
<Connecting>

**<Web kết nối gia sư - học sinh>
Software Architecture Document**

Version <1.0>

<Web kết nối gia sư - học sinh>	Version: <1.0>
Software Architecture Document	Date: <27/11/2021>
<Connecting>	

Revision History

Date	Version	Description	Author
21/11/2021	1.0	SAD document	Connecting

<Web kết nối gia sư - học sinh>	Version: <1.0>
Software Architecture Document	Date: <27/11/2021>
<Connecting>	

Table of Contents

1	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, acronyms, abbreviations	4
1.4	References	4
2	Architectural Goals and Constraints	4
2.1.	Technical platform	4
2.2	Security	4
2.3	Performance	5
2.4	Persistence	5
2.5	Reliability/Availability	5
3	Use-Case Model	5
4	Logical View	5
4.1	GUI	6
4.1.1	Authentication Component	6
4.1.2	Request component	7
4.1.3	Post component	7
4.1.4	User component	7
4.1.5	System Management component	8
4.2	Controller	9
4.2.1	Authentication component	9
4.2.2	Request component	9
4.2.3	Post component	9
4.2.4	User component	10
4.2.5	System Management component	10
4.3	Model	11
5	Deployment	12
6	Implementation View	12

<Web kết nối gia sư - học sinh>	Version: <1.0>
Software Architecture Document	Date: <27/11/2021>
<Connecting>	

Software Architecture Document

1. Introduction

Web kết nối gia sư là một trang web hướng tới mục tiêu kết nối gia sư và những học sinh phụ huynh có nhu cầu. Được phát triển dựa trên kiến trúc phân tầng dựa trên mô hình MVC với bộ ba NodeJs, ExpressJs và MongoDB và được phát triển bằng ngôn ngữ Javascript. Trang web sẽ là một nền tảng đáp ứng tốt nhu cầu tìm và lựa chọn gia sư của các quý phụ huynh cũng như tìm công việc cho các gia sư một cách thuận tiện nhất và đáp ứng được các nhu cầu, tiêu chí cụ thể của từng trường hợp.

1.1 Purpose

Tài liệu kiến trúc phần mềm này cung cấp một cái nhìn tổng quan toàn diện về kiến trúc của web Kết nối gia sư - học sinh. Nó nhằm nắm bắt và truyền đạt các quyết định kiến trúc sẽ được thực hiện trên hệ thống.

1.2 Scope

Phạm vi của SAD này là mô tả kiến trúc cung cấp cho web “Web kết nối gia sư”

1.3 Definitions, acronyms, abbreviations

SAD: Software Architecture Document

MVC: model view controller

CRUD: Create Read Update Delete

1.4 References

Web: VIBLO. Tên bài viết: Phân tích thiết kế hệ thống thông tin sử dụng biểu đồ UML (Phần 1)

link: <https://viblo.asia/p/phan-tich-thiet-ke-he-thong-thong-tin-su-dung-bieu-do-uml-phan-1-PjxMe6yNG4YL>

Web: sparxsystem.com. Tên bài viết: Package diagram | Enterprise architect user guide

link: https://sparxsystems.com/enterprise_architect_user_guide/

Tên video: PACKAGE DIAGRAM PRACTICE IN STARUML

link: https://www.youtube.com/watch?v=_q5GF-Oq6xc

web: MONA MEDIA. Tên bài viết: Mô hình MVC là gì? Các thành phần trong MVC

link: <https://monamedia.co/mvc-la-gi-ung-dung-cua-mo-hinh-mvc-trong-lap-trinh/>

2. Architectural Goals and Constraints

2.1. Technical platform

Ứng dụng kết nối Gia sư và Học sinh sẽ được triển khai trên server của Heroku

2.2 Security

- Hệ thống phải được bảo mật để bảo mật thông tin của người dùng
- Ứng dụng phải được triển khai một vài bảo mật cơ bản:
 - + Xác thực (Authentication): Đăng nhập sử dụng username và password
 - + Phân quyền (Authorization): Dựa vào hồ sơ người dùng thì sẽ phân quyền thành Gia sư hoặc Học sinh
 - + Toàn vẹn dữ liệu (Data integrity): Dữ liệu được gửi qua mạng không thể bị sửa đổi bởi một cấp

