Netfliz

Software Architecture Document

Version 1.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 27/11/2022 | 1.0 | Introduction, Use-Case Model, Logical View | Phạm Trần Gia Phú, Trần Hoàng Tín, Phạm Phước Sang, Nguyễn Đức Quang, Trần Minh Tuấn Kiệt |
| 13/12/2022 | 1.1 | Sửa Logical view, Components, Deployment, Implementation view | Phạm Trần Gia Phú, Trần Hoàng Tín, Phạm Phước Sang, Nguyễn Đức Quang, Trần Minh Tuấn Kiệt |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[**1. Introduction**](#_heading=h.gjdgxs) **4**

[**2. Architectural Goals and Constraints**](#_heading=h.30j0zll) **4**

[**3. Use-Case Model**](#_heading=h.1fob9te) **5**

[**4. Logical View**](#_heading=h.3znysh7) **5**

[4.1 Component: Quản lý tài khoản cá nhân](#_heading=h.2et92p0) 6

[4.2 Component: Quản lý tài khoản admin](#_heading=h.ltp83bd6ss7) 7

[4.3 Component: Đăng tải comment trên phim](#_heading=h.k4sd3eqhn36r) 8

[4.4 Component: Thao tác với comment trên phim](#_heading=h.ie69fgbweguh) 9

[4.5 Component: Tương tác với comment trên phim](#_heading=h.935zzc1bon58) 10

[4.6 Component: Tìm kiếm, lọc phim](#_heading=h.j7rp3cfweevw) 11

[4.7 Component: Quản lý các phim](#_heading=h.xwcuyoh1xpuo) 11

[4.8 Component: Quản lý danh sách phim yêu thích](#_heading=h.sue3gvrb6v30) 12

[4.9 Component: Cập nhật thông tin cá nhân](#_heading=h.kes17ubidyn) 13

[4.10 Component: Xem chi tiết phim, xem phim](#_heading=h.owyi5jv5738n) 14

[4.11 Component: Thay đổi trang web](#_heading=h.qb2oor4yohah) 15

[**5. Deployment**](#_heading=h.tyjcwt) **15**

[**6. Implementation View**](#_heading=h.3dy6vkm) **15**

Software Architecture Document

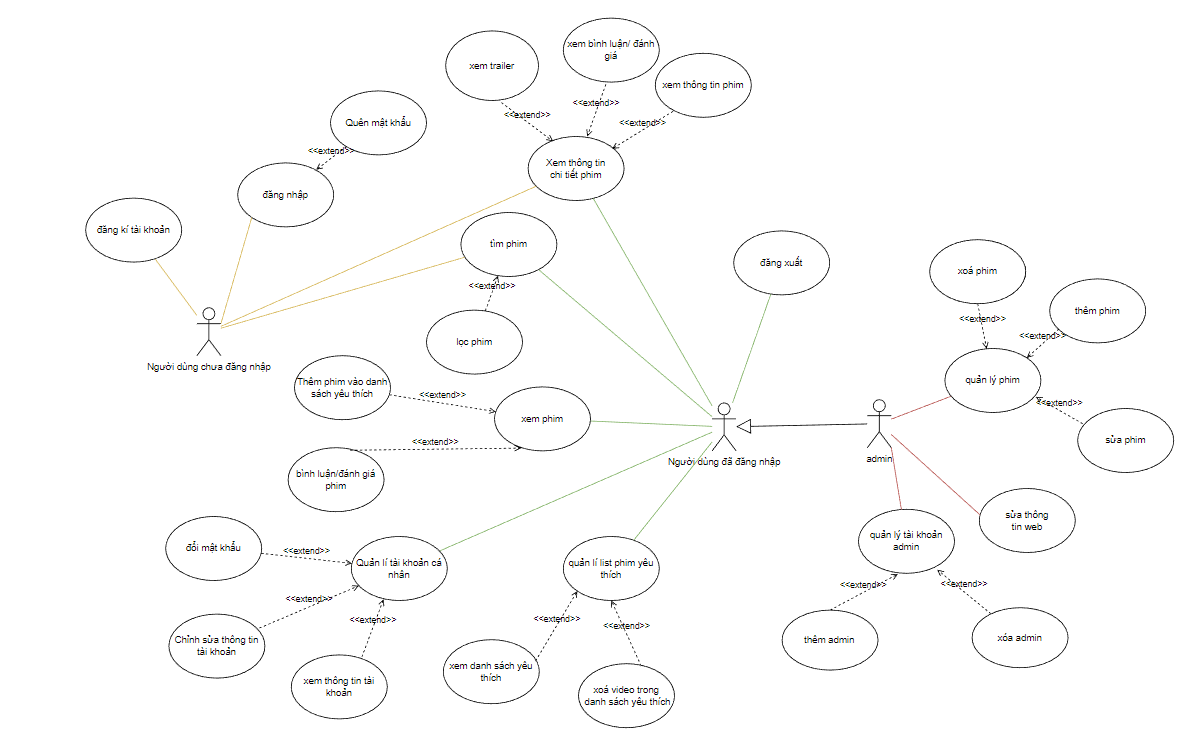
# Introduction

* 1. **Purpose:**
* Tài liệu Kiến trúc Phần mềm này giúp người đọc có cái nhìn tổng quan về kiến trúc của toàn bộ hệ thống. Nó cung cấp các mô hình về kiến trúc hệ thống dưới nhiều góc độ khác nhau.
