**BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG**

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

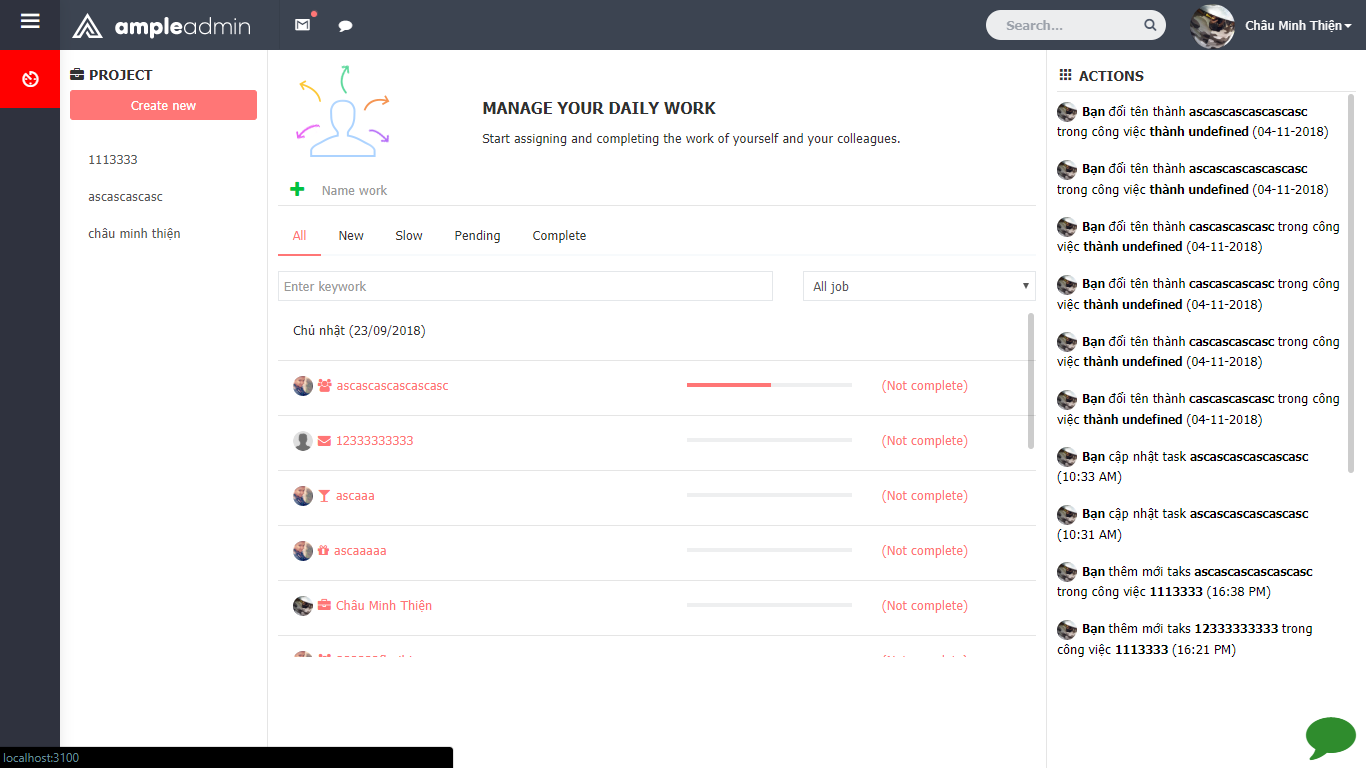
**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**LUẬN VĂN TỐT NGHIỆP**

**CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ CÔNG VIỆC BẰNG REACTJS VÀ NODEJS**



Giảng viên hướng dẫn: **ThS. Nguyễn Thị Ngọc Thanh**

Sinh viên thực hiện: **Châu Minh Thiện**

Lớp **: 03DHCNTT01**

Khoá  **: 03**

**TP. Hồ Chí Minh, ngày 22 tháng 11 năm 2018**

**MỞ ĐẦU**

Trong thời đại công nghệ thông tin, các lĩnh vực, ngành nghề có xu hướng công nghiệp hóa – hiện đại hóa để tăng năng suất và hiệu quả làm việc. Những công việc quản lý thủ công trên các giấy tờ, sổ sách đã dần chuyển sang dữ liệu số để tăng năng suất làm việc mà tiết kiệm thời gian một cách triệt để nhất.

Qua một số cuộc tham khảo em nhận thấy không chỉ các cơ quan hành chính nhà nước mà các đơn vị tư nhân hoặc các công ty nhỏ khi giao nhận việc với nhau còn sử dụng giấy tờ và truyền miệng. Như thế sẽ làm cho sự quản lý công việc trở nên khó khăn mà còn mất nhiều thời gian cho việc sắp xếp cũng như phân công trao đổi công việc với nhau.

Do đó mong muốn của em là sẽ sử dụng các kiến thức của mình để giải quyết vấn đề trên giúp cho việc quản lý công việc trở nên tiện lợi hơn và không mất nhiều thời gian.

Do kiến thức về xây dựng phần mềm trên nền tảng website còn hạn chế nên sẽ không tránh khỏi những sai sót cũng như chức năng không đáp ứng đủ với yêu cầu thực tế, em rất mong nhận được sự đóng góp chân thành của quý thầy (cô) để em càng hoàn thiện phần mềm này hơn để có thể đưa ra sử dụng ngoài thực thế.

**LỜI CẢM ƠN**

Em chân thành cảm ơn Cô Nguyễn Thị Ngọc Thanh.

Cô đã cho em cơ hội để tìm hiểu về chủ đề “**Xây dựng hệ thống quản lý công việc bằng reactjs và nodejs**” và đã hướng dẫn tận tình giúp chúng em biết được cách thức để trình bày một đồ án môn học hoàn chỉnh và hợp lí.

Bước đầu đi vào tìm hiểu nội dung còn có nhiều khó khăn, do kiến thức chuyên môn chưa sâu nên không tránh khỏi gặp nhiều sai sót, em rất mong nhận được những ý kiến đóng góp quý báu của Cô để chúng em có cơ hội mở rộng kiến thức cũng như tích góp kinh nghiệm để cải thiện những phần mềm tiếp theo được hoàn chỉnh và chính xác hơn.

Em xin chân thành cảm ơn!

**NHẬN XÉT**

**(Của giảng viên hướng dẫn tại khoa)**

**Kết Luận:**

TP.HCM ngày … tháng … năm 2018

CÁN BỘ HƯỚNG DẪN

*(ký tên)*

**NHẬN XÉT**

**(Của giảng viên phản biện)**

TP.HCM ngày … tháng … năm 2018

NGƯỜI NHẬN XÉT

*(ký tên)*

**Nội dung**

[**Chương 1: Tổng quan** 1](#_Toc530507093)

[**1.1.** **Mục tiêu đề ra.** 1](#_Toc530507094)

[**1.3.** **Phạm vi hệ thống.** 1](#_Toc530507095)

[**1.4.** **Phương pháp nghiên cứu.** 1](#_Toc530507096)

[**1.5.** **Ý nghĩa thực tiễn.** 1](#_Toc530507097)

[**Chương 2: Cơ sở lý luận** 2](#_Toc530507098)

[**2.1. Cơ sở lý thuyết.** 2](#_Toc530507099)

[**2.1.1.** **Bài toán đặt ra.** 2](#_Toc530507100)

[**2.1.2. Mô tả bài toán.** 2](#_Toc530507101)

[**2.1.3. Phân tích và thiết kế hệ thống.** 3](#_Toc530507102)

[**2.1.3.1. Lược đồ Use-Case.** 3](#_Toc530507103)

[**a. Xác định các tác nhân của hệ thống.** 3](#_Toc530507104)

[**b. Xác định các Use-Case.** 3](#_Toc530507105)

[**c. Lược đồ Use-Case.** 4](#_Toc530507106)

[**2.1.3.2. Đặc tả Use-Case** 5](#_Toc530507107)

[**a. Đặc tả use-case quản lý đăng nhập.** 5](#_Toc530507108)

[**b. Đặc tả use-case chatting.** 5](#_Toc530507109)

[**c. Đặc tả use-case quản lý công việc.** 6](#_Toc530507110)

[**d. Đặc tả use-case dự án.** 7](#_Toc530507111)

[**e. Đặc tả use-case quan lý danh mục.** 8](#_Toc530507112)

[**f. Đặc tả use-case quản lý báo cáo dự án.** 9](#_Toc530507113)

[**g. Đặc tả use-case quản lý báo cáo công việc.** 10](#_Toc530507114)

[**j. Đặc tả use-case quản lý tài khoản.** 10](#_Toc530507115)

[**k. Đặc tả use-case quản lý nhóm tài khoản.** 11](#_Toc530507116)

[**2.1.3.3. Lược đồ Activity.** 11](#_Toc530507117)

[**a. Quản lý đăng nhập.** 11](#_Toc530507118)

[**b. Quản lý Chatting.** 13](#_Toc530507119)

[**c. Quản lý công việc.** 14](#_Toc530507120)

[**d. Quản lý dự án.** 15](#_Toc530507121)

[**e. Quản lý danh mục.** 16](#_Toc530507122)

[**f. Quản lý báo cáo dự án.** 17](#_Toc530507123)

[**g. Quản lý báo cáo công việc.** 18](#_Toc530507124)

[**j. Quản lý nhóm tài khoản.** 19](#_Toc530507125)

[**k. Quản lý nhóm tài khoản.** 20](#_Toc530507126)

[**2.1.3.4. Lược đồ Sequence.** 21](#_Toc530507127)

[**a. Chức năng đăng nhập** 21](#_Toc530507128)

[**b. Chức năng trò chuyện** 23](#_Toc530507129)

[**c. Chức năng quản lý công việc.** 25](#_Toc530507130)

[**d. Chức năng quản lý dự án** 27](#_Toc530507131)

[**e. Chức năng quản lý danh mục** 29](#_Toc530507132)

[**g. Chức năng quản lý tài khoản** 31](#_Toc530507133)

[**j. Chức năng nhóm tài khoản.** 33](#_Toc530507134)

[**2.1.3.5 Cơ sở dữ liệu hệ thống.** 35](#_Toc530507135)

[**a. Xác định các thực thể hệ thống.** 35](#_Toc530507136)

[**b. Mối quan hệ giữa các thực thể.** 37](#_Toc530507137)

[**c. Lược đồ quan hệ.** 38](#_Toc530507138)

[**d. Chi tiết các trường dữ liệu.** 38](#_Toc530507139)

[**2.2 Cở sở kỹ thuật.** 45](#_Toc530507140)

[**2.2.1. Giới thiệu về Javascript.** 45](#_Toc530507141)

[**2.2.1.1. Khái niệm.** 45](#_Toc530507142)

[**2.2.1.2 Vai trò.** 45](#_Toc530507143)

[**2.2.2. Giới thiệu về ReactJs.** 46](#_Toc530507144)

[**2.2.2.1 Giới thiệu:** 46](#_Toc530507145)

[**2.2.2.2** **. Cách thức hoạt động của ReactJs** 46](#_Toc530507146)

[**2.2.2.3** **. Ưu và nhược điểm của ReactJs.** 46](#_Toc530507147)

[**2.2.3. Giới thiệu về Loopback.** 47](#_Toc530507148)

[**2.2.3.1 Giới thiệu:** 47](#_Toc530507149)

[**2.2.3.2 Ưu và nhược điểm.** 47](#_Toc530507150)

[**2.2.4. Giới thiệu về Mongodb.** 48](#_Toc530507151)

[**2.2.4.1: Giới thiệu:** 48](#_Toc530507152)

[**2.2.4.2. Đặt điểm của Mongodb** 48](#_Toc530507153)

[**2.2.4.3 Ưu điểm của MongodDB.** 48](#_Toc530507154)

[**Chương 3: Cài đặt và thử nghiệm** 49](#_Toc530507155)

[**3.1 Hướng dẫn cài đặt.** 49](#_Toc530507156)

[**3.1.1 Cài đặt môi trường.** 49](#_Toc530507157)

[**3.1.1.1 NodeJS.** 49](#_Toc530507158)

[**3.1.1.2 Loopback** 50](#_Toc530507159)

[**3.1.1.3 MongoDB** 50](#_Toc530507160)

[**3.2 Giao diện hệ thống.** 51](#_Toc530507161)

[**3.2.1. Giao diện phía người dùng.** 51](#_Toc530507162)

[**3.2.1.1 Trang chủ.** 51](#_Toc530507163)

[**3.2.1.2 Giao diện dự án.** 52](#_Toc530507164)

[**3.2.1.3 Giao diện màn hình công việc.** 53](#_Toc530507165)

[**3.2.1.4 Giao diện trò chuyện.** 56](#_Toc530507166)

[**3.2.1.5. Giao diện thông tin cá nhân.** 57](#_Toc530507167)

[**3.2.2 Giao diện người quản trị hệ thống.** 58](#_Toc530507168)

[**3.2.2.1 Giao diện quản lý tài khoản** 58](#_Toc530507169)

[**3.2.2.2 Giao diện quản lý nhóm tài khoản** 59](#_Toc530507170)

[**3.2.2.3 Giao diện quản lý danh mục** 59](#_Toc530507171)

[**Chương 4: Kết luận và hướng phát triển** 61](#_Toc530507172)

[**4.1 Kết quả làm được.** 61](#_Toc530507173)

[**4.2 Phần chưa làm được** 61](#_Toc530507174)

[**4.3 Kinh nghiệm làm được.** 61](#_Toc530507175)

[**4.4 Hướng phát triển.** 61](#_Toc530507176)

[**4.4 Kết luận.** 62](#_Toc530507177)

**DANH MỤC CÁC HÌNH**

[Hình. 1. Use case hệ thống. 4](#_Toc530339941)

[Hình. 2. Lược đồ activity quản lý đăng nhập 12](#_Toc530339942)

[Hình. 3. Lược đồ activity quản lý chatting 13](#_Toc530339943)

[Hình. 4. Lược đồ activity quản lý công việc 14](#_Toc530339944)

[Hình. 5. Lược đồ activity quán lý dự án 15](#_Toc530339945)

[Hình. 6. Lược đồ activity quản lý danh mục 16](#_Toc530339946)

[Hình. 7. Lược đồ activity quản lý báo cáo dự án 17](#_Toc530339947)

[Hình. 8. Lược đồ activity quản lý công việc 18](#_Toc530339948)

[Hình. 9. Lược đồ activity quản lý tài khoản 19](#_Toc530339949)

[Hình. 10. Lược đồ activity quản lý nhóm tài khoản 20](#_Toc530339950)

[Hình. 11. Giao diện chức năng đăng nhập 21](#_Toc530339951)

[Hình. 12. Lược đồ Sequence chức năng đăng nhập 22](#_Toc530339952)

[Hình. 13. Giao diện chức năng trò chuyện 23](#_Toc530339953)

[Hình. 14. Lược đồ Sequence chức năng trò chuyện 24](#_Toc530339954)

[Hình. 15. Giao diện chức năng quản lý công việc 25](#_Toc530339955)

[Hình. 16. Lược đồ Sequence chức năng quản lý công việc 26](#_Toc530339956)

[Hình. 17. Giao diện quản lý dự án 27](#_Toc530339957)

[Hình. 18. Lược đồ Sequence chức năng quản lý dự án 28](#_Toc530339958)

[Hình. 19. Giao diện chức năng quản lý danh mục 29](#_Toc530339959)

[Hình. 20. Lược đồ Sequence chức năng quản lý danh mục 30](#_Toc530339960)

[Hình. 21. Giao diện chức năng quản lý tài khoản 31](#_Toc530339961)

[Hình. 22. Lược đồ Sequence chức năng quản lý tài khoản 32](#_Toc530339962)

[Hình. 23. Giao diện chức năng quản lý nhóm tài khoản 33](#_Toc530339963)

[Hình. 24. Lược đồ Sequence chức năng quản lý nhóm tài khoản 34](#_Toc530339964)

[Hình. 25. Hình mối quan hệ giữa các thực thể 37](#_Toc530339965)

[Hình. 26. Lược đồ quan hệ các bảng thực thể 38](#_Toc530339966)

[Hình. 27. Hình kiểm tra version loopback 50](#_Toc530339967)

[Hình. 28. Hình ảnh kiểm tra version Database mongo 50](#_Toc530339968)

[Hình. 29. Giao diện trang chủ hệ thống 51](#_Toc530339969)

[Hình. 30. Giao diện xem thông tin dự án 52](#_Toc530339970)

[Hình. 31. Giao diện thêm mới một dự án 53](#_Toc530339971)

[Hình. 32. Giao diện màn hình công việc 53](#_Toc530339972)

[Hình. 33. Giao diện danh sách bình luận trao đởi công việc 54](#_Toc530339973)

[Hình. 34. Giao diện thêm mới công việc 55](#_Toc530339974)

[Hình. 35. Giao diện trò chuyện dạng danh sách 56](#_Toc530339975)

[Hình. 36. Giao diện trò chuyện dạng chatbox 56](#_Toc530339976)

[Hình. 37. Giao diện quản lý thông tin cá nhân 57](#_Toc530339977)

[Hình. 38. Giao diện thay đổi mật khẩu 57](#_Toc530339978)

[Hình. 39. Giao diện quản lý tài khoản 58](#_Toc530339979)

[Hình. 40. Giao diện thêm mới tài khoản 58](#_Toc530339980)

[Hình. 41. Giao diện quản lý nhóm tài khoản 59](#_Toc530339981)

[Hình. 42. Giao diện danh sách danh mục 59](#_Toc530339982)

[Hình. 43. Giao diện thêm mới danh mục 60](#_Toc530339983)

**DANH MỤC CÁC BẢNG**

[Bảng 1. Bảng chi tiết thực thể lịch sử đăng nhập 38](#_Toc530339984)

[Bảng 2. Bảng chi tiết thực thể danh mục công việc 39](#_Toc530339985)

[Bảng 3. Bảng chi tiết thực thể tin nhắn 39](#_Toc530339986)

[Bảng 4. Bảng chi tiết thực thể bình luận 40](#_Toc530339987)

[Bảng 5. Bảng chi tiết thực thể nhóm tài khoản 41](#_Toc530339988)

[Bảng 6. Bảng chi tiết thực thể ghi lịch sử hoạt động 41](#_Toc530339989)

[Bảng 7. Bảng chi tiết thực thể thông báo 42](#_Toc530339990)

[Bảng 8. Bảng chi tiết thực thể dự án 42](#_Toc530339991)

[Bảng 9. Bảng chi tiết thực thể công việc 43](#_Toc530339992)

[Bảng 10. Bảng chi tiết thực thể tài khoản 44](#_Toc530339993)

# **Chương 1: Tổng quan**

* 1. **Mục tiêu đề ra.**

Xây dựng một hệ thống cho phép quản lý công việc của một nhóm hoặc một công ty nhỏ. Người dùng có thể sử dụng hệ thống để phân công công việc của nhóm mình, giúp cho hiệu quả công việc càng cao mà thời gian giải quyết lại thấp. Hệ thống được viết trên nền tảng website và cho máy tính.

* 1. **Lý do chọn đề tài.**

Qua bốn năm học và làm việc nhóm với các bạn trong lớp em cảm thấy việc chia nhiệm vụ cho nhau tốn khá nhiều thời gian và việc trao đổi khó khăn. Vậy làm sao để giải quyết được vấn đề trên em đã quyết định sẽ xây dựng một hệ thống phân chia công việc cho một nhóm hoặc một công ty vừa và nhỏ.

Để tăng thêm việc cọ xát về lập trình web, nhằm để có thêm kinh nghiệm phát triển theo hướng lập trình viên, đề tài này vừa phù hợp với mong muốn của em có thể bước đầu phát triển và xây dựng một ứng dụng chạy trên nền tảng website.

* 1. **Phạm vi hệ thống.**

Phần mềm xây dựng cho phép một nhóm hoặc một công ty vừa và nhỏ để quản lý công việc của mình và đồng thời giao nhận nhiệm vụ cho nhau.

* 1. **Phương pháp nghiên cứu.**
* Nghiêm cứu cách thức phân công trong việc của một công ty với môi trường thực tế.
* Tham khảo các ứng dụng quản lý công việc trên thế giới.
* Tìm hiểu cách hoạt động của một website.
* Tìm hiểu và nghiêm cứu ngôn ngữ mới (ReactJs, NodeJs).
  1. **Ý nghĩa thực tiễn.**

Ứng dụng nhằm giúp cho việc quản lý công việc trở nên dễ dàng hơn và tiết kiệm thời gian hơn, đối tượng chính là công ty vừa và nhỏ.

**Chương 2: Cơ sở lý luận**

**2.1. Cơ sở lý thuyết.**

**2.1.1. Bài toán đặt ra.**

Do tình trạng công việc hiện đang quá tải mà trong khi đó việc phân chia các công việc còn quá thủ công trên giấy tờ vì thế dẫn đến tốn nhiều thời gian mà công việc lại không được xử lý nhiều. Vì vậy để giải quyết được vấn đề trên chúng ta cần áp dụng kỹ thuật công nghệ vào quá trình quản lý công việc của nhóm hoặc công ty của mình.

**2.1.2. Mô tả bài toán.**

Đối với Client, người sử dụng hệ thống sẽ có hai quyền:

* Thành viên bình thường của một nhóm trong hệ thống sẽ có các chức năng sau:
* Trao đổi công việc với nhau trược tiếp nhờ vào ứng dụng chatting online trên hệ thống mà không cần dùng phần mềm khách.
* Xem công việc của mình đã được giao trên hệ thống, sau khi hoàn thành công việc người dùng có thể cập nhật trạng thái công việc của mình cho người quản lý theo dõi. Đồng thời có thắc mắc gì thì có thể để lại bình luận phía dưới nhiệm vụ được giao.
* Tạo một dự án cho công việc của mình và thêm cách thành viên khác vào để tiện theo dõi các nhiệm vụ được giao.
* Có thể cập nhật dữ liệu của các nhiệm vụ hoặc các công việc mà mình đã tạo ra.
* Xem được danh sách và trạng thài công việc của mình.
* Đối với người quản trị của nhóm khi được người quản trị hệ thống tạo ra hoặc chỉ định đến sẽ có cách chức năng ngoài với chức năng của một thành viên bình thường:
* Cấp thêm tài khoản cho cách thành viên của mình với số lượng được quy định của người quản trị hệ thống.
* Xem báo cáo tiến độ công việc của nhóm mình.
* Cấu hình cách chức năng cho hệ thống.