<Web kết nối gia sư - học sinh>	Version: <1.0>
Software Architecture Document	Date: <27/11/2021>
<Connecting>	

2.3 Performance

Giao diện bố cục rõ ràng, thân thiện với người dùng. Tương thích với đa dạng thiết bị như pc, điện thoại, ipad

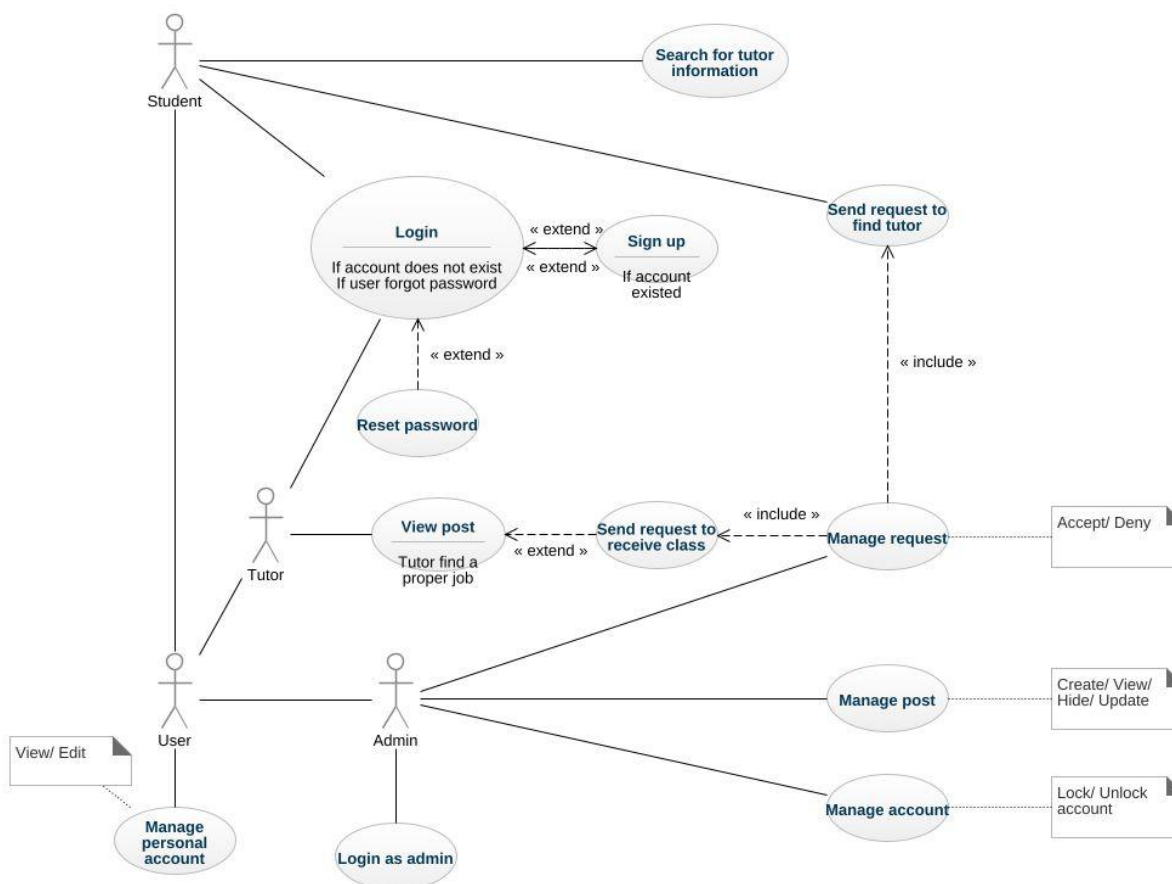
2.4 Persistence

Sự toàn vẹn của dữ liệu sẽ được giải quyết bằng cách sử dụng cơ sở dữ liệu quan hệ.

