RATING, REVIEW APPLICATION

A project report submitted in partial fulfillment of the requirements for the degree of

MASTER OF COMPUTER APPLICATION (MCA)
OF
TEZPUR UNIVERSITY

Submitted by
ALOK SHANDILYA, CSM22010

Guided by

Dr. Nabajyoti Medhi
Assistant Professor
Tezpur University

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
TEZPUR UNIVERSITY
TEZPUR 784028
ASSAM

Department of Computer Science and Engineering
Tezpur University

Certificate by the Examiner

This project report entitled ***Rating, Review Application*** is submitted by **Alok Shandilya** bearing Roll no: **CSM22010,** for partial fulfillment of the requirements and the regulations for the award of the degree of **Master's in Computer Application** in Computer Science & Engineering during the session 2022-2024 at Tezpur University has been examined.

Examiner:

Place: Tezpur
Date: ***19th December 2023***

Department of Computer Science and Engineering
Tezpur University

Certificate by the HOD

This is to certify that the project report entitled ***Rating, Review Application*** is submitted by **Alok Shandilya** bearing Roll no: **CSM22010.** He has completed her project work successfully as needed for partial fulfillment of the requirements and the regulations for the award of the degree of **Master's in Computer Application** in Computer Science & Engineering during the session 2022-2024 at Tezpur University. To the best of my knowledge, the matter embodied in this project has not been submitted to any other university/institute for the award of any Degree or Diploma.

Head of the Department
Department of Computer Science & Engineering
Tezpur University

Place: Tezpur
Date: ***19th December 2023***

Department of Computer Science and Engineering
Tezpur University

Certificate

This is to certify that the project report entitled ***Rating, Review Application*** submitted to the Department of Computer Science and Engineering, Tezpur University, in partial fulfillment for the award of the degree of **Master's in Computer Application** in Computer Science and Engineering, is a record of bonafide work carried out by **Alok Shandilya** bearing Roll no: **CSM22010** under my supervision and guidance. All help received by him from various sources has been duly acknowledged. No part of this report has been submitted elsewhere for award of any other degree.

Dr. Nabajyoti Medhi
Assistant Professor
Tezpur University

Place: Tezpur
Date: ***19th December 2023***

Acknowledgment

I would like to extend my heartful gratitude to my project guide **Dr. Nabajyoti Medhi**, Assistant Professor, Dept. Of CSE, Tezpur University, for giving us the opportunity to work under him and providing me ample guidance and support through the course of the project and for his helpful guidance as well as for providing necessary information regarding the project. I would like to express our sincere gratitude to **Dr. Tribikram Pradhan** for the continuous support, motivation, and guidance provided throughout the course of this project. Your insights and direction were truly invaluable.

I would also like to thank the Head of the Department of Computer Science and Engineering department and all the faculty members of the Dept. Of CSE, Tezpur University for the valuable guidance and cooperation throughout the project.

My thanks and appreciation also go to all other people who have directly or indirectly helped me out with their abilities.

ABSTRACT

This project presents the development of a movie rating and review application using Angular and Bootstrap for the frontend. The application is designed to offer users the ability to explore and provide feedback on a predefined set of movies. With a fixed list of movies and hardcoded login credentials, the focus is on the frontend implementation, showcasing Angular's versatility and Bootstrap's styling capabilities. Users can submit reviews along with 5-star ratings, and the application stores this information in the browser's local storage. While lacking a backend or database integration in this iteration, future enhancements are planned to extend the application's functionality.

In summary, this project demonstrates the utilization of modern web development technologies to create an intuitive and interactive user experience for movie enthusiasts. The combination of Angular and Bootstrap provides a solid foundation for frontend development, laying the groundwork for future backend integration and database management. The abstract encapsulates the essence of the application, emphasizing its current capabilities and the envisioned expansions to provide a comprehensive movie rating and review platform.

CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Project Objective

The overarching goal of this project is to *conceive, develop, and deploy an interactive movie rating and review application, leveraging the power of Angular and Bootstrap for front-end development*. At its core, the project aims to provide users with a seamless and enjoyable experience for rating and reviewing a predefined collection of movies. The application's central focus is on enabling users to express their opinions through intuitive interfaces, featuring a fixed list of movies and hard-coded login credentials for simplicity at this stage. The front-end, powered by Angular, is designed to be both dynamic and visually engaging, fostering an environment where users can easily explore, rate, and contribute reviews with a straightforward 5-star rating system. While back-end and database management functionalities are deferred to future phases, this project aims to establish a foundation for a comprehensive rating and review system that prioritizes user interaction and experience. By concentrating on the front-end intricacies and user engagement aspects, the objective is to lay the groundwork for subsequent development stages, ensuring that the application is poised for scalability and feature-rich expansions.

1.2 Background

In the digital age, user-generated content and community engagement have become integral aspects of online experiences. This project emerges from the growing need to facilitate user expression and community interaction within the realm of rating and reviews. Focusing on a broad spectrum of subjects, the application is designed to empower users to evaluate and critique various items, with an initial emphasis on movies.

