
HiFive

**Friendly Classroom
Software Architecture Document**

Version 2.0

Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

Revision History

Date	Version	Description	Author
16/11/2021	1.0	Kiến trúc phần mềm	Team
01/12/2021	2.0	Hoàn thành SAD	Team

Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

MỤC LỤC

1.	INTRODUCTION	4
1.1	PURPOSE	4
1.2	SCOPE	4
1.3	DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	4
1.4	REFERENCES	4
1.5	OVERVIEW	4
2.	ARCHITECTURAL GOALS AND CONSTRAINTS	4
3.	USE-CASE MODEL	6
4.	LOGICAL VIEW	7
4.1	STRUCTURAL DIAGRAM	7
	4.1.1 View Component	7
	4.1.2 Model Component	8
	4.1.3 Controller Component	9
4.2	PACKAGE DIAGRAM.....	14
5.	DEPLOYMENT	14
6.	IMPLEMENTATION VIEW	16
6.1	FRONT END	16
6.2	BACK END.....	18

Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

Software Architecture Document

1. Introduction

1.1 Purpose

Tài liệu này thể hiện một cái nhìn toàn diện và bao quát về kiến trúc của hệ thống, sử dụng hai quan điểm kiến trúc khác nhau để mô tả các khía cạnh của hệ thống.

Tài liệu giúp truyền đạt các quyết định quan trọng liên quan đến kiến trúc đã được thực hiện đối với dự án.

1.2 Scope

Tài liệu này cung cấp một cái nhìn tổng quan về kiến trúc của hệ thống học tập trực tuyến Friendly Classroom. Hệ thống học tập trực tuyến Friendly Classroom được phát triển bởi nhóm HiFive thuộc lớp Nhập môn Công nghệ phần mềm 19CNTN. Người đọc sẽ biết được những technology, framework và tool được sử dụng thiết kế hệ thống.

1.3 Definitions, Acronyms and Abbreviations

1.4 References

Các tài liệu được tham khảo bao gồm:

- https://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm?fbclid=IwAR2M5pRqIMp1o4HH_c0d47MylS6nbyV39b8h11BoHZyI8qaahIkDrQYoOmM
- <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-package-diagram/>
- <https://sudonull.com/post/79895-Honest-MVC-on-React-Redux-Developer-Soft-Blog>
- <https://upcommons.upc.edu/bitstream/handle/2099.1/18373/90629.pdf>

1.5 Overview

Tài liệu này là cách tiếp cận đầu tiên để trình bày thông tin của dự án Friendly Classroom trong một kiến trúc và thảo luận về kiến trúc của nó. Nó cũng cung cấp hướng dẫn cho sự phát triển trong tương lai.

Tài liệu gồm các phần như sau:

- Introduction
- Architectural Goals and Constraints
- Use-case model
- Logical View
- Deployment
- Implementation View

2. Architectural Goals and Constraints

Một số yêu cầu chính và các ràng buộc hệ thống có ảnh hưởng đáng kể đến kiến trúc hệ thống:

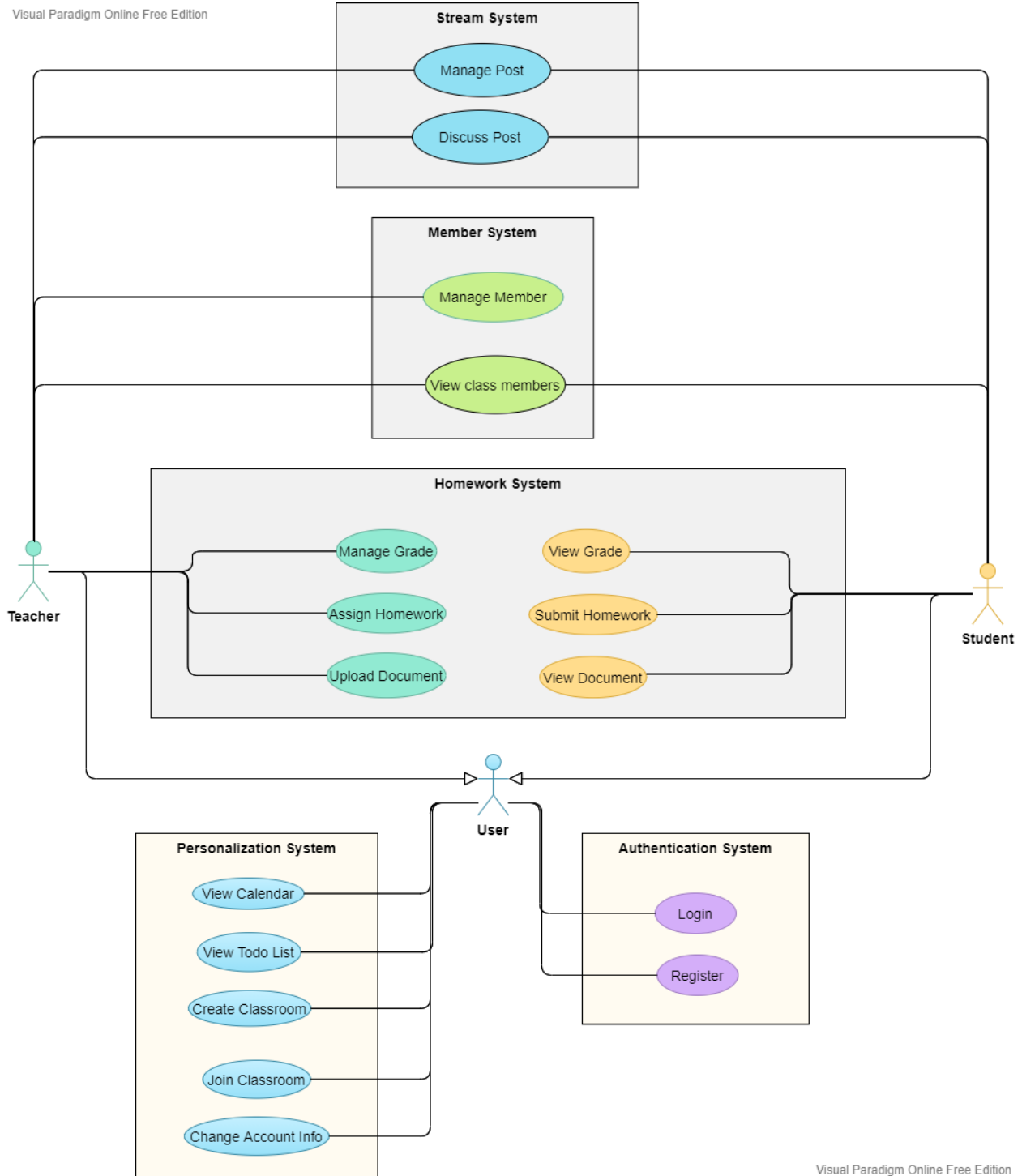
- Hệ thống sử dụng mô hình kiến trúc MVC (Model-View-Controller)
- Hệ thống Friendly Classroom sẽ được triển khai như một hệ thống client-server.
- Mọi người dùng có kết nối internet đều có thể truy cập đến hệ thống từ xa.
- Development tools: React Library, MongoDB Server, ExpressJs, NodeJs, RESTful API, Figma.
- Sử dụng thiết kế Material Design UI để tạo ra giao diện hiện đại và thân thiện.
- Hệ thống đăng ký phải đảm bảo bảo vệ dữ liệu người dùng khỏi bị tấn công hay truy cập không hợp lệ. Tất cả các truy cập từ xa đều phải kiểm soát bằng tên đăng nhập và mật khẩu.
- Dự án bao gồm 5 thành viên, hoạt động xuyên suốt với nhau, số lượng thành viên là cố định

Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

- Cách code được quy định và thiết kế theo chuẩn để có sự thống nhất giữa các thành viên.
- Thiết kế giao diện được thực hiện xong trước khi cài đặt hệ thống. Trong quá trình cài đặt sẽ có một vài trường hợp thay đổi giao diện để phù hợp với sự phát triển dự án.

Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

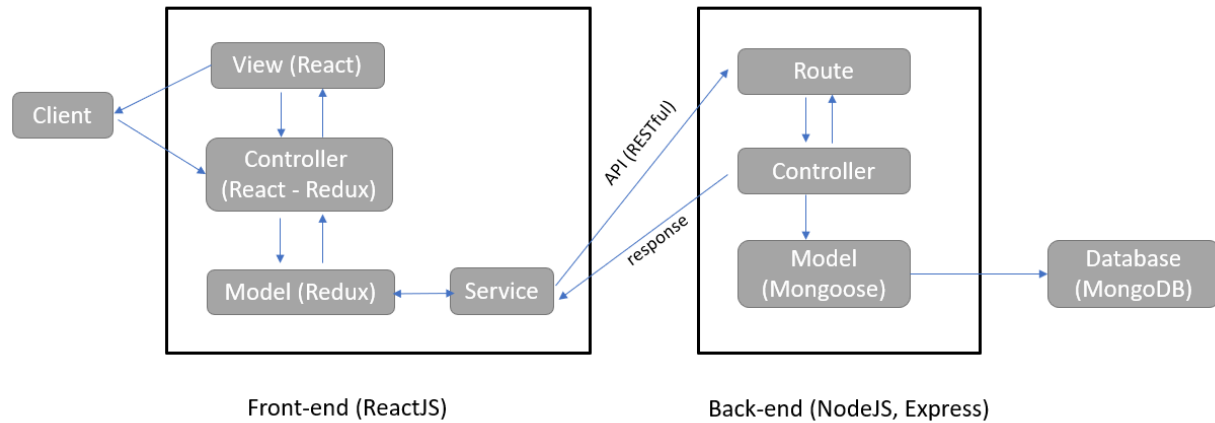
