

---

**Group 03**

---

**Movie Reviews and Ratings  
Software Architecture Document**

**Version <1.1>**

Movie Reviews And Ratings	Version: 1.1
Software Architecture Document	Date: 11/12/2021
None	

## Revision History

Date	Version	Description	Author
15/11/2021	1.0	Architecture goals and constraints Use-case model	Bui Huynh Trung Nam
21/11/2021	1.0	Introduction (Purpose, scope, References, overview)	Nguyen Phan Vu
24/11/2021	1.0	Components	Ngo Truong Tuyen
27/11/2021	1.0	Class diagram	Hoang Huu Giap
10/12/2021	1.1	Deployment, implementation view	Bui Huynh Trung Nam

Movie Reviews And Ratings	Version: 1.1
Software Architecture Document	Date: 11/12/2021
None	

## Table of Contents

<b>Introduction</b>	<b>4</b>
1.1. Purpose	4
1.2. Scope	4
1.3. Definitions, Acronyms, and Abbreviations	4
1.4. References	4
1.5. Overview	4
<b>Architectural Goals and Constraints</b>	<b>5</b>
2.1. Software requirement and objectives have significant impact on architecture:	5
2.2. Goals	5
2.3. Constraints	5
<b>Use-Case Model</b>	<b>6</b>
<b>Logical View</b>	<b>7</b>
Class diagram:	7
Class diagram description:	8
4.1. Component: Authentication UI	10
4.2. Component: Home UI	10
4.3. Component: Search Results UI	11
4.4. Component: Single Review UI	11
4.5. Component: List of Reviews UI	11
4.6. Component: About Us UI	11
4.7. Component: Contact Us UI	11
4.8. Component: User's Profile UI	11
4.9. Component: Favorites List UI	11
4.10. Component: History Activity UI	11
4.11. Component: Admin Management UI	12
4.12. Component: User Management UI	12
4.13. Component: Review Management UI	12
<b>Deployment</b>	<b>12</b>
<b>Implementation View</b>	<b>13</b>

Movie Reviews And Ratings	Version: 1.1
Software Architecture Document	Date: 11/12/2021
None	

# Software Architecture Document

## 1. Introduction

### 1.1. Purpose

- The purpose of this document is to provide a detailed architectural design of our website, it is intended to capture and communicate the important architectural decisions that have been made on the system, helping the Software developers understand the reasoning behind architecture choice decisions.
- This document will address the background for this project, and the architecturally significant functional requirements. Additionally, it helps the development team to determine how the system will be structured at the highest level. The leader can use this document to validate that the development team is meeting the agreed upon requirements during his evaluation of the team's efforts
- Simplify maintenance and product upgrades, by documenting key architectural decisions that have far-reaching effects on the system
- By modeling, it's easy to compare performance between different design architectures, so it's easy to choose the best one, even support future architecture reuse.

### 1.2. Scope

- This document presents the technical architecture of our product. It refers to the rules that developers, system architects need to follow in the process of developing, maintaining, and upgrading the system.
- Includes a lot of useful information for development orientation, future project planning. Specifically, the constraints and limitations of the architecture. Additionally, the readers will find information about the frameworks, tools and technologies used by the website.
- However, this is only the first version of it and there are many shortcomings, so it is highly encouraged at least one future revision of the document to correct inaccuracies, extend it and insert additional feedback and decisions that were made in the past.

### 1.3. Definitions, Acronyms, and Abbreviations

SAD: software architecture document

All name of methods in class diagram will use camel case convention

### 1.4. References

- [https://joinup.ec.europa.eu/sites/default/files/document/2014-11/e-TrustEx\\_Software\\_Architecture\\_Document.pdf](https://joinup.ec.europa.eu/sites/default/files/document/2014-11/e-TrustEx_Software_Architecture_Document.pdf)
- [Software Architecture - Khác biệt giữa kiến trúc và thiết kế \(viblo.asia\)](#)
- [SAD - Tài liệu kiến trúc phần mềm của các bậc "nhân giả" \(spiderum.com\)](#)
- <https://upcommons.upc.edu/bitstream/handle/2099.1/18373/90629.pdf>
- <http://www.se.rit.edu/~co-operators/SoftwareArchitectureDocumentation.pdf>
- [https://www.its.dot.gov/research\\_archives/msaa/pdf/MSAA\\_SystemDesignFINAL.pdf](https://www.its.dot.gov/research_archives/msaa/pdf/MSAA_SystemDesignFINAL.pdf)
- <https://shekhargulati.com/2021/04/24/software-architecture-document-template>
- <https://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm>
- <http://www.apexglobal.com.vn/vi/tam-quan-trong-cua-software-architecture-document/>