Đối với người quản tri hệ thống sẽ có các chức năng chính sau:

* Tạo tài khoản cho cách thành viên mới vào sử dụng hệ thống (Tài khoản được tạo là tài khoản có quyển tạo ra các thành viên của họ)
* Cập nhật thông tin toài khoản người dùng.
* Cấu hình cách chức năng cho hệ thống.

**2.1.3. Phân tích và thiết kế hệ thống.**

**2.1.3.1. Lược đồ Use-Case.**

**a. Xác định các tác nhân của hệ thống.**

Dựa vào mô tả của bài toán, ta xác định được các tác nhân sau.

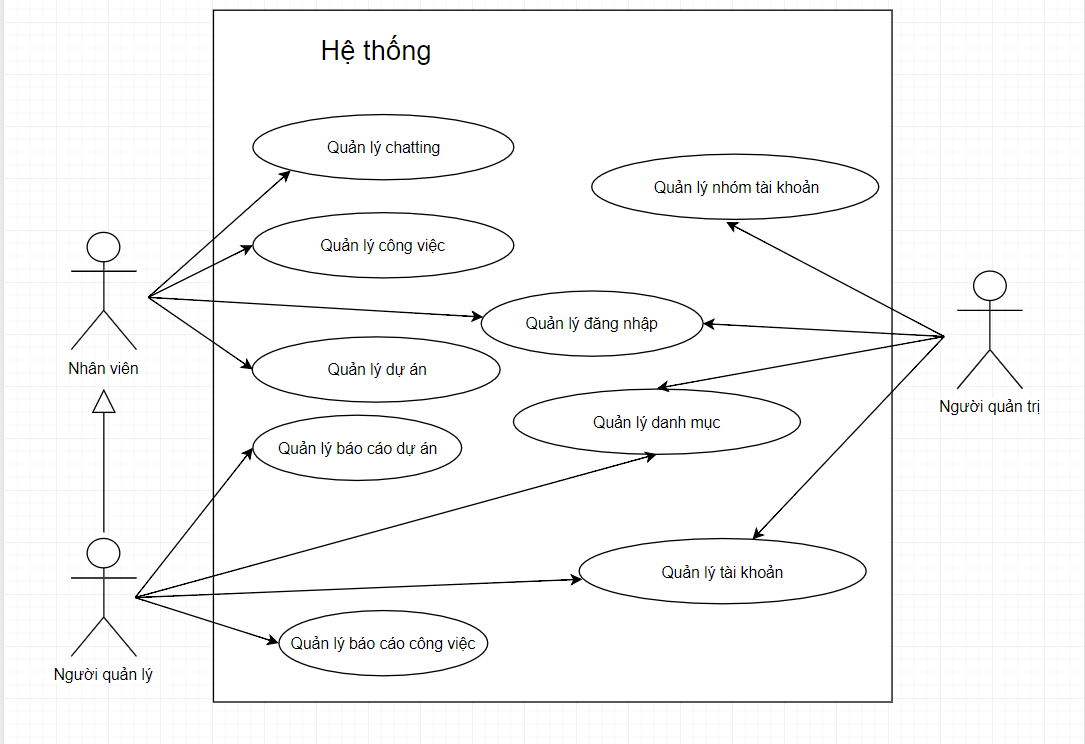
* Tác nhân người quản trị hệ thống.
* Tác nhân người quản lý.
* Tác nhân nhân viên.

**b. Xác định các Use-Case.**

Dựa vào các mô tả bài toán và qua quá trình phân tích thì tác xác định được các Use-Case sau:

* Quản lý Chatting.
* Quản lý công việc.
* Quản lý dự án.
* Quản lý tài khoản.
* Quản lý nhóm tài khoản.
* Quản lý đăng nhập.
* Quản lý báo cáo dự án.
* Quản lý báo cáo công việc

**c. Lược đồ Use-Case.**



Hình. 1. Use case hệ thống.

**2.1.3.2. Đặc tả Use-Case**

##### **a.** **Đặc tả use-case quản lý đăng nhập.**

Use case này mô tả cách đăng nhập vào hệ thống.

* Dòng sự kiến chính.
* Use case này bắt đầu khi có một actor muốn vào sử dụng hệ thống. Hệ thống sẽ yêu cầu actor nhập tên đăng nhập và mật khẩu.
* Hệ thống kiểm tra tên và mật khẩu mà actor đã nhập, nếu trùng khớp thì hệ thống sẽ cho phép actor đăng nhập vào hệ thống.
* Dòng sự kiện khác.
* Nếu actor chưa đăng nhập thì hệ thống phải yêu cầu actor đó thực hiện quá trình đăng nhập của mình và thông báo lỗi khi một trong những thông tin bắt buộc bị bỏ trống hoặc thông tin không chính xác.
* Khi actor đã thực hiện thành công quá trình đăng nhập thì actor có sử dụng được hệ thống và cập nhật thông tin cá nhân của mình.
* Các yêu cầu.
* Kiểm tra tài khoản có còn thời gian sử dụng hệ thống hay không.
* Kiểm tra tài khoản có bị người quản lý khoá hay không.
* Điểm mở rộng.

**b. Đặc tả use-case chatting.**

Use case này mô tả chức năng trò chuyện hệ thống

* Dòng sự kiện chính.
* Use case này bắt đầu khi có một actor muốn nhắn tin trực tiếp để trao đổi công việc với một actor khác có trong nhóm của hệ thống.
* Đầu tiên actor phải chọn người mình muốn trò chuyện.
* Nhập nội dung trò chuyện và gửi đi
* Người được trò chuyện sẽ sẽ được thông báo ngay lập tức và đồng thời mở cửa sổ trò chuyện để hiển thì nội dung tin nhắn.
* Dòng sự kiện khác.
* Các yêu cầu.
* Các actor chỉ được trò chuyện với các actor thuộc nhóm của mình.
* Chỉ hiển thị danh sách bạn bè có thuộc nhóm công việc của mình
* Điểm mở rộng.
* Các hành đồng trao đổi công việc với nhau phải đảm bảo chúng luôn được đồng bộ với nhau.
* Thông báo tình trạng online của các actor khác liên tục.

**c. Đặc tả use-case quản lý công việc.**

Use case này mô tả chức năng tạo công việc cho các actor của mình.

* Dòng sự kiện chính.
* Use case này bắt đầu khi có một công việc mới cần giao cho một người trong nhóm để làm.
* Để tạo một công việc và giao cho actor làm thì cần phải nhập đây đủ cách thông tin cần thiết cho hệ thống.
* Sau khi nhập đầy đủ thông tin thì cần phải chỉ định công việc giao cho ai và thời gian của công việc.
* Trong quá trình thực hiện công việc thì ở tab công việc sẽ có thể bình luận để hiểu rõ hơn nội dung công việc.
* Người chỉ định có thể cập nhật trạng thái công việc của mình sau khi hoàn thành.
* Sau khi trạng thái được 100% thì người quản lý sẽ kiểm tra lại công việc của nhân việc và đống công việc của nhân viên đó.
* Dòng sự kiện khác.
* Thông báo lỗi khi nhập thông tin không đúng hoặc có vấn đề khi khởi tạo công việc.
* Các yêu cầu.
* Khi một công việc tạo thành công thì hệ thông sẽ tự động thông báo công việc cho người được giao.
* Gửi mail xác nhận công việc cho người được giao.
* Hệ thống tự động nhắc nhở trước một ngày hết thời gian hoàn thành.
* Thời gian hoàn thảnh phải nằm trong khoản thời gian của dự án.
* Hệ thống tự động gửi mail nhắc nhở công việc trước khi công việc bắt đầu cũng như trước khi hết thời gian để hoàn thành công việc.
* Khi công việc chuyển sang trạng thái đóng thì các tác vụ liên quan đến công việc sẽ bị đóng.
* Điểm mở rộng.
* Ghi logs lại những bước mình đã làm lên hệ thống.

**d. Đặc tả use-case dự án.**

Use case này mô tả chức năng tạo dự án công việc cho nhóm của mình

* Dòng sự kiện chính.
* Use case này bắt đầu khi cần triển khai một dự án cho nhóm của mình.
* Để tạo một dự án thì actor điền đầy đủ thông tin để khởi tạo một dự án.
* Sau khi tạo dự án công việc thành công ta có thể cập nhật nội dung của dự án cũng như có thể tạo các công việc để hoản thành dự án.
* Khi các công việc trong một dự án đã hoàn thành hơn 80% thì người quản lý có thể đóng dự án nếu không phát sinh thêm công việc.
* Dòng sự kiện khác
* Thông báo lỗi khi nhập thông tin không đúng hoặc có vấn đề khi khởi tạo công việc
* Các yêu cầu.
* Khi một dự án tạo thành công thì hệ thông sẽ tự động thông báo đến các người được tham gia vào dự án.
* Gửi mail xác nhận công việc cho người được giao.
* Khi đống công việc thì các tác vụ tạo công việc, bình luận hay cập nhật dự án sẽ bị đóng theo.
* Điểm mở rộng.
* Ghi logs lại những bước mình đã làm lên hệ thống

**e. Đặc tả use-case quan lý danh mục.**

Use-case mô tả cách quản lý danh mục trong hệ thống

* Dòng sự kiện chính.
* Use-Case này bắt đầu khi một actor (người quản lý) đăng nhập vào hệ thống.
* Người quản lý sẽ chọn danh mục mà mình muốn xem.
* Người quản lý có thể chỉnh sửa thêm hay xoá danh mục vừa chọn.
* Dòng sự kiện khác.
* Thông báo lỗi khi nhập thông tin không đúng hoặc có vấn đề khi khởi tạo công việc
* Các yêu cầu.
* Đối với tài khoản là người quản trị hệ thống sẽ chỉ thấy các danh mục của người quản trị tạo ra.
* Các danh mục được người quản trị tạo ra sẽ được tạo măt định khi tạo người người quản lý.
* Đối với người quản lý sẽ thấy các danh mục được hệ thống tạo sẵn. Khi đó họ có thể cập nhật danh mục theo ý họ.
* Điểm mở rộng.

**f. Đặc tả use-case quản lý báo cáo dự án.**

Use-case mô tả cách quản lý các báo cao có trong hệ thống

* Dòng sự kiện chính.
* Use case bắt đầu khi có người quản lý vào xem các báo cáo của nhóm mình.
* Ở phần báo cáo actor có thể chọn hình thức báo cáo như là báo cáo tình trạng dự án, có bao nhiêu dự án hoàn thành…..
* Hình thức báo có sẽ theo tình trạng dự án, tên dự án.
* Báo cáo có thể in ra dưới dạng excel.
* Dòng sự kiện khác.
* Các yêu cầu.
* Chỉ có người quản lý mới có quyền xem các báo cáo của nhóm của mình.
* Điểm mở rộng.

**g. Đặc tả use-case quản lý báo cáo công việc.**

Use-case mô tả cách quản lý các báo cáo có trong hệ thống

* Dòng sự kiện chính.
* Use case bắt đầu khi có người quản lý vào xem các báo cáo của nhóm mình.
* Ở phần báo cáo actor có thể chọn hình thức báo cáo như là báo cáo tình trạng công việc, có bao nhiêu công việc hoàn thành…..
* Hình thức báo có sẽ theo tình trạng công việc, tên công việc.
* Báo cáo có thể in ra dưới dạng excel.
* Dòng sự kiện khác.
* Các yêu cầu.
* Chỉ có người quản lý mới có quyền xem các báo cáo của nhóm của mình.
* Điểm mở rộng.

**j. Đặc tả use-case quản lý tài khoản.**

Use case này mô tả hình thức quản lý tài khoản trong hệ thống.

* Dòng sự kiện chính.
* Use case này bắt đầu khi một actor muốn vào điều chỉnh tài khoản dưới quyền của mình.
* Actor có quyền tạo mới, cập nhật các thông tin tài khoản thuộc quyền quản lý của họ.
* Dòng sự kiện khác.
* Các yêu cầu.
* Đối với tài khoản người quản trị hệ thống sẽ có quyền tạo ra tài khoản người quản lý.
* Tài khoản người quản lý có quyền tạo và chỉnh sửa nhân viên thuộc quyền quản lý của mình.
* Đối vối tạo người quản lý thì người quản trị hệ thống cần chọn thời gian sử dụng cho tài khoản đó. Và đồng thời tạo một nhóm và để người mới tạo thuộc vào nhóm.
* Điểm mở rộng.

**k. Đặc tả use-case quản lý nhóm tài khoản.**

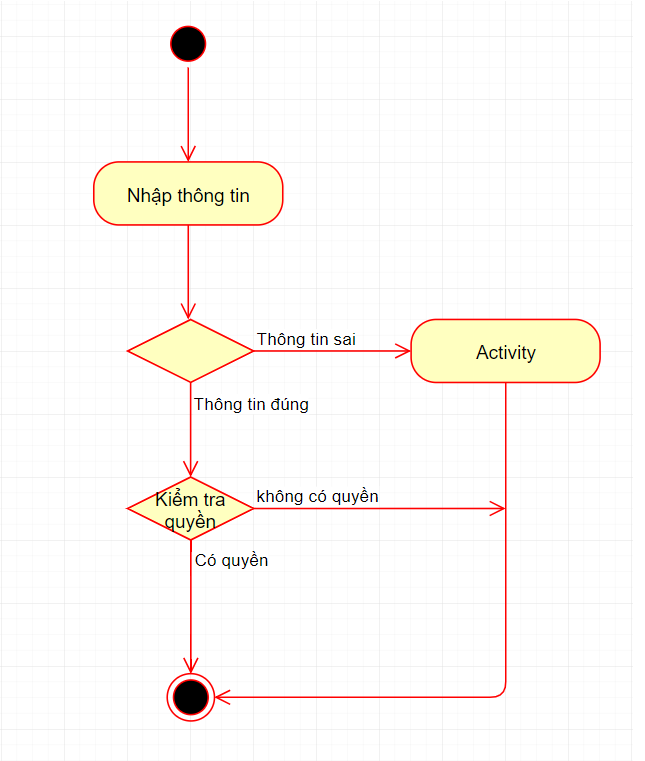
Use case này mô tả hình thức quản lý nhóm tài khoản trong hệ thống

* Dòng sự kiện chính.
* Use case này bắt đầu khi người quản trị hệ thống muốn cập nhật thông tin cho nhóm.
* Dòng sự kiện khác.
* Các yêu cầu.
* Ở use case này chỉ có người quản trị hệ thống mới có thể vào.
* Người quản trị hệ thống không thể tạo nhóm ở use case này.
* Điểm mở rộng.

**2.1.3.3. Lược đồ Activity.**

**a. Quản lý đăng nhập.**

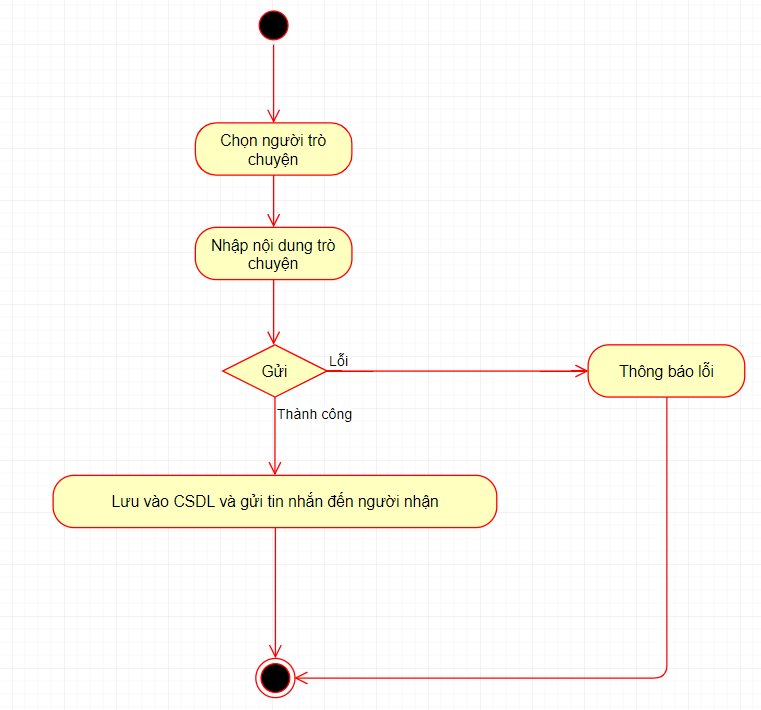
* **Chức năng.**
* Nhập tài khoản mật khẩu.
* Nếu thông tin chính xác sẽ thực hiện kiểm tra quyền tài khoản.
* Thông báo lỗi nếu thông tin không chính xác hoặc không có quyền truy cập hệ thống.
* **Lược đồ** **activity.**



Hình. 2. Lược đồ activity quản lý đăng nhập

**b. Quản lý Chatting.**

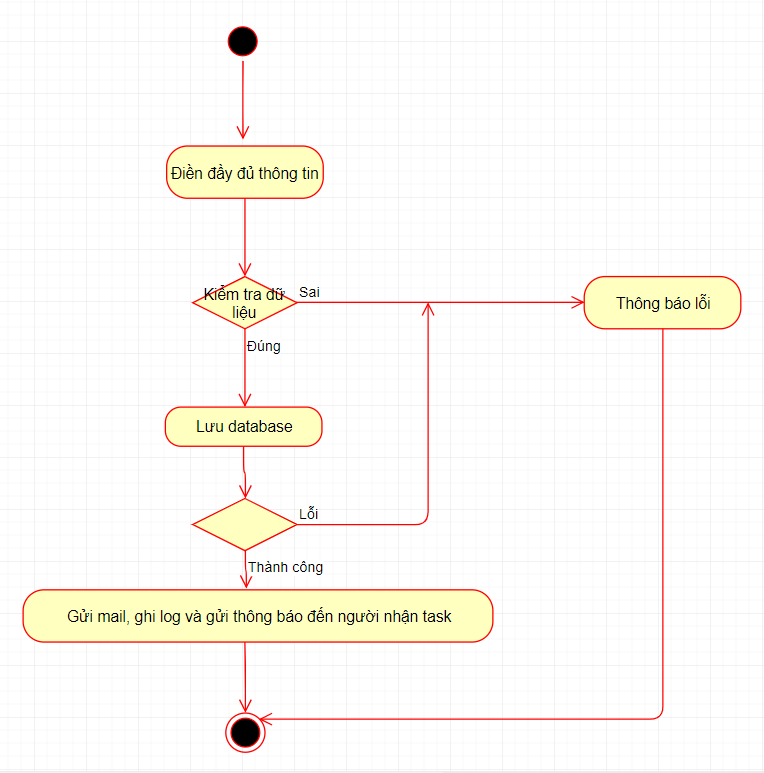
* **Chức năng.**
* Chọn người muốn trò chuyện.
* Nhập nội dung trò chuyện.
* Gửi nội dung tin nhắn
* Tin nhắn nếu được gửi thành công sẽ được lưu vào database và thông báo đến người nhận tin nhắn.
* Thông báo lỗi nếu như có lỗi.
* **Lược đồ activity.**



Hình. 3. Lược đồ activity quản lý chatting

**c. Quản lý công việc.**

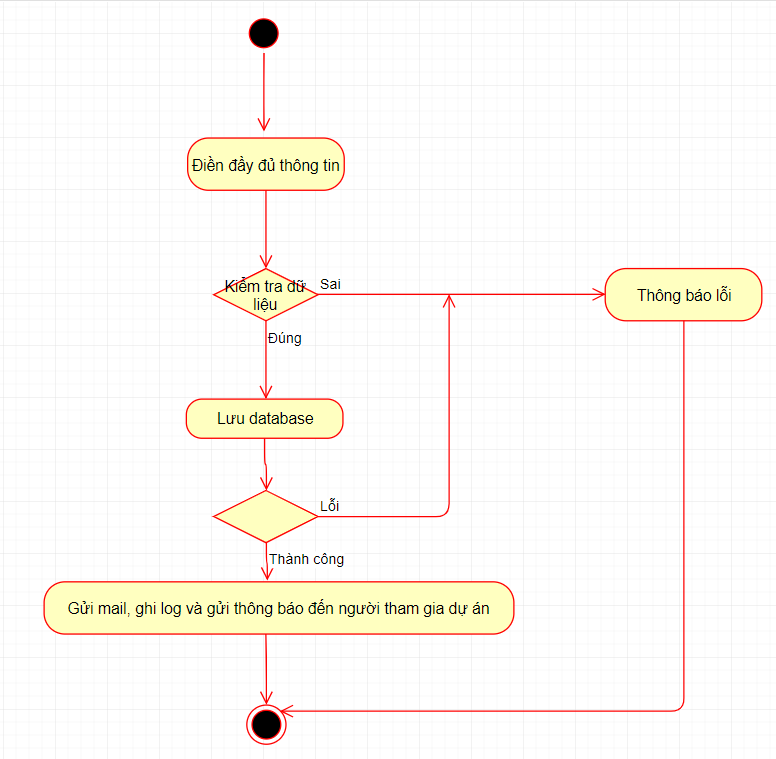
* **Chức năng.**
* Điền đầy đủ thông tin công việc.
* Kiểm tra dữ liệu vừa nhập.
* Lưu database và đồng thời gửi mail, ghi log và gửi thông báo đến người được giao công việc.
* Thông báo lỗi nêu như gặp lỗi
* **Lược đồ activity.**