As the project unfolds, the plan is to progressively introduce back-end functionalities and database management, expanding the application's versatility beyond its current boundaries. This approach aligns with the evolving landscape of online interactions, where user opinions hold significant value. By recognizing the dynamics of user-generated content and the changing expectations of online communities, this project endeavors to establish a foundation for a robust rating and review application that transcends specific content domains, laying the groundwork for an inclusive and engaging digital platform.

1.3 Problem Statement

The absence of a dedicated and versatile **Rating and Review Application** poses a significant challenge in the current digital landscape. Users lack a centralized platform where they can easily and comprehensively share their opinions and evaluations across various domains. The challenge is to develop an intuitive and user-friendly application that simplifies the process of submitting and accessing reviews, starting with a focus on movies. The goal is to create a dynamic and engaging platform that not only addresses the current void in the market but also sets the stage for future expansions into diverse content categories. The primary problem is the absence of a robust and adaptable rating and review application, hindering users from seamlessly expressing and exploring opinions within a unified digital space.

1.4 Aims and Objectives

The following are the aims focused by this project :

1. Develop a User-Centric Platform

Create a user-centric Rating and Review Application that prioritizes simplicity, accessibility, and an engaging user experience.

2. Facilitate Diverse Content Evaluation

Enable users to express opinions and reviews across a spectrum of content categories, starting with movies, and providing a foundation for expansion into additional domains.

3. Establish a Versatile Framework

Design a flexible and adaptable framework capable of accommodating evolving user preferences and content types as the application matures.

4. Encourage Community Interaction

Foster a sense of community and interaction by providing users with a platform to share, discuss, and engage with reviews and ratings from a diverse user base.

5. Lay the Foundation for Future Expansion

Build a solid foundation for the integration of back-end functionalities and database management to enhance scalability and prepare for the incorporation of additional features in subsequent phases.

The objective of this project is to develop a rating and review application.

1. Front-end Development

Develop an aesthetically pleasing and intuitive front-end interface using appropriate technologies.

2. User Authentication

Implement a secure user authentication system, initially with hard-coded credentials, for a seamless login experience.

3. Review Submission Process

Create a streamlined process for users to submit reviews, ***incorporating a 5-star rating system for evaluation.***

4. Local Storage Integration

Utilize browser local storage to store and retrieve user reviews and ratings without the need for immediate back-end infrastructure.

5. User Interaction Features

Implement features that encourage user interaction, such as the ability to like, comment on, and share reviews.

6. Scalability and Flexibility

Design the application with scalability and flexibility in mind to accommodate the addition of new content categories and features in the future.

7. Usability Testing

Conduct usability testing to gather feedback and make iterative improvements to enhance the overall user experience.

8. Documentation

Create comprehensive documentation detailing the application's architecture, functionalities, and potential areas for future development.

By achieving these aims and objectives, the project endeavors to create a robust and inclusive Rating and Review Application that caters to diverse user preferences and content categories, setting the ***stage for future growth and expansion.***

CHAPTER 2
FRONTEND DEVELOPMENT

2.1 Overview of Angular Framework Usage

Angular is a ***robust frontend framework*** that plays a pivotal role in developing the user interface of our movie rating and review application. Leveraging the power of **TypeScript**, Angular provides a *structured and scalable approach to building dynamic web applications*. With its ***modular architecture***, Angular facilitates the organization of code into ***reusable components***, making the development process more manageable and maintainable. The use of **two-way data binding** simplifies the s*ynchronization between the application's model and view*, ensuring a seamless and responsive user experience. Angular's *dependency injection* system enhances code modularity and testability, allowing for the efficient management of application components. Overall, the utilization of Angular empowers our project with a feature-rich and maintainable frontend, ensuring a smooth and interactive user interface.

2.2 Explanation of the User Interface and Design Choices

The user interface of our movie rating and review application is designed with a user-centric approach, aiming to provide an intuitive and visually appealing experience. *Angular's component-based architecture allows for the creation of modular UI elements, ensuring consistency and reusability throughout the application*. The design choices prioritize simplicity and clarity, with an uncluttered layout that guides users seamlessly through the content. User interactions are enhanced with responsive design principles, ensuring a consistent experience across various devices and screen sizes. *Bootstrap is employed for styling, offering a set of pre-designed components and a responsive grid system. This not only accelerates the development process but also contributes to a modern and aesthetically pleasing interface*. The color scheme and typography are chosen to align with the theme of movie ratings, creating a cohesive and engaging visual identity for the application. Overall, the user interface is meticulously crafted to enhance usability and deliver an enjoyable experience for users exploring and interacting with the content.