3. Use-Case Model



Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

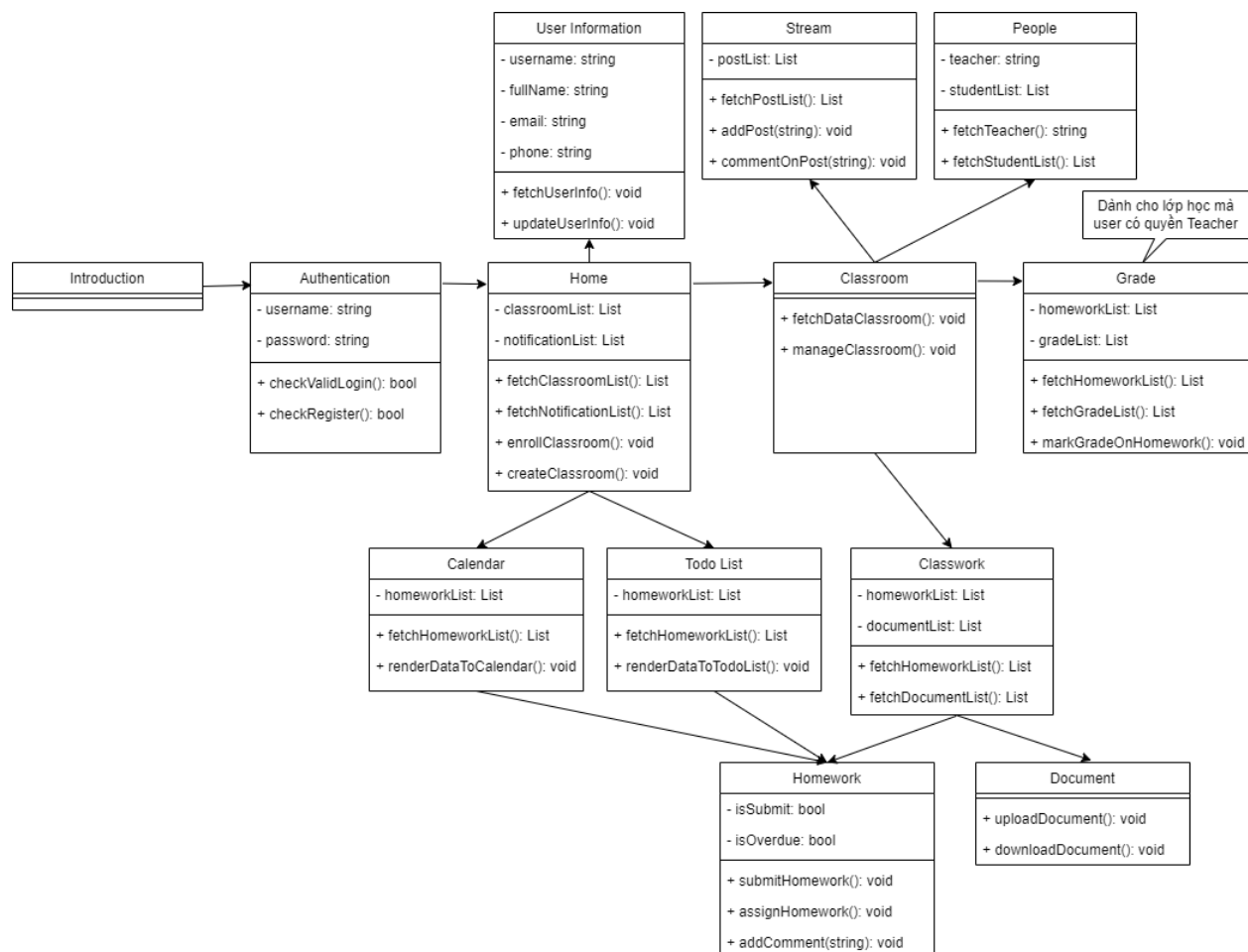
4. Logical View

4.1 Structural Diagram



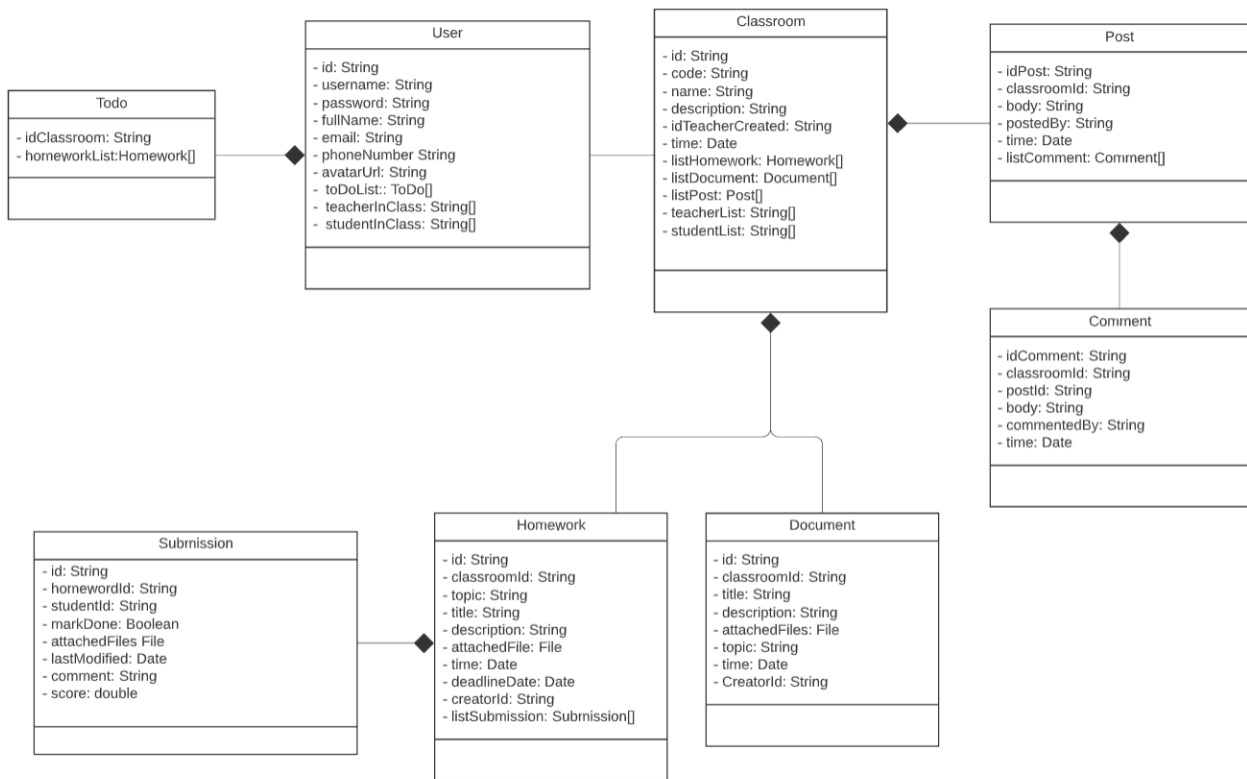
4.1.1 View Component

- Các hàm bắt đầu bằng “fetch” ở đây dùng để lấy thông tin từ server để hiển thị ra màn hình.



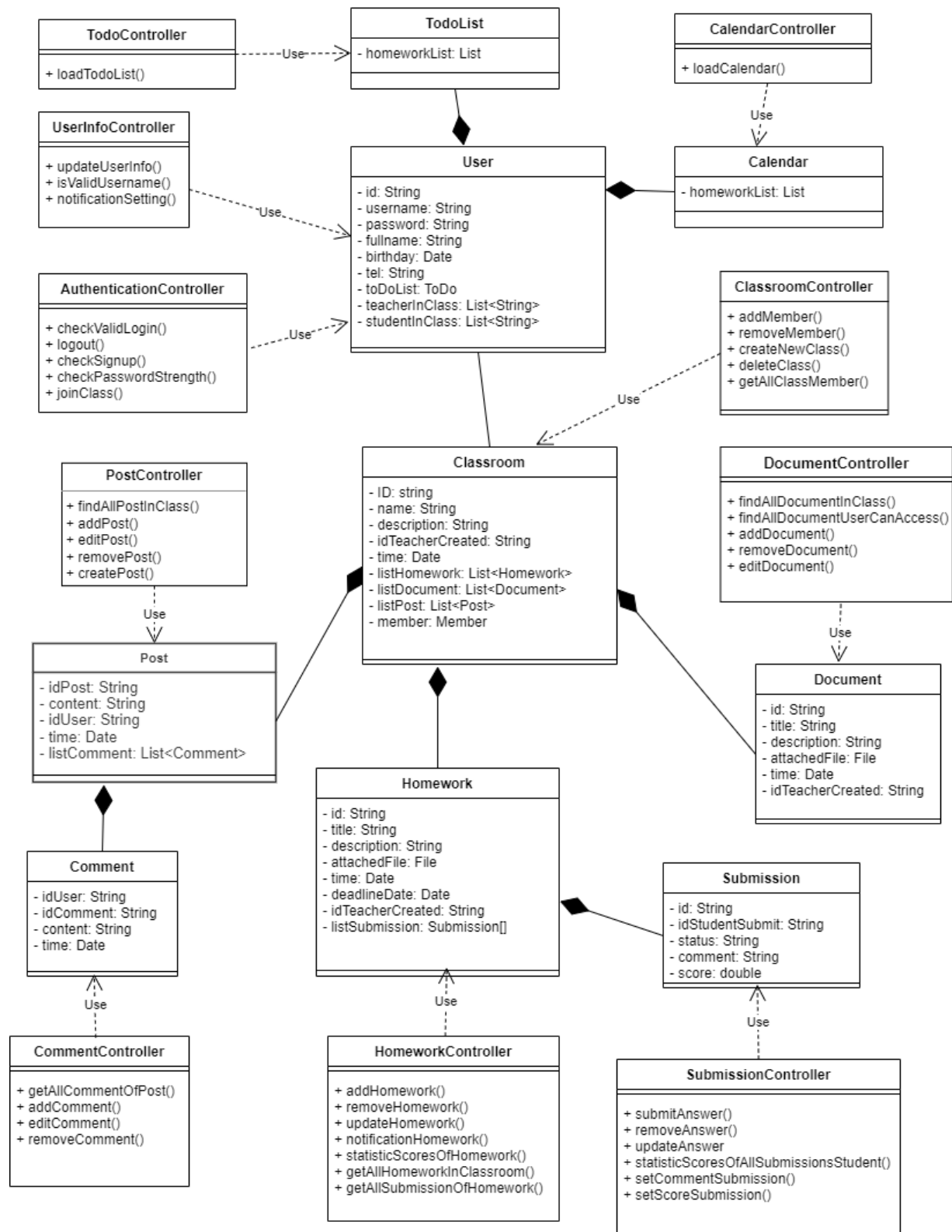
Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

