”

- Thêm yêu cầu:

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhấp chọn “Quản lý yêu cầu”.
- **Bước 2:** Người dùng nhấn nút “Thêm yêu cầu”.
- **Bước 3:** Người dùng nhập thông tin tại form.
- **Bước 4:** Hệ thống kiểm tra đầu vào. Nếu đúng thực hiện bước 5, ngược lại quay lại bước 3.
- **Bước 5:** Thông báo thành công và tải lại danh sách
- **Bước 6:** Kết thúc

Biểu đồ hoạt động

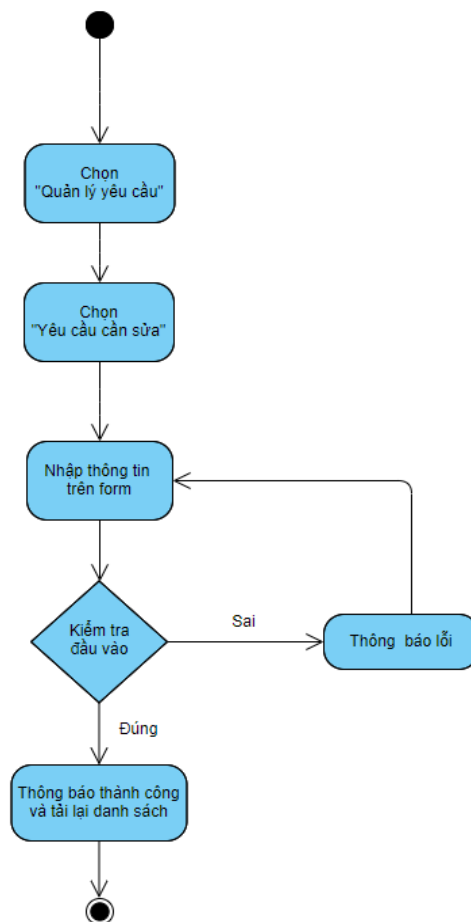


Hình 23: Biểu đồ hoạt động chức năng thêm yêu cầu

- Sửa yêu cầu:

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhập chọn “Quản lý yêu cầu”.
- **Bước 2:** Người dùng chọn yêu cầu cần sửa và nhấn nút “sửa yêu cầu”.
- **Bước 3:** Người dùng nhập thông tin tại form.
- **Bước 4:** Hệ thống kiểm tra đầu vào. Nếu đúng thực hiện bước 5, ngược lại quay lại bước 3.
- **Bước 5:** Thông báo thành công và tải lại danh sách
- **Bước 6:** Kết thúc

Biểu đồ hoạt động

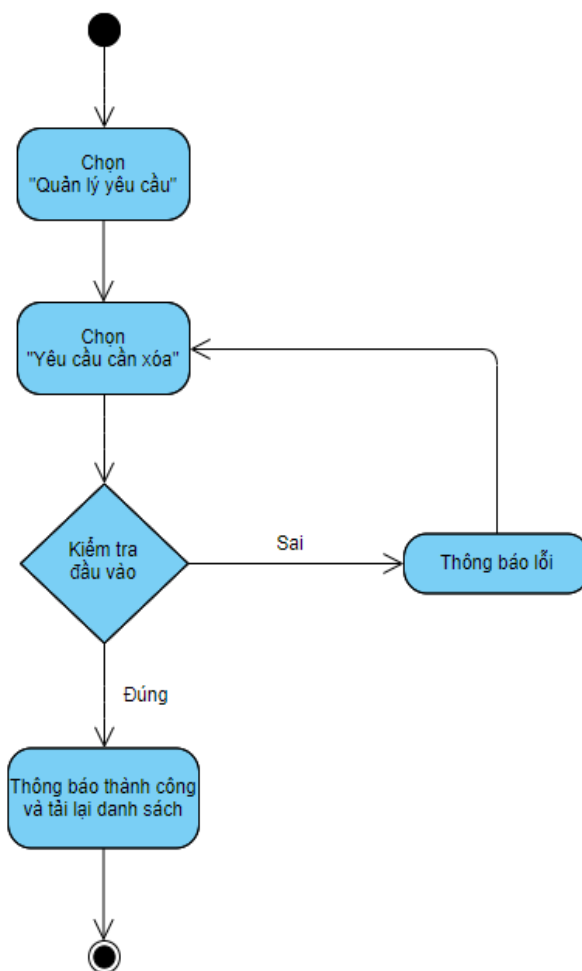


Hình 24: Biểu đồ hoạt động chức năng sửa yêu cầu

- Xóa yêu cầu:

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhấp chọn “Quản lý yêu cầu”.
- **Bước 2:** Người dùng chọn yêu cầu cần xóa và nhấn nút “Xóa yêu cầu”.
- **Bước 3:** Hệ thống kiểm tra đầu vào. Nếu đúng thực hiện bước 4, ngược lại quay lại bước 2.
- **Bước 4:** Thông báo thành công và tải lại danh sách
- **Bước 5:** Kết thúc

Biểu đồ hoạt động

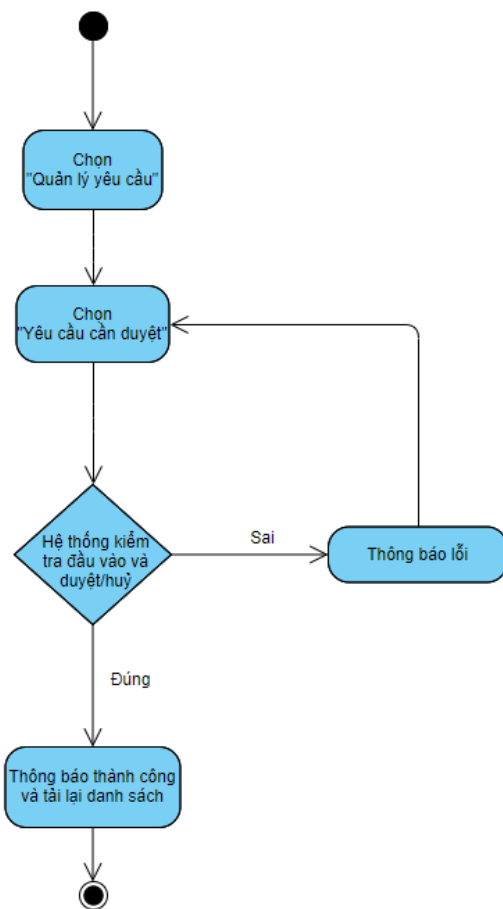


Hình 25: Biểu đồ hoạt động chức năng xóa yêu cầu

- Hủy/Duyệt yêu cầu:

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhấp chọn “Quản lý yêu cầu”.
- **Bước 2:** Người dùng chọn yêu cầu cần duyệt và nhấn nút “Duyệt yêu cầu” hoặc “Hủy duyệt yêu cầu”.
- **Bước 3:** Hệ thống kiểm tra đầu vào. Nếu đúng thực hiện bước 4, ngược lại quay lại bước 2.
- **Bước 4:** Thông báo thành công và tải lại danh sách
- **Bước 5:** Kết thúc

Biểu đồ hoạt động



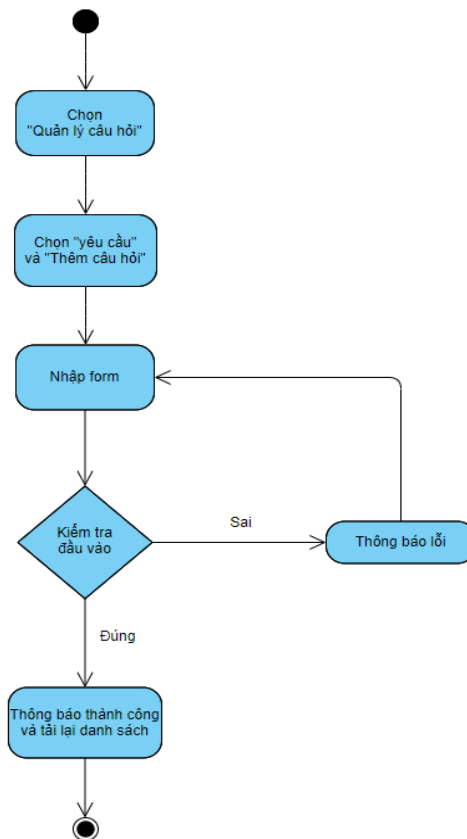
Hình 26: Biểu đồ chức năng hoạt động hủy/duyệt yêu cầu

h. Quản lý câu hỏi

- Thêm câu hỏi:

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhấp chọn “Quản lý câu hỏi”.
- **Bước 2:** Người dùng nhấn chọn yêu cầu và nhấn nút “Thêm câu hỏi”.
- **Bước 3:** Người dùng nhập thông tin tại form.
- **Bước 4:** Hệ thống kiểm tra đầu vào. Nếu đúng thực hiện bước 5, ngược lại quay lại bước 3.
- **Bước 5:** Thông báo thành công và tải lại danh sách
- **Bước 6:** Kết thúc

Biểu đồ hoạt động

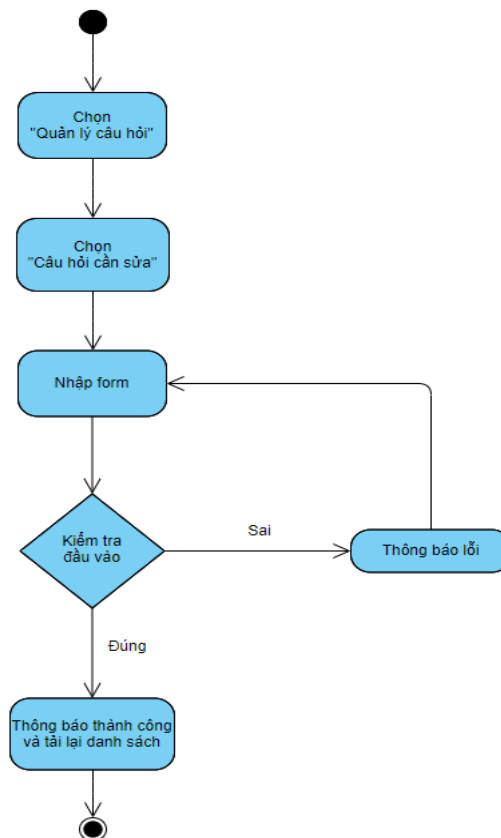


Hình 27: Biểu đồ hoạt động chức năng thêm câu hỏi

- Sửa câu hỏi:

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhấp chọn “Quản lý câu hỏi”.
- **Bước 2:** Người dùng nhấn chọn yêu cầu và nhấn nút “Sửa câu hỏi”.
- **Bước 3:** Người dùng nhập thông tin tại form.
- **Bước 4:** Hệ thống kiểm tra đầu vào. Nếu đúng thực hiện bước 5, ngược lại quay lại bước 3.
- **Bước 5:** Thông báo thành công và tải lại danh sách
- **Bước 6:** Kết thúc

Biểu đồ hoạt động

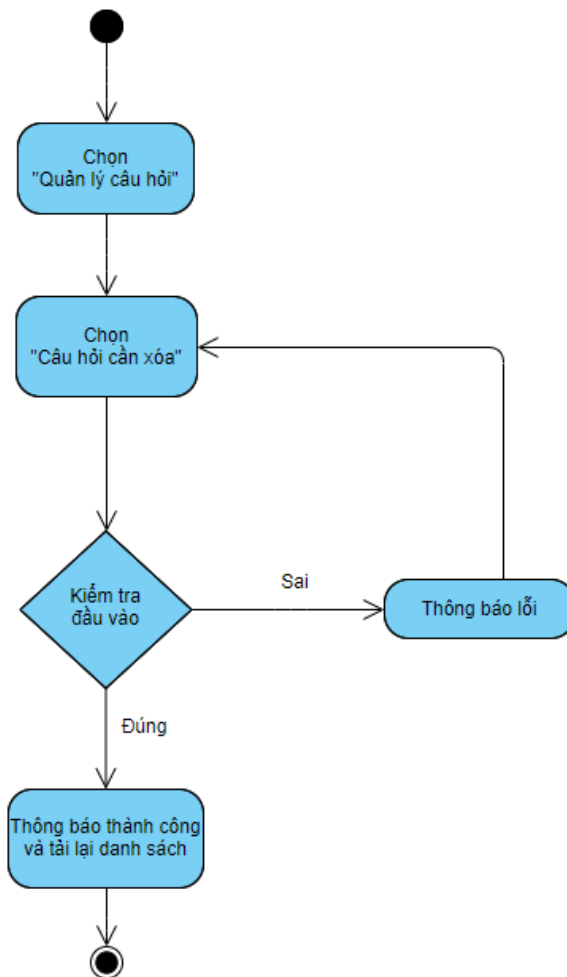


Hình 28: Biểu đồ hoạt động chức năng sửa câu hỏi

- Xóa câu hỏi:

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhấp chọn “Quản lý câu hỏi”.
- **Bước 2:** Người dùng chọn câu hỏi cần xóa và nhấn nút “Xóa câu hỏi”.
- **Bước 3:** Hệ thống kiểm tra đầu vào. Nếu đúng thực hiện bước 4, ngược lại quay lại bước 2.
- **Bước 4:** Thông báo thành công và tải lại danh sách
- **Bước 5:** Kết thúc

Biểu đồ hoạt động



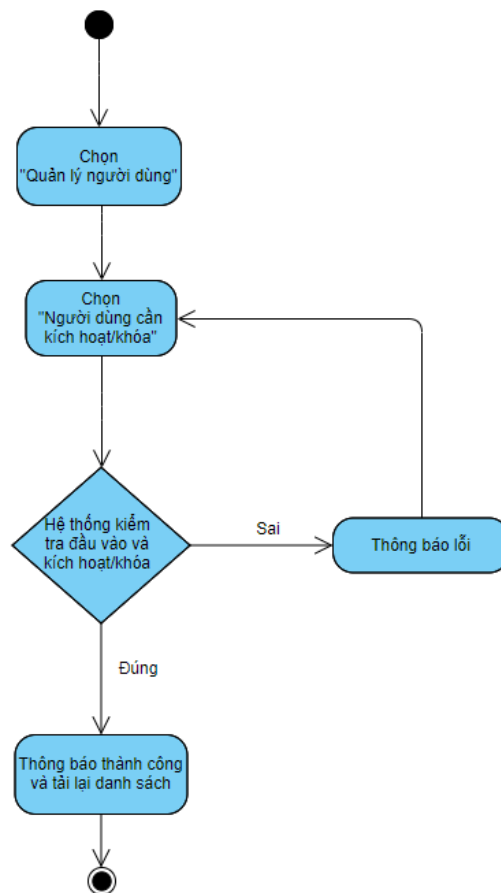
Hình 29: Biểu đồ hoạt động chức năng xóa câu hỏi

i. Quản lý người dùng

Mô tả: Hệ thống cho phép người dùng “Admin” kích hoạt hoặc khóa tài khoản

- **Bước 0:** Bắt đầu
- **Bước 1:** Người dùng nhập chọn “Quản lý người dùng”.
- **Bước 2:** Người dùng chọn người dùng cần kích hoạt hoặc khóa và nhấn nút “Kích hoạt” hoặc “Khóa”.
- **Bước 3:** Hệ thống kiểm tra đầu vào. Nếu đúng thực hiện bước 4, ngược lại quay lại bước 2.
- **Bước 4:** Thông báo thành công và tải lại danh sách
- **Bước 5:** Kết thúc

Biểu đồ hoạt động



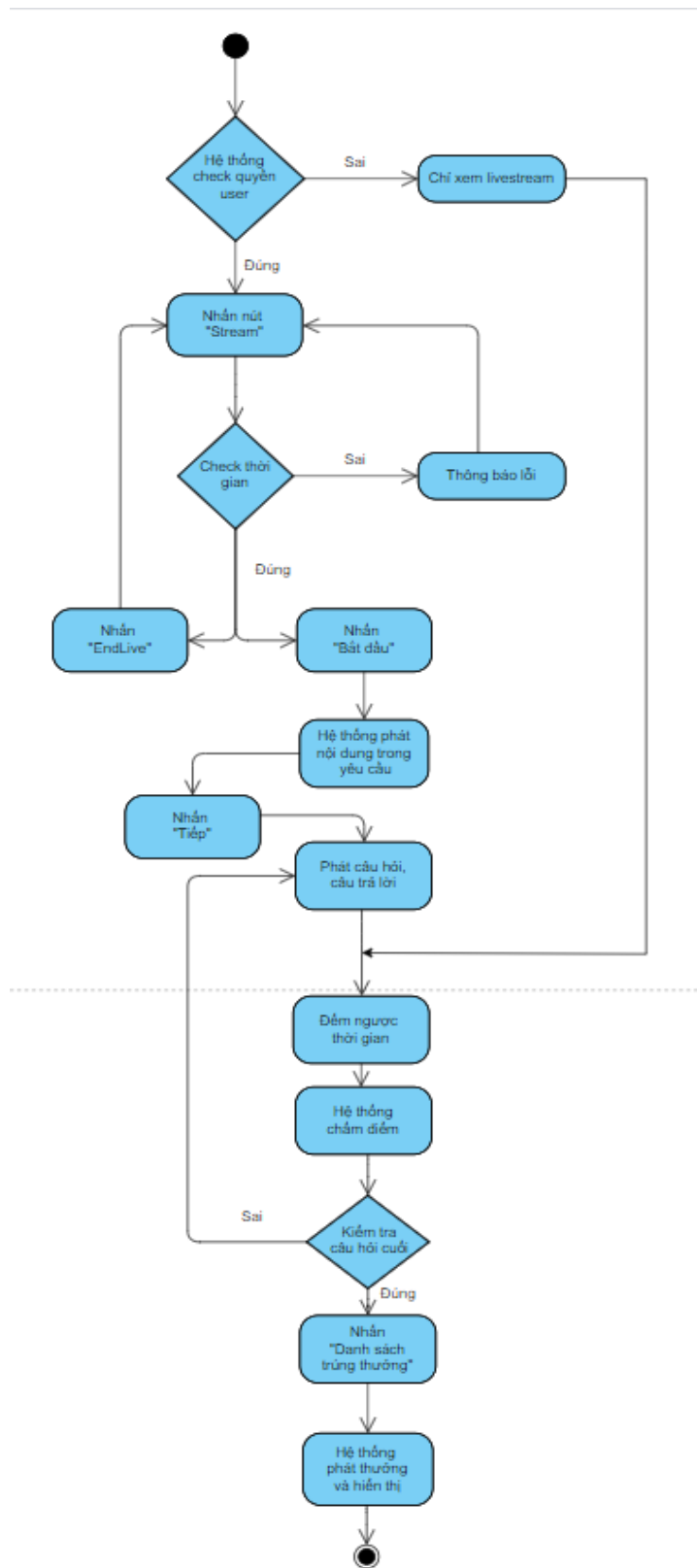
Hình 30: Biểu đồ hoạt động chức năng khóa/kích hoạt người dùng

k. Hệ thống livestream

Mô tả: Hệ thống cho phép người dùng “**Agent**” tổ chức livestream theo yêu cầu được duyệt. Người dùng “**Normal**” được tương tác chơi. Hệ thống tính điểm hiển thị danh thắng cuộc.