Hình. 4. Lược đồ activity quản lý công việc

**d. Quản lý dự án.**

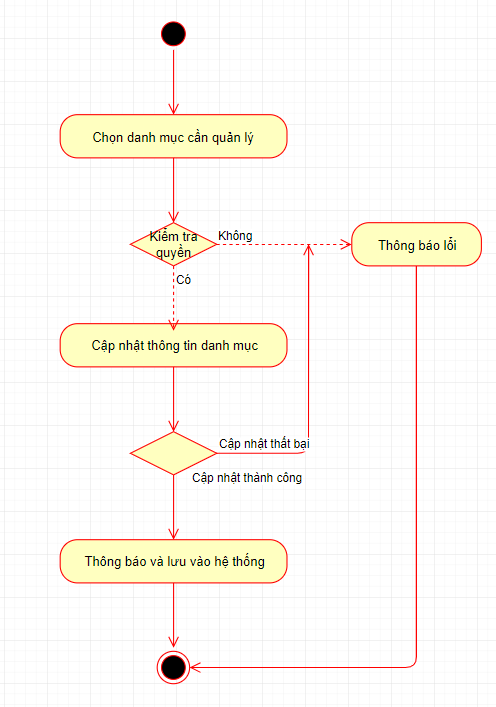
* **Chức năng.**
* Điền đầy đủ thông tin công việc.
* Kiểm tra dữ liệu vừa nhập.
* Lưu database và đồng thời gửi mail, ghi log và gửi thông báo đến người được tham gia vào dự án.
* Thông báo lỗi nêu như gặp lỗi.
* **Lược đồ activity.**



Hình. 5. Lược đồ activity quán lý dự án

**e. Quản lý danh mục.**

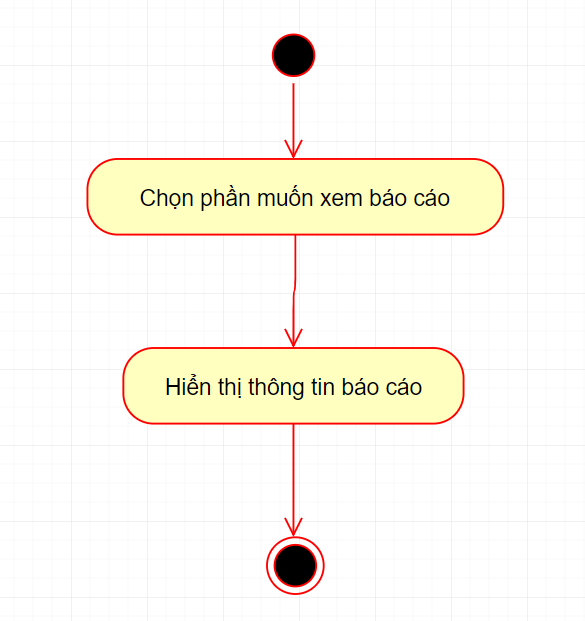
* **Chức năng.**
* Chọn danh mục cần quản lý.
* Cập nhật thông tin danh mục.
* Thông báo thành công khi cập nhật thành công.
* Thông báo lỗi khi có lỗi phát sinh.
* **Lược đồ activity.**



Hình. 6. Lược đồ activity quản lý danh mục

**f. Quản lý báo cáo dự án.**

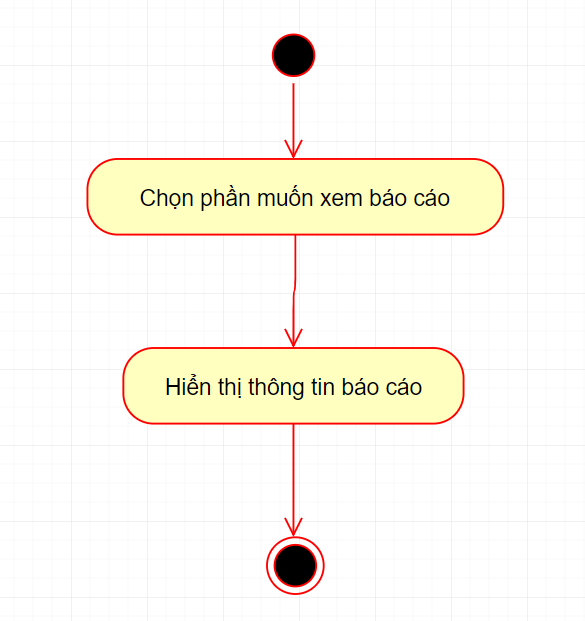
* **Chức năng.**
* Chọn phần muốn xem báo cáo.
* Hiển thị thông tin.
* **Lược đồ activity.**



Hình. 7. Lược đồ activity quản lý báo cáo dự án

**g. Quản lý báo cáo công việc.**

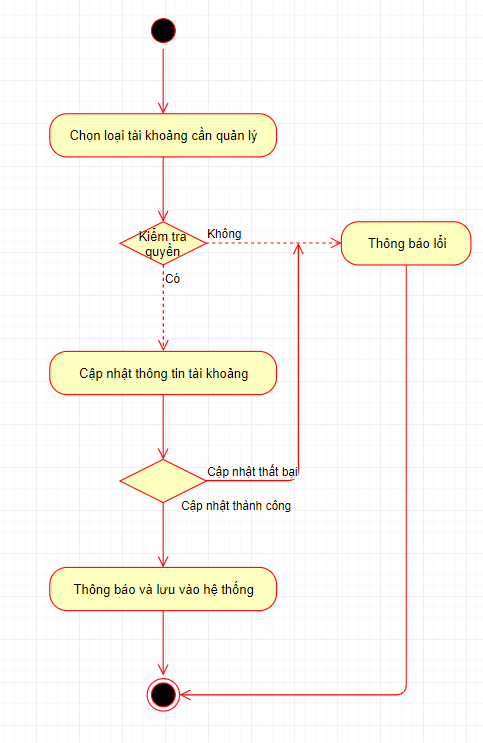
* **Chức năng.**
* Chọn phần muốn xem báo cáo.
* Hiển thị thông tin.
* **Lược đồ activity.**



Hình. 8. Lược đồ activity quản lý công việc

**j. Quản lý nhóm tài khoản.**

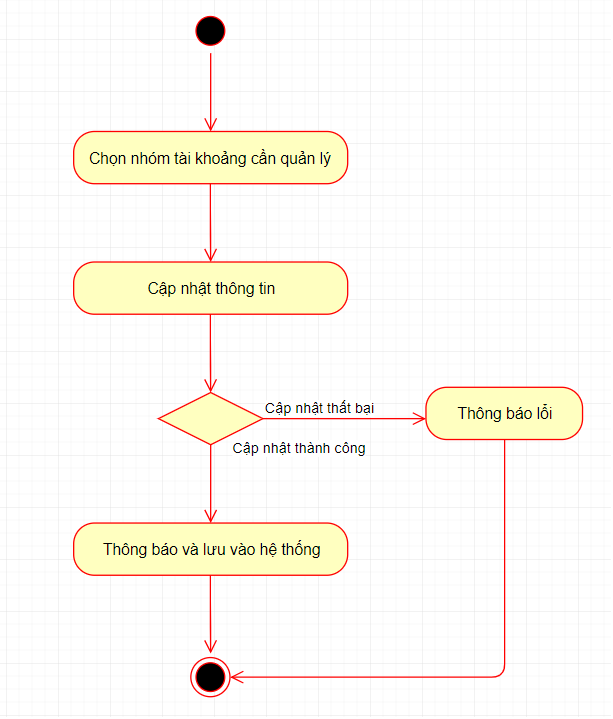
* **Chức năng.**
* Chọn tài khoản cần quản lý.
* Cập nhật thông tin tài khoản.
* Thông báo lỗi khi có lỗi phát sinh
* **Lược đồ activity.**



Hình. 9. Lược đồ activity quản lý tài khoản

**k. Quản lý nhóm tài khoản.**

* **Chức năng.**
* Chọn nhóm tài khoản.
* Cập nhật thông tin.
* Thông báo lỗi khi có lỗi phát sinh
* **Lược đồ activity.**

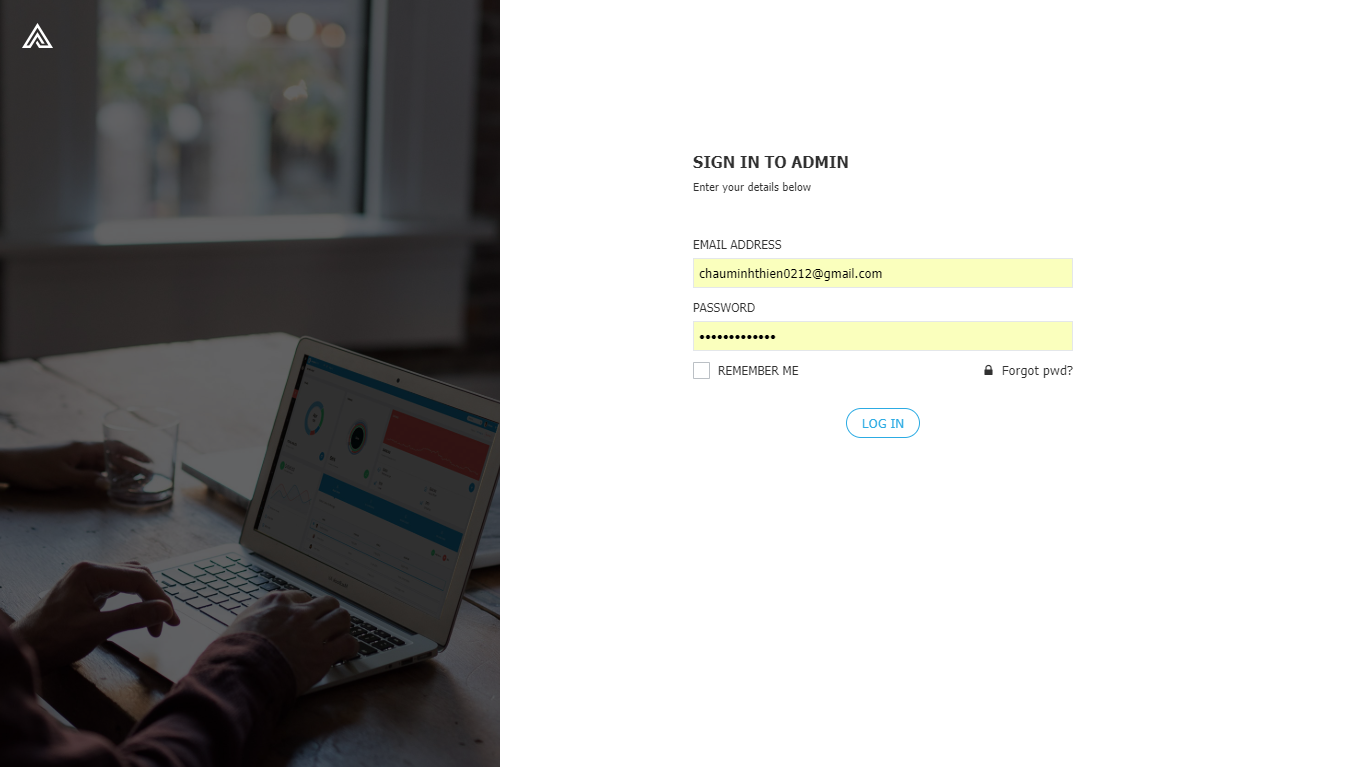


Hình. 10. Lược đồ activity quản lý nhóm tài khoản

**2.1.3.4. Lược đồ Sequence.**

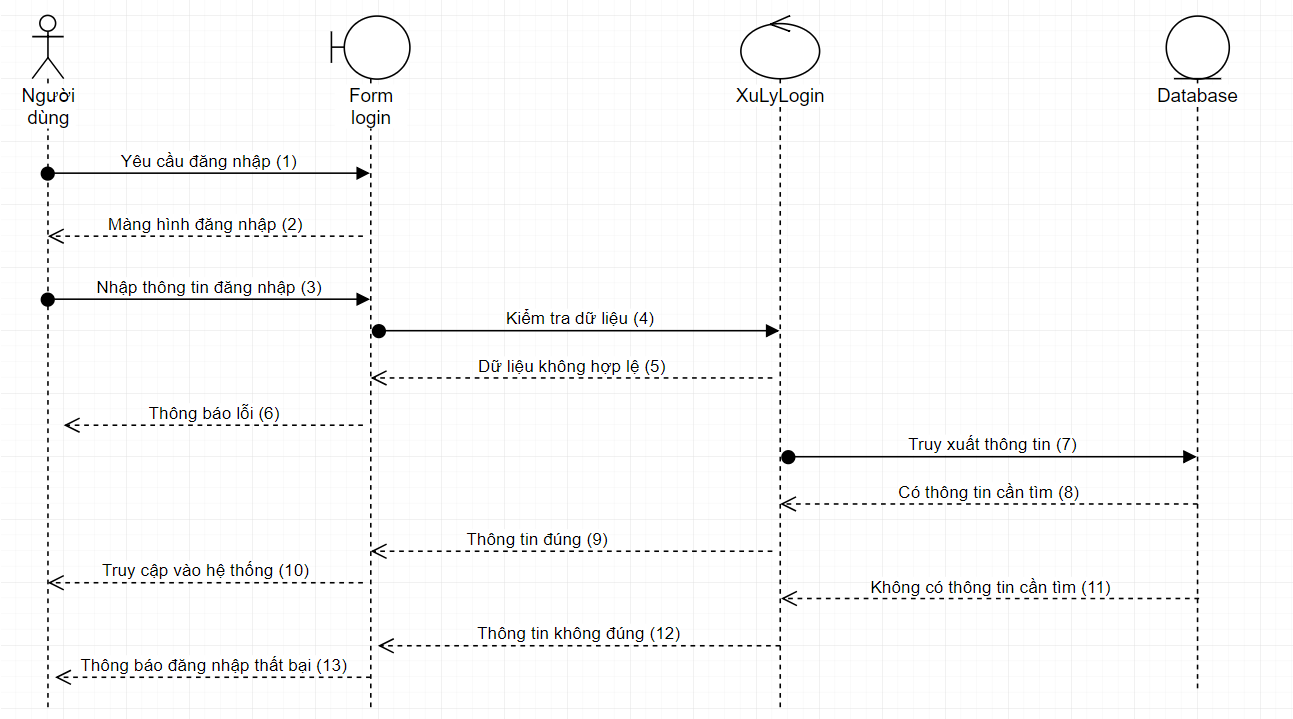
**a. Chức năng đăng nhập**

* **Giao diện.**



Hình. 11. Giao diện chức năng đăng nhập

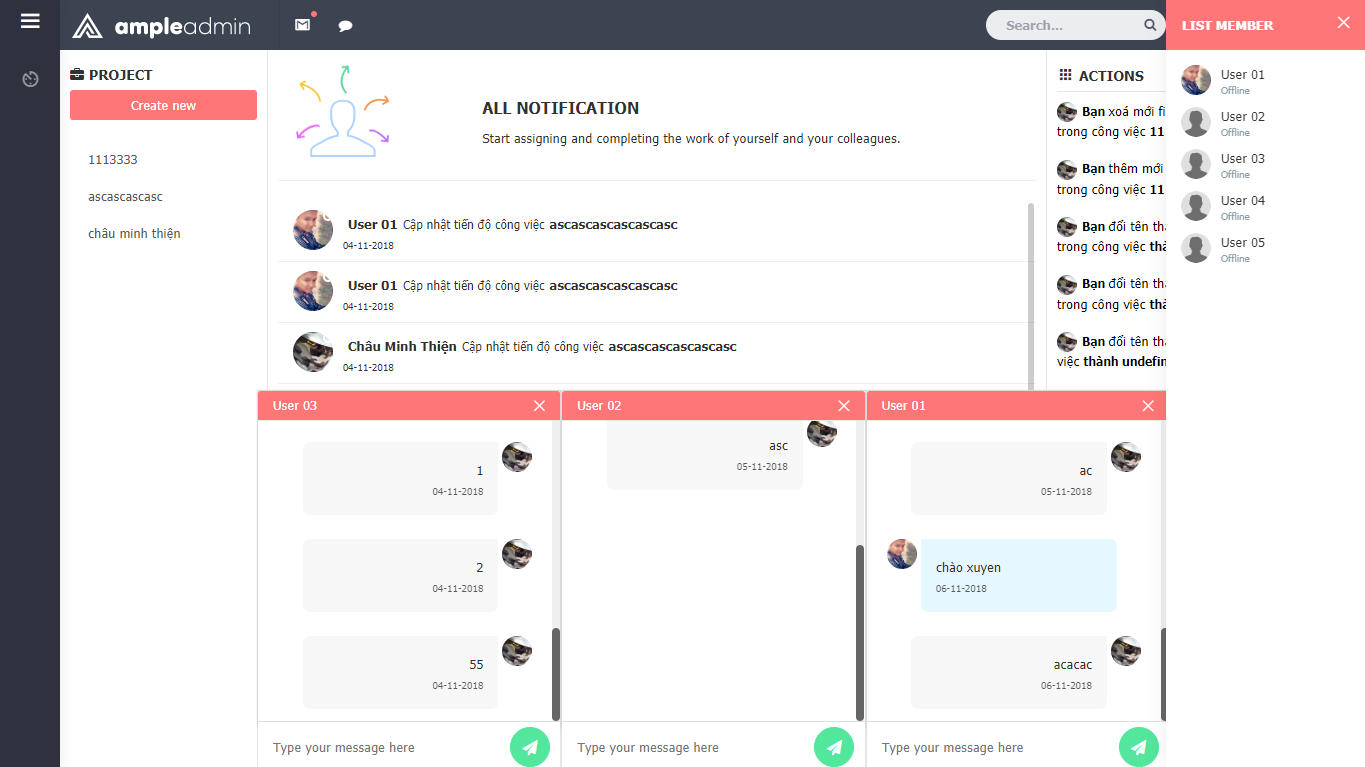
* **Các lớp.**
* Form login.
* XulyLogin
* Database
* **Các hành động.**
* Yêu cầu đăng nhập (1).
* Màn hình đăng nhập (2).
* Nhập thông tin đăng nhập (3).
* Kiểm tra dữ liệu (4).
* Dữ liệu không hợp lệ (5).
* Thông báo lỗi (6).
* Truy xuất thông tin (7).
* Có thông tin cần tìm (8).
* Thông tin đúng (9).
* Truy cập vào hệ thống (10).
* Không có thông tin cần tìm (11).
* Thông tin không đúng (12).
* Thông báo đăng nhập thất bại (13).
* **Lược đồ.**