2.3 Implementation Details Using Bootstrap for Styling

In the implementation of our movie rating and review application, Bootstrap emerges as a key tool for achieving a polished and responsive user interface. Bootstrap is a front-end framework that streamlines the styling process with a **collection of pre-designed components and a flexible grid system**. These components, such as navigation bars, cards, and modals, are utilized to establish a consistent visual language across the application. The grid system ensures a responsive layout, adapting seamlessly to various screen sizes and devices.

For instance, Bootstrap's grid system aids in creating a responsive and organized movie catalog, allowing users to explore content intuitively. The application's rating and review submission forms are styled using Bootstrap's form components, providing a clean and user-friendly interface for feedback. Additionally, the integration of Bootstrap's utility classes facilitates quick and effective styling adjustments.

The decision to leverage Bootstrap aligns with the project's goal of efficient and visually cohesive frontend development. It not only expedites the styling process but also ensures a modern and professional aesthetic. By harnessing Bootstrap's styling capabilities, our application achieves a harmonious balance between functionality and design, enhancing the overall user experience.

CHAPTER 3
FEASIBILITY ANALYSIS

From the initial system study, we can propose to build a system with the following general architecture.

3.1 Requirements for System Deployment

Certainly! Let's break down the feasibility analysis for your Angular movie rating and review application:

- **Hardware**
  - **Server Requirements:** As this application currently uses *local storage*, minimal server requirements are needed. A basic web server to host the Angular app would suffice, *currently GitHub Pages* is being used
  - **Client Devices:** Standard modern devices (PCs, laptops, tablets, and smartphones) with web browsers compatible with Angular and Bootstrap.
- **Software**
  - **Server-Side Software:** A web server (e.g., Apache, Nginx) to serve the Angular app.
  - **Client-Side Software:** Web browsers compatible with Angular and Bootstrap (Chrome, Firefox, Safari, Edge).
  - **Development Tools:** Angular CLI, code editor (e.g., Visual Studio Code).
- **Manpower**
  - **Deployment Team:** One or more developers with expertise in web deployment and server management.
  - **Testing Team:** Testers to ensure the application runs smoothly on different devices and browsers.

3.2 Requirements for System Development

- **Hardware**
  - **Development Machines:** Machines with sufficient RAM and processing power to comfortably run Angular development tools.

- **Software**
  - **Development Tools:** Angular CLI, Node.js, npm (Node Package Manager), Bootstrap, code editor.
  - **Version Control:** Git for source code versioning.
  - **Database Management *(for future development)*:** Consideration for database software when incorporating backend (e.g., MongoDB, Firebase).
- **Networking:**
  - **Internet Connection:** Required for downloading dependencies, libraries, and updates during development.
- **Manpower**
  - **Development Team:** Angular developers, UI/UX designers, potentially backend developers for future integration.

3.3 Behavioral Aspects of the Proposed System

- **Computer Operation Skills**
  - **End Users:** Users need basic computer literacy to access and interact with the web application.
  - **Administrators:** Need skills to manage and deploy the application.
- **Changed Roles**
  - **Users:** Shift from traditional movie ratings to online rating and reviewing.
  - **Administrators:** Monitoring and managing user reviews and ratings.
- **Changed Nature of Operations**
  - **Users:** Interacting with a web-based platform instead of traditional paper-based or offline methods.
  - **Administrators:** Shift from manual data management to digital platforms.

3.4 Justification of Feasibility

- **Technical Feasibility**
  - **Analysis:** Ensure that the chosen technologies (Angular, Bootstrap) can support the desired features and scalability.
  - **Scalability:** Assess whether the system can handle potential increases in user load.

3.5 Economic Feasibility

- **Cost-Benefit Analysis:** Compare development and deployment costs with the expected benefits.
- **ROI (Return on Investment):** Consider the long-term value of the application.