2.5 Reliability/Availability

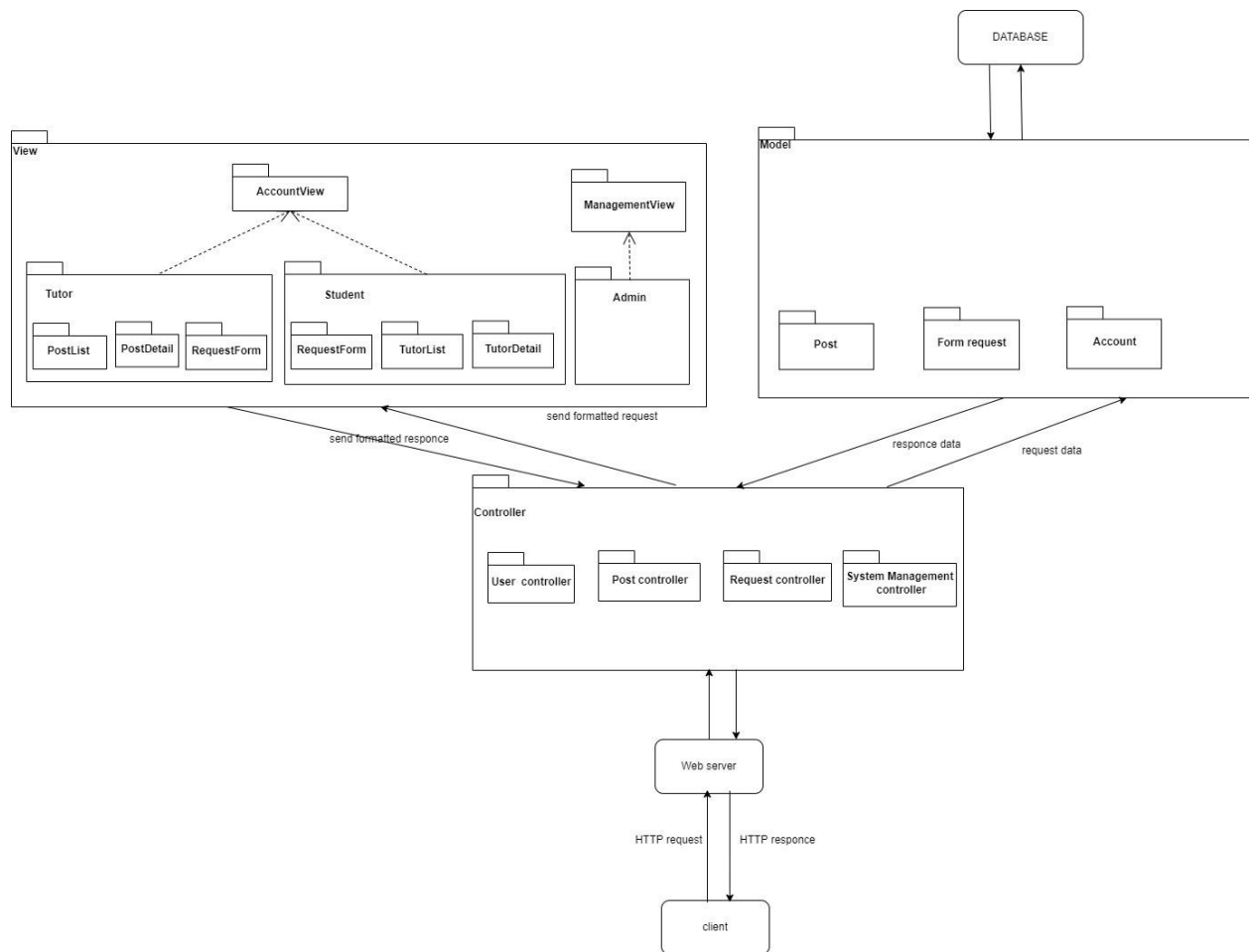
- Hệ thống cần đáp ứng được tính sẵn có. Kiến trúc phải được đảm bảo có thể chuyển đổi dự phòng. Đảm bảo được độ tin cậy cao
- Thời gian tối đa cho bất kì một hoạt động bảo trì nào là 8 tiếng.
- Hệ thống có thể hoạt động tốt 24/7, 24h trên 1 ngày và 7 ngày trên 1 tuần.