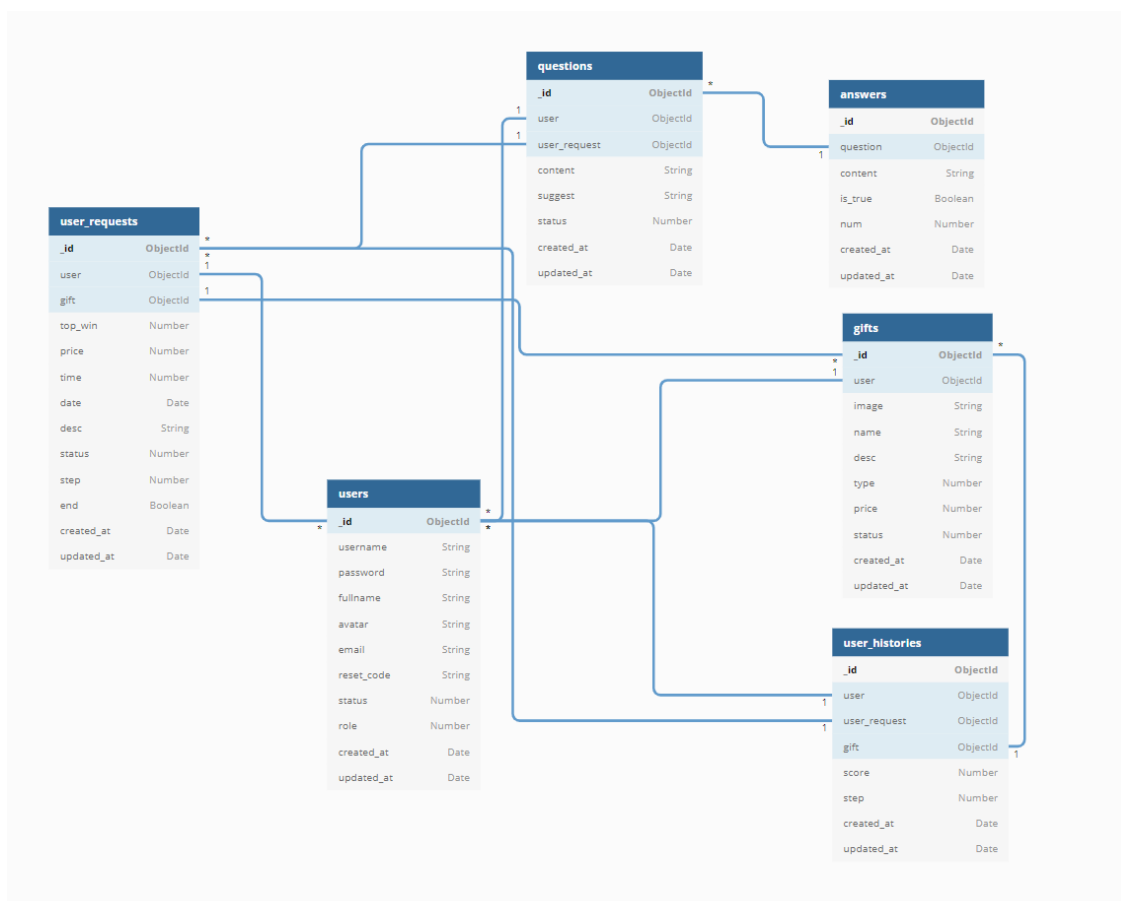
- **Bước 0:** Bắt đầu và trang chủ hệ thống
- **Bước 1:** Hệ thống check quyền người dùng check quyền livestream. Nếu đúng thực hiện bước 2. Nếu sai người dùng chỉ có thể xem livestream.
- **Bước 2:** Người dùng “**Agent**” nhấp chọn “Stream”.
- **Bước 3:** Hệ thống check thời gian Stream < 30p trước thời gian cấu hình khung giờ. Nếu đúng thực hiện bước 4. Nếu sai thông báo lỗi và quay lại bước 2.
- **Bước 4:** Hệ thống yêu cầu bật camera. Nếu hỗ trợ thực hiện bước 5. Nếu không hỗ trợ báo lỗi quay lại bước 2.
- **Bước 5:** Người dùng “**Agent**” có thể bấm “End Live” thì thực hiện lại bước 2, bấm “Bắt đầu” thì thực hiện bước 6.
- **Bước 6:** Hệ thống phát các nội dung trong yêu cầu và phát câu hỏi nếu “**Agent**” nhấn “Tiếp” thực hiện bước 7.
- **Bước 7:** Đến các câu hỏi hệ thống đếm người thời gian cho phép các “**Normal**” tương tác trả lời câu hỏi. Khi hết giờ thực hiện bước 8.
- **Bước 8:** Hệ thống chấm điểm và thông báo lượt tương tác cho câu hỏi đó. Nếu hết câu hỏi thực hiện bước 9. Nếu còn quay lại bước 6.
- **Bước 9:** Người dùng “**Agent**” nhấn Danh sách trúng thưởng hệ thống sẽ gửi toàn bộ danh sách người thắng cuộc cho toàn bộ người dùng “**Normal**”. Và phá thưởng cho người dùng thắng cuộc theo cấu hình của yêu cầu từ phía “**Agent**”.
- **Bước 10:** Kết thúc

Biểu đồ hoạt động



Hình 31: Biểu đồ hoạt động chức năng livestream

3.3. Thiết kế database



Hình 32: Mô hình thực thể database

3.3.1. Bảng users (người dùng)

STT	Thuộc tính	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	_id	ObjectId	Bắt buộc *	Mã định danh
2	username	String	Bắt buộc *	Tên đăng nhập
3	password	String	Bắt buộc *	Mật khẩu mã hóa md5
4	fullname	String	Bắt buộc *	Họ và tên người dùng
5	avatar	String	null	Ảnh đại diện người dùng
6	email	String	Bắt buộc *	Email của người dùng
7	reset_code	String	null	Mã code để quên mật khẩu
8	status	Number	1	Trạng thái của người dùng Giá trị