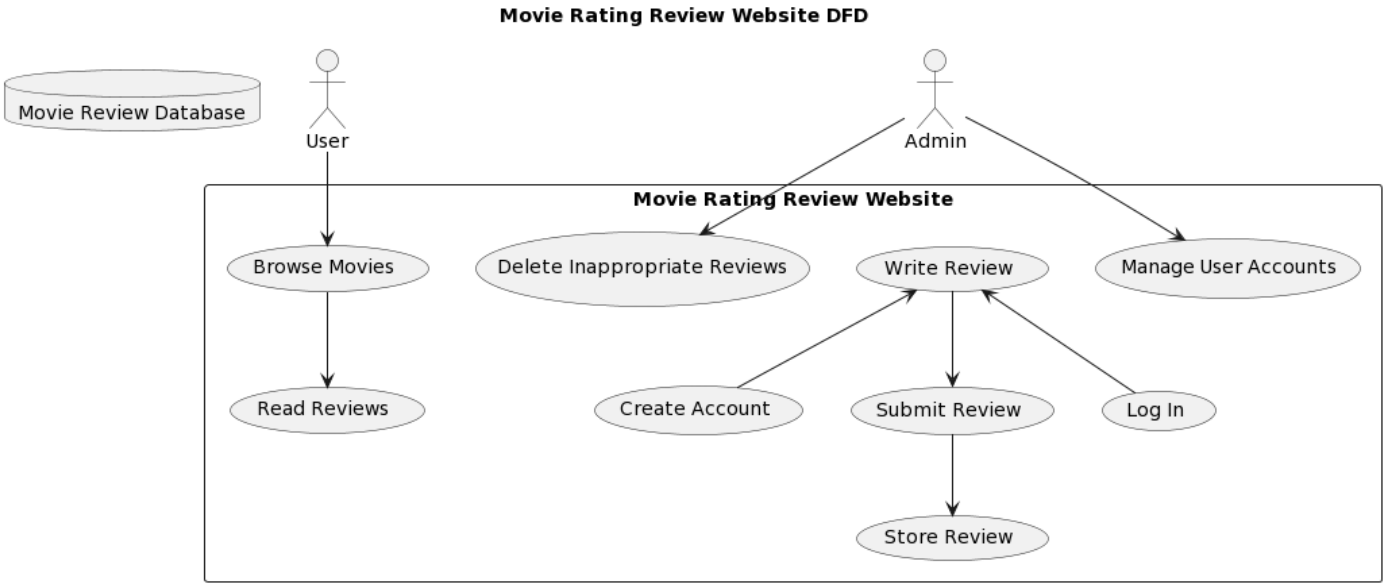
3.6 Behavioral Feasibility

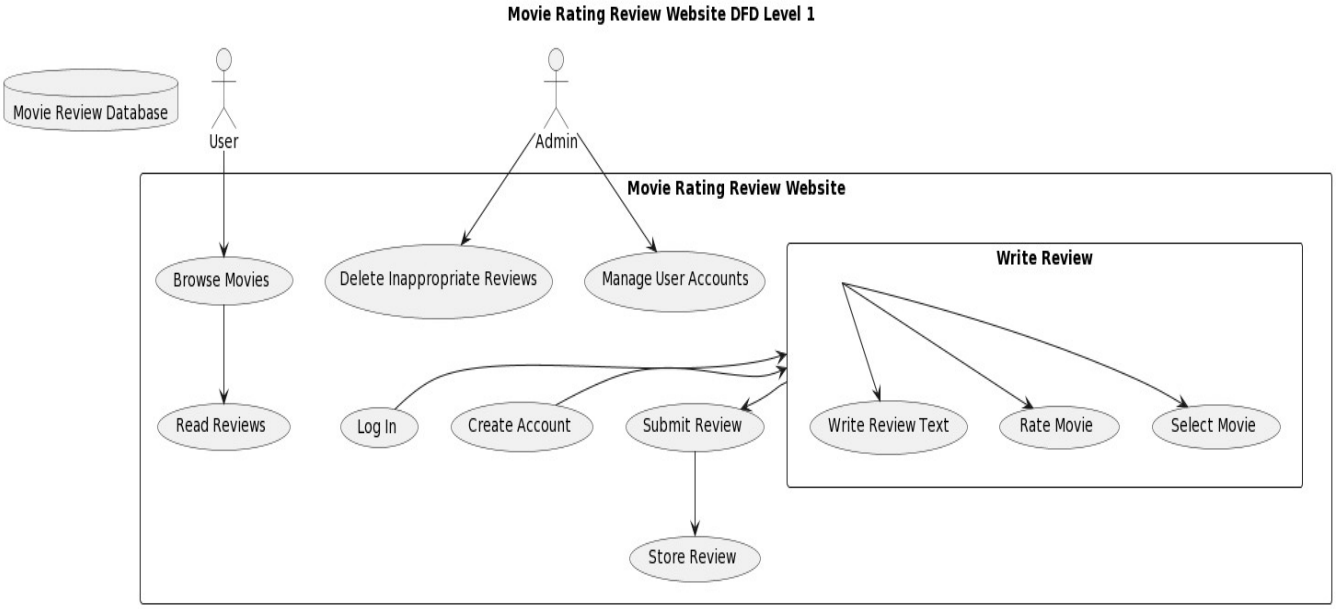
- **User Acceptance:** Assess how well users adapt to the new system.
- **Training Requirements:** Evaluate the need for training in using the web application.