* Tài liệu này được dùng để bao quát và truyền tải các quyết định quan trọng khi chọn kiến trúc phần mềm cho hệ thống.
  1. **Scope:**
* Tài liệu Kiến trúc Phần mềm này được sử dụng trong quá trình phát triển website xem phim trực tuyến Netfliz.
  1. **Overview:**
* Tài liệu Kiến trúc Phần mềm này bao gồm:
  + Architectural Goals and Constraint: Các mục tiêu và ràng buộc của kiến trúc phần mềm.
  + Use-case Model: Chứa mô hình use-case
  + Logical View: Cung cấp danh sách các components trong hệ thống và mối quan hệ giữa chúng, bao gồm package diagram và class diagram.
  + Deployment: Phương pháp triển khai các thành phần của ứng dụng
  + Implementation View: Cấu trúc các thư mục chứa code và cách cài đặt các components

# Architectural Goals and Constraints

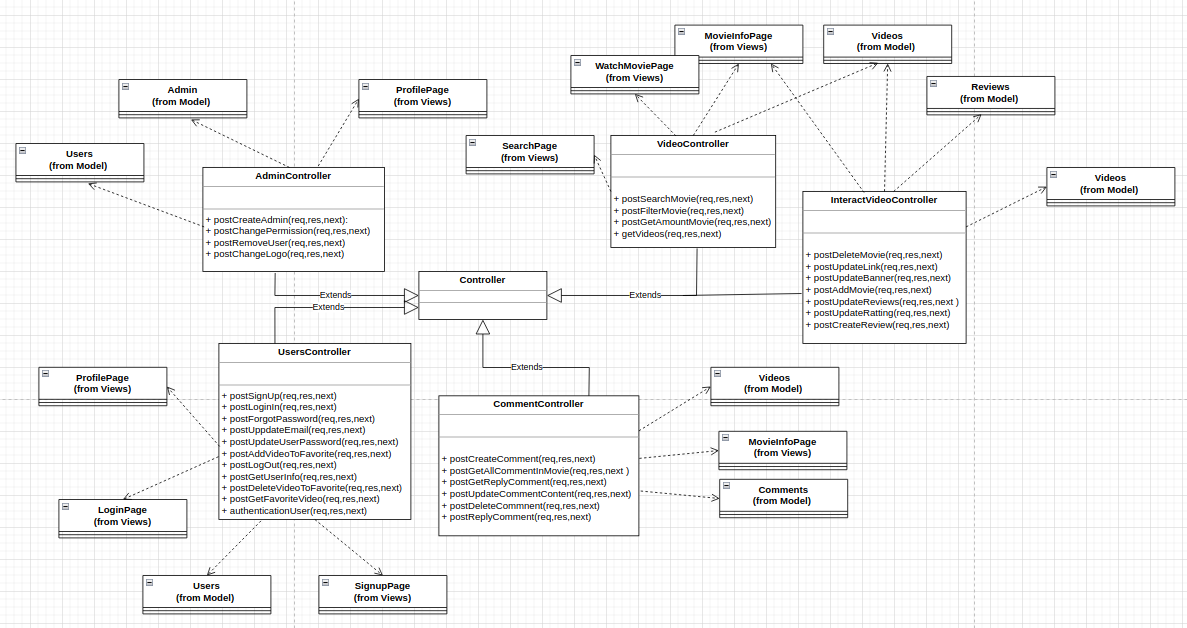
* Ngôn ngữ lập trình: HTML, CSS, Javascript, …
* Môi trường của trang web: tương thích với các giao diện khác nhau như điện thoại, tablet, PC, laptop và các trình duyệt khác nhau như Google Chrome, Microsoft Edge, …
* Hiệu năng:
  + Hệ thống sẽ hỗ trợ tối đa 1000 người dùng đồng thời dựa trên cơ sở dữ liệu trung tâm tại bất kỳ thời điểm, và tối đa 200 người dùng đồng thời với các máy chủ cục bộ.
  + Hệ thống đáp ứng 90% lượng trao đổi của người dùng trong 2 phút, đảm bảo cung cấp việc truy cập vào trang web NetfliZ với độ trễ không hơn 15 giây và trả lại kết quả tìm kiếm phim, lọc phim với độ trễ dưới 2 giây.
* Bảo mật: Thông tin của người dùng sẽ được bảo mật cao.
* Dữ liệu: Hệ thống sử dụng cơ sở dữ liệu postgres, dịch vụ lưu trữ firebase.
* Trợ giúp: Nếu có bất kỳ thắc mắc hay gặp khó khăn trong quá trình sử dụng, người dùng có thể chọn “Help ?” để đọc các hướng dẫn sử dụng chi tiết và mục FAQ (Frequently asked question).
* Hướng dẫn: Người dùng có thể dễ dàng thao tác trên trang web sau 2 phút hướng dẫn.