				- Hoạt động: 1 - Khóa: -1
9	role	Number	1	Quyền của người dùng Giá trị - Normal: 1 - Agent : 2 - Admin : 3
10	created_at	Date	Current Date	Ngày tạo
11	updated_at	Date	Current Date	Lần cập nhật gần nhất

3.3.2. Bảng gifts (quà tặng)

STT	Thuộc tính	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	_id	ObjectId	Bắt buộc *	Mã định danh
2	user	ObjectId	Bắt buộc *	Khóa ngoại bảng Users
3	image	String	null	Ảnh của quà tặng
4	name	String	Bắt buộc *	Tên quà tặng
5	desc	String	Bắt buộc *	Mô tả quà tặng
6	type	Number	1	Loại quà tặng Giá trị - Tiền mặt: 1 - Sản phẩm :2
7	price	Number	Bắt buộc *	Giá trị của quà tặng
8	status	Number	1	Trạng thái của người dùng Giá trị - Hoạt động: 1 - Khóa: -1 - Xóa: 0
9	created_at	Date	Current Date	Ngày tạo

10	updated_at	Date	Current Date	Lần cập nhật gần nhất
----	------------	------	--------------	-----------------------

3.3.3. Bảng *user_requests* (yêu cầu của agent)

STT	Thuộc tính	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	_id	ObjectId	Bắt buộc *	Mã định danh
2	user	ObjectId	Bắt buộc *	Khóa ngoại bảng Users
3	gift	ObjectId	Bắt buộc *	Khóa ngoại bảng Gifts
4	top_win	Number	0	Top chia thưởng
5	price	Number	0	Giá trị tiền buổi phát sóng cho nhà cung cấp phần mềm.
6	time	Number	Bắt buộc *	Khung giờ phát sóng Giá trị 1. 9h – 10h 2. 15 – 16h 3. 20h – 21h
7	date	Date	Bắt buộc *	Ngày phát sóng
8	desc	String	<i>Chuỗi rỗng</i>	Mô tả yêu cầu
9	status	Number	-2	Trạng thái yêu cầu Giá trị -2: Lưu -1: Bị từ chối 0: Đang chờ duyệt 1: Đã duyệt 2: Đã phát sóng yêu cầu
10	step	Number	-1	Yêu cầu đang được phát đến câu số bao nhiêu?
11	end	Boolean	false	Yêu cầu đã kết thúc chưa?
12	created_at	Date	Current Date	Ngày tạo

13	updated_at	Date	Current Date	Lần cập nhật gần nhất
----	------------	------	--------------	-----------------------

3.3.4. Bảng questions (câu hỏi)

STT	Thuộc tính	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	_id	ObjectId	Bắt buộc *	Mã định danh
2	user	ObjectId	Bắt buộc *	Khóa ngoại bảng Users
3	user_request	ObjectId	Bắt buộc *	Khóa ngoại bảng User_requests
4	content	String	Bắt buộc *	Nội dung câu hỏi
5	suggest	String	<i>Chuỗi rỗng</i>	Gợi ý câu hỏi
6	status	Number	Bắt buộc *	Trạng thái câu hỏi Giá trị 1. Hoạt động 2. Khóa
7	created_at	Date	Current Date	Ngày tạo
8	updated_at	Date	Current Date	Lần cập nhật gần nhất

3.3.5. Bảng answers (câu trả lời)

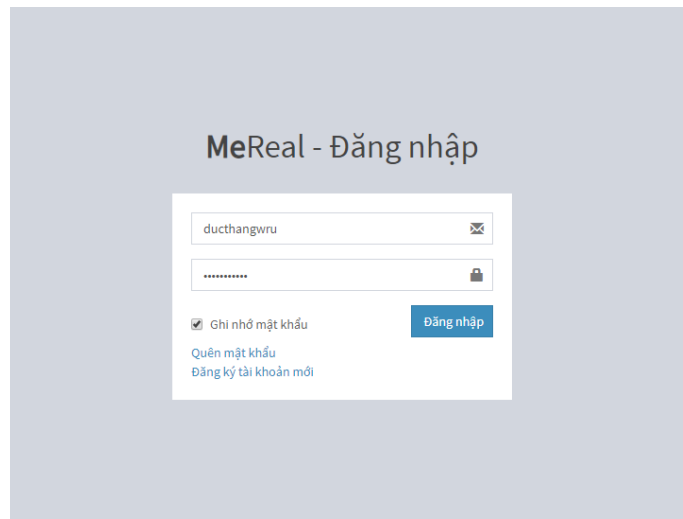
STT	Thuộc tính	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	_id	ObjectId	Bắt buộc *	Mã định danh
2	question	ObjectId	Bắt buộc *	Id câu hỏi Khóa ngoại bảng Questions
3	content	String	Bắt buộc *	Nội dung câu hỏi
4	is_true	Boolean	Bắt buộc *	Đáp án câu hỏi
5	num	Number	Bắt buộc *	Số người lựa chọn câu trả lời là đáp án của mình
6	created_at	Date	Current Date	Ngày tạo
7	updated_at	Date	Current Date	Lần cập nhật gần nhất

3.3.6. Bảng *user_histories* (lịch sử chơi)

STT	Thuộc tính	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	_id	ObjectId	Bắt buộc *	Mã định danh
2	user	ObjectId	Bắt buộc *	Id người chơi Khóa ngoại bảng Users
3	user_request	ObjectId	Bắt buộc *	Id của yêu cầu phát sóng Khóa ngoại bảng User_requests
4	gift	ObjectId	null	Quà tặng đạt được Khóa ngoại bảng Gifts
5	score	Number	0	Số điểm đạt được
6	step	Number	0	Đang trả lời đến câu bao nhiêu
7	created_at	Date	Current Date	Ngày tạo
8	updated_at	Date	Current Date	Lần cập nhật gần nhất