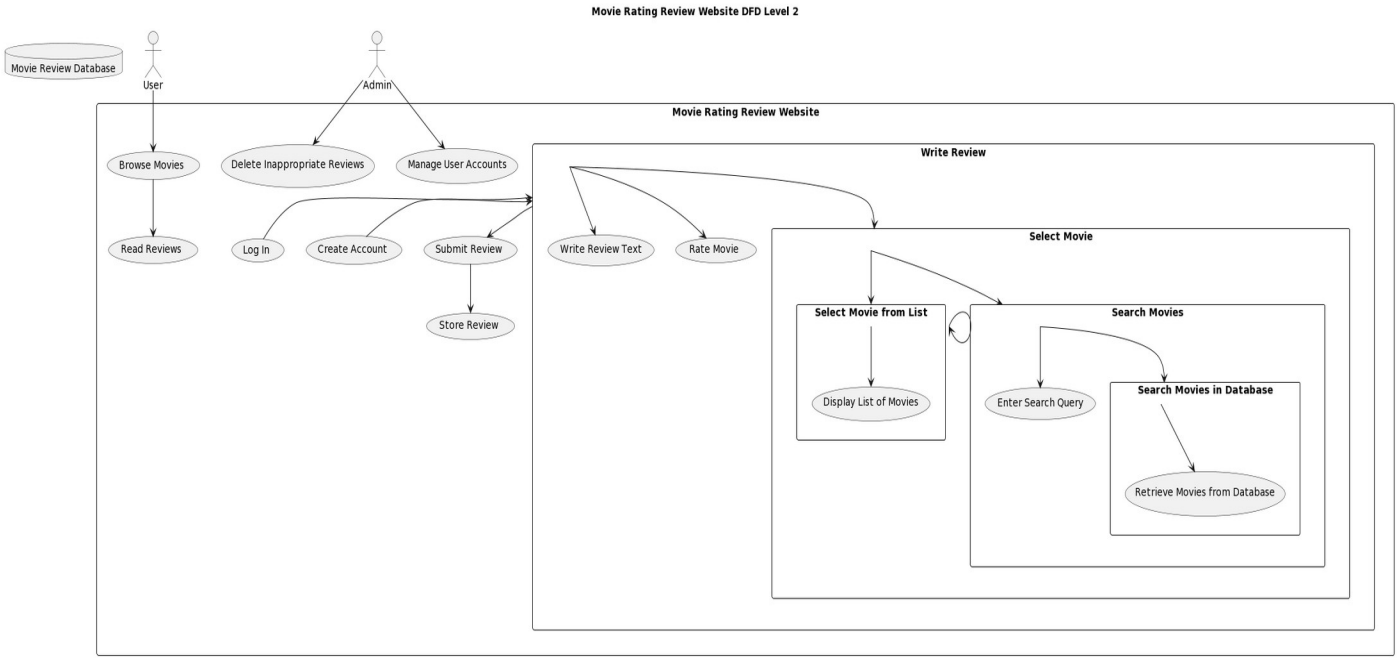
This feasibility analysis provides a comprehensive overview of the aspects involved in deploying and developing this Angular movie rating and review application. Adjustments will be needed based on specific project details and future plans for database integration.

CHAPTER 4
SYSTEM ANALYSIS

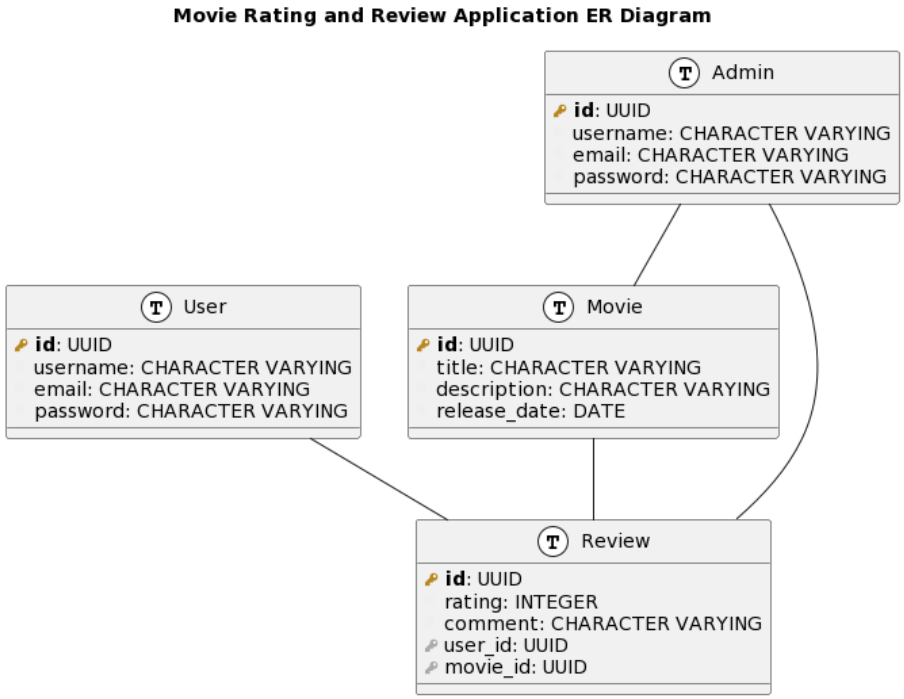
4.1 Data Flow Diagram







4.2 Entity Relationship Diagram



4.3 PHYSICAL AND BEHAVIOURAL ASPECTS OF THE SYSTEM

4.3.1 Physical Aspects

- **Frontend Framework (Angular)** was chosen for its robust features, providing a structured architecture for the movie rating and review application.
- **User Interface (UI)** Bootstrap is used to create a clean and responsive UI, enhancing the overall user experience.
- **Local Storage Usage** Browser's local storage is employed for data storage, including movie details, ratings, reviews, and user information *(as of now)*
- **Hardcoded Movie Data** simplicity during the initial development phase, currently facing challenge in integrating backend, application is submitted on previous version