Hình. 12. Lược đồ Sequence chức năng đăng nhập

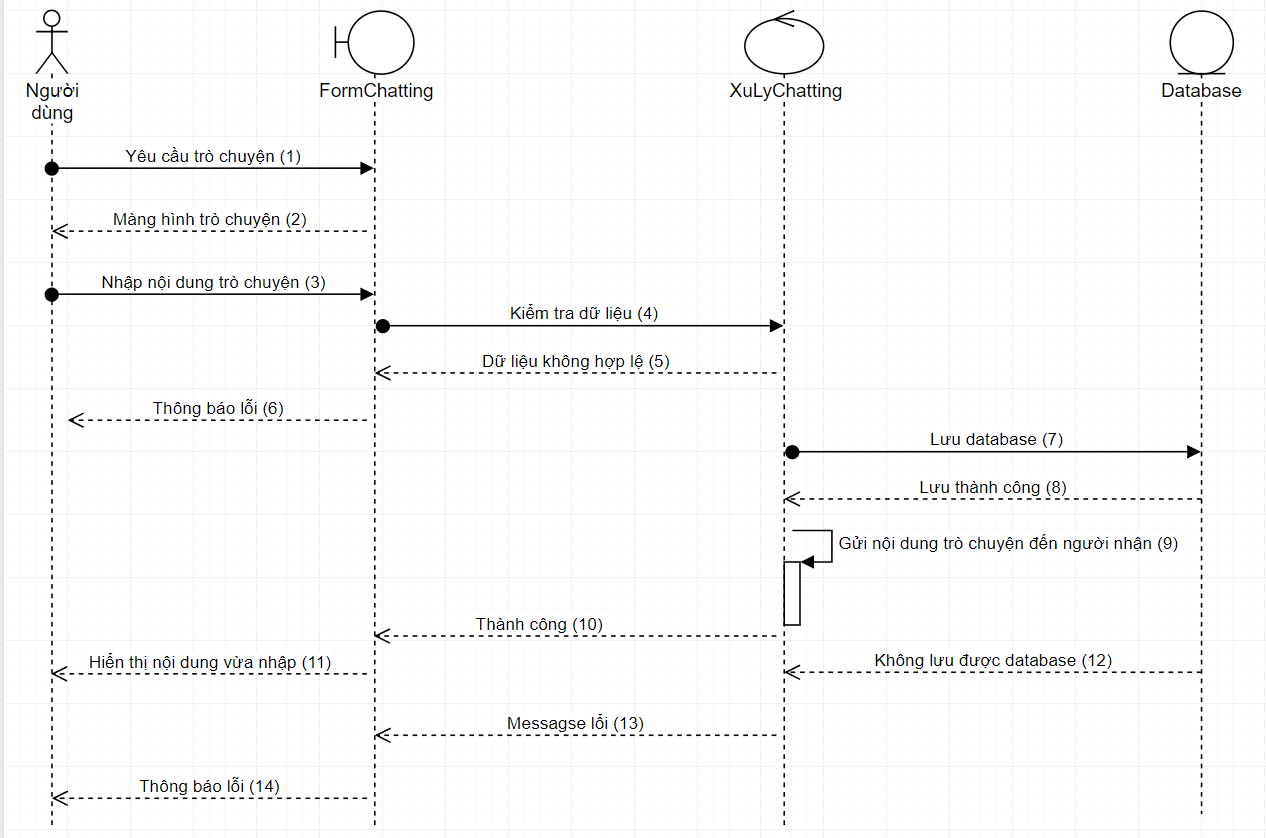
**b. Chức năng trò chuyện**

* **Giao diện.**



Hình. 13. Giao diện chức năng trò chuyện

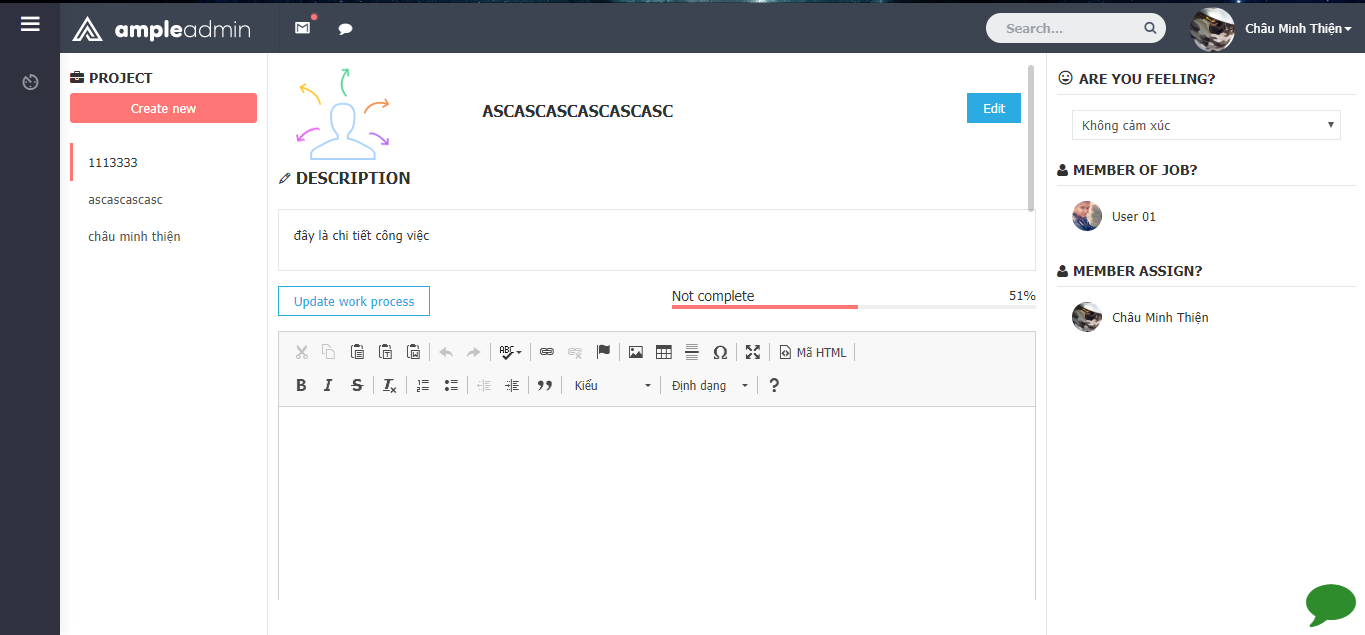
* **Các lớp.**
* FormChatting.
* XuLyChatting.
* Database
* **Các hành động.**
* Yêu cầu trò chuyện (1).
* Màn hình trò chuyện (2).
* Nhập nội dung trò chuyện (3).
* Kiểm tra dữ liệu (4).
* Dữ liệu không hợp lệ (5).
* Thông báo lỗi (6).
* Lưu database (7).
* Lưu thành công (8).
* Gửi nội dung trò chuyện đến người nhận (9).
* Thành công (10).
* Hiển thị nội dung vừa nhập (11).
* Không lưu được database (12).
* Messagse lỗi (13).
* Thông báo lỗi (14).
* **Lược đồ.**



Hình. 14. Lược đồ Sequence chức năng trò chuyện

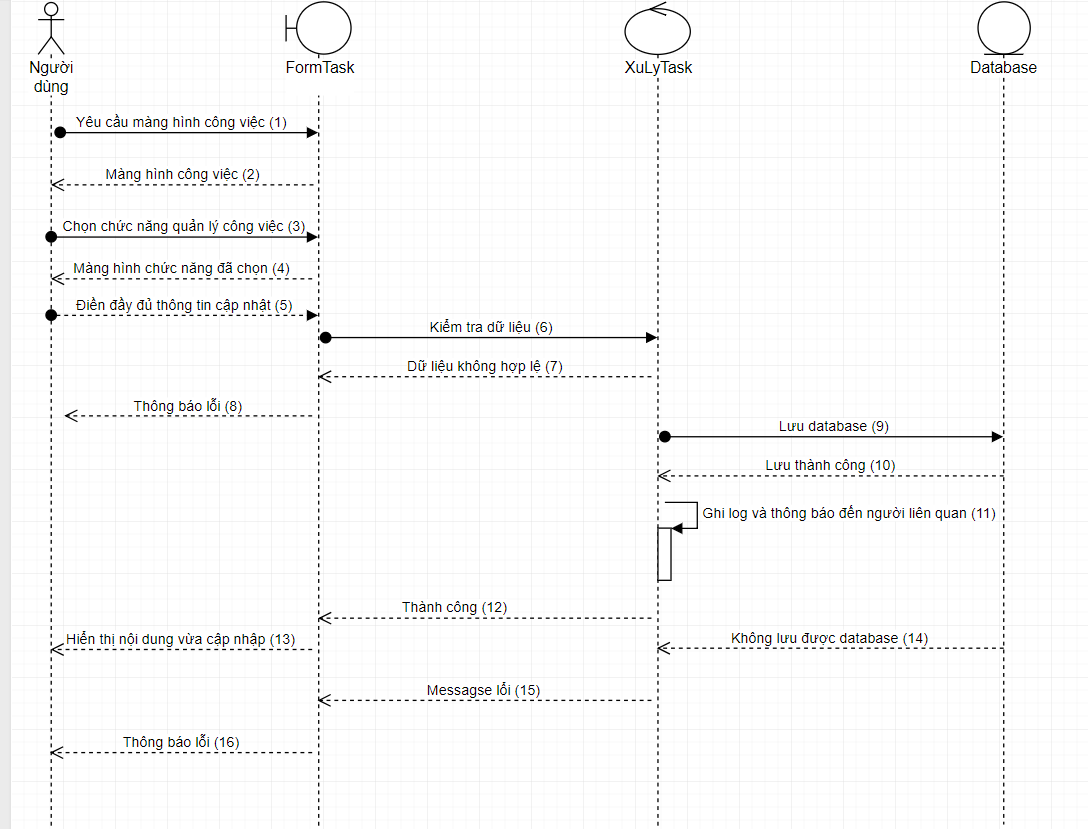
**c. Chức năng quản lý công việc.**

* **Giao diện.**



Hình. 15. Giao diện chức năng quản lý công việc

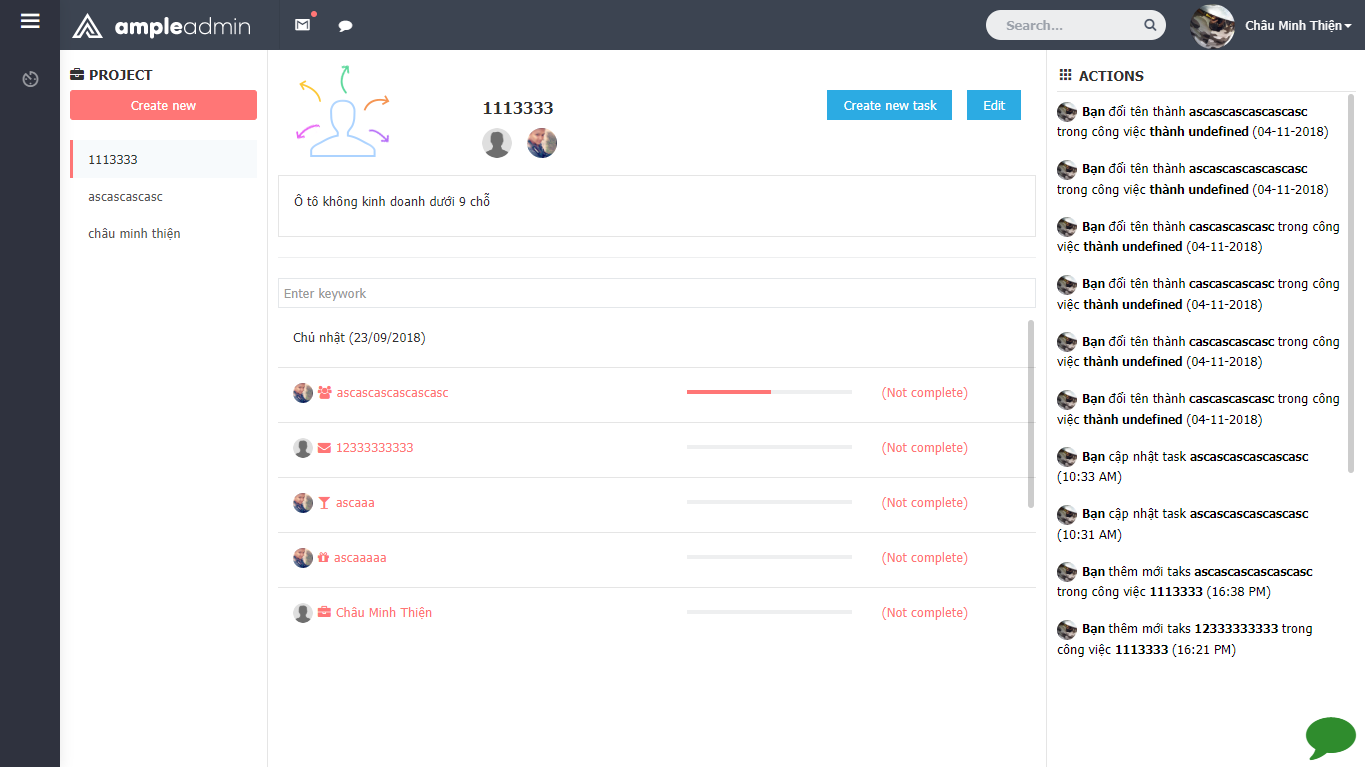
* **Các lớp.**
* FormTask.
* XuLyTask
* Database
* **Các hành động.**
* Yêu cầu màn hình công việc (1).
* Màn hình công việc (2).
* Chọn chức năng quản lý công việc (3).
* Màn hình chức năng đã chọn (4).
* Điền đầy đủ thông tin cập nhật (5).
* Kiểm tra dữ liệu (6).
* Dữ liệu không hợp lệ (7).
* Thông báo lỗi (8).
* Lưu database (9).
* Lưu thành công (10).
* Ghi log và thông báo đến người liên quan (11).
* Thành công (12).
* Hiển thị nội dung vừa cập nhập (13).
* Không lưu được database (14).
* Messagse lỗi (15).
* Thông báo lỗi (16).
* **Lược đồ.**



Hình. 16. Lược đồ Sequence chức năng quản lý công việc

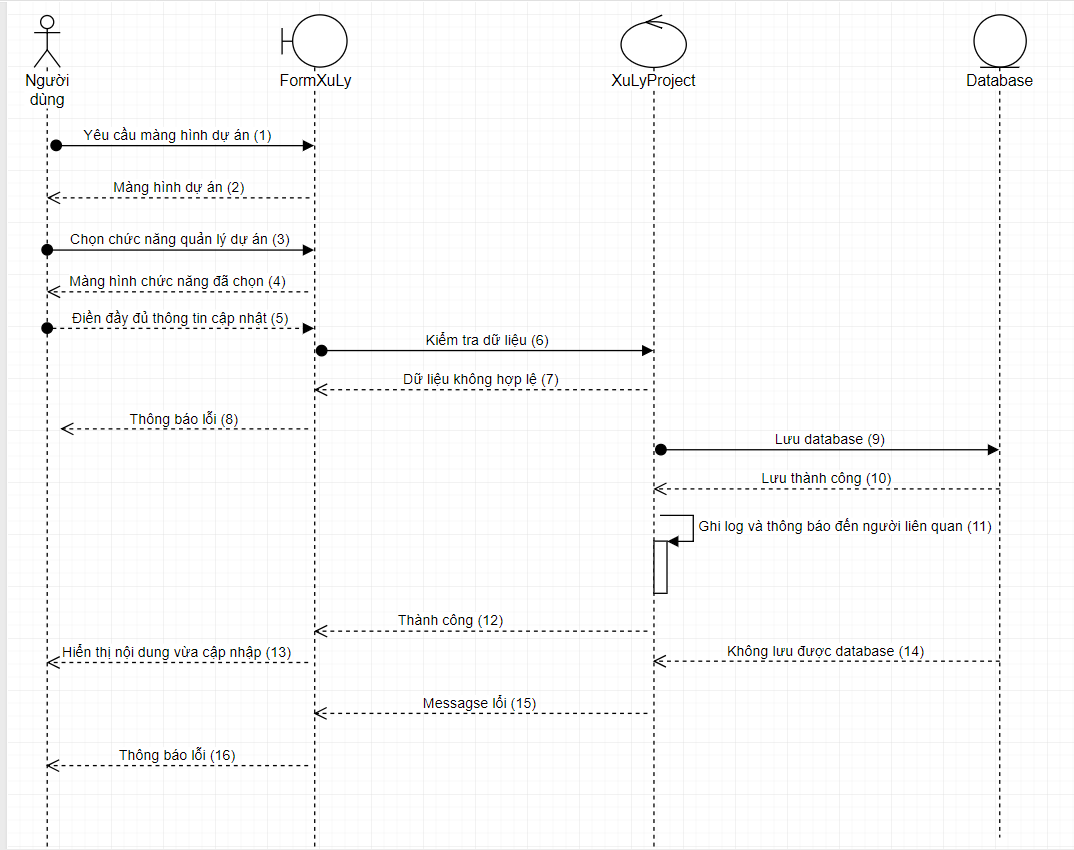
**d. Chức năng quản lý dự án**

* **Giao diện.**



Hình. 17. Giao diện quản lý dự án

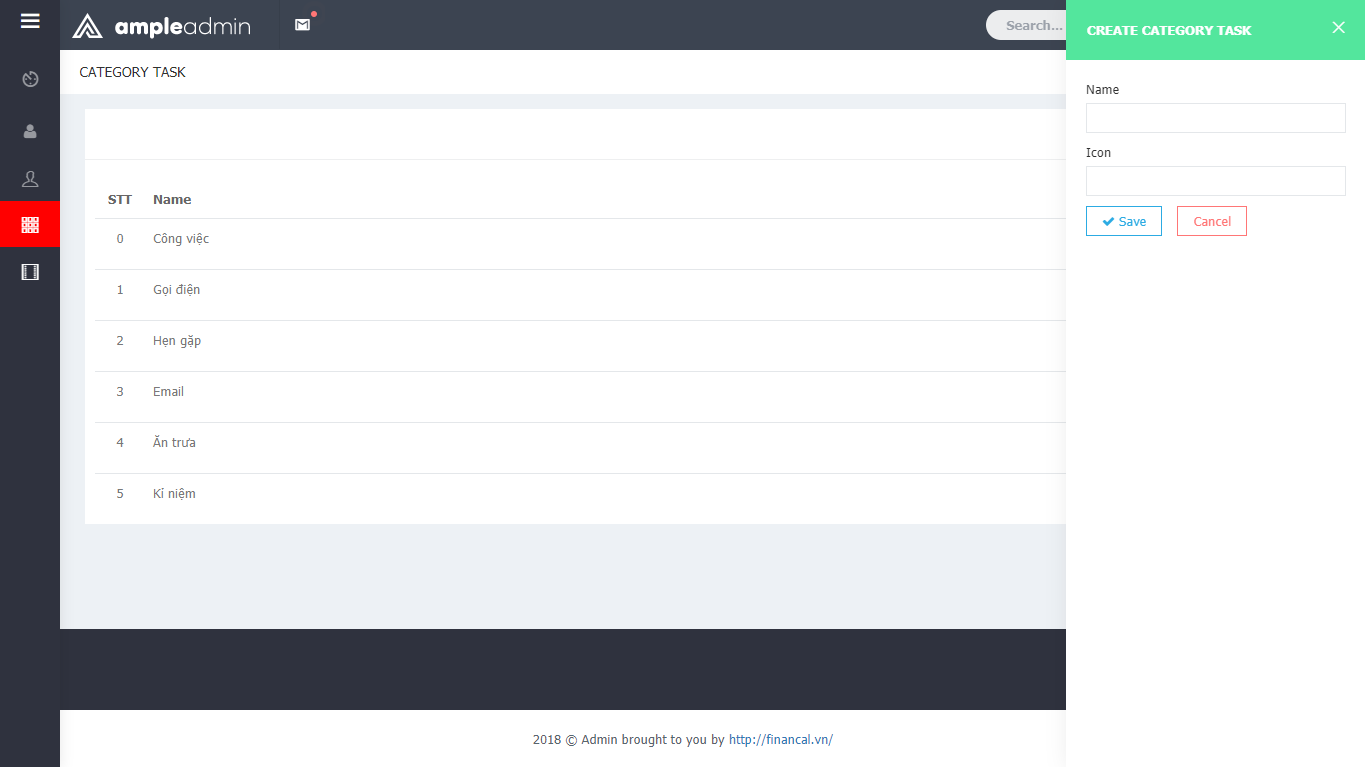
* **Các lớp.**
* FormXuLy.
* XuLyProject.
* Database
* **Các hành động.**
* Yêu cầu màn hình dự án (1).
* Màn hình dự án (2).
* Chọn chức năng quản lý dự án (3).
* Màn hình chức năng đã chọn (4).
* Điền đầy đủ thông tin cập nhật (5).
* Kiểm tra dữ liệu (6).
* Dữ liệu không hợp lệ (7).
* Thông báo lỗi (8).
* Lưu database (9).
* Lưu thành công (10).
* Ghi log và thông báo đến người liên quan (11).
* Thành công (12).
* Hiển thị nội dung vừa cập nhập (13).
* Không lưu được database (14).
* Messagse lỗi (15).
* Thông báo lỗi (16).
* **Lược đồ.**



Hình. 18. Lược đồ Sequence chức năng quản lý dự án

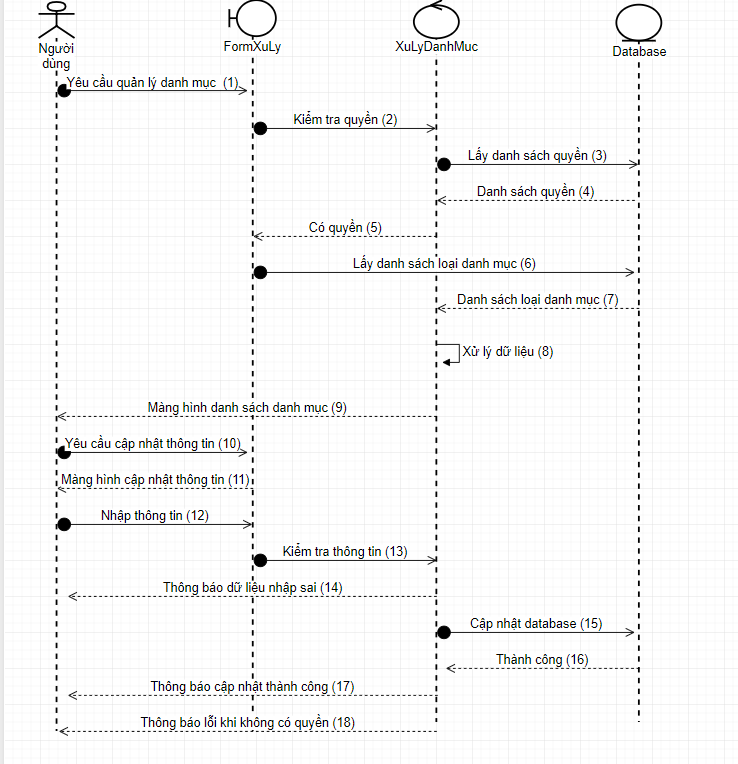
**e. Chức năng quản lý danh mục**

* **Giao diện.**



Hình. 19. Giao diện chức năng quản lý danh mục

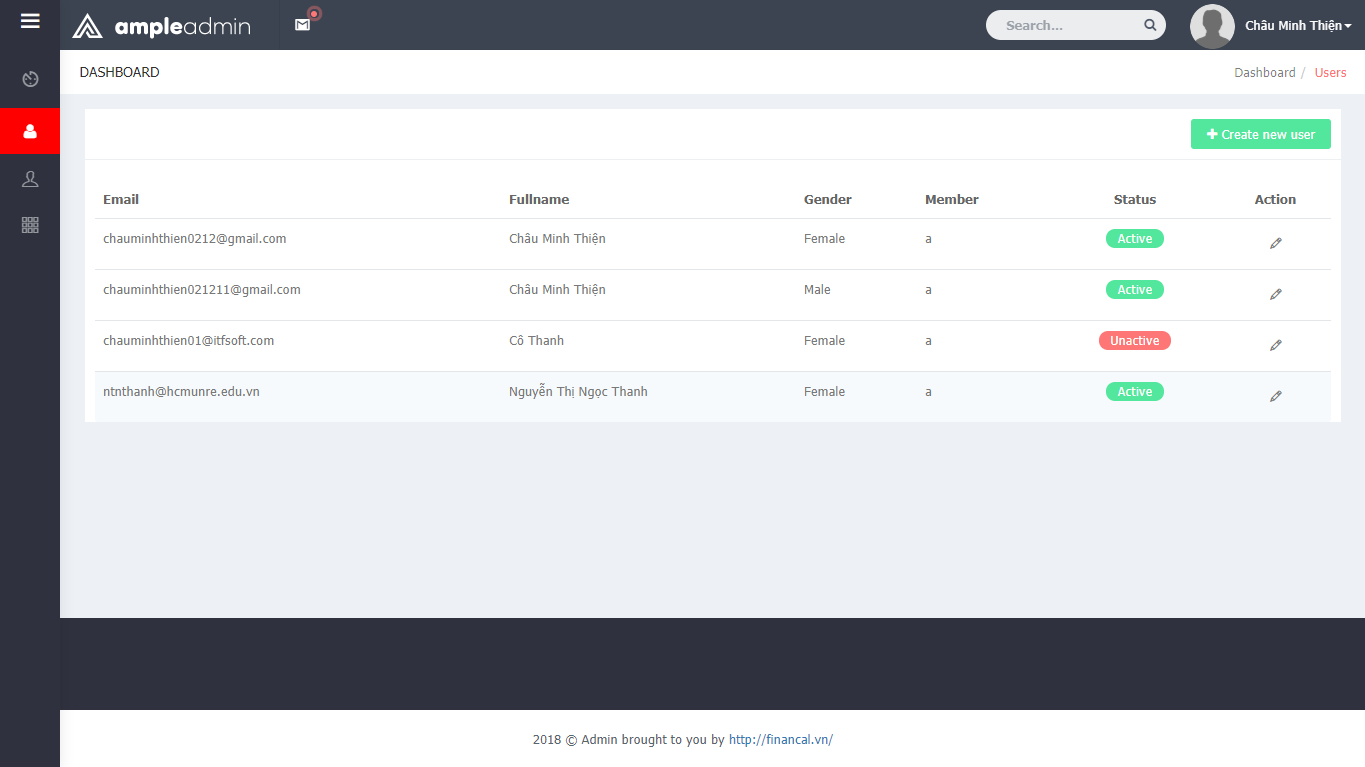
* **Các lớp.**
* FormXuLy.
* XuLyDanhMuc.
* Database.
* **Các hành động.**
* Yêu cầu quản lý danh mục (1).
* Kiểm tra quyền (2).
* Lấy danh sách quyền (3).
* Danh sách quyền (4).
* Có quyền (5).
* Lấy danh sách loại danh mục (6).
* Danh sách loại danh mục (7).
* Xử lý dữ liệu (8).
* Màn hình danh sách danh mục (9).
* Yêu cầu cập nhật thông tin (10).
* Màn hình cập nhật thông tin (11).
* Nhập thông tin (12).
* Kiểm tra thông tin (13).
* Thông báo dữ liệu nhập sai (14).
* Cập nhật database (15).
* Thành công (16).
* Thông báo cập nhật thành công (17).
* Thông báo lỗi khi không có quyền (18**)**
* **Lược đồ.**