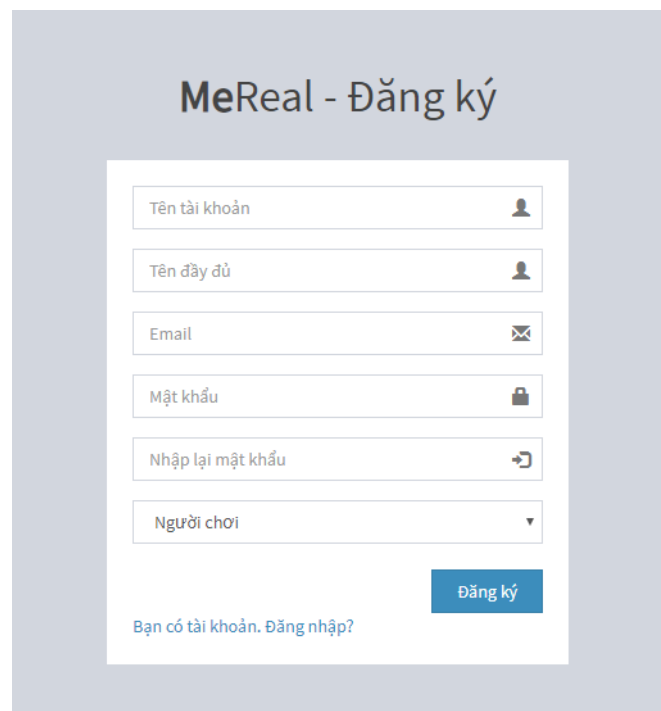
CHƯƠNG 4: GIỚI THIỆU GIAO DIỆN ỨNG DỤNG

4.1. Màn hình đăng nhập , đăng ký, quên mật khẩu



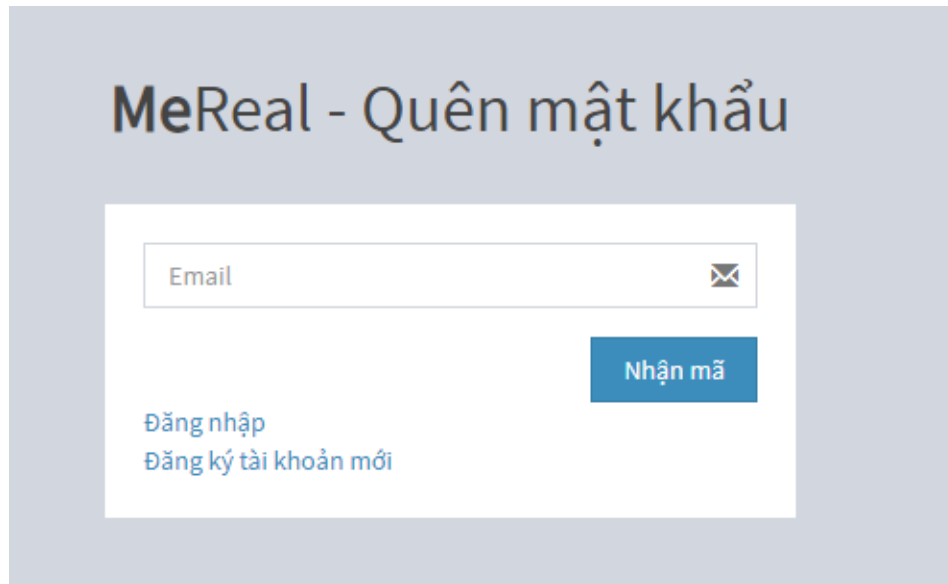
The image shows a login form titled "MeReal - Đăng nhập". It features a text input field for the username "ducthangwru" with an eye icon, and a password input field with a lock icon. Below the password field is a checkbox labeled "Ghi nhớ mật khẩu" (Remember password) which is checked. To the right of the checkbox is a blue button labeled "Đăng nhập" (Login). Below the checkbox are two links: "Quên mật khẩu" (Forgot password) and "Đăng ký tài khoản mới" (Register new account).

Hình 33: Giao diện màn hình đăng nhập



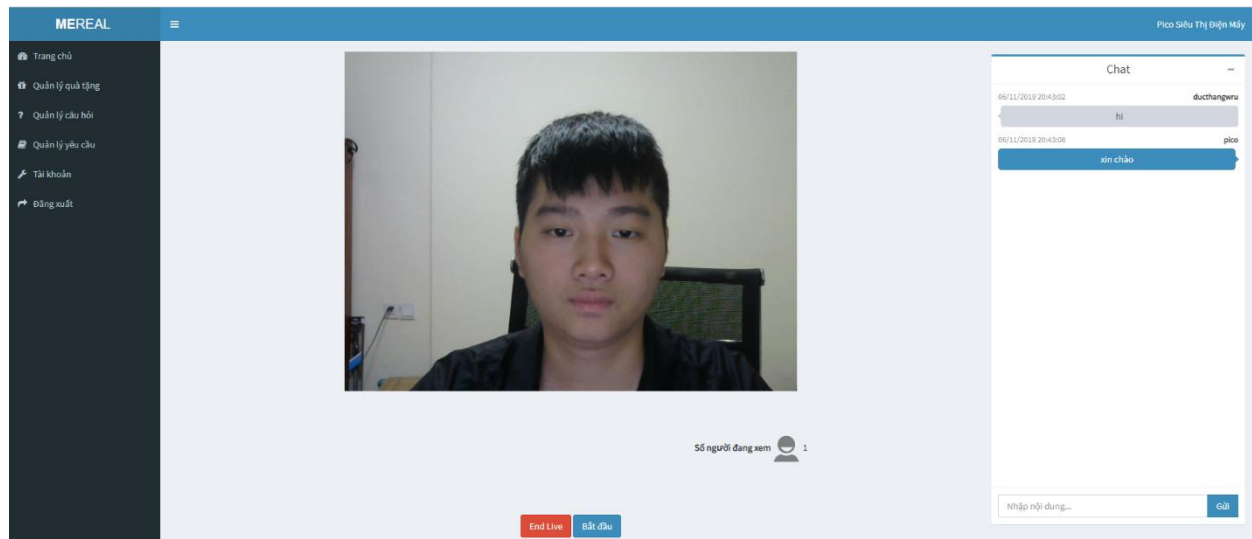
The image shows a registration form titled "MeReal - Đăng ký". It features several input fields: "Tên tài khoản" (Username) with a person icon, "Tên đầy đủ" (Full name) with a person icon, "Email" with an envelope icon, "Mật khẩu" (Password) with a lock icon, and "Nhập lại mật khẩu" (Repeat password) with a circular arrow icon. Below these is a dropdown menu labeled "Người chơi" (Player) with a downward arrow. To the right of the dropdown is a blue button labeled "Đăng ký" (Register). At the bottom left, there is a link: "Bạn có tài khoản. Đăng nhập?" (You have an account. Login?).