3. Use-Case Model



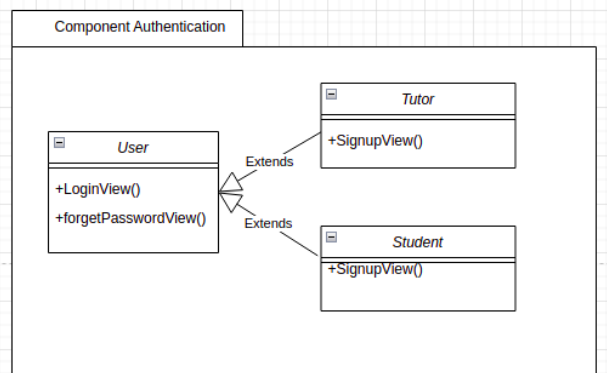
4. Logical View

<Web kết nối gia sư - học sinh>	Version: <1.0>
Software Architecture Document	Date: <27/11/2021>
<Connecting>	



4.1 GUI

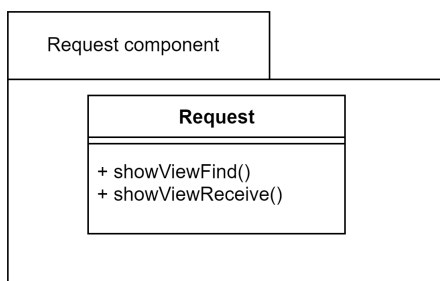
4.1.1 Authentication Component



<Web kết nối gia sư - học sinh>	Version: <1.0>
Software Architecture Document	Date: <27/11/2021>
<Connecting>	

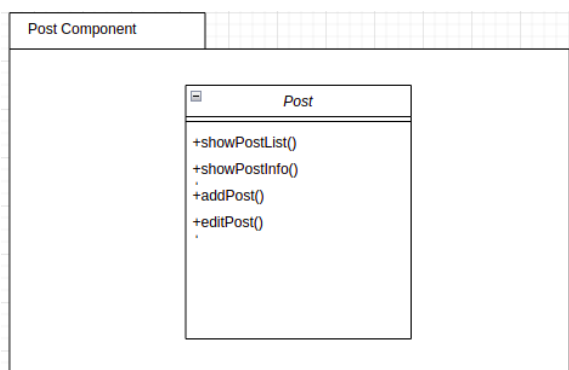
- Giao diện login: Đăng nhập vào hệ thống, người dùng với vai trò là student/ tutor sẽ đăng nhập với giao diện này
- Giao diện signup dành cho Student: Giao diện đăng kí nếu người dùng chọn đăng ký với vai trò là student
- Giao diện signup dành cho Tutor: Giao diện đăng kí nếu người dùng chọn đăng ký với vai trò là tutor
- Giao diện forgot password: Giao diện được sử dụng khi người dùng bấm vào nút quên mật khẩu

4.1.2 Request component



- Hiển thị form điền thông tin tìm kiếm gia sư
- Hiển thị giao diện form điền thông tin để yêu cầu nhận lớp