Hình. 20. Lược đồ Sequence chức năng quản lý danh mục

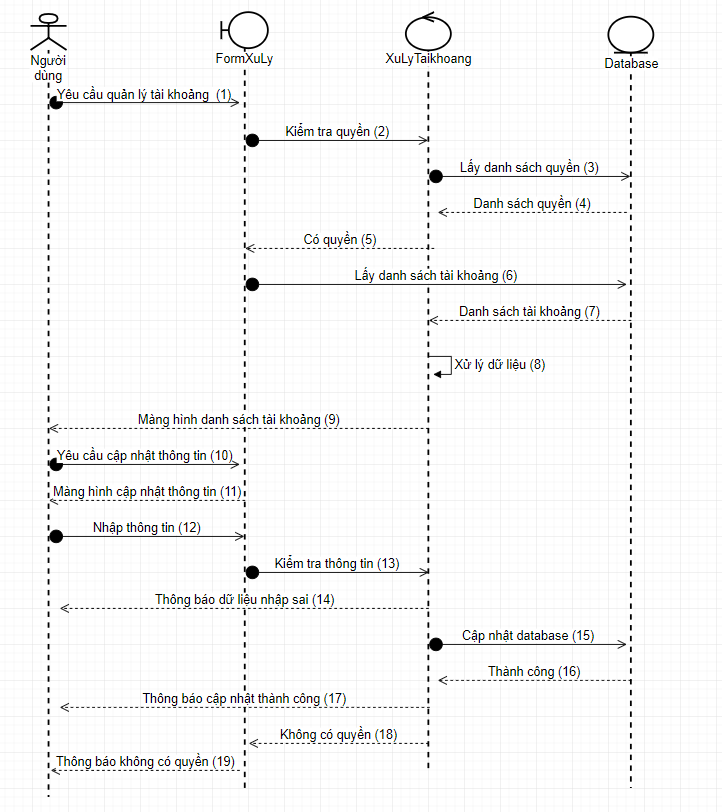
**g. Chức năng quản lý tài khoản**

* **Giao diện.**



Hình. 21. Giao diện chức năng quản lý tài khoản

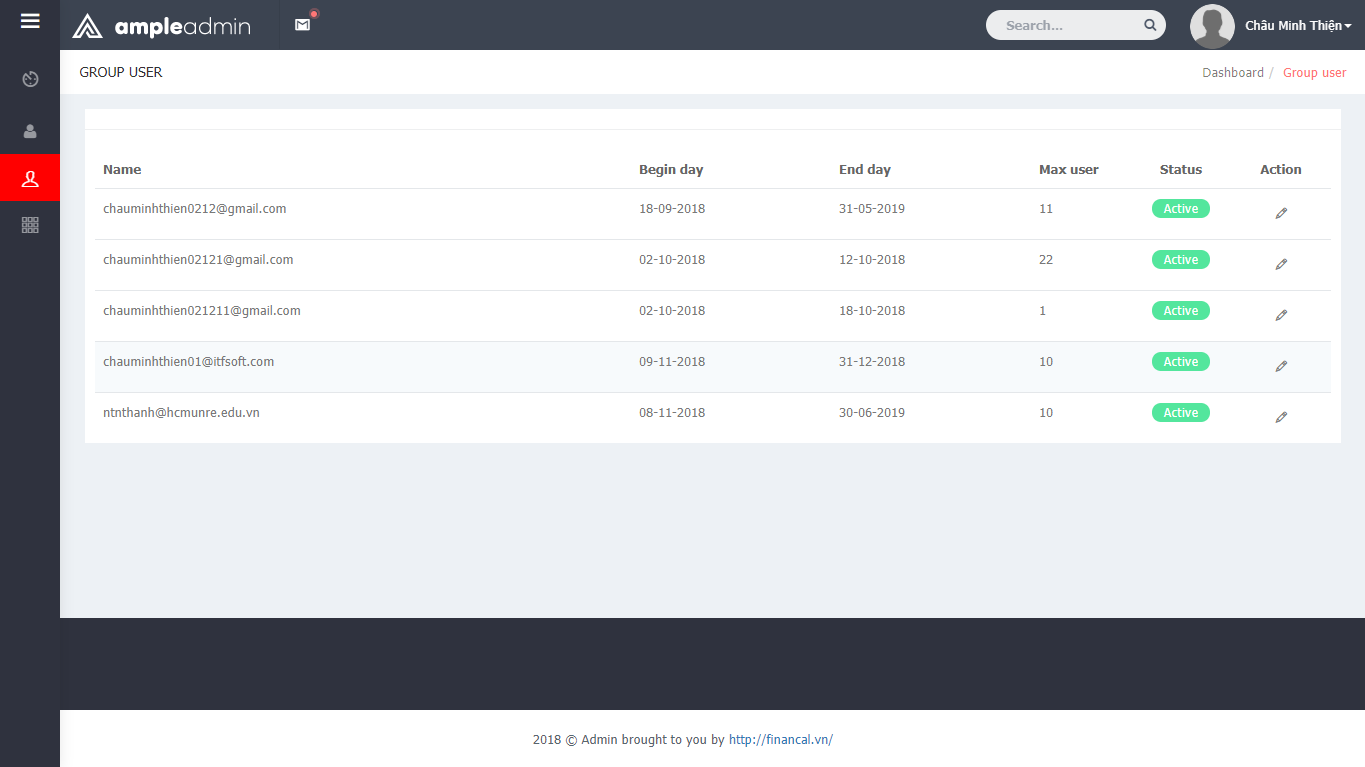
* **Các lớp.**
* FormXuLy.
* XuLyDanhMuc.
* Database
* **Các hành động.**
* Yêu cầu quản lý tài khoản (1).
* Kiểm tra quyền (2).
* Lấy danh sách quyền (3).
* Danh sách quyền (4).
* Có quyền (5).
* Lấy danh sách tài khoản (6).
* Danh sách tài khoản (7).
* Xử lý dữ liệu (8).
* Màn hình danh sách tài khoản (9).
* Yêu cầu cập nhật thông tin (10).
* Màn hình cập nhật thông tin (11).
* Nhập thông tin (12).
* Kiểm tra thông tin (13).
* Thông báo dữ liệu nhập sai (14).
* Cập nhật database (15).
* Thành công (16).
* Thông báo cập nhật thành công (17).
* Không có quyền (18).
* Thông báo không có quyền (19)
* **Lược đồ.**



Hình. 22. Lược đồ Sequence chức năng quản lý tài khoản

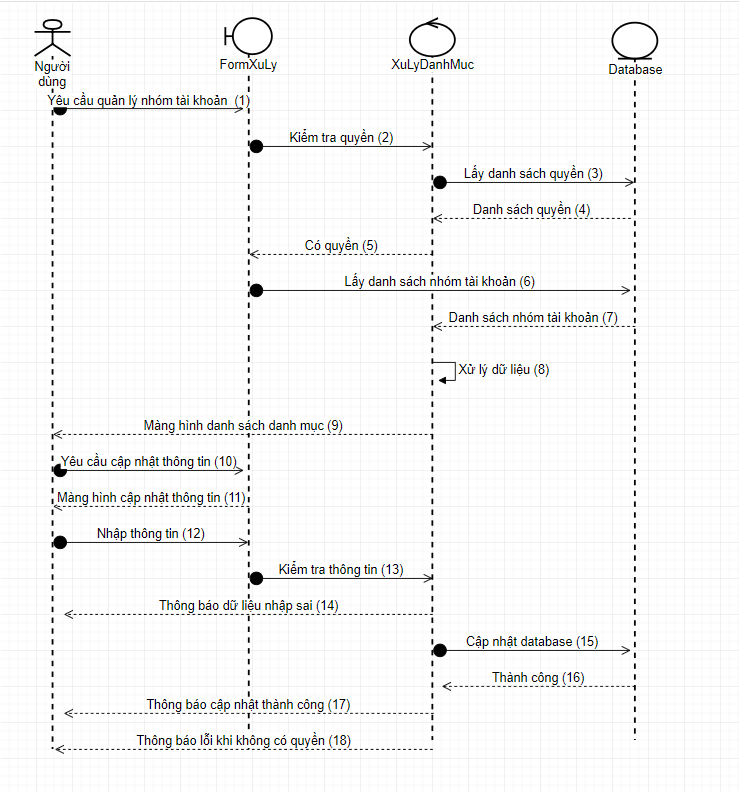
**j. Chức năng nhóm tài khoản.**

* **Giao diện.**



Hình. 23. Giao diện chức năng quản lý nhóm tài khoản

* **Các lớp.**
* FormXuLy.
* XuLyDanhMuc.
* Database
* **Các hành động.**
* Yêu cầu quản lý nhóm tài khoản (1).
* Kiểm tra quyền (2).
* Lấy danh sách quyền (3).
* Danh sách quyền (4).
* Có quyền (5).
* Lấy danh sách nhóm tài khoản (6).
* Danh sách nhóm tài khoản (7).
* Xử lý dữ liệu (8).
* Màn hình danh sách danh mục (9).
* Yêu cầu cập nhật thông tin (10).
* Màn hình cập nhật thông tin (11).
* Nhập thông tin (12).
* Kiểm tra thông tin (13).
* Thông báo dữ liệu nhập sai (14).
* Cập nhật database (15).
* Thành công (16).
* Thông báo cập nhật thành công (17).
* Thông báo lỗi khi không có quyền (18).
* **Lược đồ.**



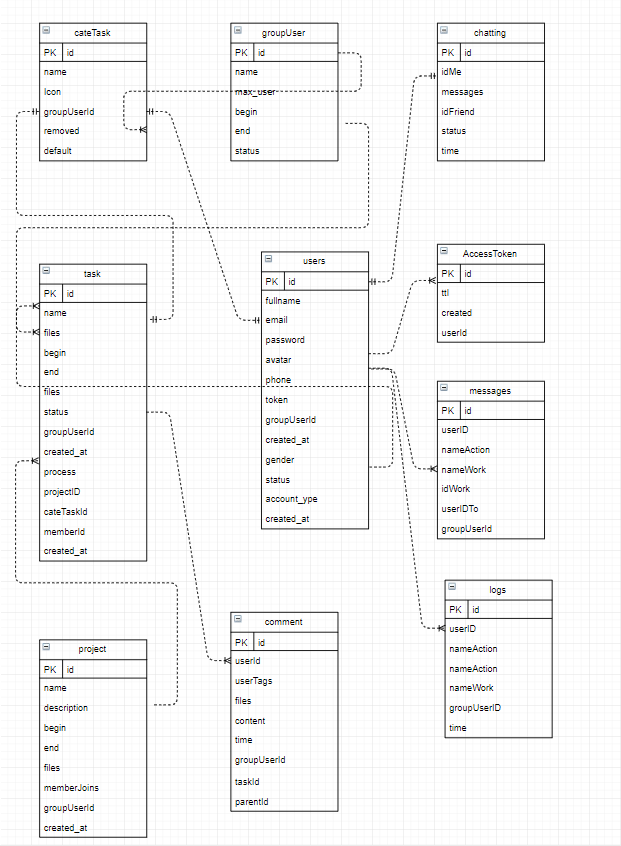
Hình. 24. Lược đồ Sequence chức năng quản lý nhóm tài khoản

**2.1.3.5 Cơ sở dữ liệu hệ thống.**

**a. Xác định các thực thể hệ thống.**

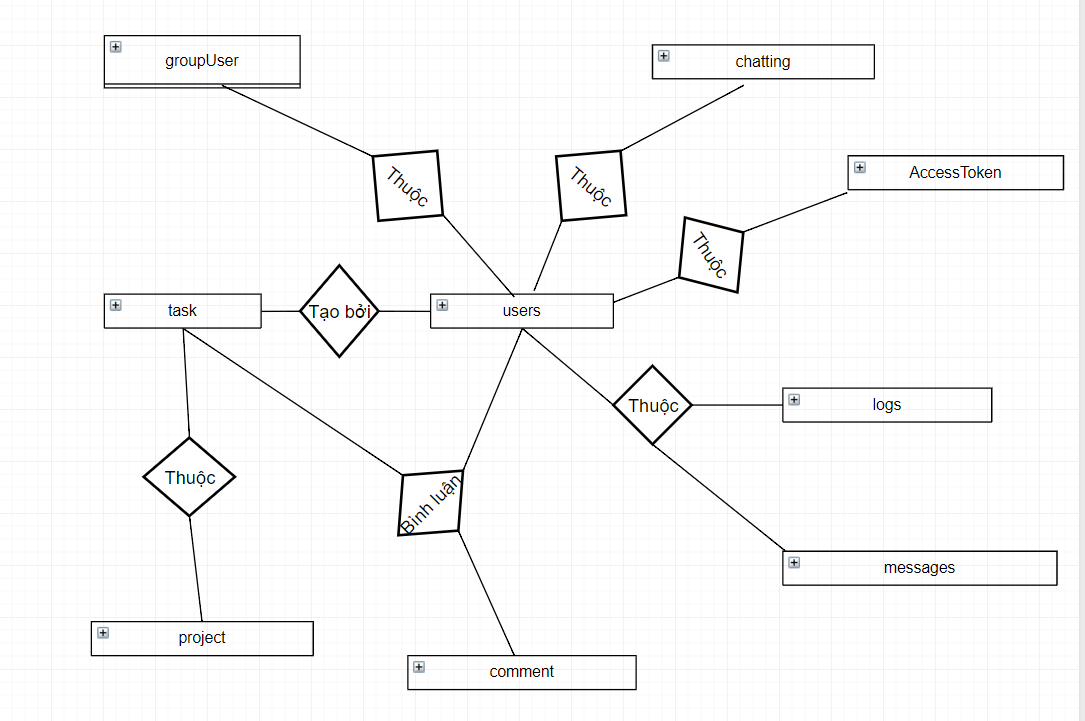
* Thực thể lịch sử đăng nhập.
* Dùng để lưu trữ các lịch sử đăng nhập vào hệ thống.
* Bảng lưu trữ trong CSDL: AccessToken(id, ttl, created, userId);
* Thực thể danh mục công việc.
* Dùng để lưu các danh mục của công việc. Người quản trị hệ thống và có thể thao tác với dữ liệu này.
* Bảng lưu trữ trong CSDL: cateTask(id, name, icon, groupUserId, removed, default).
* Thực thể tin nhắn.
* Dùng để lưu trữ nội dung người quản lý cuộc trò chuyện của các thành viên trong nhóm. Người quản lý và nhân viên có thể thao tác.
* Bản lưu trữ trong CSDL: chatting(id, idMe, message, idFriend, status, time).
* Thực thể bình luận.
* Dùng để lưu trữ nội dung cuộc bình luận về công việc của các thành viên trong nhóm. Người quản lý và nhân viên có thể thao tác.
* Bản lưu trữ trong CSDL: comment(id, userId, useTags, files, content, time, groupUserId, taskId, parentId).
* Thực thể nhóm tài khoản.
* Dùng để lưu trữ các nhóm tài khoản có trong hệ thống. Chỉ có người quản trị hệ thống mới được quyền thao tác.
* Bản lưu trữ trong CSDL: groupUser(id, name, max\_use, begin, end, status).
* Thực thể ghi nhận lịch sử.
* Dùng để lưu trữ các hành động của người dùng lên hệ thống, người sử dụng chỉ có quyền xem dữ liệu.
* Bản lưu trữ trong CSDL: logs(id, userID, nameAction, nameWork, groupUserID, time).
* Thực thể thông báo.
* Dùng để lưu các thông báo từ hệ thống.
* Bản lưu trữ trong CSDL: messages(id, userID, nameAction, nameWork, idWork, userIDTo, groupUser).
* Thực thể dự án công việc.
* Dùng để lưu thông tin dự án. Người quản lý và nhân viên có thể thao tác với cơ sở dữ liệu.
* Bản lưu trữ trong CSDL: project(id, name, description, begin, end, files, memberJoins, groupUserId, created\_at).
* Thực thể công việc.
* Dùng để lưu thông tin công việc của một dự án. Người quản lý và nhân viên có thể thao tác với cơ sở dữ liệu.
* Bản lưu trữ trong CSDL: task(id, name, files, begin, end, description, status, process, groupUserId, projectID, cateTaskId, memberId, created\_at).
* Thực thể tài khoản.
* Dùng để lưu thông tin tài khoản để đăng nhập vào hệ thống.
* Bản lưu trữ trong CSDL: users(id, fullname, email, password, avatar, phone, token, groupUserId, gender, status, account\_ype, created\_at).

**b. Mối quan hệ giữa các thực thể.**



Hình. 25. Hình mối quan hệ giữa các thực thể

**c. Lược đồ quan hệ.**



Hình. 26. Lược đồ quan hệ các bảng thực thể

**d. Chi tiết các trường dữ liệu.**

* Thực thể lịch sử đăng nhập.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên bảng: AccessToken** | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Id (khoá) | string |  |
| 2 | Ttl | number |  |
| 3 | Created | date |  |
| 4 | userId | String | Khoá ngoại bảng users |

Bảng 1. Bảng chi tiết thực thể lịch sử đăng nhập

* Thực thể danh mục công việc.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên bảng: cateTask** | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Id (khoá) | string |  |
| 2 | name | string |  |
| 3 | icon | string |  |
| 4 | removed | int | 0: chưa xoá, 1 đã xoá |
| 5 | default | Int | 0: do người quản trị hệ thống tạo, dùng để khởi tạo khi tao group.  1: item do người quản lý tạo |
| 6 | groupUserID | string | Khoá ngoại đến bảng groupUser |

Bảng 2. Bảng chi tiết thực thể danh mục công việc

* Thực thể tin nhắn.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên bảng: chatting** | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Id (khoá) | string |  |
| 2 | idMe | string |  |
| 3 | message | string |  |
| 4 | idFriend | string |  |
| 5 | status | Int | 0: chưa xem  1: đã xem |
| 6 | time | number |  |

Bảng 3. Bảng chi tiết thực thể tin nhắn

* Thực thể bình luận.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên bảng: comment** | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Id (khoá) | string |  |
| 2 | userId | string |  |
| 3 | userTags | string | Người được tag vào bình luận |
| 4 | content | string | Nội dung bình luận |
| 5 | groupUserID | string | Khoá ngoại đến bảng groupUser |
| 6 | time | number | Thời gian bình luận |
| 7 | taskId | string | Khoá ngoại đến bảng task |
| 8 | parentId | string | Trả lời cho 1 bình luận |

Bảng 4. Bảng chi tiết thực thể bình luận

* Thực thể nhóm tài khoản.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên bảng: groupUser** | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Id (khoá) | string |  |
| 2 | name | string |  |
| 3 | max\_user | Int | Số lượng tài khoản tối đa được tạo sử dụng cho nhóm |
| 4 | begin | number | Ngày bắt đầu |
| 5 | end | number | Ngày kết thúc |
| 6 | status | int | 0: group không bị khoá  1: group bị khoá. |

Bảng 5. Bảng chi tiết thực thể nhóm tài khoản

* Thực thể ghi nhận lịch sử.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên bảng: logs** | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Id (khoá) | string |  |
| 2 | userID | string |  |
| 3 | nameAction | string | Tên hành động |
| 4 | nameWork | string | Tên công việc |
| 5 | nameTask | string | Tên task |
| 6 | groupUserID | string |  |
| 7 | time | number |  |

Bảng 6. Bảng chi tiết thực thể ghi lịch sử hoạt động

* Thực thể thông báo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên bảng: messages** | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Id (khoá) | string |  |
| 2 | userID | string |  |
| 3 | nameAction | string | Tên hành động |
| 4 | nameWork | string | Tên công việc |
| 5 | idWork | string | Tên task |
| 6 | userIdTo | string |  |
| 7 | groupUserID | string |  |
| 8 | link | string |  |
| 9 | time | number |  |
| 10 | status | Int | 0: chưa xem  1: đã xem |

Bảng 7. Bảng chi tiết thực thể thông báo

* Thực thể dự án công việc.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên bảng: project** | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Id (khoá) | string |  |
| 2 | name | string |  |
| 3 | description | string |  |
| 4 | begin | number | Ngày bắt đầu dự án |
| 5 | end | number | Ngày kết thúc dự án |
| 6 | files | any | Danh sách file đính kém |
| 7 | memberJoins | any | Danh sách thành viên tham gia |
| 8 | groupUserID | string |  |
| 9 | createAt | number |  |

Bảng 8. Bảng chi tiết thực thể dự án

* Thực thể công việc.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên bảng: task** | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Id (khoá) | string |  |
| 2 | name | string |  |
| 3 | description | string |  |
| 4 | begin | number | Ngày bắt đầu công việc |
| 5 | end | number | Ngày kết thúc công việc |
| 6 | files | any | Danh sách file đính kém |
| 7 | projectId | any | Thuộc dự án náo |
| 8 | groupUserID | string |  |
| 9 | createAt | number |  |
| 10 | cateTaskId | String |  |
| 11 | memberId | String | Người được phân công |

Bảng 9. Bảng chi tiết thực thể công việc

* Thực thể tài khoản.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên bảng: users** | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | Id (khoá) | string |  |
| 2 | email | string |  |
| 3 | password | string |  |
| 4 | avatar | number |  |
| 5 | fullname | number |  |
| 6 | phone | any |  |
| 7 | token | any |  |
| 8 | address | string |  |
| 9 | groupUserID | number |  |
| 10 | gender | String |  |
| 11 | status | String | Người được phân công |
| 12 | account\_type |  | 0: người quản trị hệ thống.  1: người quản lý.  2: nhân viên |
| 13 | created\_at | String | Người tạo tài khoản |

Bảng 10. Bảng chi tiết thực thể tài khoản

**2.2 Cở sở kỹ thuật.**

**2.2.1. Giới thiệu về Javascript.**

**2.2.1.1. Khái niệm.**

Là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa ra. Đối tượng có sẵn ở đây là từ những đối tượng trong 1 trang Html, Javascript sẽ viết tiếp những kịch bản cho các đối tượng đó phù hợp với mong muốn của người sử dụng.