# Use-Case Model

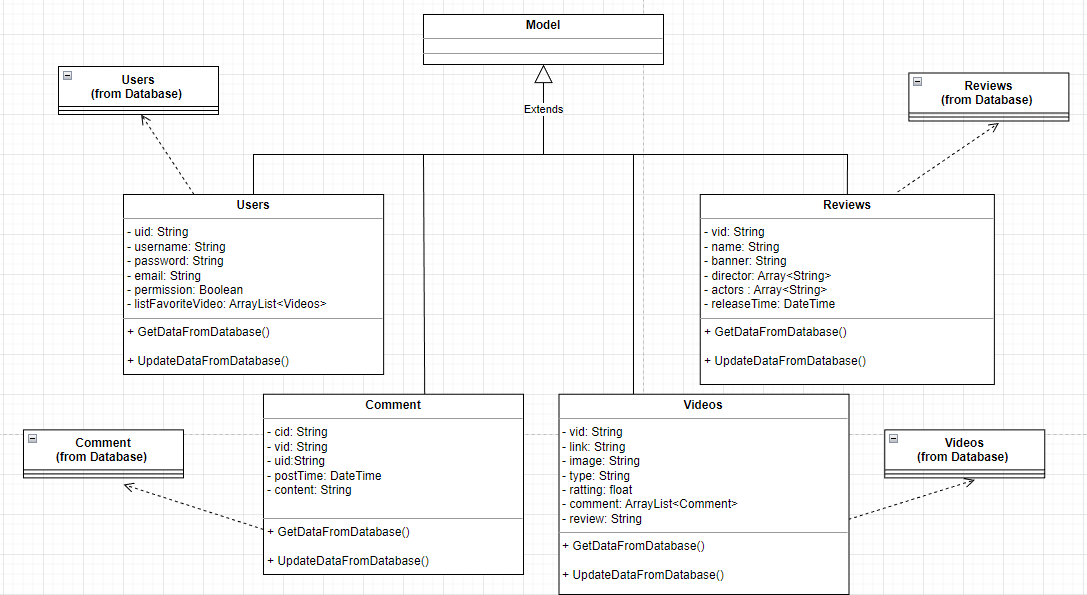
**

# Logical View

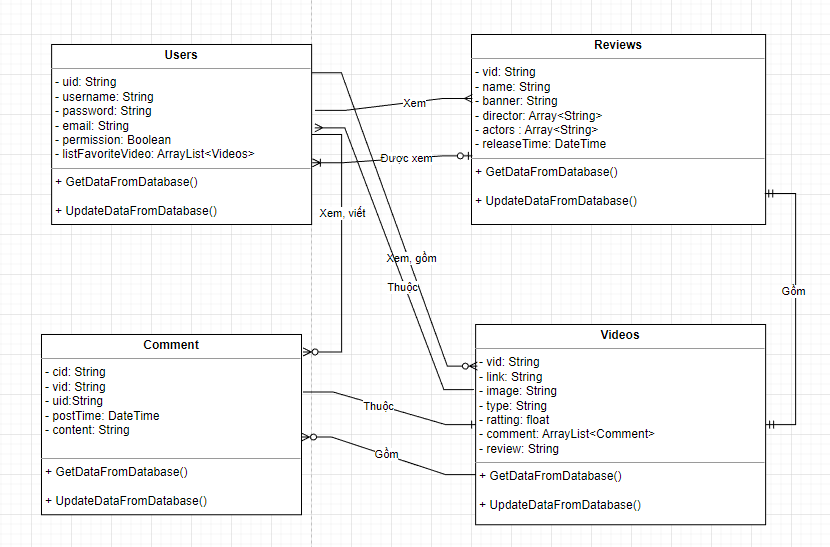
**Controller**

****

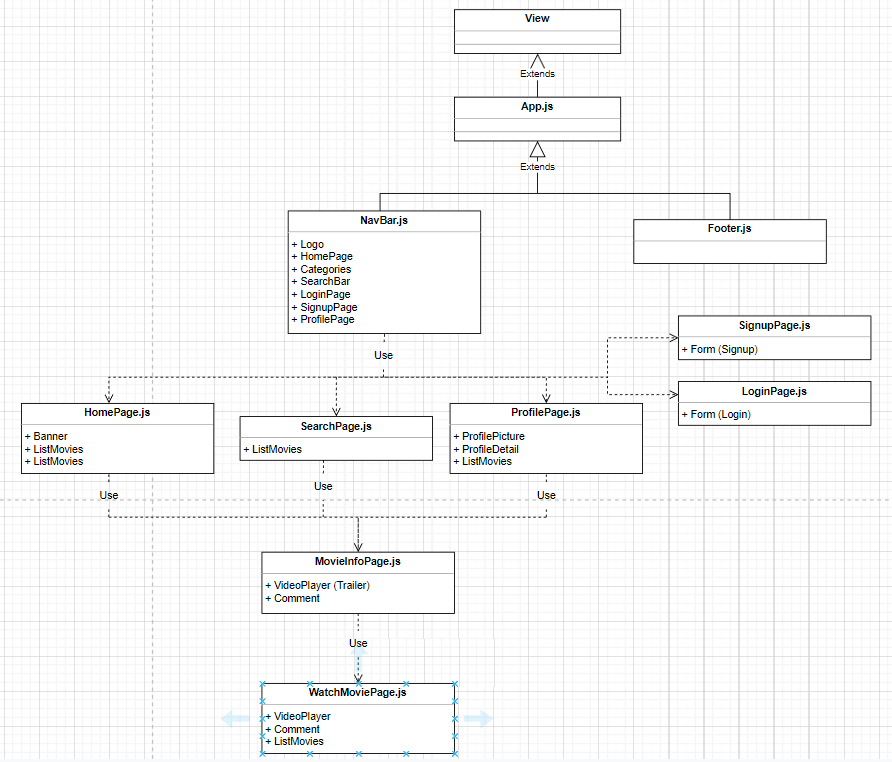
**Model**

****

**Database**

****

**View**

****

## 

## Component: Controller

**Mô tả:** Thành phần Controller bao gồm các lớp sau để điều khiển sự tương tác của nó và các thành phần khác với nhau.