### 1.5. Overview

- After reading this document, you will know the work schedule of the development team, understand more about not only how the product works, but also how it is structured and works for each audience. used

Movie Reviews And Ratings	Version: 1.1
Software Architecture Document	Date: 11/12/2021
None	

through observing the architectural models of the system

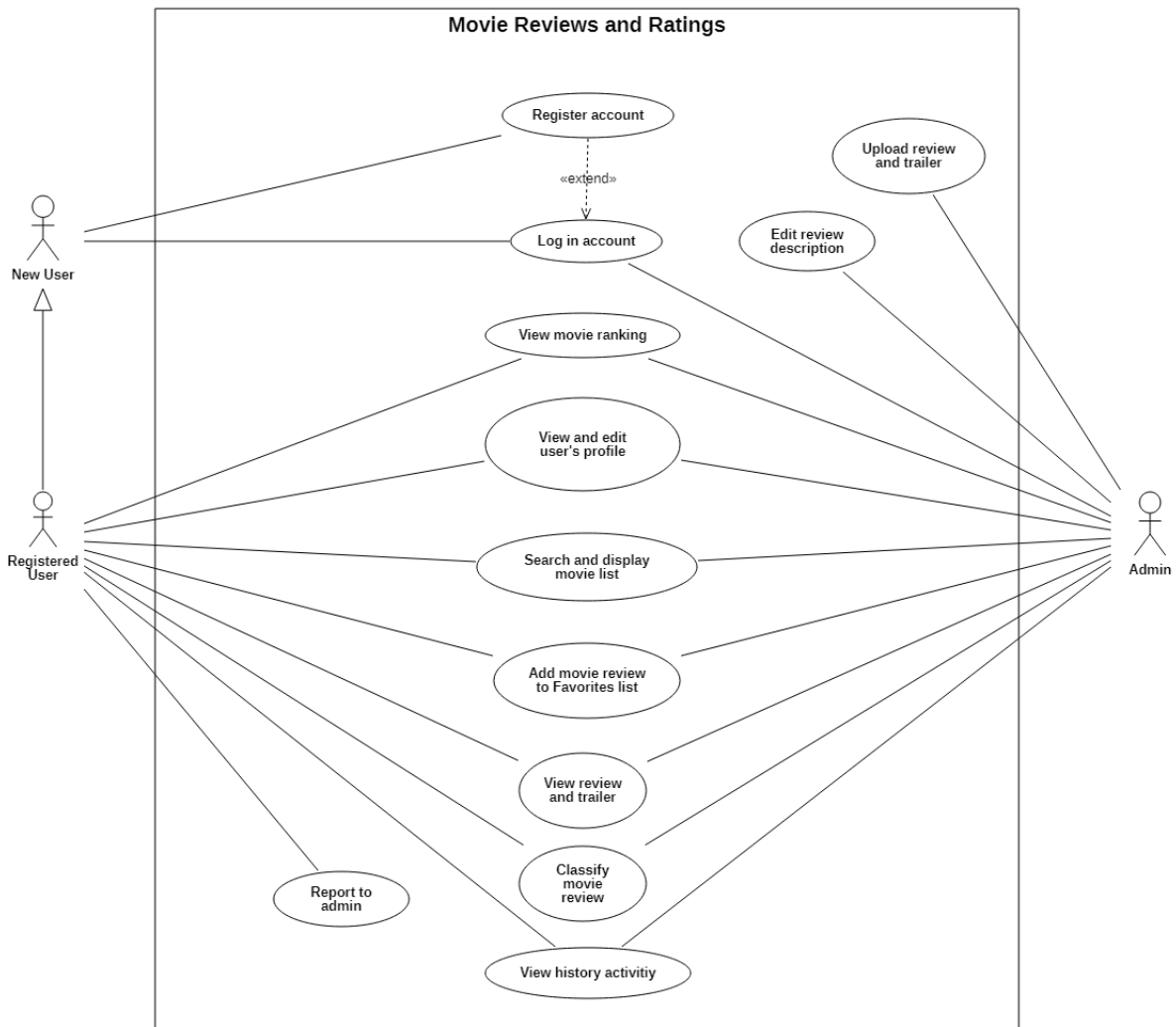
## 2. Architectural Goals and Constraints