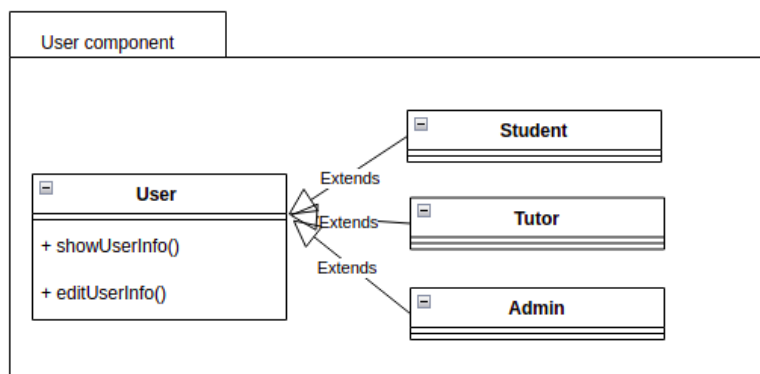
4.1.3 Post component



- Giao diện Post list: Hiển thị danh sách các bài đăng
- Giao diện Post info: Hiển thị thông tin cho một bài đăng
- Giao diện thêm bài đăng: Admin điền đầy đủ form để tạo bài đăng mới
- Giao diện chỉnh sửa bài đăng: Admin được quyền chỉnh sửa các thông tin trong một bài đăng

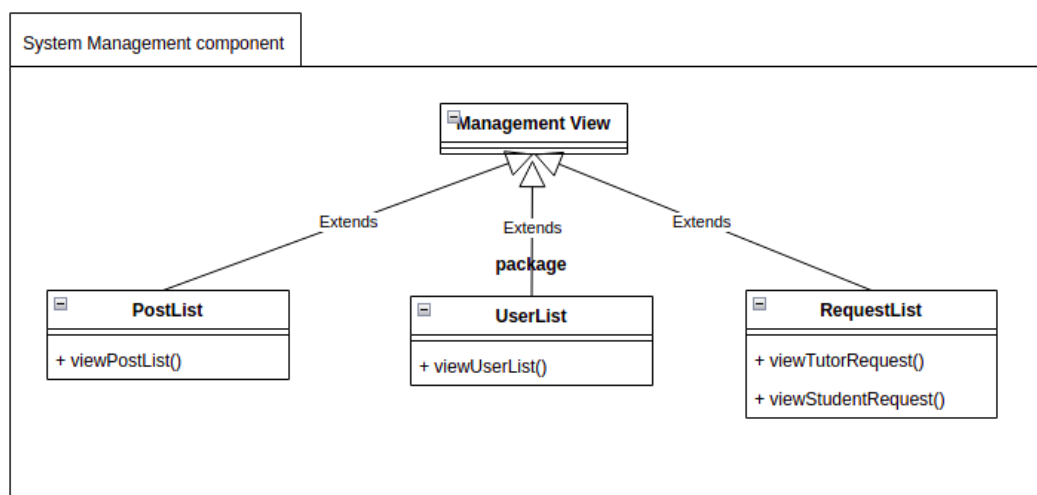
4.1.4 User component

<Web kết nối gia sư - học sinh>	Version: <1.0>
Software Architecture Document	Date: <27/11/2021>
<Connecting>	



- Giao diện hiển thị thông tin người dùng cho cả student, tutor và admin
- Giao diện chỉnh sửa thông tin người dùng