**2.2.1.2 Vai trò.**

Nếu 1 trang web có giao diện chỉ được xây dựng từ Html và Css, người dùng

sẽ thấy rất khô khan, nhàm chán. Javascrript có thể bổ sung hiệu ứng cho các thao tác khiến nó sinh động hơn, đơn cử như việc bấm vào một button và nó dần dần mất đi chứ không lập tức biến mất.

* Đồng thời có những thao tác ở người dùng đòi hỏi phải sử dụng Javascrip mới có thể tiến hành. Như việc yêu cầu khi người dùng nhập vào nội dung ở Textbox 1 thì Textbox 2 sẽ hiện nội dung tương ứng.
* Những ứng dụng to lớn của Javascript khiến người ta không thể quên nó được.

Hiện nay có rất nhiều libraries và framework được viết từ Javascript như:

* ReactJS: Một thư viện dùng để xây dựng ứng dụng Single Page

(Kiểu website hạn chế load lại toàn trang, khi cần thay đổi nội dung nào

đó chỉ thay đổi 1 phần của trang nên theo tổng thể vẫn là ở 1 trang).

* NodeJS: Một thư viện được phát triển phía Server dùng để xây dựng

ứng dụng realtime (Ứng dụng thời gian thực: một mô hình xử lý mà tính

đúng đắn của hệ thống không chỉ phụ thuộc vào kết quả tính toán lôgic

mà còn phụ thuộc vào thời gian mà kết quả này phát sinh ra.).

* ExtJS: Một Framework dùng xây dựng ứng dụng quản lý (Web

Applications).

* jQuery: Một thư viện rất mạnh về hiệu ứng (như 1 cửa sổ trượt ra, mờ

dần, chớp tắt .v.v…).

**2.2.2. Giới thiệu về ReactJs.**

**2.2.2.1 Giới thiệu:**

React.js là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Nếu như AngularJS là một Framework cho phép nhúng code javasscript trong code html thông qua các attribute như ng-model, ng-repeat...thì với react là một library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS.Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn.

* + - 1. **. Cách thức hoạt động của ReactJs**
* Khi phía client request lên server thì phía server (không quan trọng là node.js, php, asp.net, ruby...) Nói chung là nó không quan tâm tới server viết bằng ngôn ngữ gì.
* Server sẽ trả về cho client và trong đó sẽ có mã script của react.
* Và lúc đó react mới compile code và hiển thị cho người dùng. Khi React mới ra đời thì cơ chế hoạt động của React là khi server trả dữ liệu về cho người dùng thì lúc đó mới tải về mã React. Và lúc đó React mới rennder ra HTML. Chính vì vậy mà các image hay keyword được render bởi React sẽ không tốt cho SEO. Chính vì vậy mà React mới ra 1 kỹ thuật mới là Server sẽ render mã React thành HTML rồi mới trả về phía Client. Như vậy GG sẽ bắt được khi phía server trả HTML về phía client. Tuy nhiên, điều quan trọng hiện giờ GG đều bắt được các image hoặc keyword bằng cả 2 cách này nên bạn có thể áp dụng cả 2 cách đều được.
  + - 1. **. Ưu và nhược điểm của ReactJs.**
* **Ưu điểm:**
* Reactjs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật lên DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí.
* Reactjs giúp việc viết các đoạn code JS dễ dàng hơn.
* Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.
* Render tầng server.
* **Nhược điểm:**
* Reactjs chỉ phục vụ cho tầng View.
* Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.
* React khá nặng nếu so với các framework khác React có kích thước tương tương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh.
* Khó tiếp cận cho người mới học Web.

**2.2.3. Giới thiệu về Loopback.**

**2.2.3.1 Giới thiệu:**

Đây là một opensource framework dạng mobile backend as a service (mBaaS) rất mạnh để thiết kế Rest API.

Loopback được viết dựa trên nên tản express giúp cho việc tạo API một cách dễ dàng và nhanh chống.

**2.2.3.2 Ưu và nhược điểm.**

* **Ưu điểm.**
* Nhanh chóng tạo các API REST end-to-end động.
* Kết nối với nhiều hệ cơ sở dữ liệu khác nhau.
* Viết trên nền tảng Nodejs nên tốc độ xử lý nhanh.
* Document hướng dẫn chi tiết.
* **Nhược điểm.**
* Tài liệu hướng dẫn khá ít cho người mới bắt đầu.
* Đồi hỏi kiến thức javascript.

**2.2.4. Giới thiệu về Mongodb.**

**2.2.4.1: Giới thiệu:**

MongoDB là một chương trình cơ sở dữ liệu mã nguồn mở được thiết kế theo kiểu hướng đối tượng trong đó các bảng được cấu trúc một cách linh hoạt cho phép các dữ liệu lưu trên bảng không cần phải tuân theo một dạng cấu trúc nhất định nào. Chính do cấu trúc linh hoạt này nên MongoDB có thể được dùng để lưu trữ các dữ liệu có cấu trúc phức tạp và đa dạng và không cố định (hay còn gọi là Big Data).

**2.2.4.2. Đặt điểm của Mongodb**

* Kho lưu định hướng Document: Dữ liệu được lưu trong các tài liệu kiểu JSON.
* Lập chỉ mục trên bất kỳ thuộc tính nào.
* Các truy vấn đa dạng.
* Cập nhật nhanh hơn.

**2.2.4.3 Ưu điểm của MongodDB.**

* Ít Schema hơn: MongoDB là một cơ sở dữ liệu dựa trên Document, trong đó một Collection giữ các Document khác nhau. Số trường, nội dung và kích cỡ của Document này có thể khác với Document khác.
* Cấu trúc của một đối tượng là rõ ràng.
* Không có các Join phức tạp.
* MongoDB dễ dàng để mở rộng.
* Sử dụng bộ nhớ nội tại để lưu giữ phần công việc, giúp truy cập dữ liệu nhanh hơn.

**Chương 3: Cài đặt và thử nghiệm**

**3.1 Hướng dẫn cài đặt.**

**3.1.1 Cài đặt môi trường.**

**3.1.1.1 NodeJS.**

**a. Window**

Đối với windows, bạn chỉ cần vào trang chủ Nodejs.org để download và cài đặt Nodejs. Để kiểm tra đã cài đặt được nodejs hay chưa bạn cần mở chương trình "Node.js Command Prompt" lên bằng cách vào Start gõ search từ "prompt" rồi gõ node -v.

Nếu xuất ra version của Nodejs tức là bạn đã cài đặt thành công. Tiếp theo là kiểm tra NPM - Công cụ quản lý package của NodeJS

npm -v

**b.Linux**

* Đầu tiên, bạn nên update tất cả package của hệ điều hành để đảm bảo việc cài đặt NodeJS không gặp vấn đề.

sudo apt-get update

* Cài Nodejs.

sudo apt-get install nodejs

* Cài đặt NPM.

sudo apt-get install npm

* Để kiểm tra NPM và NodeJS đã cài đặt được chưa.

nodejs -v

npm -v

**3.1.1.2 Loopback**

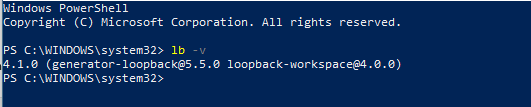
Nếu như chưa cài nodeJS và NPM thì vui lòng xem cách cài đặt phía trên.

Mở màn hình Command và chạy câu lệnh

npm install -g loopback-cli

Kiểm tra version để biết cài đặt thành công hay chưa bằng câu lệnh:

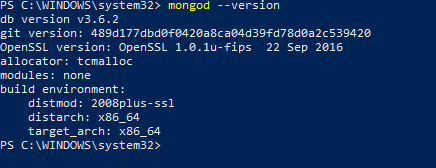
Lb -v



Hình. 27. Hình kiểm tra version loopback

**3.1.1.3 MongoDB**

* Down load mongodb <https://www.mongodb.com/download-center/community>
* Download thành công cài đặt với các giá trị mặc định.
* Sau khi cài đặt xong kiểm tra cài đặt thành công chưa.



Hình. 28. Hình ảnh kiểm tra version Database mongo

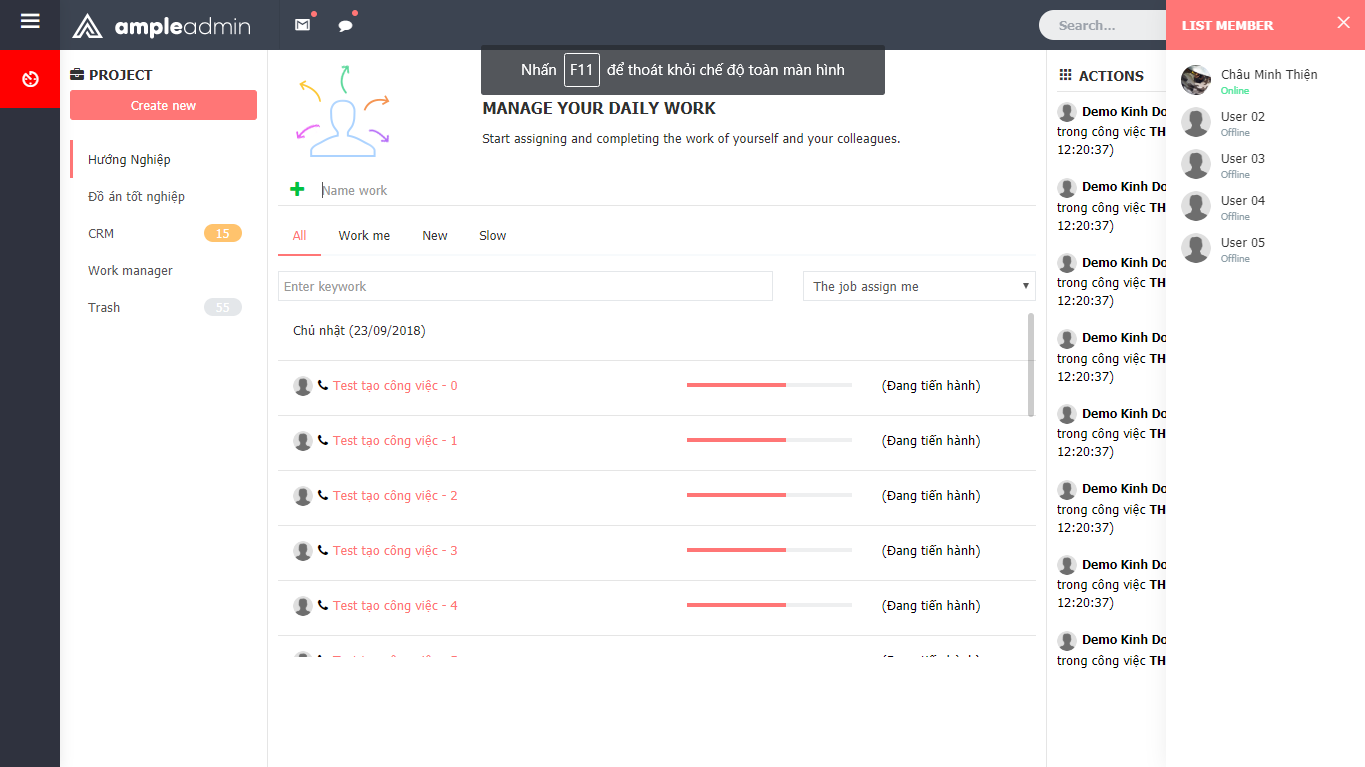
**3.2 Giao diện hệ thống.**

**3.2.1. Giao diện phía người dùng.**

**3.2.1.1 Trang chủ.**

Ở trang chủ người dùng sẽ thấy tổng quan hệ thống bao gồm.

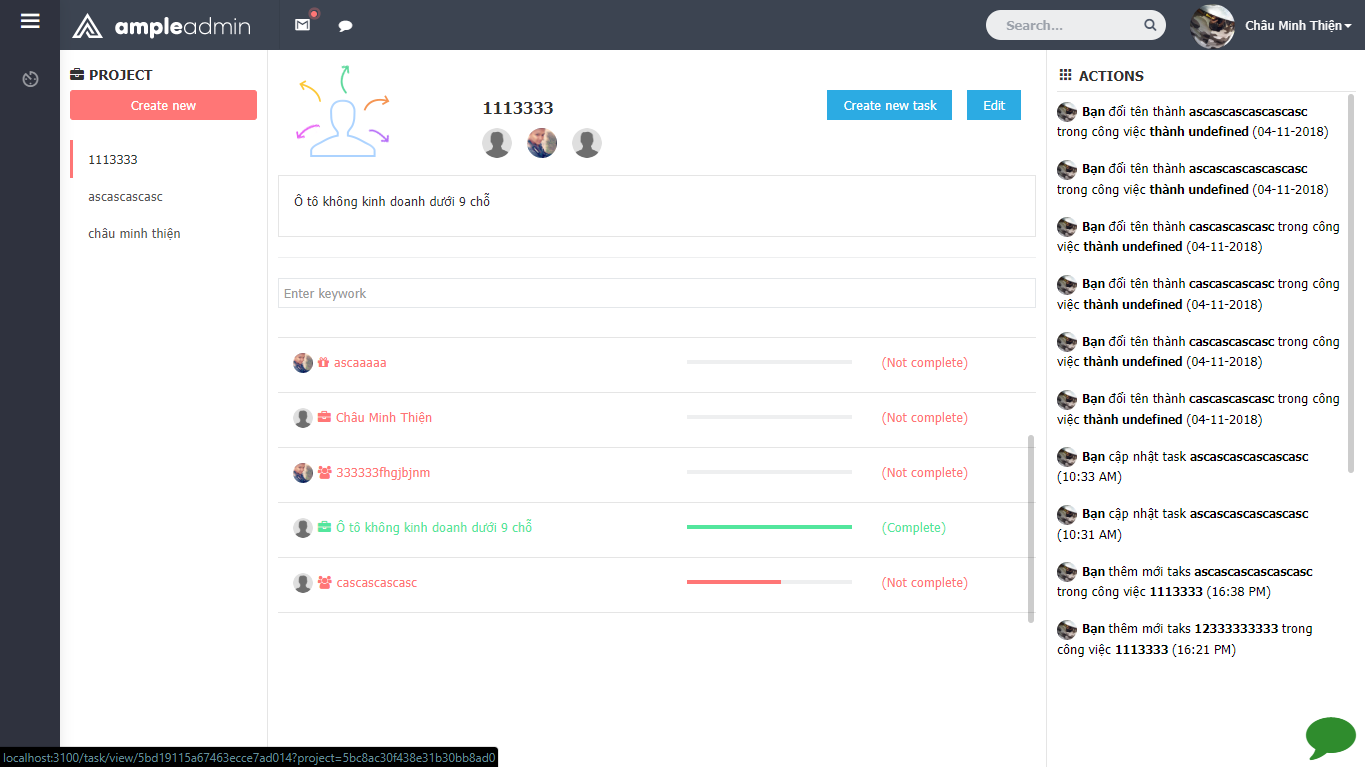
* Danh sách dự án.
* Lịch sử hoạt động của mình.
* Danh sách các công việc của mình làm.
* Danh sách bạn bè trong nhóm.



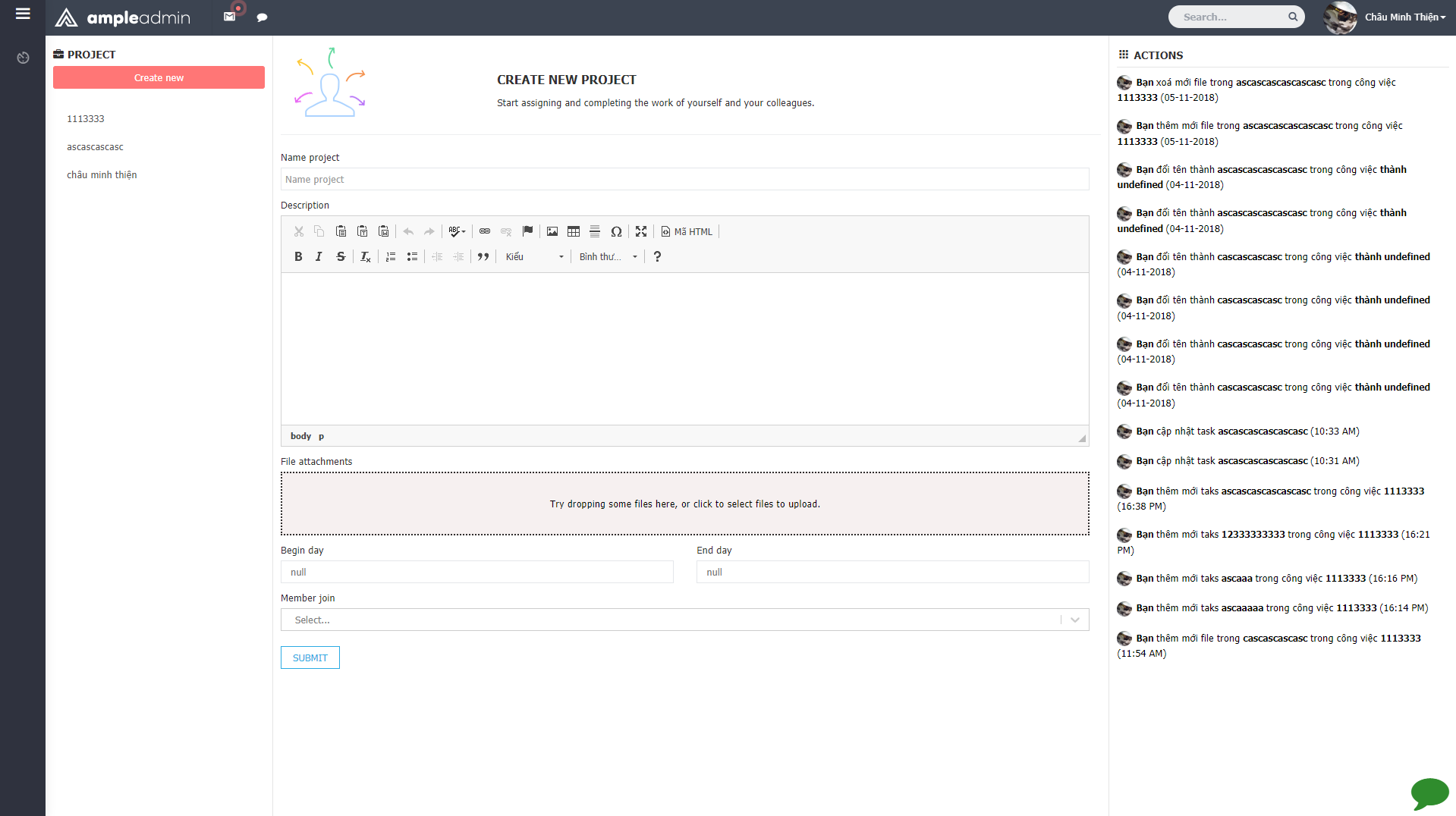
Hình. 29. Giao diện trang chủ hệ thống

**3.2.1.2 Giao diện dự án.**

Ở giao diện này sẽ được hiển thị thông tin đầy đủ nội dung của dự án đang làm. Phía dưới là danh sách cách công việc và trạng thái của công việc.