Hình 34: Giao diện màn hình đăng ký

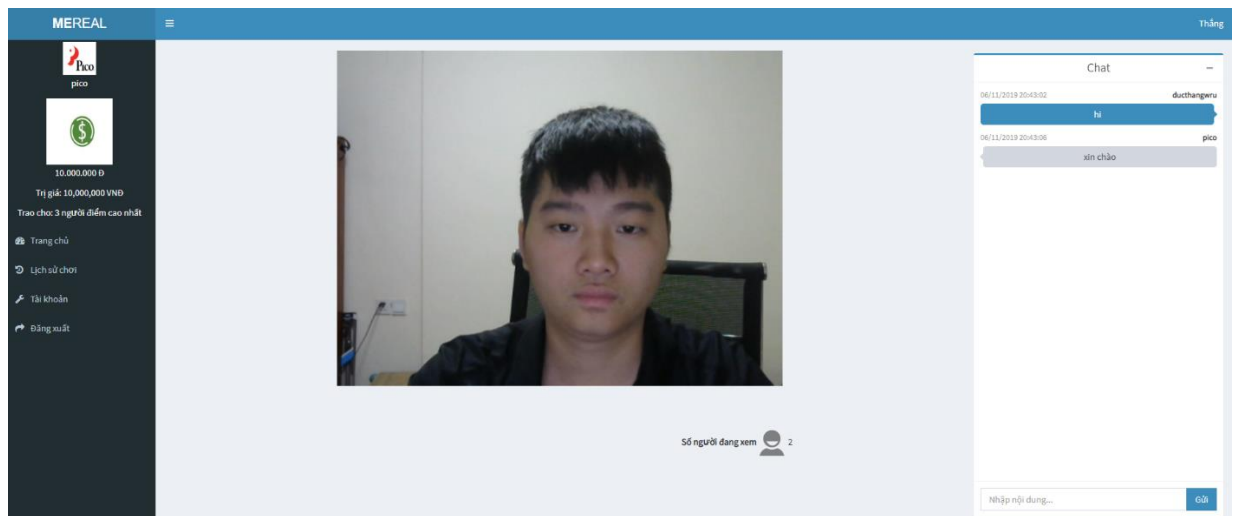


Hình 35: Giao diện màn hình chức năng quên mật khẩu

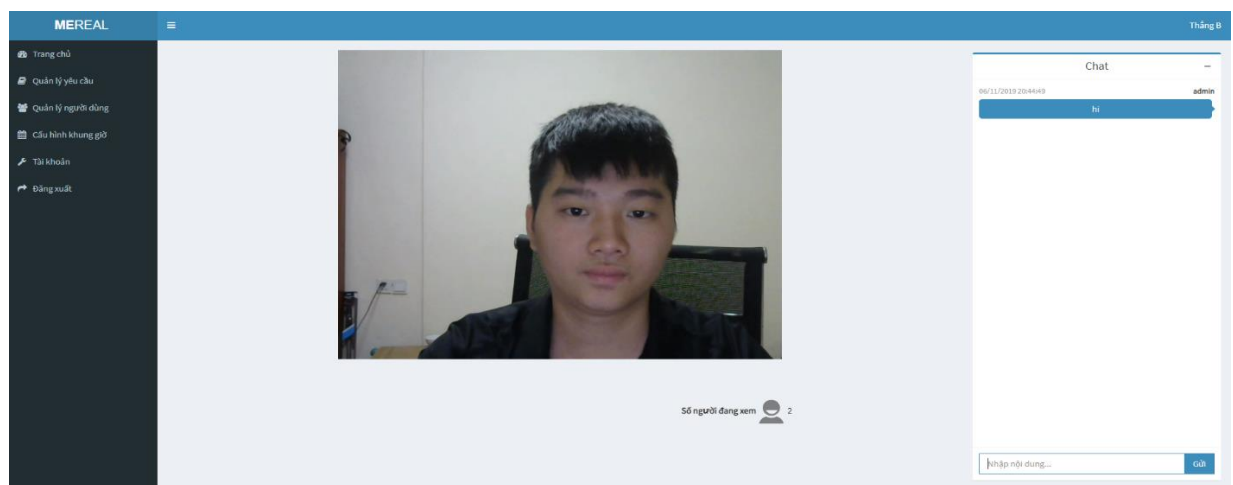
4.2. Màn hình trang chủ



Hình 36: Màn hình trang chủ dành cho Agent



Hình 37: Màn hình trang chủ dành cho Normal



Hình 38: Màn hình trang chủ dành cho Admin

4.3. Màn hình quản lý quà tặng



Hình 39: màn hình quản lý quà tặng của Agent

4.4. Màn hình quản lý câu hỏi

QUẢN LÝ CÂU HỎI

Chọn yêu cầu:

Search:

STT	Nội dung	Gợi ý	Trạng thái	Thao tác
1	Đèn đỏ là dừng hay đi?		Hoạt động	<input type="button" value="Gr"/> <input type="button" value="Re"/>
2	Đèn xanh đi hay ở?		Hoạt động	<input type="button" value="Gr"/> <input type="button" value="Re"/>
3	Đèn vàng là gì?		Hoạt động	<input type="button" value="Gr"/> <input type="button" value="Re"/>

Showing 1 to 3 of 3 entries

DANH SÁCH CÂU TRẢ LỜI "Đèn đỏ là dừng hay đi?"