- **2.1. Software requirement and objectives have significant impact on architecture:**
  - Security - privacy: Obviously, in this project we will use account to login and experience several features → to enhance satisfaction of user, security and privacy have a big impact on our project
  - Maintainability: We need and create a system that can be easy to fix, find and debug → definitely we need a system with a clear architecture.
  - Performance: We try to create a system that can be used by many users at the same time or we can interact real-time on it → Performance is priority on our architecture.
- **2.2. Goals**
  - Developer team:
    - Improve knowledge of developers on architecture
    - Know how to create a architecture when having requirements
    - Understand relationship between components in architecture

⇒ The way create, improve, modified architecture is reasonable and more effective
  - User: with a effective architecture:
    - Enhance user's experience
    - Enhance interact between users
    - Provide an environment to relax
- **2.3. Constraints**
  - Type of project: web application
  - Environment: popular browsers nowadays such as google chrome, edge, cốc cốc,...
  - Front-end language: HTML, CSS, Ajax (will update in working process)
  - Back-end language: JavaScript, NodeJs, ExpressJs (will update in working process)
  - Database: Firebase Realtime Database (will update in working process)
  - For user: need a smartphone or computer, stable internet
  - The way team working: Agile, separate work into many sprint, each sprint take 2 weeks to finish, meeting/discussion frequently
  - Schedule: [Group03\\_Gantt Chart](#)
  - Only use free source code

Movie Reviews And Ratings	Version: 1.1
Software Architecture Document	Date: 11/12/2021
None	

### 3. Use-Case Model



Movie Reviews And Ratings	Version: 1.1
Software Architecture Document	Date: 11/12/2021
None	

## 4. Logical View

### Class diagram:



Movie Reviews And Ratings	Version: 1.1
Software Architecture Document	Date: 11/12/2021
None	

○ **Class diagram description:**

- **Class RegisteredUser:** this class is used to store basic information of a user like name, phone number, address... And it provides a lot of functions to make the user's experience better.
  - getUserID: get the user id of this user
  - getUserPassword: get this user's password from database
  - getUserEmail: get user's email from database
  - getUsername: get user's full name from database
  - getUserTelephone: get user's phone number from database
  - getUserDOB: get user's date of birth from database
  - getUserAddress: get user's address from database
  - getUserGender: get user's gender from database like mail, female, others
  - setUserPassword: set this user's password to database
  - setUserEmail: set user's email to database
  - setUsername: set user's full name to database
  - setUserTelephone: set user's phone number to database
  - setUserDOB: set user's date of birth to database
  - setUserAddress: set user's address to database
  - setUserGender: set user's gender to database.
  - login and logout methods is used when user login and logout in website.
  - writeComments: users can write comments in any movie reviews.
- **Class NormalUser:** this class provides methods for any registered user but not an admin, NormalUser is managed by admin
  - accountStatus (attribute): use to show the account status of this user, false is banned and true is active
  - reportToAdmin (method): use to contact the admin if this user needs any help during use of our website.
  - getAccountStatus (method): use to get the account status of this user
  - viewMovie (method): allow user to see our movie reviews
  - viewFavoriteList (method): every logged user will have one favorite list, and they can add any movie to this list, in order to see later.
- **Class Admin:** this class provides methods for any admin account
  - addMovie (method): allow admin to add new movie review to the system
  - removeMovie (method): admin can delete any movie review from system
  - updateMovie (method): admin can update any information of movie review
  - changeAccountStatus (method): admin can ban or activate a normal user's account and confirm registration from the unregistered users.
  - deleteUser (method): admin can delete a user permanently.
  - updateUserInfo (method): admin can update user's information
  - viewUserInfo (method): admin can see any information of registered users, except password (hashed).
- **Class UnregisteredUser:** this class provides methods for a new user to create an account.
  - getRegistered (method): use for user registration in order to create an account