4.1.5 System Management component

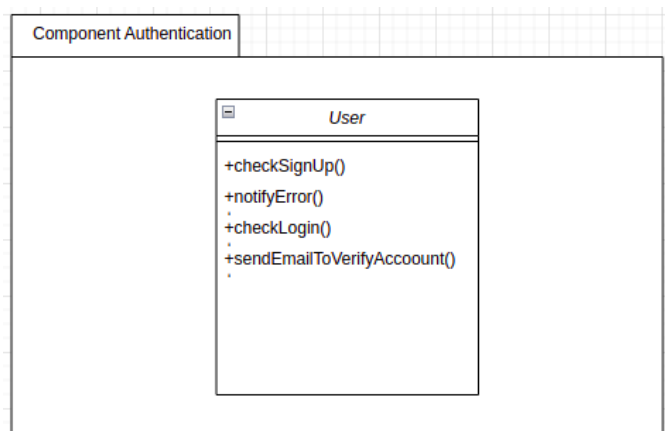


- Giao diện dành cho người quản lý hệ thống
 - + Giao diện post list: Hiển thị danh sách các bài đăng
 - + Giao diện user list: Hiển thị danh sách các người dùng trong hệ thống
 - + Giao diện request list: Hiển thị danh sách các request đến từ gia sư hoặc học sinh đang chờ duyệt

<Web kết nối gia sư - học sinh>	Version: <1.0>
Software Architecture Document	Date: <27/11/2021>
<Connecting>	

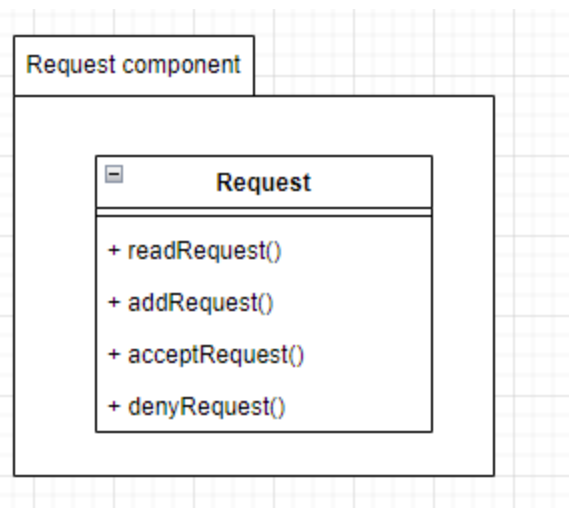
4.2 Controller

4.2.1 Authentication component



- checkSignUp(): Kiểm tra thông tin đăng ký trong form đăng ký có hợp lệ chưa, còn thiếu sót thông tin nào không, nếu chưa điền đầy đủ form sẽ có thông báo lỗi notifyError()
- checkLogin(): Kiểm tra username và password đăng nhập vào đã hợp lệ chưa, tài khoản đã có trong hệ thống hay chưa, nếu chưa có thì phải đăng ký
- sendEmailToVerifyAccount(): Nếu người dùng bấm vào nút quên mật khẩu ở screen đăng nhập, hệ thống gửi thông báo xác thực tài khoản qua mail

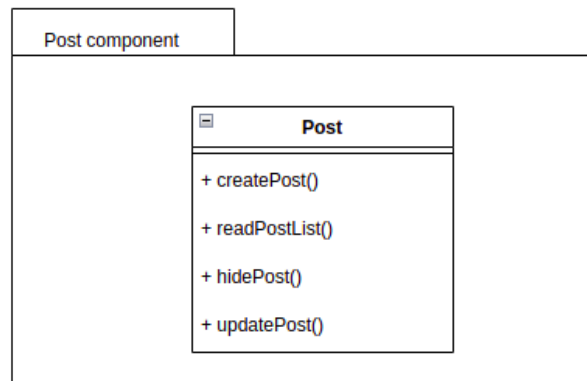
4.2.2 Request component



- readRequest() đọc các request được trả về từ database
- addRequest(): thêm một form request mới vào DataBase để chờ duyệt
- acceptRequest(): chấp nhận form yêu cầu tìm tutor của student gửi lên
- denyRequest(): từ chối form yêu cầu tìm tutor của student gửi lên

