<Company Name>
<Company Name>

<MovieHub>
Vision

Version <1.0>

Revision History

Date	Version	Description	Author
27/05/2025	1.0	Initial version of the Vision document	Moga Diana
27/05/2025	1.1	Completed Functional & Non-functional Requirements sections	Moga Diana
28/05/2025	1.2	Added User Environment and Introduction sections	Moga Diana

Table of Contents

Vision

1. Introduction

The purpose of this document is to collect, analyze, and define the high-level features and objectives of the **MovieHub** application. It focuses on the core capabilities required by both administrative and general users, and the reasons these needs exist. The document outlines how MovieHub addresses these requirements and serves as a foundation for the use-case and supplementary specifications that will follow.

This Vision document provides a structured overview of the application, including its purpose, scope, terminology, references, and organization. It helps align all stakeholders on the system's goals and defines the context in which the product will be developed and used.

1.1 Purpose

The purpose of this Vision document is to define the high-level features and objectives of the MovieHub application. This document outlines the needs of users and administrators, the system's functionality, and the benefits of its implementation.

1.2 Scope

MovieHub is a movie management and discovery application that enables administrators to manage data on movies, actors, and directors, while allowing users to browse content, leave reviews, and track movies via a personalized watchlist. The application supports complex data relationships and ensures security, usability, and scalability.

1.3 Definitions, Acronyms, and Abbreviations

- **CRUD** – Create, Read, Update, Delete
- **JWT** – JSON Web Token
- **ORM** – Object Relational Mapping
- **PgAdmin** – PostgreSQL database administration tool
- **UI** – User Interface

1.4 References

- Assignment Specification (2025)
- Spring Boot and Hibernate Documentation
- PostgreSQL Reference Manual
- REST API Design Best Practices

1.5 Overview

This document outlines the motivation behind the MovieHub application, its intended stakeholders and users, as well as the high-level requirements that define the system's architecture and functionality.

2. Positioning

2.1 Problem Statement

The problem of	*of manually managing and tracking film data and user engagement*
affects	*movie database administrators and end users*
the impact of which is	*inefficiency, data inconsistency, and lack of personalized experiences*
a successful solution would be	*an integrated platform that supports movie cataloging, relationship mapping, user engagement, and watchlist-based notifications.*

2.2 Product Position Statement

For	*users*
Who	*want to explore and track movies, and administrators who need to manage film-related data,*
The MovieHub application	*is a movie management and discovery platform*
That	*provides a centralized interface for CRUD operations, user reviews, and personalized notifications.*
Unlike	*simple review or listing websites,*
Our product	*integrates user watchlists, automatic notifications, and full administrative control with a modern, secure backend and user-friendly UI.*

3. Stakeholder and User Descriptions

3.1 Stakeholder Summary

Name	**Description**	**Responsibilities**
Admin	*Authorized staff maintaining movie database*	*Manage movies, actors, directors; perform CRUD operations; ensure data accuracy*
Developer	*System architects and coders*	*Build and maintain backend, API, frontend, and database*
QA	*Test engineerirs*	*Ensure application quality and performance via testing*
Product Owner	*Oversees project direction*	Approve features, monitor progress, align development with business goals

3.2 User Summary

Name	**Description**	**Responsibilities**	**Stakeholder**
Registered User	*General movie app user*	*Register, log in, browse content, write reviews, manage watchlist*	*Product Owner*
Guest Visitor	*Non-registered visitor*	*Need to register*	*Product Owner*

3.3 User Environment

Users will access the MovieHub application through modern desktop. The frontend is developed using **Vite**, providing a fast and efficient user experience. Administrators will access a dedicated admin interface within the same application.

Currently, the backend and PostgreSQL database are hosted **locally** during development. Email notification functionality will be send on email. Future deployment may involve migrating to a cloud platform for production use.

4. Product Requirements

- **Platform Requirements**: Web-based; responsive for desktop and mobile.
- **Database**: PostgreSQL with ORM (Hibernate).
- **Security**: JWT-based authentication; role-based access control.
- **Performance**: Scalable for thousands of users with minimal latency.
- **UI/UX**: Modern and intuitive interface with admin dashboard and user features.
- **Notification System**: Email alerts via SMTP when watched movies are released.
- **API**: RESTful, secure, and modular.