- AdminController: Xử lý các tác vụ của quản trị viên đối với trang web

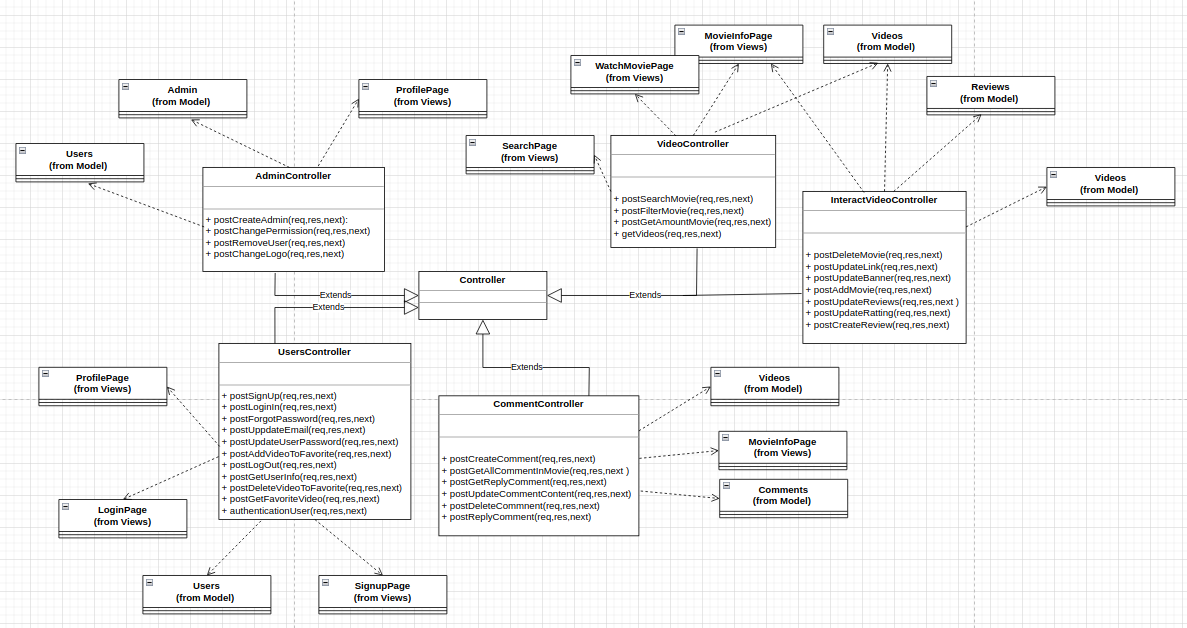
- VideoController: Quản lý các video

- UsersController: Xử lý các yêu cầu của người dùng

- CommentController: Quản lý bình luận

- InteractVideoController: Xử lý các tác vụ liên quan đến video

**class diagram:**

****

**Giải thích key-class:**

- AdminController:

* postCreateAdmin(req,res,next): tạo tài khoản admin
* postChangePermission(req,res,next): cấp quyền admin cho user
* postRemoveUser(req,res,next): xoá user
* postChangeLogo(req,res,next): đổi logo của trang web

- VideoController:

* postSearchMovie(req,res,next): Xử lí dữ liệu và hiển thị kết quả tìm kiếm phim
* postFilterMovie(req,res,next): Xử lí dữ liệu và hiển thị kết quả lọc phim theo chủ đề
* postGetAmountMovie(req,res,next): Trả về số lương phim trong DB
* getVideos(req,res,next): Xử lí dữ liệu phim để hiển thị video

- UsersController:

* postSignUp(req,res,next): Lấy dữ liệu và xử lí việc đăng kí
* postLoginIn(req,res,next): Lấy dữ liệu và xử lí việc đăng nhập
* postFogortPassword(req,res,next): lấy dữ liệu và xử lí việc quên mật khẩu.
* postUpdateEmail(req,res,next): lấy dữ liệu và xử lí việc cập nhật email.
* postUpdateUserPassword(req,res,next): lấy dữ liệu và xử lí việc cập nhật mật khẩu mới của người dùng.
* postAddVideoToFavorite(req,res,next): lấy dữ liệu và xử lí việc thêm video vào mục yêu thích của người dùng.
* postLogOut(req,res,next): Lấy dữ liệu và xử lí việc đăng xuất.
* postGetUserInfo(req,res,next): Lấy dữ liệu và xuất thông tin người dùng.
* postDeleteVideoToFavorite(req,res,next): Lấy dữ liệu và xử lí việc xoá video khỏi danh sách yêu thích.
* postGetFavoriteVideo(req,res,next): Lấy dữ liệu và xử lí việc hiển thị danh sách phim yêu thích của người dùng.
* authenticationUser(req,res,next): Lấy dữ liệu và xử lí việc xác minh người dùng.

- CommentCotroller:

* postCreateComment(req,res,next): Lấy dữ liệu và xử lí việc tạo bình luận của người dùng.
* postGetAllCommentInMovie(req,res,next): Lấy dữ liệu và hiển thị tất cả các bình luận của 1 bộ phim.
* postGetReplyComment(req,res,next): Lấy dữ liệu và xử lí việc hiển thị trả lời của bình luận khác của người dùng.
* postUpdateCommentContent(req,res,next): Lấy dữ liệu và xử lí việc chỉnh sửa bình luận của bản thân người dùng.
* postDeleteComment(req,res,next): Lấy dữ liệu và xử lí việc xoá bình luận của người dùng.
* postReplyComment(req,res,next): Lấy dữ liệu và xử lí việc trả lời bình luận của người dùng với người dùng khác.

- InteractVideoController:

* postDeleteMovie(req,res,next): Lấy id của video được chọn để xóa và server xử lý việc xóa video trên database.
* postUpdateLink(req,res,next): Lấy id của video được chọn để sửa liên kết và server xử lý việc thay đổi đường dẫn của video.
* postUpdateBanner(req,res,next): Lấy file ảnh và server xử lý việc upload ảnh lên database và hiển thị trên web.
* postAddVideo(req,res,next): Lấy dữ liệu, thông tin của video và server xử lý việc đăng tải, thêm video mới vào database, từ đó hiển thị lên trang web.
* postUpdateReview(req,res,next): Lấy id của video được chọn cùng với nội dung review được cập nhập và server xử lý việc cập nhật review của bộ phim đó trên database
* postUpdateRating(req,res,next): Lấy id của video được chọn để cập nhật đánh giá và server cập nhật rating của bộ phim đó
* postCreateReview(req,res,next): Lấy id của video được chọn để tạo mới review và server thực hiện tạo review cho video đó, chỉ được thực hiện 1 lần.

