Project Plan

MoviMingle

Contents

[Project Overview 3](#_Toc159844264)

[Technologies, Methods, and Tools 3](#_Toc159844265)

[Learning Outcomes Focus 3](#_Toc159844266)

[Development Strategy and Methodology 4](#_Toc159844267)

[Sprint Breakdown and Deliverables 4](#_Toc159844268)

[Sprint 1: Weeks 1-3 4](#_Toc159844269)

[Sprint 2: Weeks 4-6 4](#_Toc159844270)

[Sprint 3: Weeks 7-9 4](#_Toc159844271)

[Sprint 4: Weeks 10-12 4](#_Toc159844272)

[Sprint 5: Weeks 13-15 5](#_Toc159844273)

[Finalization Phase: Weeks 16-17 5](#_Toc159844274)

[Testing and Quality Assurance 5](#_Toc159844275)

[Documentation 5](#_Toc159844276)

[Submission Package 5](#_Toc159844277)

# Project Overview

* Objective: Develop an application to enable users to collaboratively select movies to watch by combining their preferences and incorporating a review system.
* Duration: 15 Weeks (5 Sprints of 3 Weeks Each), Final Deadline: June 23rd
* Developer: Solo Project by Kaloyan Preslavski
* Academic Submission: For the fulfillment of the specified learning outcomes in software engineering.

# Goal of the project

The goal of the project is to create a collaborative movie watching application for groups of people, focusing on the learning outcomes for this semester. Everything done will be documented and uploaded as evidence in my portfolio. The main objectives of the project are to ensure the functionality of the application for a large group of users, ranging from 1 million to more than 100 million users. Together with that I will need to keep the security in check.

# Technologies, Methods, and Tools

* Backend: Spring Boot. Because of its characteristics, such as having an embedded server support for load balancing, auto-configuration and the use of annotations I think it is the best choice in my case, since I have a limited time and I am familiar with the use of spring boot from previous semesters.
* Frontend: React. As one of the leading JavaScript libraries for building user interfaces, particularly single-page applications, It will play a not so important part of this project for the duration of this semester, since my goals are not oriented towards building the best looking interface, but just the basic functionalities. I have also chosen it partly because of my familiarity with the technology.
* Cloud Services: As I am in the initial phase of developing the software I have not decide on a cloud service provider, however, it will most likely be one of the suggested providers by Fontys (ASW, Google cloud, or Azure).
* Security: For the security, I will not be implementing my own, since it takes time and is very hard to get right. Since the idea of the semester is to focus on developing the important parts of the application, a good timesaver will be to implement OAuth2 with JWT for secure authentication.
* Documentation: MS Office

# Learning Outcomes Focus

* Scalable Architectures: Emphasis on backend scalability and adaptability for growing user base and data volume.
* DevOps Practices: Implementation of Continuous Integration/Continuous Deployment (CI/CD) pipelines and containerization.
* Cloud-Native Development: Utilization of cloud services for hosting, storage, and computation to ensure scalability and resilience.
* Security by Design: Integration of security practices from the outset, focusing on authentication, data integrity, and privacy.
* Distributed Data Management: Strategies for handling and storing large datasets efficiently while ensuring compliance with data protection regulations.

# Development Strategy and Methodology

* Approach: Agile development methodology, with work divided into five 3-week sprints, allowing for iterative design, development, and testing.
* Documentation: Comprehensive documentation of requirements, design decisions, code, testing procedures, and deployment instructions will be maintained throughout the project.

# Sprint Breakdown and Deliverables

## Sprint 1: Weeks 1-3

Activities: For the initial phase of the project development I will focus on gathering information about the learning outcomes and the technologies I plan on using. Then the initial research will be applied to the walking skeleton.

Deliverables: Project plan document, system architecture design, and initial backend codebase (walking skeleton).

## Sprint 2: Weeks 4-6

Activities: This sprint on focus will be the development of the functionality.

Deliverables: Backend functionality for managing favorite movies and user linking, initial cloud service integration.

## Sprint 3: Weeks 7-9

Activities: Next up is the implementation of the voting system, random movie selection logic, and starting security measures implementation.

Deliverables: Voting and movie selection backend functionalities, security strategy outline.

## Sprint 4: Weeks 10-12

Activities: In the final two sprints will be the time to focus on the frontend development for user interface, enhancement of security measures, and initiation of frontend-backend integration will be done.

Deliverables: Initial version of the frontend application, enhanced security features in the backend.

## Sprint 5: Weeks 13-15

Activities: This sprint I will focus on finalizing the development, conducting comprehensive testing (unit, integration, user acceptance), and compiling project documentation.

Deliverables: Final application (frontend and backend), complete project documentation, and testing reports.

## Finalization Phase: Weeks 16-17

Week 16: Deployment preparation, final adjustments based on feedback, and documentation finalization.

Week 17: Submission of final project code and comprehensive documentation by June 23rd.

# Testing and Quality Assurance

A multi-level testing strategy including unit tests for backend logic, integration tests for API endpoints, and user acceptance tests for the frontend interface.

Continuous testing throughout the development phases to ensure reliability and security.

# Documentation

Project Documentation: Detailed documentation covering system design, API documentation, deployment instructions, and user guide.

Progress Reporting: Bi-weekly progress reports to document challenges, solutions, and milestones achieved.

# Submission Package

Final Codebase: Including all source code for the backend and frontend, configuration files for CI/CD pipelines, and deployment scripts.

Comprehensive Documentation: System architecture, API documentation, user manual, and final project report summarizing the development process, learning outcomes achievement, and future work directions.