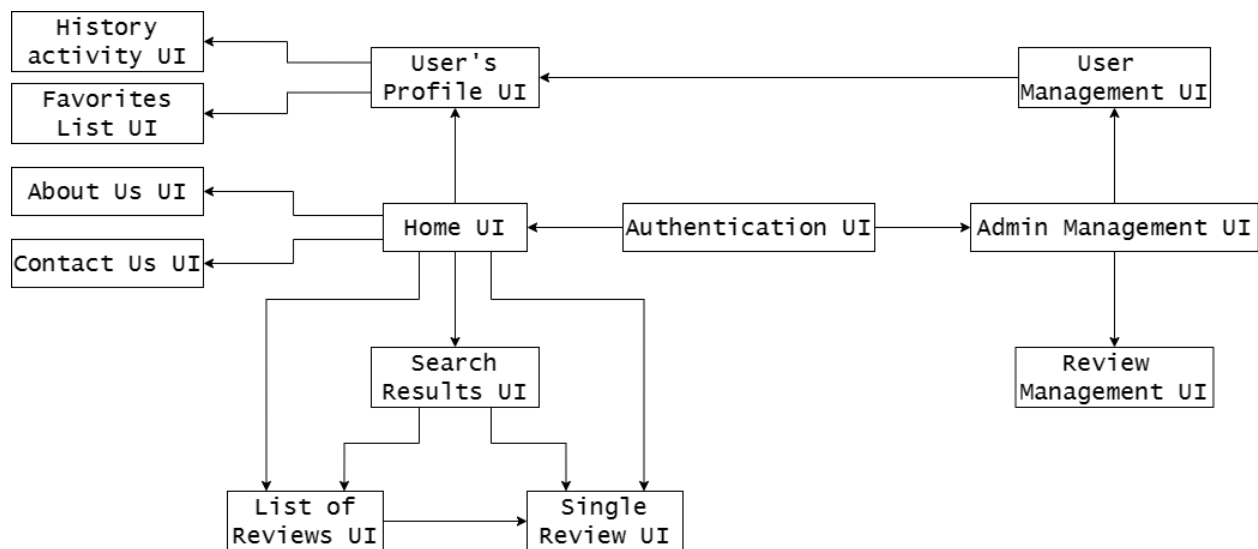


Movie Reviews And Ratings	Version: 1.1
Software Architecture Document	Date: 11/12/2021
None	

- **Class Movie:** this class is used to store movie information and some methods
  - movieID (attribute): store movie ID
  - name (attribute): store movie's name
  - overview (attribute): store the overview information of this movie review
  - genreID (attribute): store the ID of the genre or category of this movie review
  - director (attribute): store movie's directors
  - duration (attribute): store movie's duration like 2 hours or more
  - setMovieID (method): set an ID for movie
  - setMovieName (method): set name for movie
  - setMovieOverview (method): set overview information for movie
  - setMovieType (method): set genreID for movie
  - setMovieDirector (method): set directors for movie
  - setMovieDuration (method): set movie's duration time for movie
  - getMovieID (method): get the movie's ID
  - getMovieName (method): get the movie's name
  - getMovieOverview (method): get the movie's overview string
  - getMovieType (method): get the genre ID of this movie
  - getMovieDirector (method): get all directors of movie
  - getMovieDuration (method): get the duration of this movie
  - updateDetails (method): update information for one specific movie
  
- **Class FavoriteList:** any normal user has 1 favorite list that store all the movie review that user added
  - addedTime (attribute): store the time that one movie is added to this list
  - addNewMovie (method): add new movie review to list
  - isExists (method): check if this movie exists in the list or not, if no user can add to the list and otherwise, the user can't.
  - removeMovie (method): remove a movie review from the favorite list.
  