Search:

STT	Nội dung	Đúng/Sai	Thao tác
1	Dừng	Đúng	<input type="button" value="Gr"/> <input type="button" value="Re"/>
2	Đi	Sai	<input type="button" value="Gr"/> <input type="button" value="Re"/>

Showing 1 to 2 of 2 entries

Hình 40: Màn hình quản lý câu hỏi dành cho Agent

4.5. Màn hình quản lý yêu cầu

QUẢN LÝ YÊU CẦU

Search:

STT	Mô tả	Đại lý	Quà tặng	Bảo giá (VNĐ)	Top win	Khung giờ	Ngày	Trạng thái	Thao tác
1	live	trendatio	10.000.000 Đ	51.000.000	3	20:00-23:59	11/10/2019	Đã duyệt	
2	Livestream 50 triệu	trendatio	10.000.000 Đ	50.000.000	2	09:00-10:00	11/10/2019	Lưu	<input type="button" value="Gr"/> <input type="button" value="Re"/>

Showing 1 to 2 of 2 entries

Hình 41: Màn hình quản lý yêu cầu dành cho Agent

QUẢN LÝ YÊU CẦU

Search:

STT	Mô tả	Đại lý	Quà tặng	Bảo giá (VNĐ)	Top win	Khung giờ	Ngày	Trạng thái	Thao tác
1	live	trendatio	10.000.000 Đ	51.000.000	3	20:00-23:59	11/10/2019	Đã duyệt	<input type="button" value="Gr"/> <input type="button" value="Re"/> <input type="button" value="Del"/>

Showing 1 to 1 of 1 entries

Hình 42: Màn hình quản lý yêu cầu dành cho Admin

4.6. Màn hình quản lý người dùng

QUẢN LÝ NGƯỜI DÙNG						
<div>Search: <input type="text"/></div>						
STT	Tên đăng nhập	Họ tên	Email	Quyền	Trạng thái	Thao tác
1	duchanganu	Thắng	duchanganu97@gmail.com	Người dùng	Hoạt động	✕ ✕
2	admin	Thắng B	thangnguyen111@gmail.com	Quản trị viên	Hoạt động	✕ ✕
3	tiendatleo	Đạt	tiendatleo@gmail.com	Người dùng	Hoạt động	✕ ✕
Showing 1 to 3 of 3 entries						

Hình 43: Màn hình quản lý người dùng của Admin

4.7. Màn hình cấu hình khung giờ

CẤU HÌNH THỜI GIAN			
<div>Search: <input type="text"/></div>			
STT	Thời gian từ	Thời gian đến	Giá tối thiểu (VNĐ)
1	09:00	10:00	10,000,000
2	15:00	16:00	20,000,000
3	20:00	23:59	50,000,000
Showing 1 to 3 of 3 entries			

Hình 44: Màn hình xem danh sách cấu hình khung giờ của Admin

4.8. Màn hình xem danh sách lịch sử chơi

LỊCH SỬ CHƠI				
<div>Search: <input type="text"/></div>				
STT	Người tổ chức	Số điểm	Quà tặng	Thời gian
1	pico	3	10,000,000 Đ	20/10/2019 22:16:42
Showing 1 to 1 of 1 entries				

Hình 45: Màn hình xem danh sách lịch sử chơi

4.9. Màn hình thay đổi thông tin tài khoản

The screenshot shows a web application interface for MEREAL. On the left is a dark sidebar with the MEREAL logo and a menu containing: 'Trang chủ', 'Lịch sử chơi', 'Tài khoản', and 'Đăng xuất'. The main content area is titled 'THÔNG TIN TÀI KHOẢN' and features a profile picture of a man in a blue shirt. Below the photo are four input fields: a dropdown menu for 'Chọn tệp' (currently showing 'Không có tệp nào được chọn'), a text field for 'Thùng', an email field containing 'duc thang vu97@gmail.com', and a text field for 'duc thang vu'. At the bottom of the form are two buttons: 'Đổi mật khẩu' (Change password) and 'Lưu' (Save).

Hình 46: Màn hình cập nhật thông tin tài khoản

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Với sự hướng dẫn nhiệt tình của các thầy cô trong Khoa Công nghệ thông tin, đặc biệt là sự chỉ bảo tận tình của thầy Kiều Tuấn Dũng trong thời gian làm đồ án của mình, nay em đã hoàn thành đồ án và đạt được một số kết quả sau:

- Nắm được cơ bản về một số quy trình khi tiến hành xây dựng một hệ thống trong thực tế (khảo sát thực tế, thu thập dữ liệu,..., triển khai, thử nghiệm).
- Tìm hiểu về phân tích và thiết kế cơ sở dữ liệu. Từ đó, áp dụng vào việc phân tích bài toán “*Livestream tương tác trao thưởng*”.
- Tìm hiểu được các công nghệ phát triển platform và tìm hiểu được các công nghệ mới như NodeJS, MongoDB, WebRTC,...

Mặc dù đã có nhiều cố gắng trong suốt quá trình làm đồ án, nhưng do thời gian có hạn và thiếu kinh nghiệm trong thực tế nên cũng không thể tránh khỏi những nhược điểm và thiếu sót. Vì vậy, em mong nhận được sự góp ý của quý thầy cô và các bạn để em có thể phát triển hệ thống này hoàn thiện hơn.

5.2 Hướng phát triển đề tài

Sau khi hoàn thành đồ án và xây dựng được web ”Livestream tương tác”

Một số hướng để em có thể phát triển hoàn thiện hơn đó là:

- Tìm hiểu sâu hơn về WebRTC để có thể đáp ứng nhiều hơn nữa nhu cầu của người sử dụng, phát triển và tối ưu hóa hệ thống.
- Khảo sát chi tiết hơn nhu cầu của người dùng để hoàn thiện các chức năng
- Xây dựng ứng dụng trên thiết bị di động

TÀI LIỆU THAM KHẢO

1. khoatd92 (26/04/2016). WebRTC basic - Phần 1: Tìm hiểu về NAT, STUN, TURN vs ICE (<https://kipalog.com/posts/WebRTC-basic---Phan-1--Tim-hieu-ve-NAT--STUN--TURN-vs-ICE>)
2. thanhtv.96 (Sep 3rd, 2018). MongoDB là gì? Cơ sở dữ liệu phi quan hệ (<https://viblo.asia/p/mongodb-la-gi-co-so-du-lieu-phi-quan-he-bJzKmgol9N>)
3. Hồ Sỹ Hùng (08 tháng 06, 2015). Node.js là gì và tại sao tôi nên học lập trình Node.js? (<https://techmaster.vn/posts/33428/nodejs-la-gi-va-tai-sao-toi-nen-hoc-lap-trinh-nodejs>)
4. Template Website: <https://adminlte.io/themes/AdminLTE/index2.html>
5. WebRTC Live Video Stream Broadcasting from One-to-Many (Aug 27, 2015) (<https://www.pubnub.com/blog/webrtc-live-video-stream-broadcasting-from-one-to-many/>)