 

| **FPT-APTECH COMPUTER EDUCATION** |
| --- |
| eProject Document |
| [Project name] |
|  |
| | **<Group Name>** | | | --- | --- | | **Group Member** | Nghiêm Tiến Anh TH2208053  Đinh Doanh Việt TH2209073  Trịnh Đình Quốc TH2209208 | | **Instructor** | Nguyễn Tuân | | **Batch** | FPT Aptech Hanoi | | **Semester** | 4 | |
|  |

| - Hanoi, <month>/<year> - |
| --- |

<This is the template document. Replace any text in <> with your text. When you are done, there should be no <> or text surrounded by <> in this document. Remember, this purpose of this document is to be useful: not just work to get a grade.>

*<The Table of Contents goes here>*

# Introduction

*This document provides an overview of the movie website development project, including the organization of the document, the main contents and the structure of the system. The project is designed to meet the needs of online entertainment, allowing users to watch movies, TV shows and other video content from many genres and countries.*

# Problem Definition

## Problem Abstraction

*Nowadays, users' demand for online entertainment is increasing, especially watching movies and TV shows at home. However, existing websites are often limited in terms of content, streaming quality, and user experience. This project aims to develop a movie website that is capable of providing rich, high-quality content and a friendly interface to meet users' entertainment needs.*

## The Current System

*The current system mainly consists of online movie websites with basic functions such as movie searching, streaming, and downloading. However, many websites face limitations in bandwidth, video quality, and unfriendly user experience. Some sites do not have appropriate movie recommendation functions or detailed movie classification, making it difficult for users to find their favorite content.*

## The Proposed System

*The proposed system is an online movie website with modern features, supporting high-quality movie viewing, integrating movie suggestions based on users' viewing history, and classifying movies by genre, country, and year of production. This system will have a friendly, easy-to-use interface, along with the ability to expand to accommodate a large number of users accessing at the same time.*

## Boundaries of the System

*This movie website system will provide basic functions for users to search and watch their favorite movies online. However, the current scope of the system does not include some advanced functions such as checking viewing history, selecting image quality, subtitles, and movie trailers. These features may be added in later versions of the system. Currently, the website focuses on providing a seamless movie viewing experience with the ability to search for movies by genre, country, and year of release.*

## Development Environment

*The movie website system is built with the following technologies:*

* *Frontend: React is used for the user interface, along with TailwindCSS for styling, creating a modern and user-friendly experience.*
* *Backend:*
* *Java: Used to call APIs and handle some business logic within the system.*
* *Spring Boot: Manages the payment section, ensuring security and stability in online transactions.*
* *NodeJS: Serves as the main backend, managing requests from the frontend and coordinating data across different parts of the system.*
* *Database: MongoDB is used to store user information, viewing history, and data related to movies.*
* *Authentication & Hosting: Firebase provides user authentication and storage solutions, as well as quick deployment capabilities.*
* *API Testing: Postman is used to test APIs, ensuring that endpoints function accurately and efficiently.*

*Hardware Requirements:*

* *Server: Minimum specifications include a 4-core CPU, 8GB RAM, and a 100GB SSD for processing and storage needs.*
* *Development Workstation: A personal computer with similar specifications to allow developers to code, test, and deploy efficiently.*

*This development environment is designed to meet performance and scalability requirements, ensuring a smooth and secure experience for users on the movie website.*

# Requirements and Business Flow

*<In this section you will provide the requirements you developed against. This section should be useful, so only provide enough information (requirements, diagrams) to help with the implementation of the software. Focus on the “non-obvious” aspects. If you are dealing with a business process, activity diagrams are good. Making a set of requirements that is compact, simple and covers all the key aspects of the software can. Don’t do the work if you don’t need it!>*

## Customer Requirement Specification

*<Summarize the customer requirement into the compact form>*

## Activity Diagram

*<Business processes should be modeled carefully. Use activity diagrams to show important businesses. Focus on non-trivial ones>*

## Use Case Diagram

*<Put here the overall use case diagram. If the system can be partitioned into several subsystems, you can use multiple diagrams to show the overall functionalities of the system>*

## Use Case Specification

*<Write down all non-trivial use cases. This should reflect what you get when your team does the system analysis. Use the template to write the detailed specification for use cases>*