- **Class Review:** store review detail for the class Movie
  - reviewID (attribute): store review ID
  - rating (attribute): store the rating of this movie from 0 to 5 stars.
  - description (attribute): the description of this movie review
  - setReviewID (method): set an ID for this review
  - setReviewDescription (method): input review description
  - setRating (method): rate for the movie review from 0 to 5 stars
  - getReviewID (method): get the ID of one specific movie
  - getReviewRating (method): get the ratings of this movie
  - getReviewDescription (method): get description of movie
  - sortMovieByRating (method): sort all the movie by rating (star) in ascending order
  
- **Class Genre:** store movie review's categories information
  - genreID (attribute): ID of this category
  - genreName (attribute): the name of this categories
  - sortByName (method): use to sort all movie in this category by name
  - getCategoryName (method): get the name of this category
  - findMovieReview (method): find a movie in the category
  - isExists (method): check if one movie is in this category or not
  - getNumberOfMovie (method): get the total of movie in this category

Movie Reviews And Ratings	Version: 1.1
Software Architecture Document	Date: 11/12/2021
None	

- addNewMovie (method): add new movie to category
- removeMovieReview (method): delete a movie review from genre
- **Class Comment:** any registered user can comment, this class use to store all comment of users
  - commentID (attribute): ID of this comment
  - content (attribute): the summary or the anything user want to write to this movie review
  - userID (attribute): the userID of the user who writes this comment
  - movieID (attribute): the movieID that the user comment in
  - setComment (method): make comment on any movie review
  - deleteComment (method): user who can delete their comments
- **Class Likes:** to keep track of the user's reaction to the movie review
  - userID (attribute): the userID of the user who react (like or dislike)
  - movieID (attribute): the movieID that the user react
  - like (attribute): to show that the user likes (true) or dislikes (false) this movie review.



- **4.1. Component: Authentication UI**
  - Responsibility: Display the UI for users to authenticate (including registration and login).
  - Pre-page: No.
  - Post-page: Home UI, Admin Management UI.
- **4.2. Component: Home UI**
  - Responsibility: Display the overview of website, including menu bar, search bar, list of reviews,
  - Pre-page: Authentication UI.
  - Post-page: User's Profile UI, About Us UI, Contact Us UI, Search Results UI, List of Reviews UI, Single Review UI.