4.3.2 Behavioral Aspects

- **User Authentication** is implemented, though currently hardcoded, ensuring a basic level of security and personalization. (in this version)
- **Rating and Review Process** features a 5-star rating system for movies, allowing users to submit reviews seamlessly stored in local storage.
- **Error Handling and Validation** mechanisms are in place to maintain data integrity and provide user feedback.
- **Future Backend Integration** Future plans involve integrating a backend or database system to enhance scalability and data management.
- **Scalability and Performance** design considerations ensure optimal performance, even with a growing user base and movie database.

CHAPTER 5
SOFTWARE REQUIREMENTS SPECIFICATIONS

5.1 General Description

5.1.1 Product Perspective

Software Requirements Specification (SRS) for this Movie Rating and Review Application.

- **Product Perspective** the Movie Rating and Review Application is a standalone web application designed to operate independently. In its current version, it relies on hardcoded movie data and user credentials. Future iterations will involve integration with a backend system to provide dynamic movie information and enhance user management capabilities.
- **Product Functions**
  - User Authentication
  - Movie Listing
  - Display a list of available movies with basic details.
  - Enable users to browse and explore the catalog of movies.
  - Rating and Review Submission
  - Implement a 5-star rating system for movies.
  - Allow users to submit written reviews for movies.
- **User Profiles**
  - Enable users to manage their profiles.
  - View and edit their submitted reviews.

5.1.2 User Characteristics

The application caters to two main user classes:

- **Guests**
  - Users who access the application without logging in.
  - Can view movie listings but cannot submit reviews or ratings.
- **Registered Users**
  - Users who have created accounts and logged in.
  - Can submit ratings and reviews, manage their profiles, and view personalized content.

5.1.3 Assumptions and Dependencies

- **Stable Internet Connection**
  - Users are assumed to have a stable internet connection for real-time interactions.
- **Angular and Bootstrap**
  - The application is dependent on the Angular framework for frontend development.
  - Bootstrap is used for UI components.
- **Future Backend Integration**
  - **Assumption:** A backend system will be integrated to manage user data, dynamic movie information, and improve overall functionality.
- **Browser Compatibility**