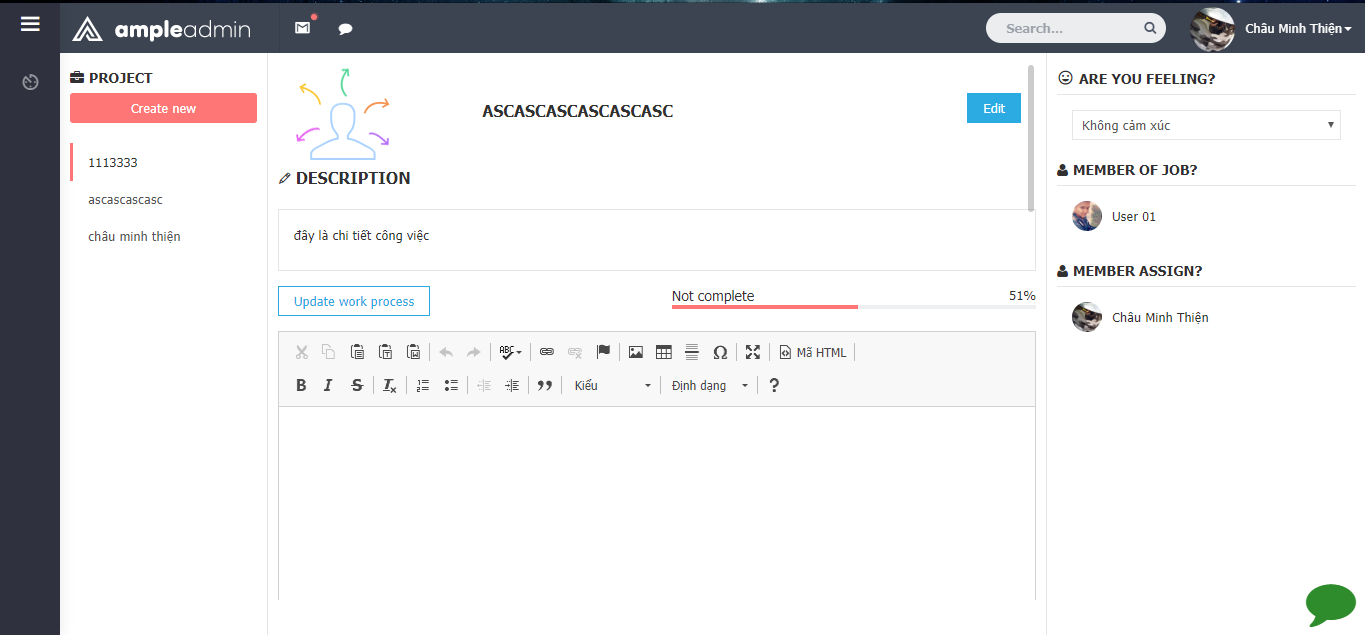


Hình. 30. Giao diện xem thông tin dự án

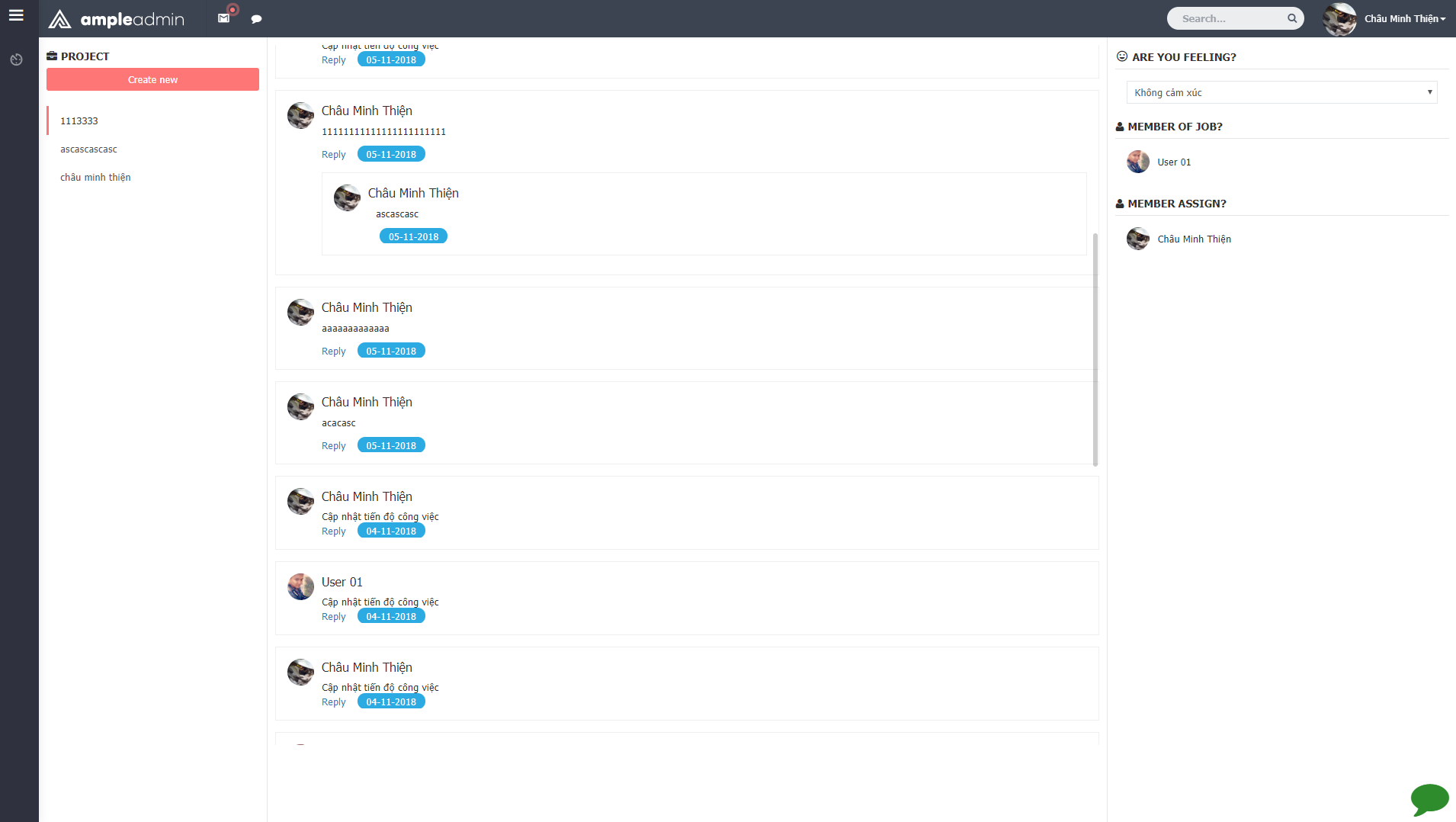


Hình. 31. Giao diện thêm mới một dự án

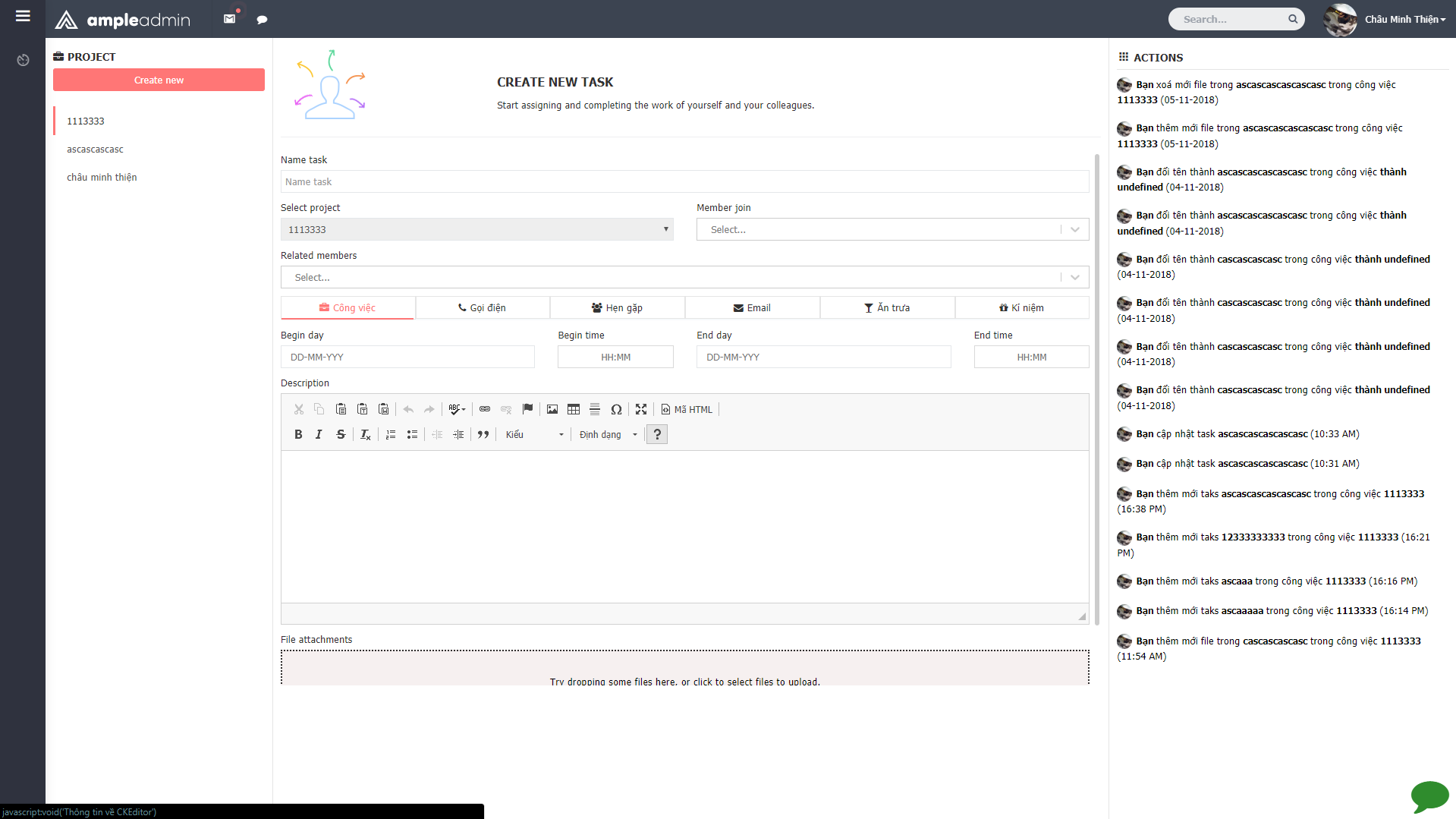
#### **3.2.1.3 Giao diện màn hình công việc.**



Hình. 32. Giao diện màn hình công việc



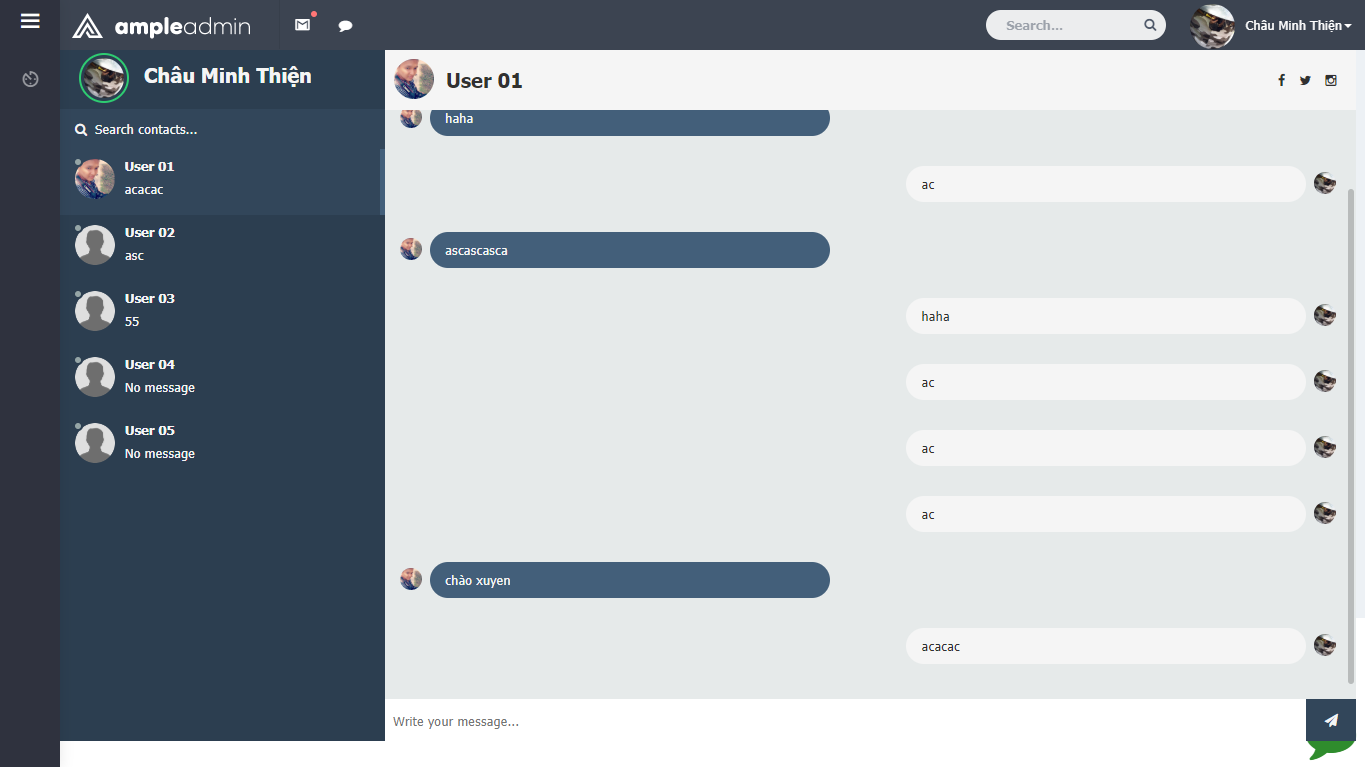
Hình. 33. Giao diện danh sách bình luận trao đổi công việc



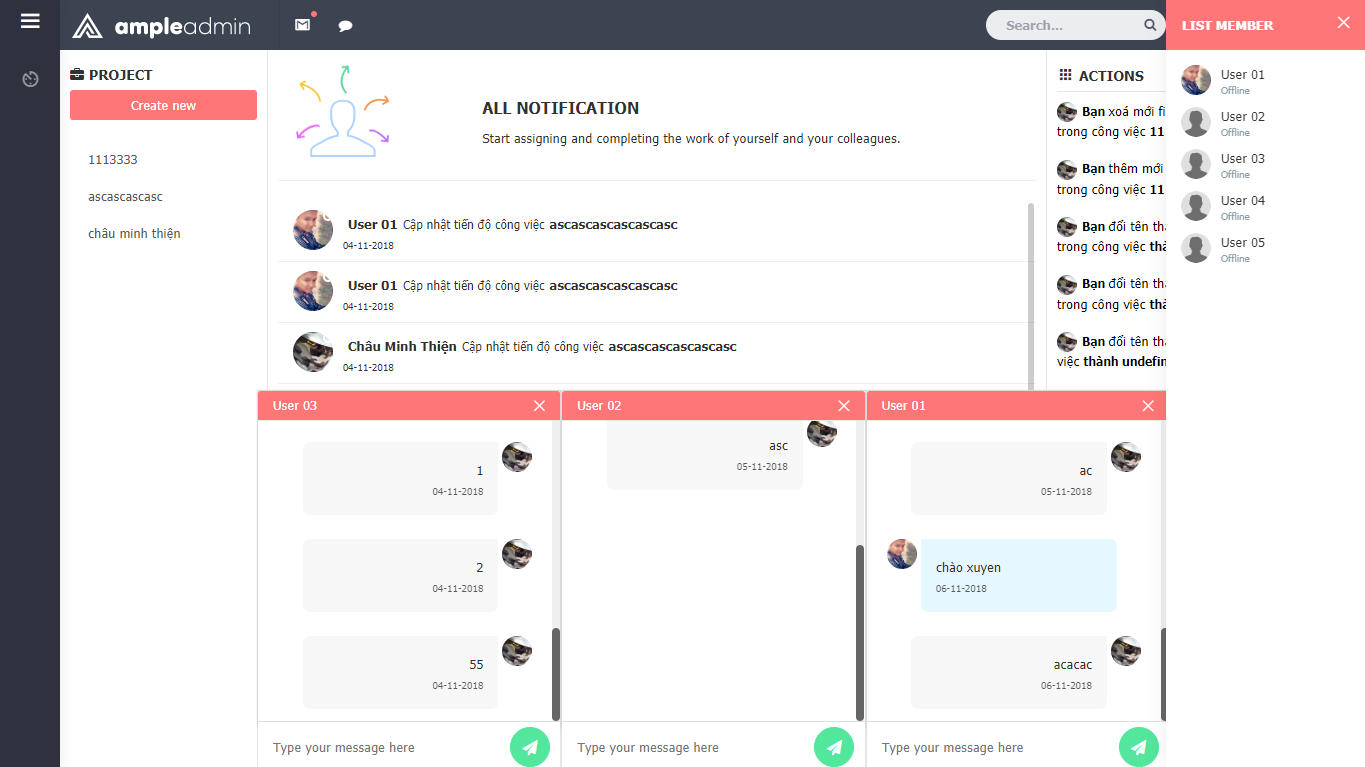
Hình. 34. Giao diện thêm mới công việc

#### **3.2.1.4 Giao diện trò chuyện.**

Ở giao diện này sẽ có 2 phần. đó là giao diện màn hình trò chuyện với nhiều người và một giao diện hiển thị cuộc trò chuyện trực quan hơn.