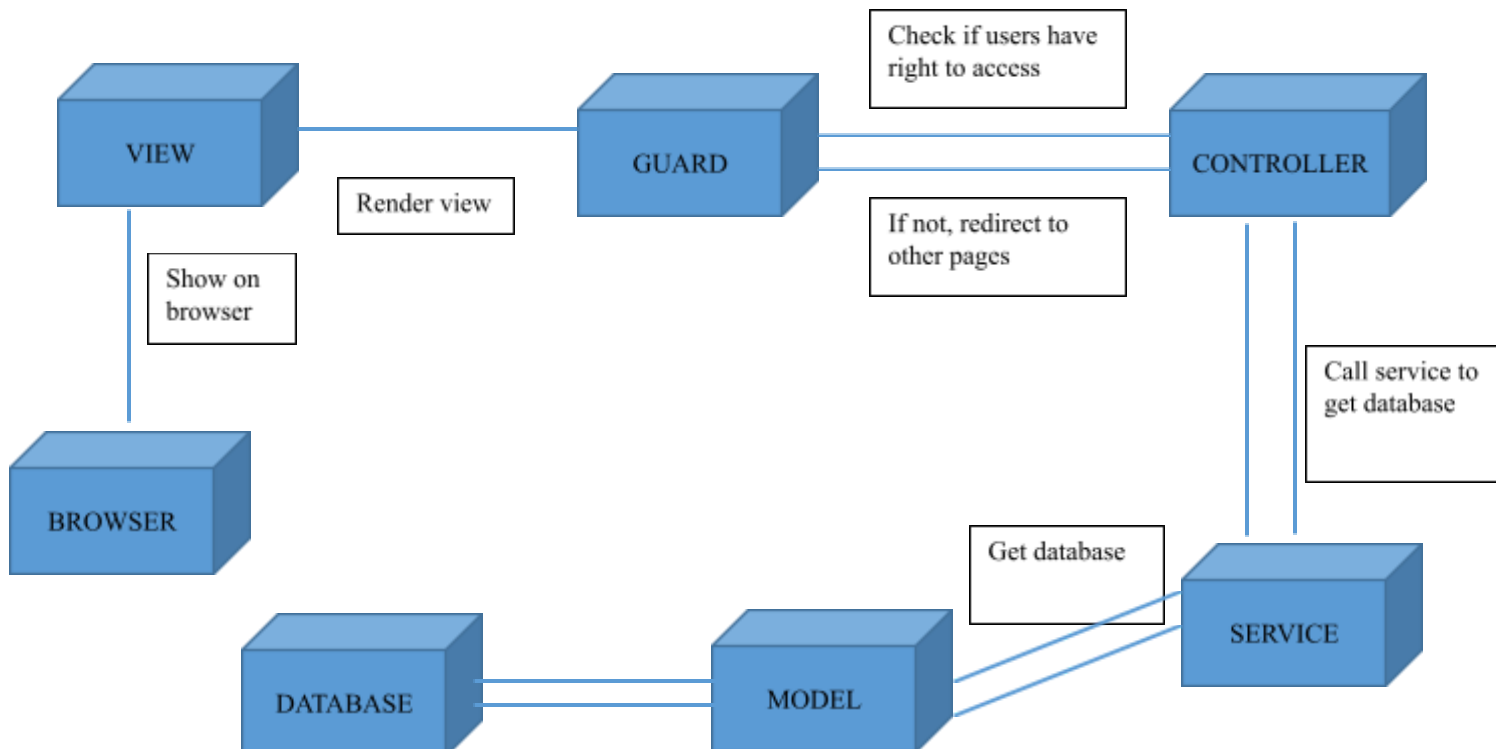
Movie Reviews And Ratings	Version: 1.1
Software Architecture Document	Date: 11/12/2021
None	

- **4.3. Component: Search Results UI**
  - Responsibility: Display the results of searching, they are reviews which the user wants to see.
  - Pre-page: Home UI.
  - Post-page: Single Review UI, List of Reviews UI..
- **4.4. Component: Single Review UI**
  - Responsibility: Display the content of a review, including title, trailer, content, comment, etc.
  - Pre-page: Home UI, Search Results UI, List of Reviews UI.
  - Post-page: No.
- **4.5. Component: List of Reviews UI**
  - Responsibility: Display the list of reviews which are the same kind or not.
  - Pre-page: Home UI, Search Results UI.
  - Post-page: Single Review UI.
- **4.6. Component: About Us UI**
  - Responsibility: All information of the owner of the website will be shown here.
  - Pre-page: Home UI.
  - Post-page: No.
- **4.7. Component: Contact Us UI**
  - Responsibility: This page shows why a user should contact you and describe how you can help to solve your user's problems. They include an email and phone number so that the users can quickly find the required information. Furthermore, a short form that uses fill will help we understand who's contacting us.
  - Pre-page: Home UI.
  - Post-page: No.
- **4.8. Component: User's Profile UI**
  - Responsibility: Display the basic information of the user such as username, name, age, email, etc.
  - Pre-page: Home UI, User Management UI.
  - Post-page: Favorites List UI, History Activity UI.
- **4.9. Component: Favorites List UI**
  - Responsibility: This page will store the list of reviews which users like.
  - Pre-page: User's Profile UI.
  - Post-page: No.
- **4.10. Component: History Activity UI**
  - Responsibility: All activities of the user, such as seeing a movie review, etc., will be stored and displayed in this page.
  - Pre-page: Home UI.