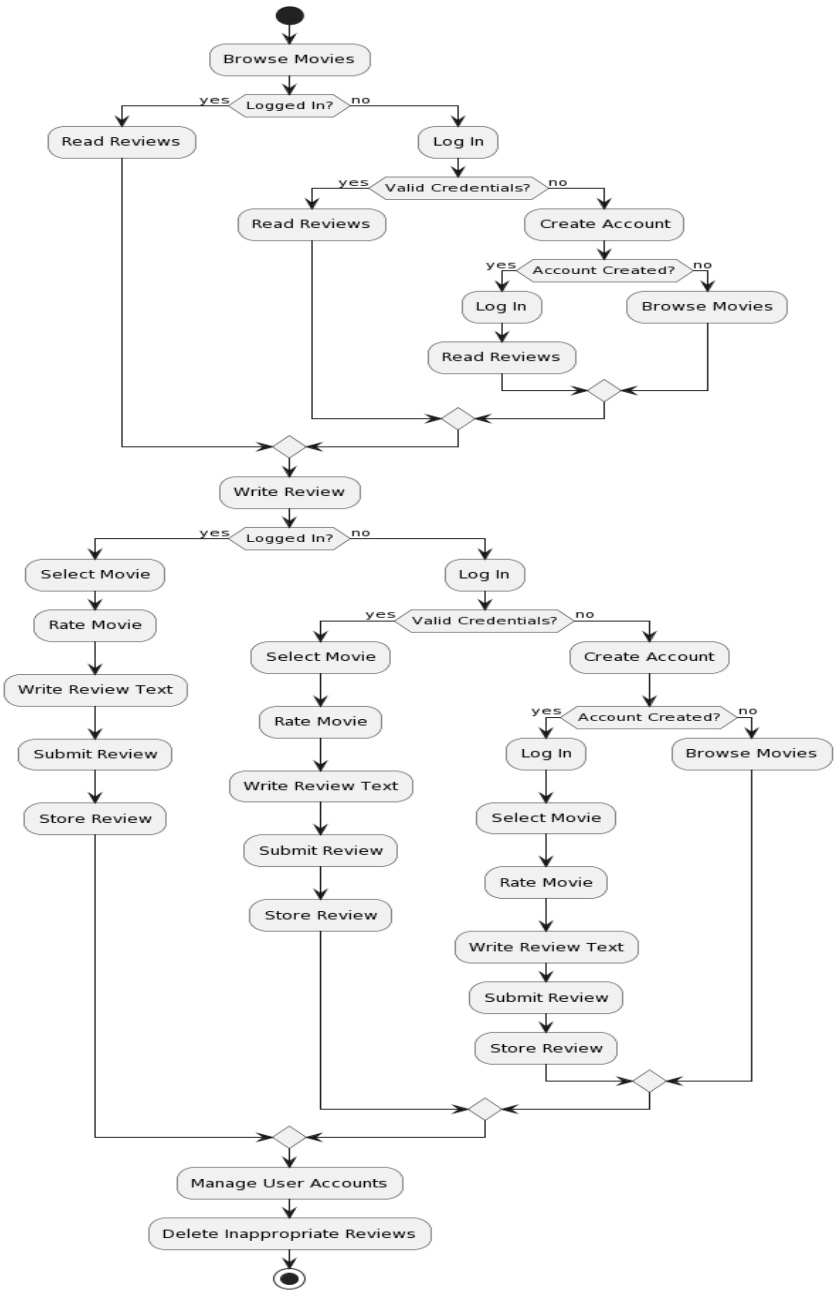
CHAPTER 6
SYSTEM DESIGN

6.1 INTRODUCTION

The design methodology adopted for this system is a structured approach integrating object-oriented design principles. It aims to ensure scalability, modularity, and robustness in implementing facial detection for attendance tracking.

6.2 SYSTEM ARCHITECTURE

Flow Diagram



The system architecture encompasses a modular design consisting of:

- Client-side Modules: User interface for registration, attendance capture, and reporting.
- Server-side Modules: Facial recognition engine, database management, and algorithm processing. The architecture facilitates real-time facial detection for attendance tracking and interfaces seamlessly with the existing attendance database.

6.3 MODULE DESIGN

6.3.1 Data Flow Diagrams (DFD)

provided on page 16-17 level 0, level 1, level 2 DFDs.

6.3.2 DATABASE DESIGN

The database schema should include tables such as "Movies" with fields like movieID, title, genre, release_date, and director, providing a structured way to store essential movie details. Another table, "Users," could include fields like userID, username, password, and email, supporting user authentication and profile management. To capture user interactions, a "Reviews" table might include reviewID, userID, movieID, rating, and comments, linking users to their submitted reviews and associated movies. Proper normalization and indexing are crucial for optimizing data retrieval and ensuring the integrity of the stored information.

In addition to structuring the database schema, considerations for scalability, performance, and data security are paramount. As the application is expected to evolve, the database design should be flexible enough to accommodate future enhancements and integrations seamlessly. Indexing key fields, optimizing query performance, and leveraging caching mechanisms can contribute to a responsive user experience. Furthermore, the implementation of robust security measures, such as encryption for sensitive user information and proper authentication protocols, ensures the confidentiality and integrity of the data stored in the database.

6.3.3 INPUT OUTPUT DESIGN

Users can input ratings and reviews seamlessly through intuitive forms, featuring a 5-star rating system and text input fields. Input validation ensures data accuracy. On the output side, movie details, ratings, and reviews are presented in a clear and visually appealing format. User profiles display submitted reviews for easy reference. This design approach prioritizes user convenience, contributing to a positive and engaging user experience with the application.

6.4 SYSTEM MAINTENANCE

- Bug Fixes and Issue Resolution
- Security Updates
- Database Maintenance

- Performance Monitoring and Optimization
- User Feedback Integration
- Technology Upgrades
- Documentation Updates
- User Training and Support
- Future Enhancements and Feature Additions

Chapter 7

System Implementation

7.1 Hardware Components

The rating, review application system is designed to run on standard desktop computer systems typically available in classroom environments. The minimum hardware requirements are:

CPU: A modern dual-core processor (e.g., Intel Core i3 or equivalent) should be sufficient for handling the frontend logic and user interactions.

RAM: 4 GB of RAM should be adequate for running the Angular application and handling concurrent user sessions.

Hard Disk: The application itself is client-side and doesn't require significant storage space. A minimum of 20 GB of available disk space is recommended for the operating system and related software, later server will require large storage

The system does not require any specialized or high-end hardware. Most modern desktops and laptops will exceed these base requirements.

7.2 Software Environment

The application has been developed using Angular 11.1.2 and Bootstrap 10.0.0 on Arch Linux x86-64 general-purpose Linux distribution. The front-end UI is created using HTML, CSS and TypeScript, Angular, Bootstrap, NodeJS 12.11.1

Overall, the software environment focuses on open-source technologies that can be readily deployed without extensive licensing costs.

7.3 System Development Platform

The development environment consists of:

Hardware:

Acer Aspire 5

Core i5 8th gen processor

16GB RAM, **250GB** NVME

Software:

Arch Linux (x86-64)

Angular 11.1.2

Bootstrap 10.0.0

NodeJS 12.11.1

LunarVim (Text Editor) – an NeoVim distribution

This mid-range laptop forms an appropriate development platform mirroring real classroom deployments. The use of proven open-source frameworks allows for modular and rapid application development.