*<Use case template:*

| **USE CASE SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | <UC001> | **Use-case Version** | | <1.0> |
| **Use-case Name** | <Name> | | | |
| **Author** | <Members> | | | |
| **Date** | Dd/mm/yyyy | **Priority** | <High\Normal\Low> | |
| **Actor:**  <Lit all actors>  **Summary:**  *<Briefly describe the use case>*  **Goal:**  *<Briefly describe the goal of use case>*  **Triggers**  *<What leads this use case?>*  **Preconditions:**  *<List the required pre-conditions for this use case>*  **Post Conditions:**  *<List the required post-condition for this use case>*  **Main Success Scenario:**  *<List the main steps for this use case to reach the goal successfully >*    **Alternative Scenario:**  *<List the other steps for this use case to reach the goal in some alternatives condition >*  **Exceptions:**  *<list the exceptions of this use case>*  **Relationships:**  *<List the relationships that use case relates to>*  **Business Rules:**  *<Any concern about the business>* | | | | |

## Other Concerns<Optional>

*<You can list here all other concerns about the business or the requirements if needed>*

# Design

*[This section shows the design of the system. This could be a part of the Developers Manual]*

## System Architecture

*<Explain and present the architecture of the system using texts or diagrams>*

## Class Diagram

*<Provide class diagrams for the project>*

## Class Diagram Explanation

*<Provide brief explanation about the class diagram above. You do not need to explain “obvious” parts of your class diagram. For example, I know what a “Login” class is. Don’t say “The login class was created to store login information.”>*

## Sequence Diagram (Optional)

*<for important and complex interactions, protocols or algorithms, sequence diagrams should be drawn for clearing the details and supporting the system implementation. This section is optional>*

## Collaboration Diagram (Optional)

*<for important and complex interactions, collaboration diagrams should be drawn for clearing the details and supporting the system implementation. This section is optional>*

## State Diagram (Optional)

*<put all state diagrams here>*

## Entity Relationship Diagram

*<Provide the ERD Diagram for the system here. If your team uses file or in-memory storage facility instead of database, replace this section by ‘Data Structures’>*

## Database Design

*<Provide the detailed database design for the system here. If your team uses file or in-memory storage facility instead of database, remove this section, use the ‘Data Structures’ section >*

## Algorithms (optional)

*<Provide the detailed description about algorithms used in the system. You can use Flow Chart or Activity Diagram to represent algorithms. Focus on the important and complex algorithms>*

## Others (optional)

*<Any design concerns or diagrams can be put here>*

# System Prototype

*<Put the system prototype or mock UI here. Focus on* ***important forms*** *and the* ***screen flows*** *between forms. If you use RAD and .NET, prototyping really help you to reach your development goals quickly>*

# Management and Project Planning

## Management Approach

*< Briefly describe the management approach that your team selects. Is your team self-managed or managed by one leader? How do you assign tasks to team members? How often do you meet? What do you do during meetings? Etc.>*

## Project Plan

*<The detailed project plan is put here. You can use WBS Excel sheet, Sprint Backlog (see Sprint Backlog\_Template.xls), Task sheet, Gantt chart, etc. to present your team’s plan. You can capture the Gantt chart in PMS if you use it to plan your project>*

## Task Sheet

*<Write down the tasks in Task Sheet-compatible format, this Task Sheet works as the activity report of the project or the plan of the project (not recommended); see eProject Guide for detailed Task Sheet>*

## Meeting Minutes (Optional)

*<Put all minutes of your team meetings here>*

# Checklists

## Check List of Validation

*< Put the checklist here; describe how it is used and the resulted checklist>*

## Submission Checklist

*< Put the checklist here; describe how it is used and the resulted checklist>*

# Screenshots

*<Capture some intuitive and main screens of the software and put them here>*

# Coding Convention

*<Provide the coding convention for your team. If you simply want to use the existing code standard(s) such as ‘Java Code Convention’, you can refer to it\them by name or URL>*

# Other Concerns<Optional>

*<If you have any other information you want to add to this document, place it here. This could include thoughts on the eProject, improvements, etc.>*

# Appendix

## Glossary [Optional]

*<Place all definitions or abbreviation used in this document >*

## References [Optional]

*<Place all referenced materials used in this document >*

## Others<Optional>