Table of Contents

Quality Assurance (QA) Plan

Esma Nur Akçınar

Alime Feyza Şahin

Merve İçkilli

Tarık Şamil Şahin

1. Quality Assurance Strategy

Overview:

The quality assurance strategy ensures that the Movie Desktop Application is reliable, secure, and performs efficiently. The QA process involves unit testing, integration testing, and usability testing to identify and resolve issues before deployment.

Testing Methodologies:

Unit Testing: Tests individual components using PyTest (e.g., database queries, user authentication).

Integration Testing: Ensures different modules (UI, database) work together.

Usability Testing: Evaluates user experience with feedback collection.

Automated vs. Manual Testing:

Areas Covered by Automated Testing

Unit tests: Core functionalities will be covered by automated unit tests

Database integration: Database interactions

Basic UI interactions: Form submissions, page navigation

Areas Covered by Manual Testing

Visual design and layout inspection

Exploratory testing

End-user scenarios

Usability testing

2. Quality Factors & Metrics

Quality Factor	**Description**	**Measurement Metric**
Performance	The application's response time when querying the database and updating the UI.	Average response time (ms): Measured by logging query execution times and UI update delays.
Security	Ensuring user authentication and data protection against unauthorized access.	Number of security vulnerabilities detected: Identified through input validation checks, database query security (e.g., preventing SQL injection), and error handling reviews.
Usability	The ease of navigation and interaction with the desktop application.	User satisfaction score: Gathered through user feedback and surveys after testing.
Maintainability	The ability to modify and extend the codebase efficiently.	Code complexity score: Measured using Cyclomatic Complexity and code linting tools.

3. Test Plan

Test Cases:

Test_Case_1: User enters valid login credentials. User redirected to the main page.

Test_Case_2: User enters incorrect login credentials. Error message is displayed.

Test_Case_3: Fetch top 100 movies from database. Movie list is retrieved successfully.

Test_Case_4: Add a movie to watchlist. Movie appears in the watchlist.

Test_Case_5: Search for a movie. Matching results are displayed.

Bug Tracking:

Bugs and issues will be tracked using GitHub Issues.

Bug Description: Identify the issue.

Steps to reproduce: Document the exact behavior and steps to reproduce it.

Severity level: Categorize the bug based on its severity and impact on the application.

Assigned Developer: Review the bug list and distribute the assignments among developers.

Task Matrix

Tasks	Esma Nur	Feyza	Merve	Tarık Şamil
Quality Assurance Strategy	• ☑	• □	• □	• ☑
Quality Factors & Metrics	• ☑	• ☑	• ☑	• □
Test Plan	• □	• ☑	• ☑	• ☑