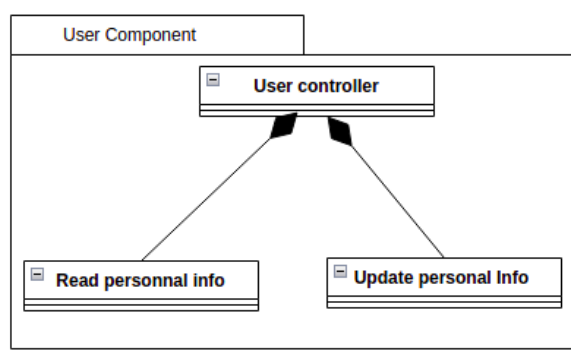
4.2.3 Post component

<Web kết nối gia sư - học sinh>	Version: <1.0>
Software Architecture Document	Date: <27/11/2021>
<Connecting>	



- Controller xử lý các thao tác tên Post như
 - + Tạo mới một bài đăng, gọi API kết nối tới database để thêm post vào post list
 - + Thay đổi trạng thái bài đăng, ẩn bài đăng
 - + Đọc danh sách các bài đăng ở database để lấy danh sách hiển thị
 - + Cập nhật lại thông tin của một bài đăng

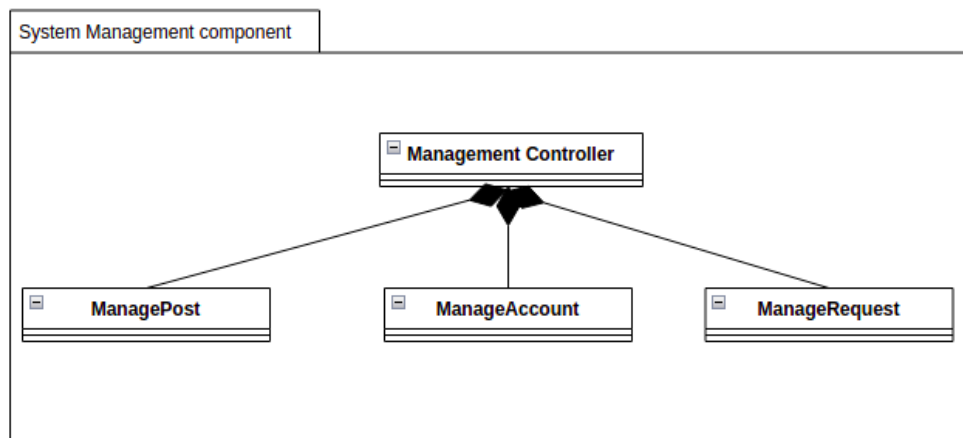
4.2.4 User component



- Controller xử lý các thao tác liên quan đến user như:
 - + Read personal info: Đọc thông tin cá nhân của các user từ database và hiển thị thông tin user
 - + Update personal info: Cập nhật các thông tin vừa được chỉnh sửa xuống database, bao gồm thông tin mật khẩu mới được thay đổi