Movie Reviews And Ratings	Version: 1.1
Software Architecture Document	Date: 11/12/2021
None	

- Post-page: No.
- **4.11. Component: Admin Management UI**
  - Responsibility: As an administrator, we can control the settings, including posting permissions, privacy, etc.
  - Pre-page: Authentication UI.
  - Post-page: User Management UI, Review Management UI.
- **4.12. Component: User Management UI**
  - Responsibility: This page will be able to see all users and as an administrator, we can delete any user or edit their information easily.
  - Pre-page: Admin Management UI.
  - Post-page: Review Management UI..
- **4.13. Component: Review Management UI**
  - Responsibility: This page will be able to see all reviews. We can control some setting, such as posting, editing, deleting.
  - Pre-page: Admin Management UI.
  - Post-page: No.

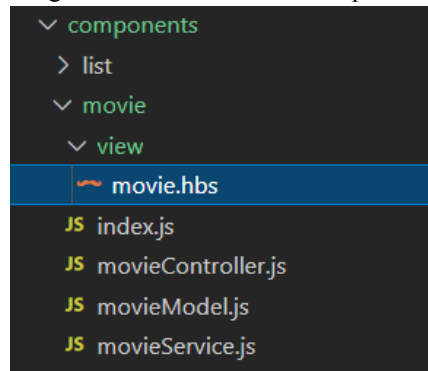
## 5. Deployment



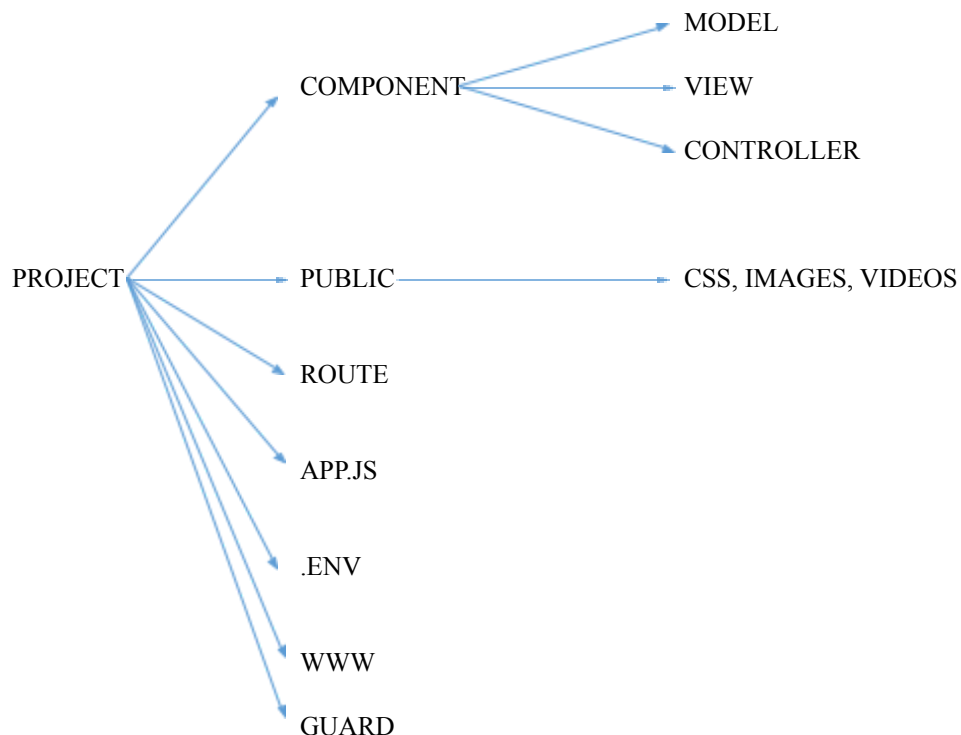
Movie Reviews And Ratings	Version: 1.1
Software Architecture Document	Date: 11/12/2021
None	

## 6. Implementation View

We will show a structure folder for the main file/folder. As above, we use MVC to operate system, so we will categorize each feature into component We will have many components, and in each component is MVC:



Our system will have many middle wares in file app.js to call corresponding components. Public folder will contain a source of image, video, css,.. for components. Route will direct to linked page



Movie Reviews And Ratings	Version: 1.1
Software Architecture Document	Date: 11/12/2021
None	