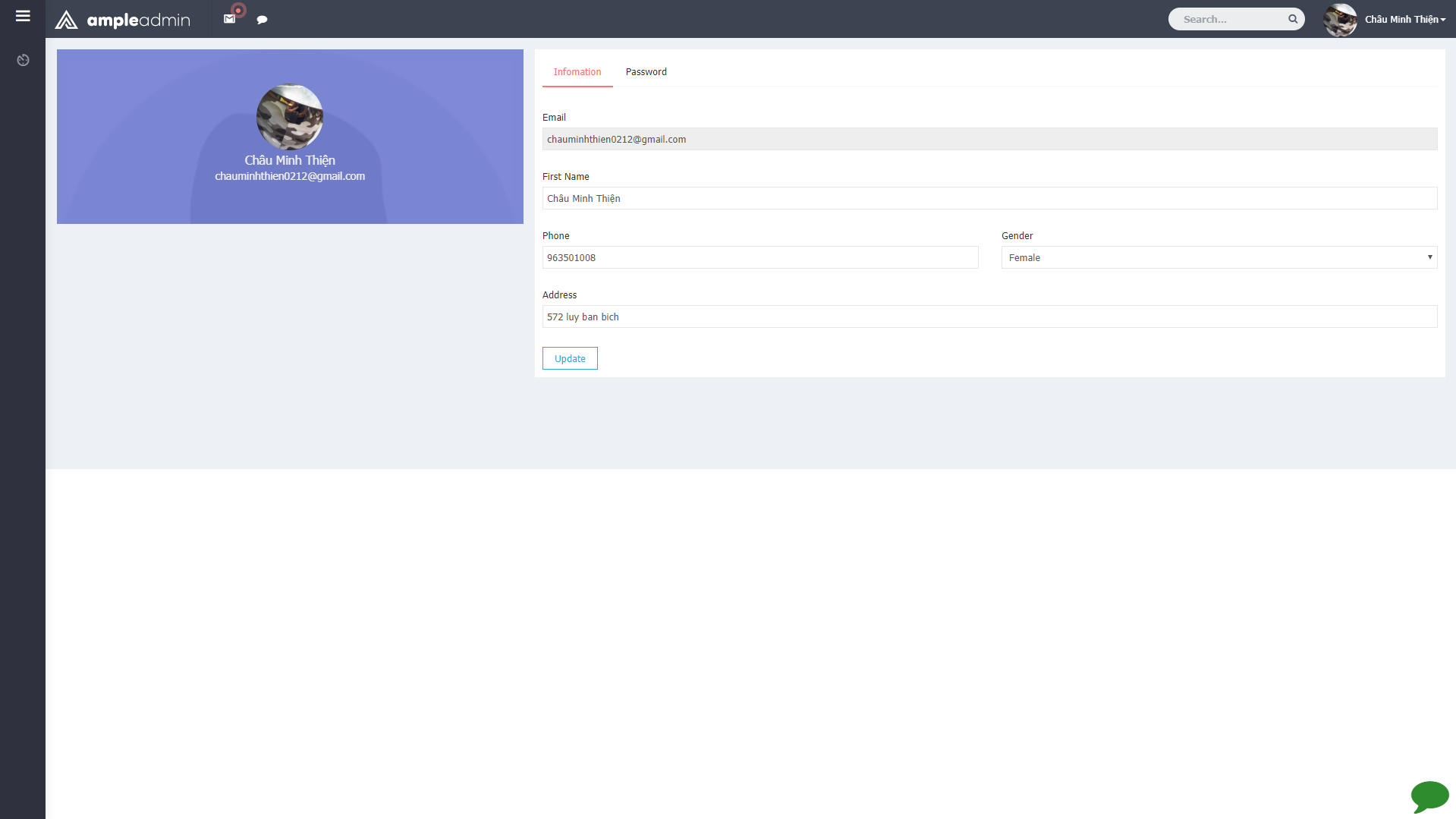


Hình. 35. Giao diện trò chuyện dạng danh sách

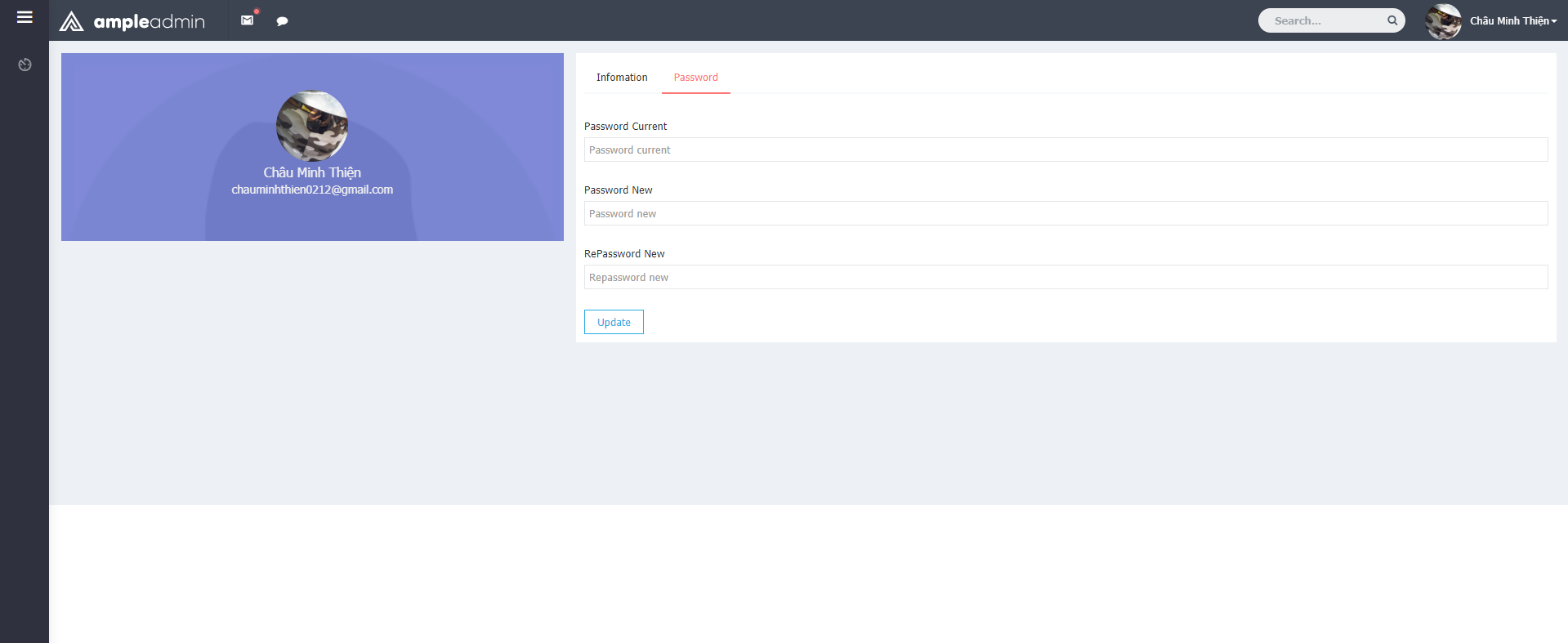


Hình. 36. Giao diện trò chuyện dạng chatbox

#### **3.2.1.5. Giao diện thông tin cá nhân.**



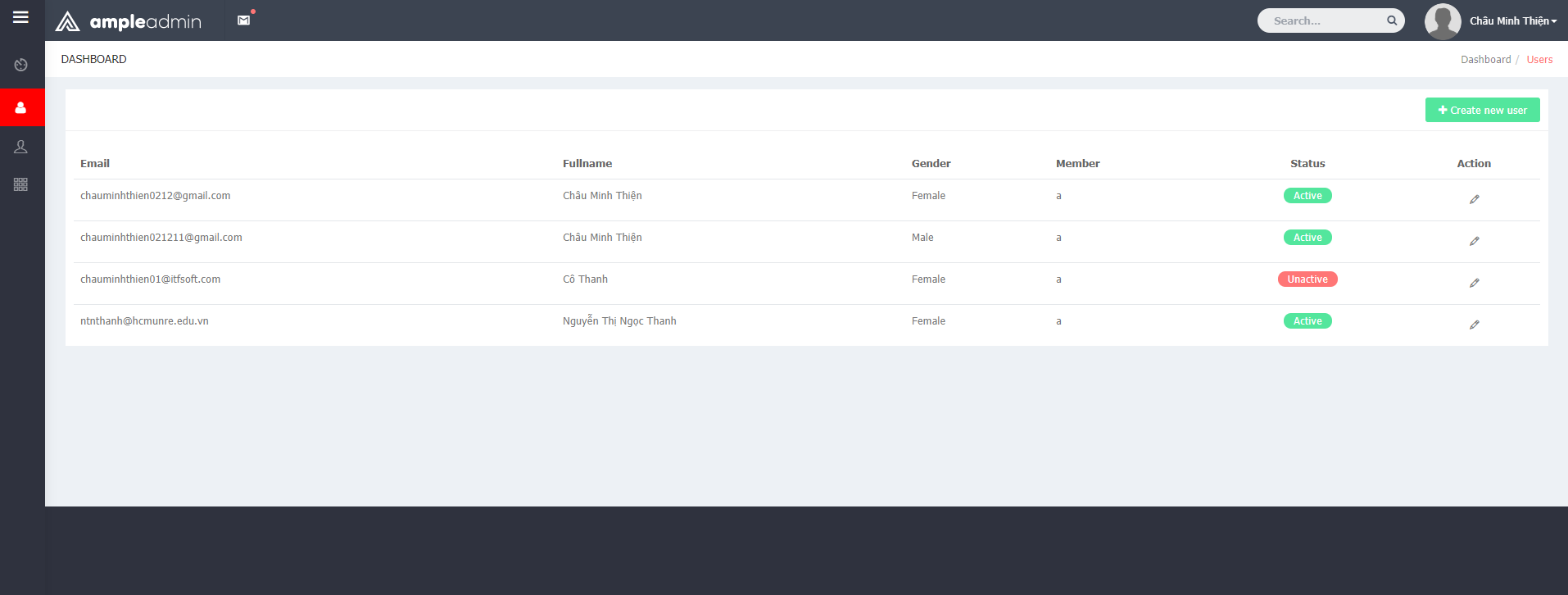
Hình. 37. Giao diện quản lý thông tin cá nhân



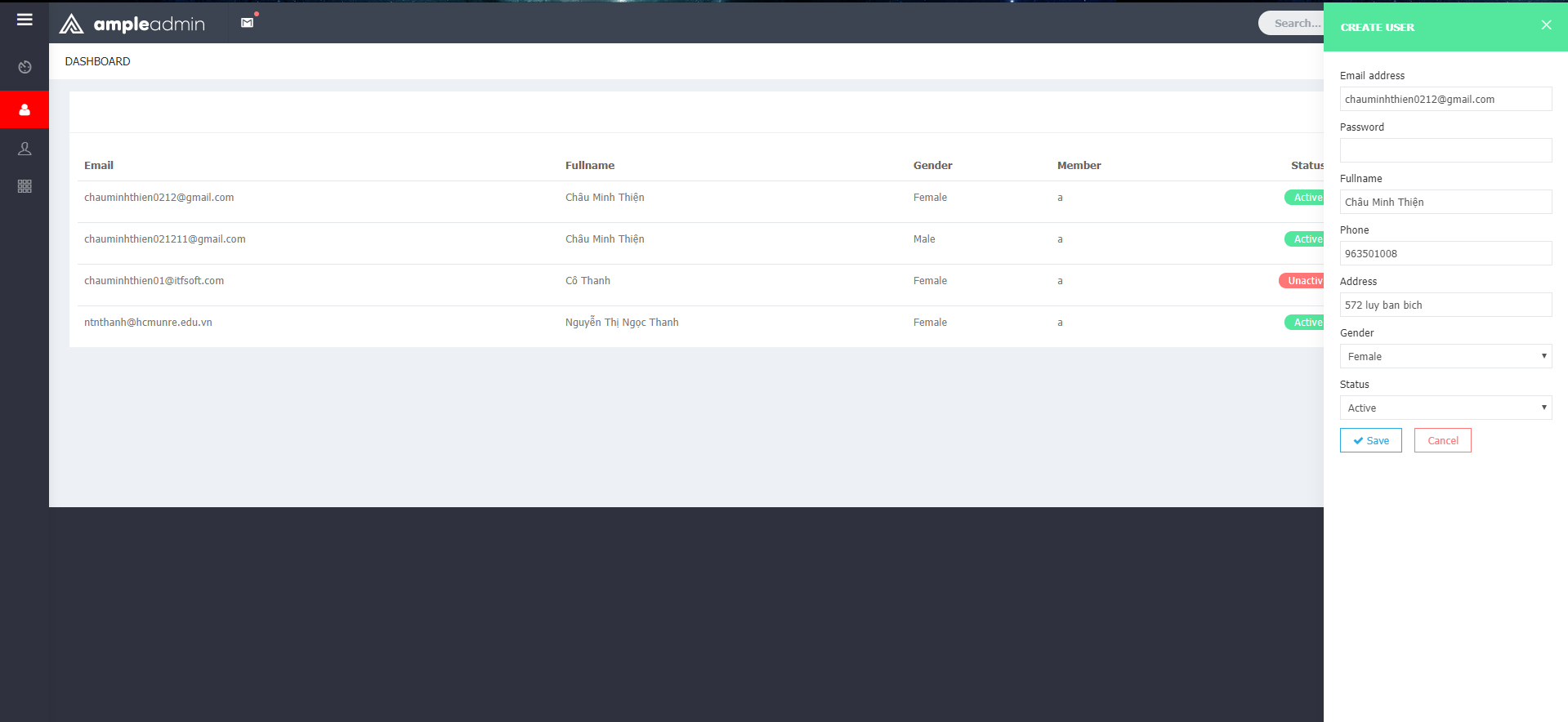
Hình. 38. Giao diện thay đổi mật khẩu

### **3.2.2 Giao diện người quản trị hệ thống.**

#### **3.2.2.1 Giao diện quản lý tài khoản**

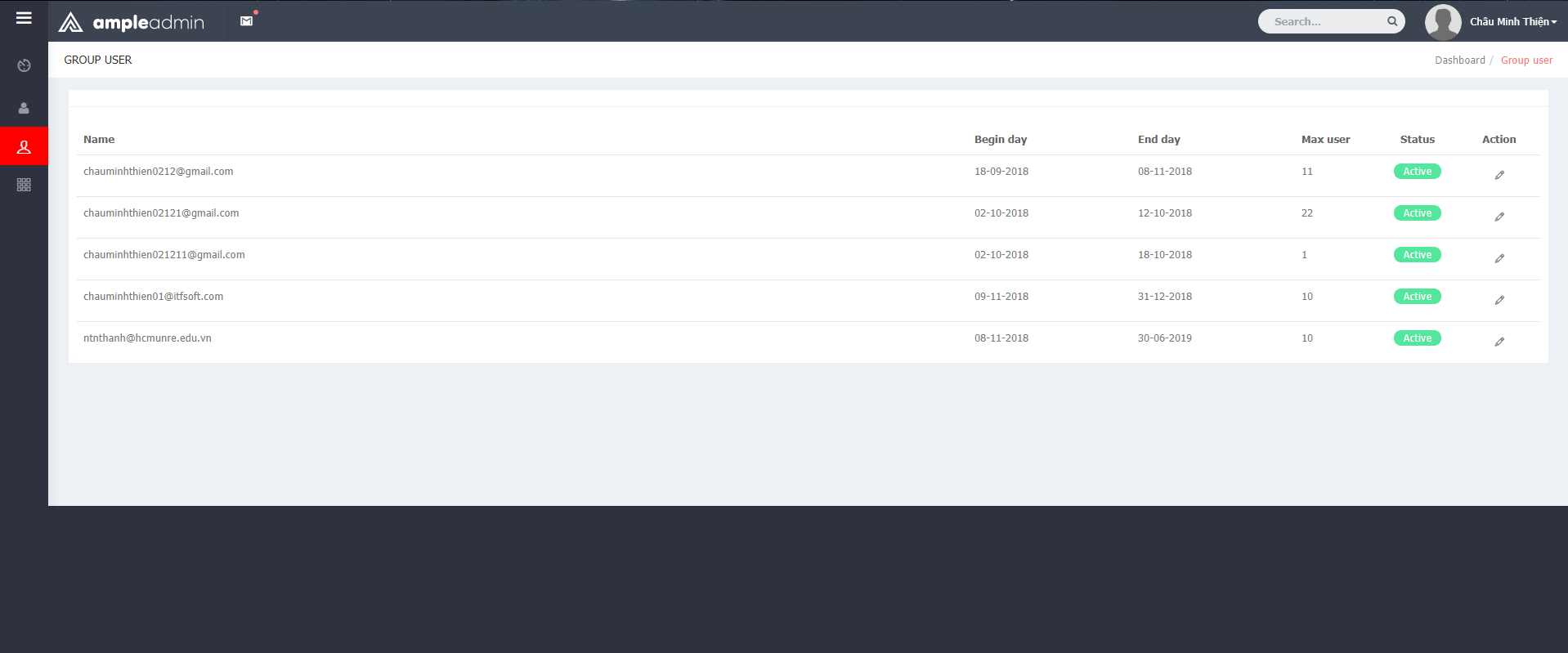


Hình. 39. Giao diện quản lý tài khoản



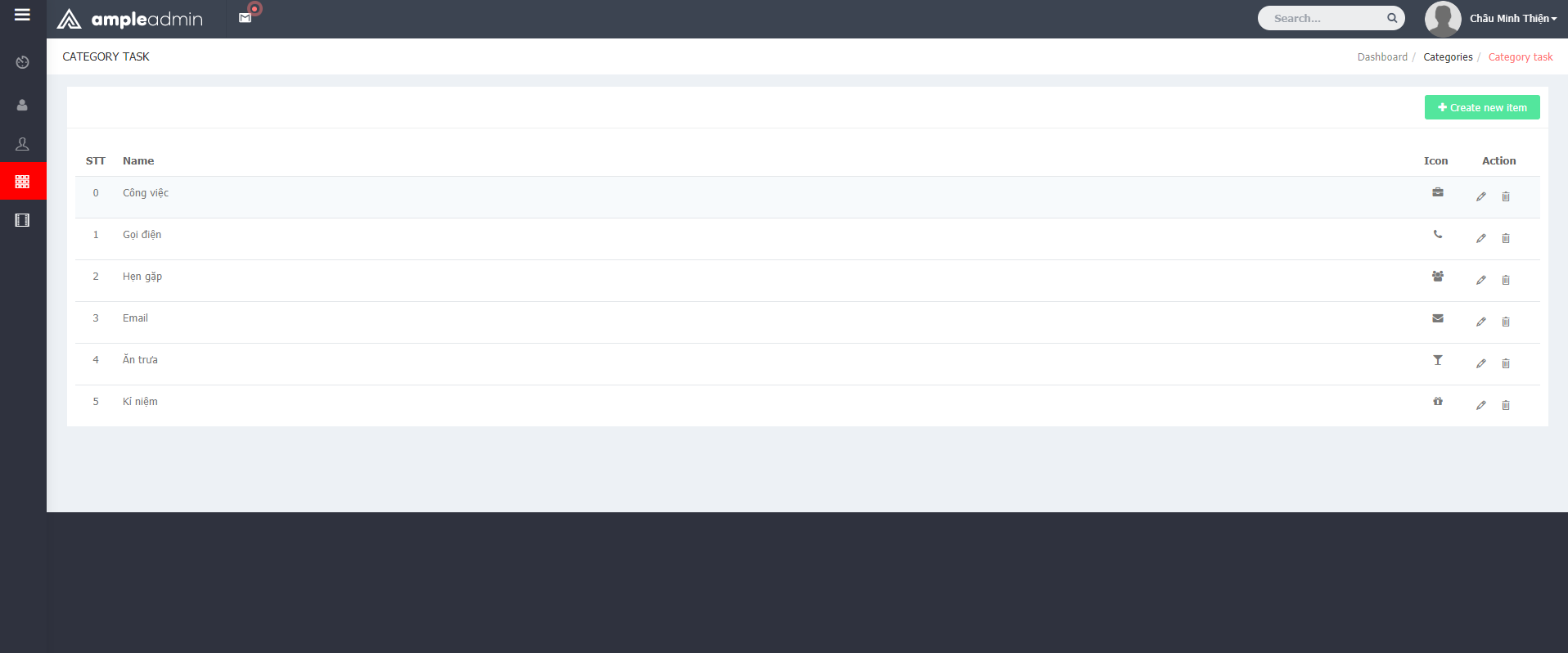
Hình. 40. Giao diện thêm mới tài khoản

#### **3.2.2.2 Giao diện quản lý nhóm tài khoản**

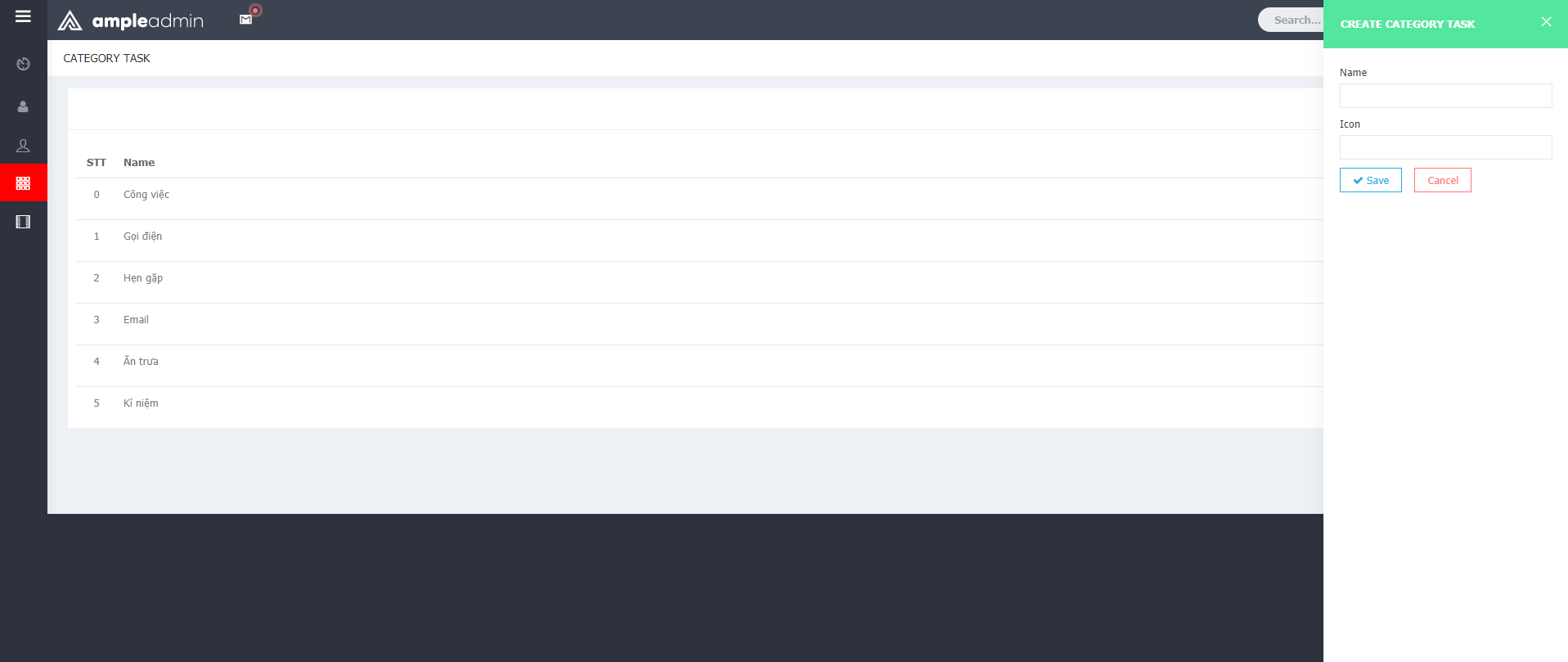


Hình. 41. Giao diện quản lý nhóm tài khoản

#### **3.2.2.3 Giao diện quản lý danh mục**



Hình. 42. Giao diện danh sách danh mục



Hình. 43. Giao diện thêm mới danh mục

**Chương 4: Kết luận và hướng phát triển**

**4.1 Kết quả làm được.**

* Xây dựng thành công hệ thống quản lý công việc.
* Xử lý thành công thời gian thực trên ứng dụng.
* Giải quyết được tình trạng tự động thông báo khi sắp hết thời gian.
* Xây dưng thành công hệ thống chạy trên ứng dụng máy tính.

**4.2 Phần chưa làm được**

* Giao diện còn đơn giản.
* Hệ thống bảo mật tương đối.
* Tốc độ xử lý chưa tốt.
* Chưa xây dựng được hệ thống chạy trên mobile.
* Hệ thống mới xây dựng nên chưa được quảng bá rộng rãi cho người sử dụng.

**4.3 Kinh nghiệm làm được.**

* Biết thêm về cách làm thề nào để xây dựng một ứng dựng theo mô hình Front-end, Backend và cách cấu hình server để chạy cách ứng dụng xử lý thời gian thực.
* Những điều cần lưu ý khi xây dựng một ứng dụng web, củng cố một số kiến thức về HTML, CSS và học được nhiều điều mới từ Javascript đặc biệt là NodeJs và ReactJs
* Biết sử dụng thêm một hình thức database theo dạng no-sql đó là MongoDB.
* Định hướng được con đường trở thành một lập trình Web javascript full stask.

**4.4 Hướng phát triển.**

* Xử lý bảo mật mức tốt nhất.
* Xây dựng theo app trên mobile.
* Cải thiện tốc độ xử lý cho hệ thống.
* Cải thiện giao diện cho hài hòa và bắt mắt hơn.
* Trong lương lai gần sẽ hoàn thiện các tính năng hướng dẫn một cách chi tiết cho người dùng mới, cải thiện hiệu năng và đồng bộ khi tải dữ liệu lên trang.

**4.4 Kết luận.**

Qua bài luận văn lần này em được củng cố lại những kiến thức được học ở trường và biết áp dụng vào một dự án thực tế. Em cảm thấy yêu thích công việc của em hiện tại đó là một web developer, em sẽ cố gắng trau dồi thêm những kiến thức mới để có thể xây dựng nhiều hệ thống giúp cho đất nước càng phát triển thêm.

**DANH MỤC TÀI LIỆU THAM KHẢO**

**Facebook**. - https://reactjs.org/docs/hello-world.html.

**GIT**. - https://git-scm.com/doc.

**Loopback**. - https://loopback.io.

**NodeJs** - https://nodejs.org/en/docs/.

**W3schools**. - https://www.w3schools.com/js/.

**Tp.HCM, ngày tháng năm 2018**

**Trưởng Khoa**