4.1.2 Model Component



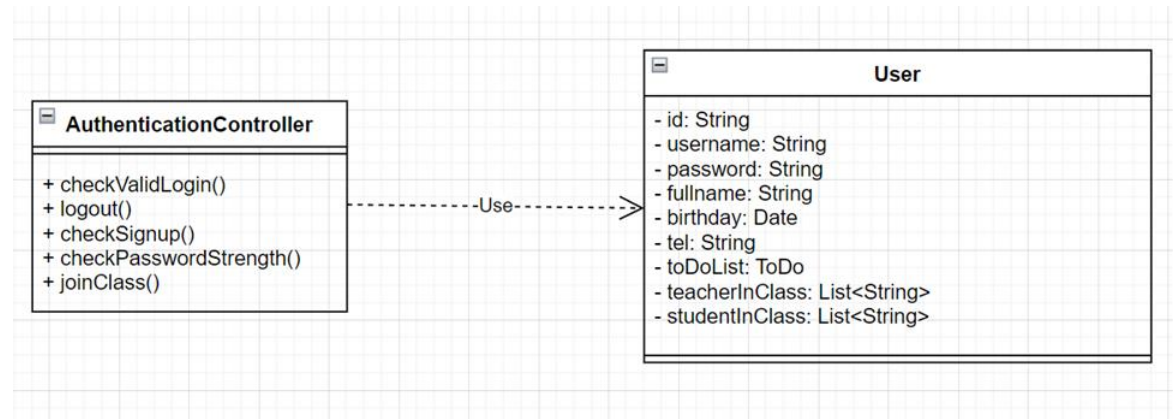
Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

4.1.3 Controller Component



Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

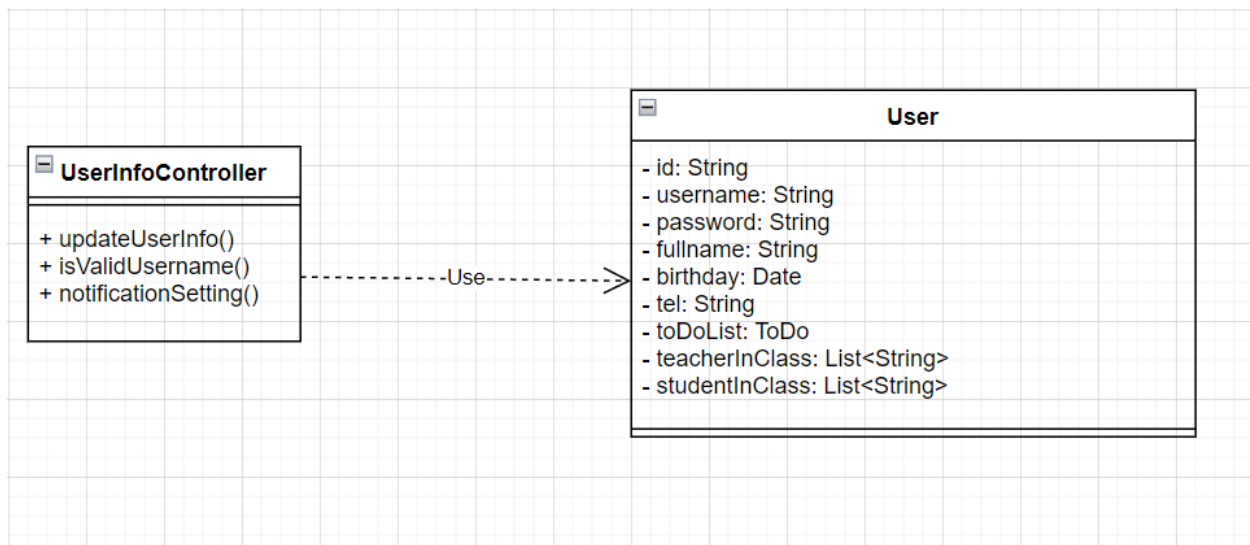
4.1.3.1 Authentication Controller



Kiểm soát việc sử dụng tài khoản

- `checkValidLogin()`: Kiểm tra thông tin đăng nhập của User có tồn tại hay không
- `logout()`: Kiểm soát việc đăng xuất
- `checkSignup()`: Kiểm tra thông tin đăng ký hợp lệ hay không
- `checkPasswordStrength()`: Kiểm tra mật khẩu có đủ mạnh hay không
- `joinClass()`: Kiểm soát việc tham gia vào một lớp học

4.1.3.2 UserInfoController

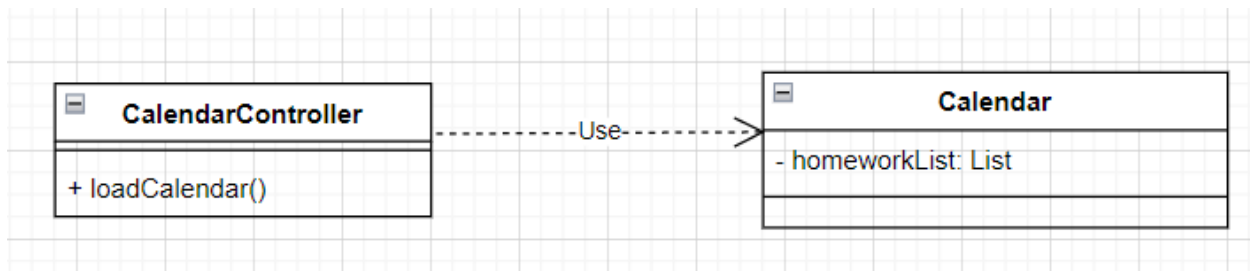


Kiểm soát việc xem, chỉnh sửa các thông tin chi tiết của tài khoản

- `updateUserInfo()`: Cập nhật thông tin cá nhân của người dùng
- `isValidUsername()`: Kiểm tra "username" có hợp lệ hay không
- `notificationSetting()`: Cài đặt thông báo cho tài khoản