## Component: Model

**Mô tả:** Component model gồm các lớp sau thể hiện các thuộc tính cấu trúc dữ liệu khi ứng dụng trao đổi với database server

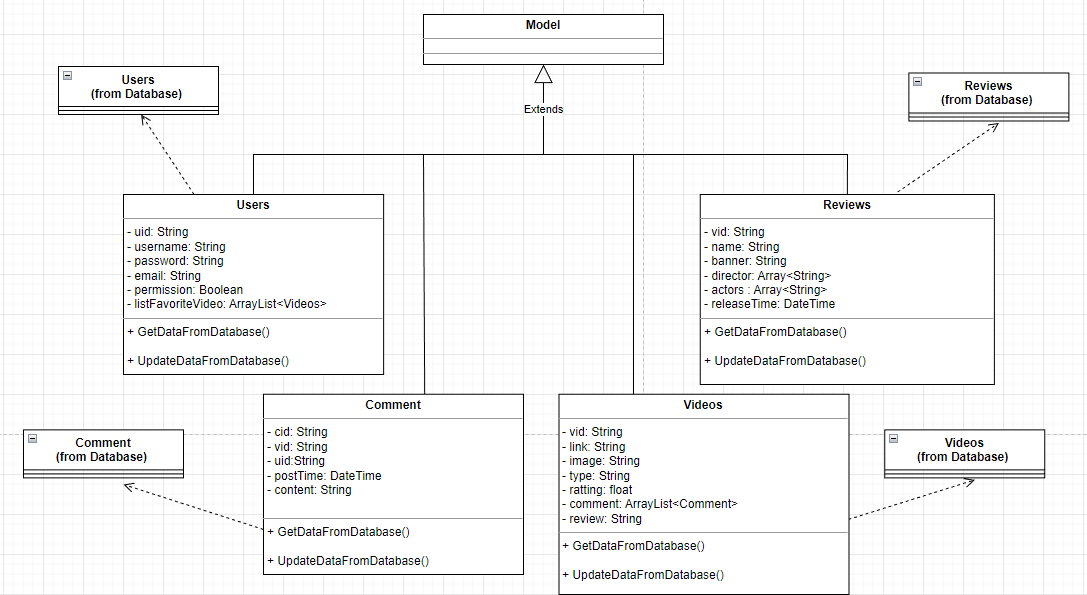
- Users: Model bao gồm các thuộc tính được lưu trữ, giao tiếp của người dùng.

- Reviews: Model bao gồm các thuộc tính được lưu trữ, giao tiếp của giới thiệu, thông tin thêm của phim

- Comment: Model bao gồm các thuộc tính được lưu trữ về những bình luận thuộc phim của người dùng.

- Videos: Model bao gồm các thuộc tính được lưu trữ của phim.

**Class diagram:**



**Giải thích key-class:**

* Users:
* uid: id của người dùng.
* username: tên đăng nhập của người dùng.
* password: mật khẩu người dùng.
* email: dùng để khôi phục mật khẩu khi quên.
* permission: true nếu người dùng là admin, false khi là người dùng bình thường (khởi tạo mặc định là false).
* listFavoriteVideo: danh sách video yêu thích của người dùng.
* GetDataFromDatabase(): lấy dữ liệu các thông tin của người dùng.
* UpdateDataFromDatabase(): cập nhật lại dữ liệu về thông tin của người dùng.
* Reviews:
* vid: id của phim.
* name: tên phim.
* banner: ảnh đai diện của phim.
* director: đạo diễn của phim.
* actors: diễn viên của phim.
* releaseTime: ngày phát hành.
* GetDataFromDatabase(): lấy dữ liệu về phần giới thiệu, thông tin thêm của phim.
* UpdateDataFromDatabase(): cập nhật lại dữ liệu về phần giới thiệu, thông tin thêm của phim.
* Comment:
* cid: id của bình luận.
* vid: id của phim được bình luận.
* uid: id của người dùng đã bình luận.
* postTime: thời gian đăng bình luận.
* content: nội dung của bình luận.
* GetDataFromDatabase(): lấy dữ liệu về những bình luận thuộc phim của người dùng.
* UpdateDataFromDatabase(): cập nhật lại dữ liệu về những bình luận thuộc phim của người dùng.
* Videos:
* vid: id của phim.
* link: chứa link lưu video phim.
* image: link ảnh poster của phim
* type: thể loại phim
* rating: đánh giá của phim
* comment: danh sách bình luận của phim.
* review: chứa thông tin của phim
* GetDataFromDatabase(): lấy dữ liệu của phim.
* UpdateDataFromDatabase(): cập nhật lại dữ liệu của phim.

## 

## Component: View

**Mô tả:** Component view gồm các lớp sau thể hiện các thuộc tính hiển thị của phần mềm

- App.js: Đây là trang chính mà trình duyệt sẽ render đầu tiên (tất cả các components chính đều được gọi từ trang này)

-NavBar.js: Thanh tiện ích của ứng dụng web với các chức năng như tìm kiếm, đăng nhập, đăng ký, … và luôn trên đầu màn hình ở mọi trường hợp

- Footer.js: Chân trang của ứng dụng web, hiển thị các thông tin thêm về trình duyệt

- HomePage.js: Trang chủ của ứng dụng web và được hiển thị đầu tiên mỗi khi truy cập vào.