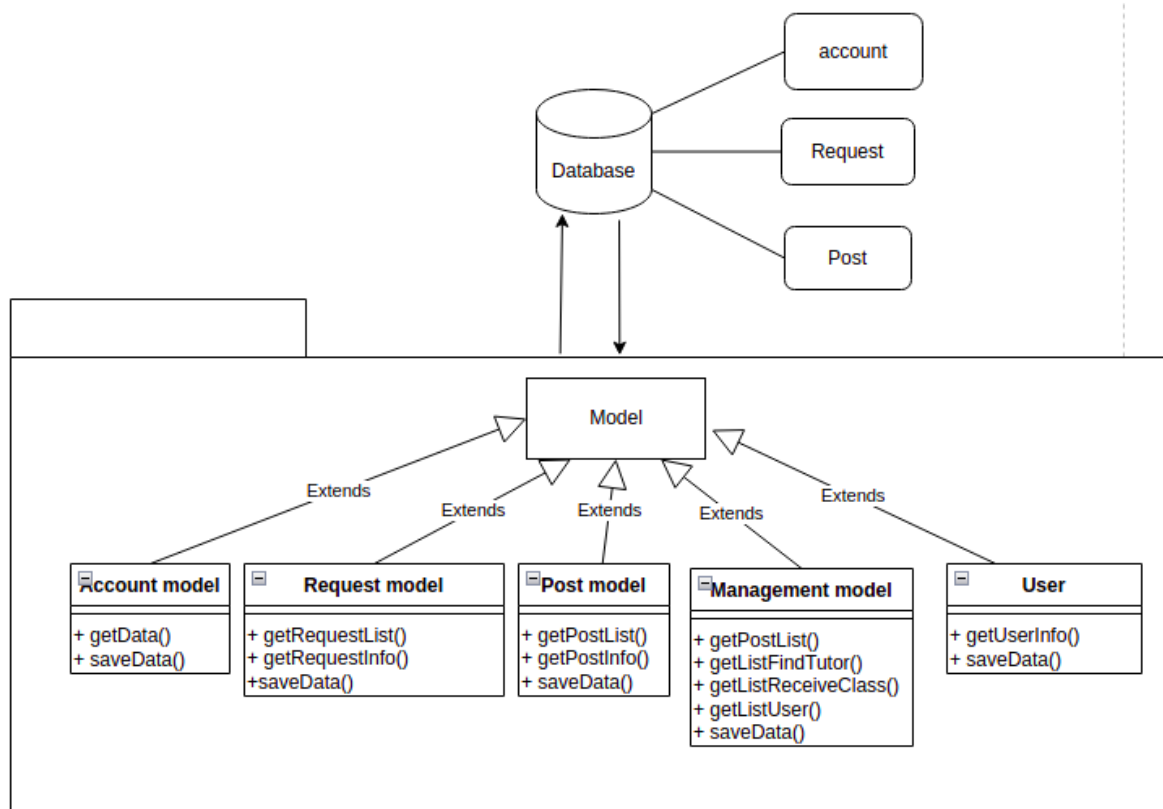
4.2.5 System Management component

<Web kết nối gia sư - học sinh>	Version: <1.0>
Software Architecture Document	Date: <27/11/2021>
<Connecting>	



- Người quản trị hệ thống quản lý :
 - + Post Management: Quản lý các bài post, đọc danh sách các bài post, xử lý các thao tác như thêm, xóa, sửa
 - + Account Management: Quản lý tài khoản người dùng, lock hoặc unlock account
 - + Request Management: Quản lý danh sách các request, accept hoặc deny request

4.3 Model



<Web kết nối gia sư - học sinh>	Version: <1.0>
Software Architecture Document	Date: <27/11/2021>
<Connecting>	

- Account model:
 - + getData(): lấy danh sách tất cả các account từ database
 - + saveData(): có nhiệm vụ thêm người dùng mới vào database
- Request model:
 - + getRequestList(): lấy danh sách tất cả các request từ database
 - + getRequestInfo(): lấy thông tin chi tiết của một request
 - + saveData(): khi có 1 request mới ta cần lưu chúng để admin có thể xem và xem xét để duyệt hay không
- Post model:
 - + getPostList(): lấy danh sách các bài đăng
 - + getPostInfo(): lấy thông tin chi tiết của một bài đăng từ database
 - + saveData(): lưu 1 bài post, lưu danh sách bài post
- Management model:
 - + getPostList(): lấy danh sách các bài request đã được duyệt
 - + getListFindTutor(): lấy danh sách các request tìm kiếm gia sư từ học sinh
 - + getListReceiveClass(): lấy danh sách các request nhận lớp từ gia sư
 - + getListUser(): lấy danh sách người dùng trong hệ thống
 - + saveData(): có nhiệm vụ lưu những thay đổi vào hệ thống
- User model:
 - + getData(): lấy dữ liệu của người dùng như username, mật khẩu, sđt, giới tính, loại phân quyền..
 - + saveData(): lưu dữ liệu của một người dùng và update nó nếu nó được thay đổi vào database

5. Deployment

None (PA4)

6. Implementation View

None (PA4)