4.1.3.3 CalendarController

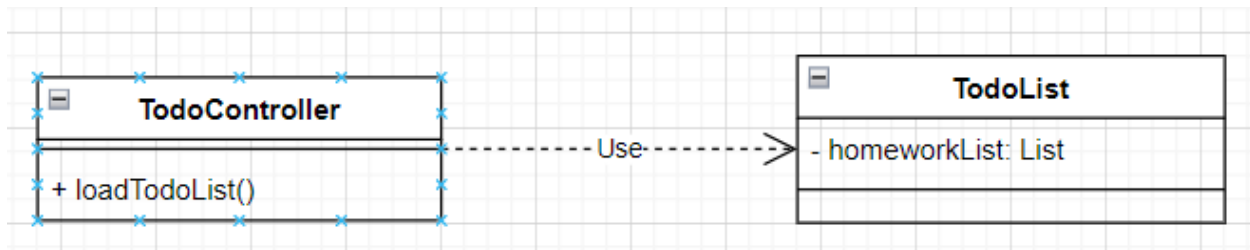
Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	



Kiểm soát việc thể hiện các bài tập và deadline trên Calendar

- loadCalendar(): load danh sách các deadline để hiển thị lên Calendar của một người dùng

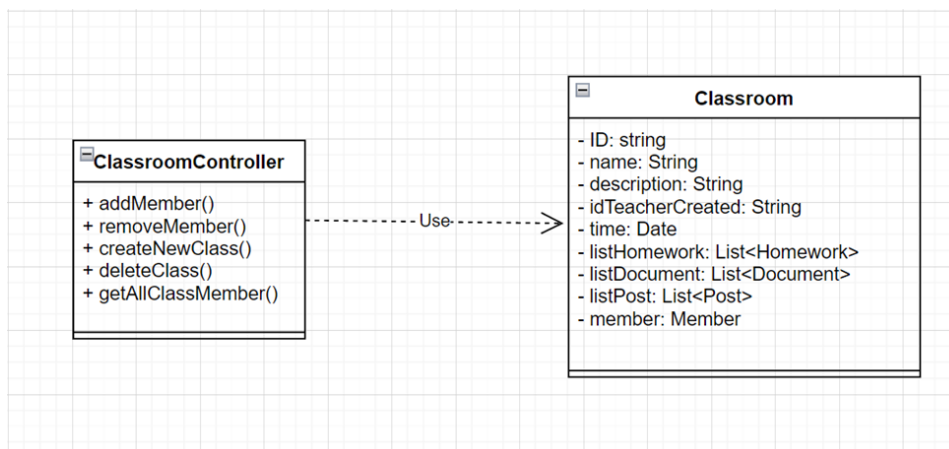
4.1.3.4 TodoController



Kiểm soát việc thể hiện các bài tập cần làm trên TodoList

- loadTodoList(): load danh sách các deadline để hiển thị lên mục Todo List của một người dùng

4.1.3.5 ClassroomController

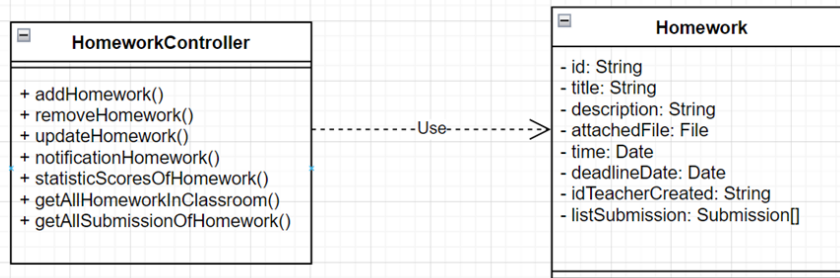


Kiểm soát chung lớp học

- addMember(): thêm một member cho lớp học
- removeMember(): xoá một member trong lớp hiện có
- createNewClass(): tạo một lớp học mới
- deleteClass(): xoá một lớp học
- getAllClassMember(): lấy danh sách thành viên lớp

4.1.3.6 HomeworkController

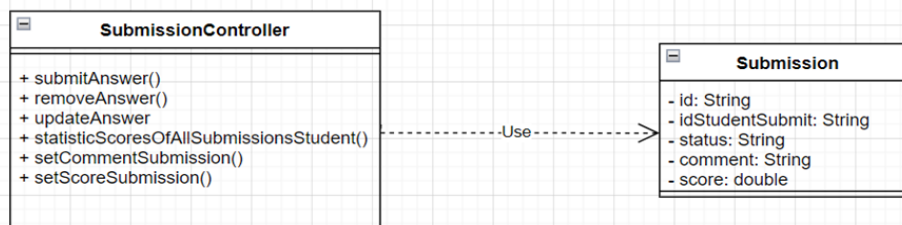
Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	



Kiểm soát việc thêm/xóa bài tập, truy xuất thông tin bài tập

- addHomework(): thêm bài tập vào CSDL
- removeHomework(): xóa bài tập khỏi CSDL
- updateHomework(): edit lại bài tập
- notificationHomework(): thông báo cho tất cả user trong lớp là lớp vừa có bài tập mới
- statisticScoresOfHomework(): thống kê điểm trong bài tập

4.1.3.7 SubmissionController



Kiểm soát các bài nộp

- submitAnswer(): thêm bài nộp vào CSDL
- removeAnswer(): xóa bài nộp khỏi CSDL
- updateAnswer(): sửa 1 bài nộp
- statisticScoresOfAllSubmissionsStudent: thống kê điểm tất cả bài nộp
- setCommentSubmission(): bình luận, nhận xét về bài nộp (dành cho giáo viên)
- setScoreSubmission(): chấm điểm cho bài nộp (dành cho giáo viên)