- SearchPage.js: Trang hiển thị những thông tin mà người dùng nhập vào thanh tìm kiếm

- ProfilePage.js: Trang thông tin cá nhân của người dùng

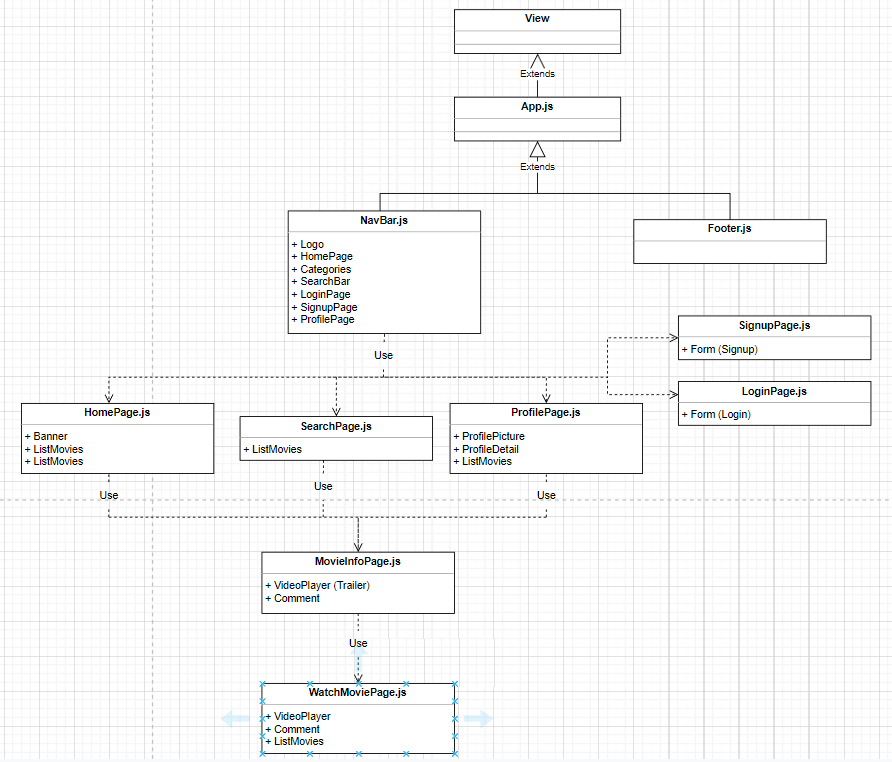
- LoginPage.js: Trang đăng nhập

- SignupPage.js: Trang đăng kí

- MovieInfoPage.js: Trang hiển thị những thông tin của bộ phim như trailer, review, ratings,...

- WatchMoviePage.js: Trang để người dùng thực hiện xem phim.

**Class diagram:**

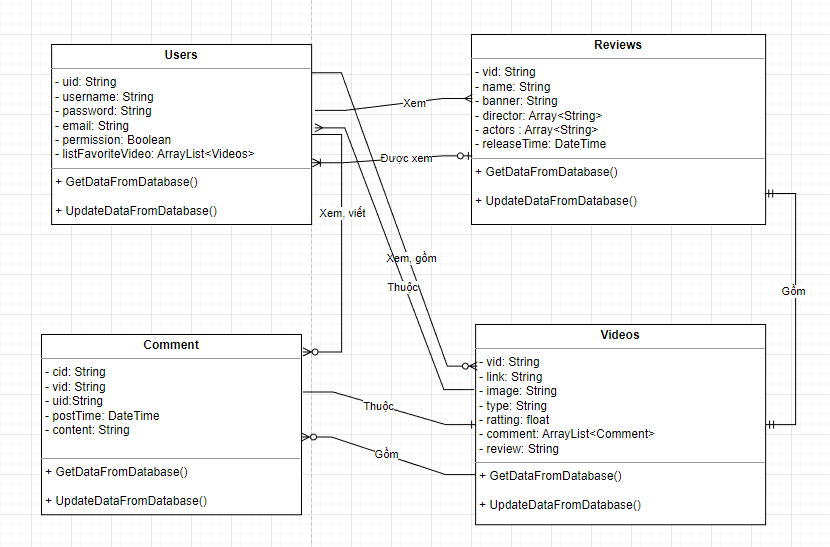
****

## Component: Database

**Mô tả:** Thành phần Database để chứa dữ liệu của hệ thống, gồm các tables sau:

* Users: chứa các thuộc thông tin về lớp người dùng
* Reviews: chứa các thông tin cho người dùng có cái nhìn sơ lược về phim (trailer và thông tin về phim)
* Comment: lưu thông tin về các bình luận của 1 người dùng trên 1 phim
* Videos: lưu dữ liệu về phim chính thức

**Class diagram:**



**Giải thích key-class:**

- Users:

* **uid (khóa chính)**: ID của người dùng
* username: tên đăng nhập
* password: mật khẩu người dùng
* email: dùng để khôi phục mật khẩu khi quên
* permission: true nếu người dùng là admin, false khi là người dùng bình thường (khởi tạo mặc định là false)
* listFavoriteVideo: danh sách video yêu thích của người dùng
* GetDataFromDatabase(): trả về toàn bộ dữ liệu của 1 user để cho backend xử lý
* UpdateDataFromDatabase(): chỉnh sửa data từ database

- Reviews:

* vid: mã số định danh của phần trailer phim
* name: Tên phim
* banner: ảnh đại diện của phim
* director: tên đạo diễn
* actors: danh sách tên diễn viên
* releaseTime: ngày phát hành
* GetDataFromDatabase(): trả về toàn bộ dữ liệu của 1 user để cho backend xử lý
* UpdateDataFromDatabase(): chỉnh sửa data từ database

- Comment:

* cid: ID của bình luận
* vid: ID của phim được bình luận
* uid: ID của người dùng bình luận này
* postTime: ngày đăng bình luận
* content: nội dung của bình luận
* GetDataFromDatabase(): trả về toàn bộ dữ liệu của 1 user để cho backend xử lý
* UpdateDataFromDatabase(): chỉnh sửa data từ database

- Video:

* vid: ID của phim
* link: link lưu video phim
* image: link ảnh poster của phim
* type: thể loại phim
* ratting: đánh giá của phim
* comment: danh sách bình luận của phim
* review: Thông tin phim
* GetDataFromDatabase(): trả về toàn bộ dữ liệu của 1 user để cho backend xử lý
* UpdateDataFromDatabase(): chỉnh sửa data từ database

# Deployment

● Web Browser: Trình duyệt lướt web của người dùng

● Execution environment:

● Netfliz Server: ứng dụng được triển khai trên môi trường server phù hợp cho phép nhận và trả request thông qua biên folder **server** đã bao gồm **models, controller**,.. các tài nguyên để thực hiện giao tiếp với Client

● Netfliz Client: ứng dụng sẽ được deploy trên môi trường phù hợp, cho phép liên kết với server.

● Dữ liệu người dùng tạo ra, có sẵn trong chương trình sẽ được lưu trữ trên Cloud thông qua kết nối Internet. Do đó, thiết bị sử dụng yêu cầu luôn có kết nối mạng ổn định.

○ User: lưu trữ thông tin người dùng (họ tên, tài khoản, danh sách phim yêu thích, …)

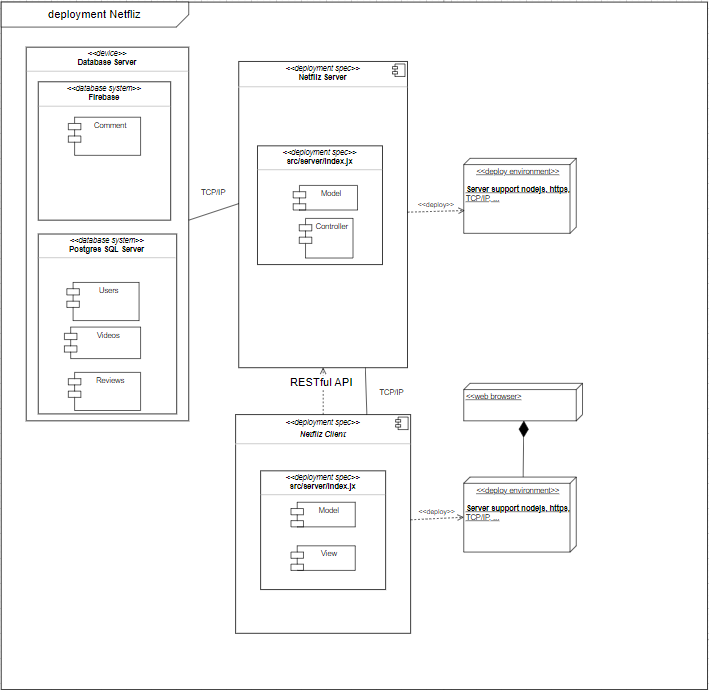
○ Videos: lưu trữ thông tin phim

○ Comment: lưu trữ bình luận của phim

○ Reviews: lưu trữ thông tin thêm của phim

● DatabaseServer: lưu trữ cơ sở dữ liệu người dùng, phim, comment và reviews. Cơ sở dữ liệu được sử dụng trong ứng dụng là SQL cho Video, Reviews, Users với dịch vụ hỗ trợ là Postgres SQL, và NoSQL cho Comment với dịch vụ hỗ trợ Firebase.

**Deployment diagram:**



# Implementation View