4.1.3.8 PostController



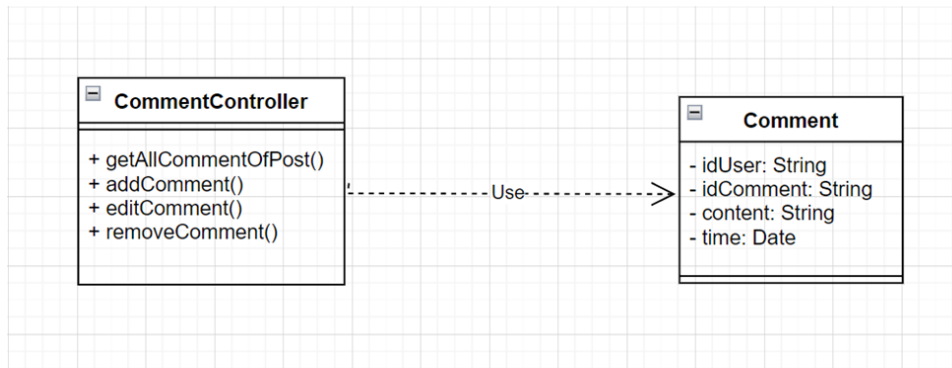
Kiểm soát các bài đăng trong lớp học

- findAllPostInClass(): trả về toàn bộ post thuộc về 1 lớp

Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

- addPost(): thêm 1 post vào CSDL
- editPost(): edit 1 post
- removePost(): xóa 1 post khỏi CSDL

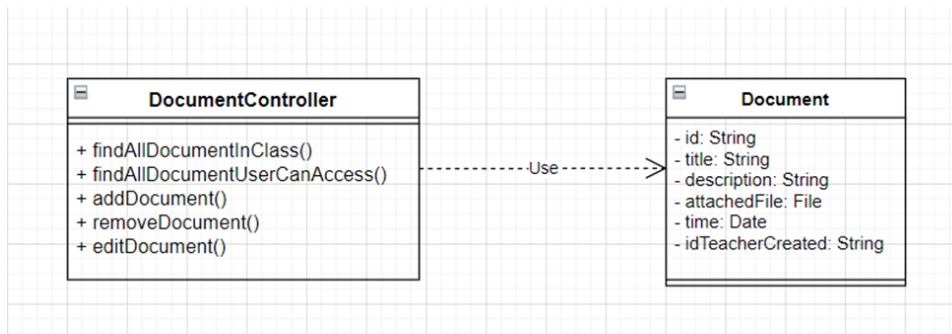
4.1.3.9 CommentController



Kiểm soát các bình luận

- getAllCommentOfPost(): trả về toàn bộ bình luận của 1 Post
- addComment(): thêm 1 comment vào CSDL
- editComment(): sửa 1 comment
- removeComment(): xóa 1 comment khỏi CSDL

4.1.3.10 DocumentController

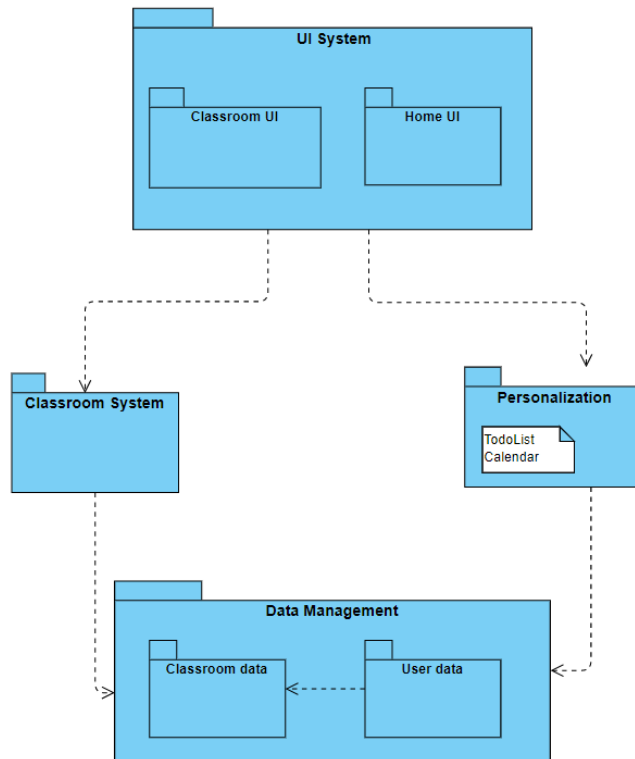


Kiểm soát việc thêm xóa sửa và truy vấn tài liệu

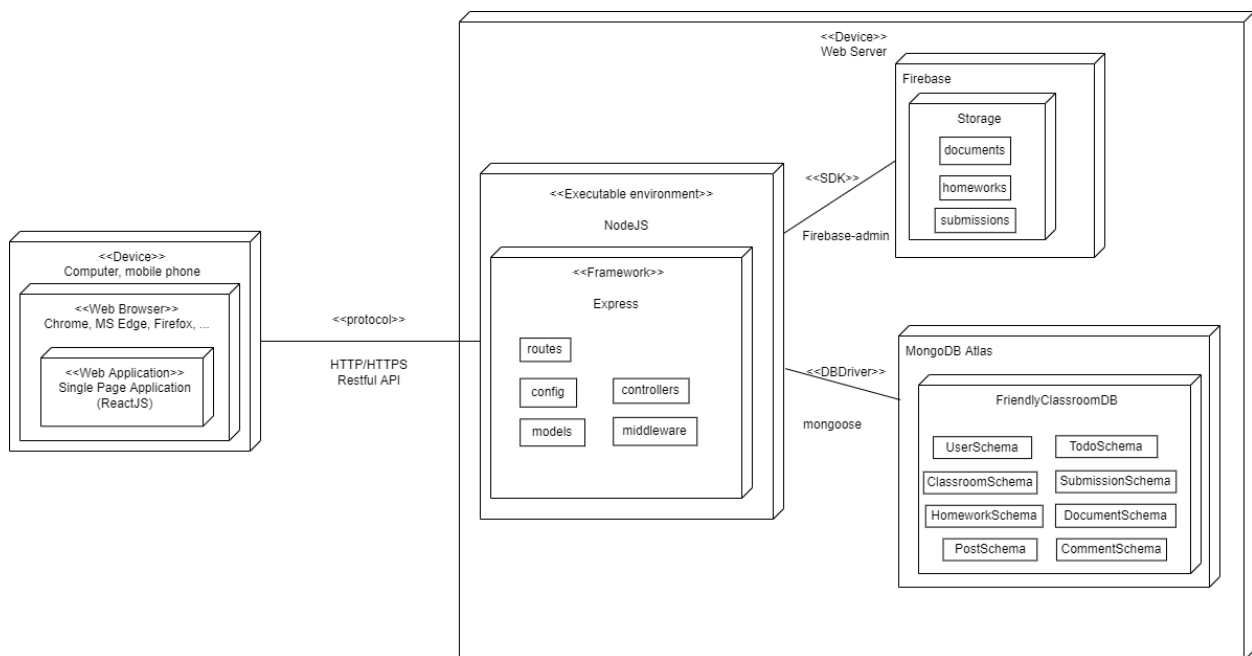
- findAllDocumentInClass(): trả về danh sách toàn bộ tài liệu của 1 lớp
- findAllDocumentUserCanAccess(): trả về danh sách toàn bộ tài liệu mà user có thể truy cập được (User đó phải theo học lớp chứa tài liệu đó)
- addDocument(): thêm 1 document vào CSDL
- removeDocument(): xóa 1 document khỏi CSDL
- EditDocument(): sửa lại nội dung 1 document

Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

4.2 Package Diagram



5. Deployment



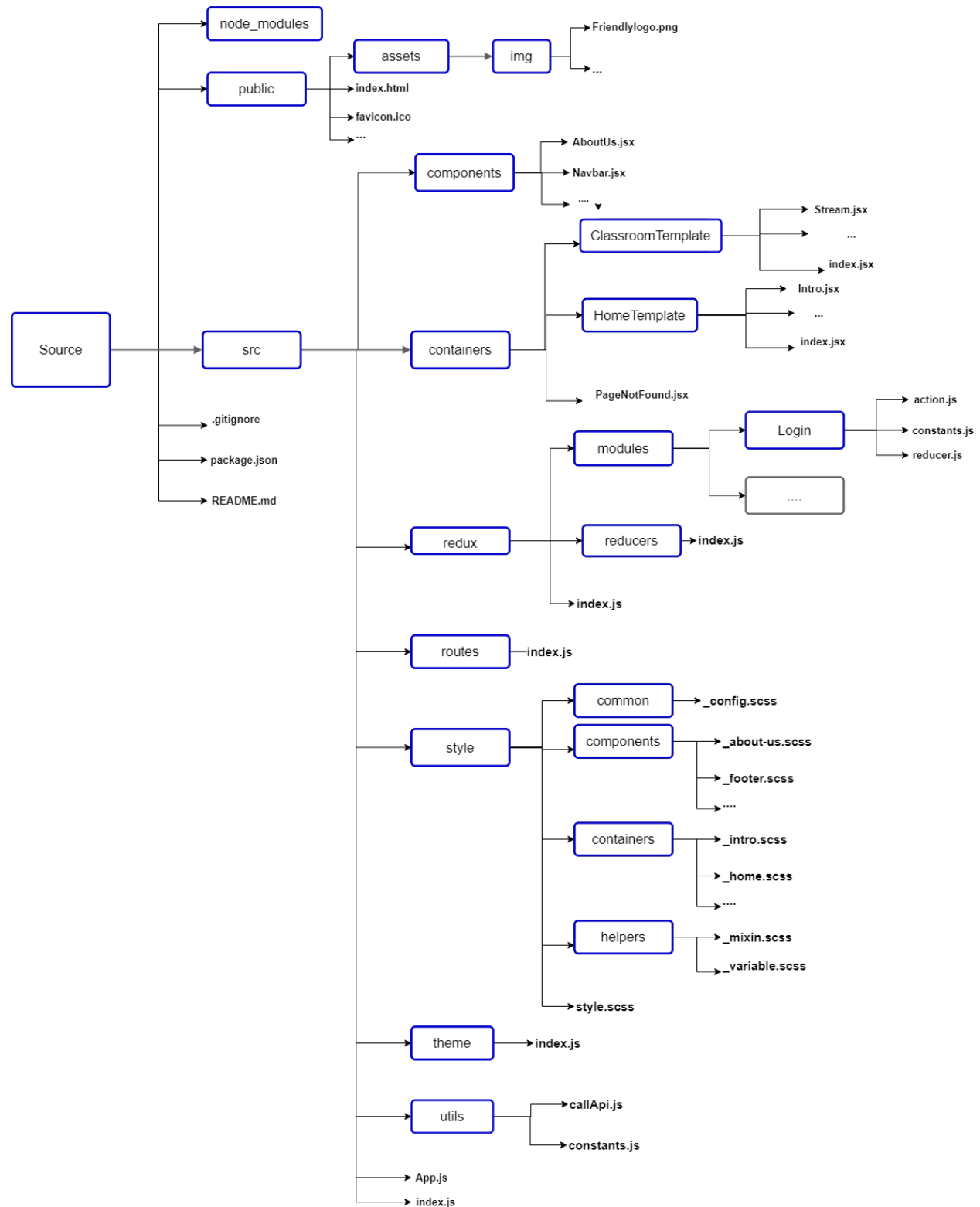
Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

- Bên phía Client, trang web có thể chạy trên các trình duyệt khác nhau như Chrome, MS Edge, ... Trang web dùng JSON Restful API để liên kết với server lấy ra dữ liệu tương ứng. Trang web sẽ được deploy lên Vercel.
- HiTeam sử dụng framework Express trên môi trường NodeJS để phát triển Server, cùng với middleware JWT để authentication người dùng và tạo ra accessToken. Đối với database, ta dùng MongoDB Atlas để lưu trữ thông tin dưới dạng text như User, Post,... và Firebase Storage để lưu trữ các file media như là avatar người dùng hay các file tài liệu. Dùng Mongoose dùng để giao tiếp với MongoDB Atlas và Firebase-admin để giao tiếp với Firebase Storage. Team sẽ sử dụng Heroku để triển khai và quản lý Webserver.

Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

6. Implementation View

6.1 Front End

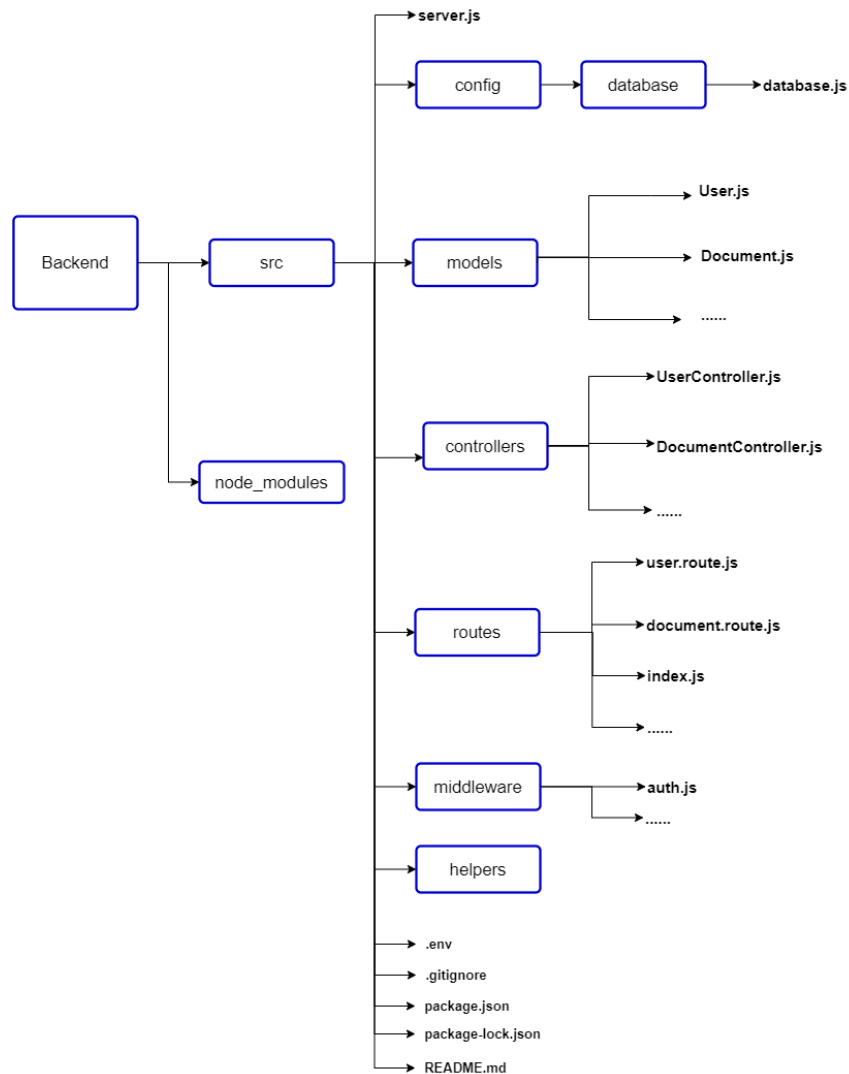


Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

- Thư mục **node_modules**: chứa các dependencies được cài đặt cho dự án
- Thư mục **public**:
 - + Thư mục **assets/img**: chứa các file ảnh sẽ sử dụng cho trang web
 - + File **index.html**: trang index chung để render tất cả giao diện
 - + Các file favico.ico, logo.png, ...
- File **.gitignore**: chứa các thư mục và file không định push lên github
- File **package.json**: lưu tên và version các dependencies cài đặt trong dự án
- File **README.md**: tóm tắt các thông tin về đồ án của nhóm
- Thư mục **components**: chứa các file jsx của component riêng lẻ để tạo nên một page trong web như AboutUs.jsx, Navbar.jsx, ...
- Thư mục **containers**: chứa các file jsx đại diện cho từng page trong web, các page này được tạo ra từ tổ hợp nhiều component được code trong thư mục **components** trên
 - + Thư mục **HomeTemplate**: chứa các page thuộc phạm vi chung của user như Intro.jsx, Login.jsx, ... File **index.jsx** sẽ tổng hợp tất cả các page này và chia route cho từng page.
 - + Thư mục **ClassroomTemplate**: chứa các page thuộc phạm vi trong 1 lớp học như Stream.jsx, Classwork.jsx, ... File **index.jsx** sẽ tổng hợp tất cả các page này và chia route cho từng page.
 - + File **PageNotFound.jsx**: trang web 404 nếu không tìm thấy đường dẫn phù hợp
- Thư mục **redux**: kho lưu trữ và quản lý tất cả các state của ứng dụng tại một store. Các action gọi API của trang web cũng được định nghĩa tại thư mục này. Bao gồm:
 - + Thư mục **modules**: mỗi thư mục trong đây tương ứng các state, lời gọi API được sử dụng trong một component. Chẳng hạn với thư mục Login:
 - file **action.js**: chứa tất cả action tương ứng cho xử lý Login
 - file **constant.js**: chứa các constant sử dụng trong action
 - file **reducer.js**: reducer quản lý state tương ứng dùng cho chức năng Login và xử lý các action
 - + Thư mục **reducers**: file index.js chứa root reducer quản lý tổng hợp và quản lý tất cả các reducer của project
 - + file **index.js**: Sử dụng react-redux và redux-thunk để tạo store có sử dụng middleware
- Thư mục **routes**: file **index.js** khai báo các object chứa tương ứng component cùng đường dẫn để chuyển đổi qua lại giữa các component
- Thư mục **style**: chứa các file scss cho project
 - + Thư mục **common**: các file trong đây chứa các đoạn mã css được dùng phổ biến cho dự án. Ví dụ: _config.scss chứa các đoạn style cài đặt lúc khởi tạo cho toàn bộ project
 - + Thư mục **components**: chứa các file scss tương ứng cho từng component trong thư mục components ở trên
 - + Thư mục **containers**: chứa các file scss tương ứng cho từng container trong thư mục containers ở trên
 - + helpers:
 - file **_mixin.scss**: tái sử dụng những đoạn code css lặp đi lặp lại, tăng hiệu suất code
 - file **_variables.scss**: chứa những biến dùng chung cho project như font-size, màu sắc... giúp dễ sửa đổi và mở rộng dự án
 - + file **style.scss**: tổng hợp các file scss bên trên
- Thư mục **theme**: cài đặt theme cho project (có sử dụng Material UI)
- Thư mục **utils**: các hàm js được sử dụng lặp đi lặp lại như callApi, các hằng số....
- File **App.js**: chứa App component là component chính trong React, như một container chứa tất cả component còn lại, tại đây cũng có cài đặt Routing
- File **index.js**: render component App bằng cách DOM đến div chứa id là 'root' tại file index.html, tạo Single Page Application

Friendly Classroom	Version: 2.0
Software Architecture Document	Date: 01/12/2021
Kiến trúc phần mềm	

6.2 Back End



- Thư mục **node_modules**: chứa các dependencies được cài đặt cho dự án
- File server.js: entry point của server
- Thư mục models: chứa tất cả các file Mongoose Schema
- Thư mục routes: chứa tất cả các route được tạo bằng cách sử dụng Express Router kết hợp với thư mục controllers
- Thư mục controllers: chứa các chức năng xử lý cụ thể cho từng API
- Thư mục config: dùng để cấu hình các API / dịch vụ của bên thứ ba
- File package.json: chứa tất cả các chi tiết npm của dự án, các lệnh chạy scripts và các dependencies
- File .gitignore: chứa các thư mục và file không muốn push lên github