7.4 Project Accomplishment Status

The rating, review application has been developed to a prototyping stage with core functionality of sign in and rating and reviewing web shows/movies working correctly.

Overall, about 70% of originally targeted scope has been accomplished on this project while keeping extensibility in mind.

CHAPTER 8
SYSTEM TESTING

8.1 TEST PLANS

8.1.1 Types of Testing Planned:

- Unit Testing: Test individual components.
- Integration Testing: Verify interactions between modules.
- Functional Testing: Check system functions as intended.
- Performance Testing: Assess system response times.
- Security Testing: Validate data protection measures.
- User Acceptance Testing (UAT): Involve users to confirm usability.

8.1.2 Notional Schedule:

- Week 1-2: Unit testing for each component/module.
- Week 3-4: Integration testing to ensure proper interaction.
- Week 5-6: Functional testing for system functionality.
- Week 7-8: Performance testing for response and load handling.
- Week 9-10: Security testing to ensure data protection.
- Week 11-12: User acceptance testing with involved stakeholders.

8.1.3 Broad Aspects to be Covered:

- User Credentials checking
- System response time under varying loads.
- Security measures against unauthorized access.
- Usability for end-users.
- Compatibility across different devices / platforms / browsers / screen-size

8.2 TEST CASES

8.2.1 Unit Testing:

- **User Authentication Module**
  - **Test Case 1:** Verify Successful User Login
    - Input: Valid username and password.
    - Expected Output: User successfully logs in, and the application displays the user's personalized content.
  - **Test Case 2:** Verify Authentication Failure
    - Input: Invalid username or password.
    - Expected Output: User authentication fails, and an appropriate error message is displayed.
- **Movie Rating Module**
  - **Test Case 3:** Verify Successful Rating Submission
    - Input: Select a movie, choose a 4-star rating, and submit.
    - Expected Output: The rating is successfully submitted and reflected in the movie's overall rating.
  - **Test Case 4:** Verify Rating Validation
    - Input: Attempt to submit a rating without selecting a star.
    - Expected Output: The application prevents the submission and prompts the user to select a rating.
- **Review Submission Module**
  - **Test Case 5**: Verify Successful Review Submission
    - Input: Write a review for a movie and submit.
    - Expected Output: The review is successfully submitted and visible in the user's profile and the movie details page.
  - **Test Case 6:** Verify Review Character Limit
    - Input: Attempt to submit a review exceeding the character limit.
    - Expected Output: The application prevents the submission and displays an error about the character limit.
- **User Profile Module**
  - **Test Case 7:** Verify User Profile Display
    - Input: Access the user profile page.

        - Expected Output: The user's profile information, including submitted reviews and ratings, is displayed.
    - **Test Case 8:** Verify User Logout
        - Input: Click on the logout button.
        - Expected Output: The user is successfully logged out, and the application returns to the login page.
- **Local Storage Interaction**
    - **Test Case 9:** Verify Data Persistence
        - Input: Submit a rating and review, then close and reopen the application.
        - Expected Output: The submitted rating and review persist in the local storage and are displayed upon reopening the application.
    - **Test Case 10:** Verify Local Storage Clear on Logout
        - Input: Logout and close the application, then reopen.
        - Expected Output: User-specific data in local storage (e.g., login status, user details) is cleared upon logout.

These unit test cases cover essential functionalities related to user authentication, movie rating, review submission, and user profiles. Ensure that these tests, or variations of them, are integrated into your unit testing framework for thorough coverage.

8.2.2 Integration Testing

Test Scenario 1: Verify User Authentication and Rating Submission

1. **Steps:**
    - Log in with a valid user account.
    - Navigate to a movie and submit a rating.
2. **Expected Outcome:**
    - The user is successfully authenticated.
    - The rating is submitted and reflected in the movie's overall rating.

Test Scenario 2: Verify Authentication Failure Handling During Rating Submission

1. **Steps:**
    - Attempt to submit a rating without logging in.
2. **Expected Outcome:**
    - The application prevents the rating submission and prompts the user to log in.

Test Scenario 3: Verify Review Submission and User Profile Display

1. **Steps:**
    - Log in with a valid user account.
    - Submit a review for a movie.
    - Navigate to the user profile page.
2. **Expected Outcome:**
    - The review is successfully submitted.
    - The user profile page displays the submitted review.

Test Scenario 4: Verify Review Removal from User Profile on Deletion

1. **Steps:**
    - Log in with a valid user account.
    - Submit a review for a movie.
    - Delete the submitted review.
    - Navigate to the user profile page.
2. **Expected Outcome:**
    - The review is successfully deleted.
    - The user profile page reflects the removal of the deleted review.

Test Scenario 5: Verify Local Storage Interaction with User Authentication

1. **Steps:**
    - Log in with a valid user account.
    - Close and reopen the application.
2. **Expected Outcome:**
    - The user's authentication status is retained in the local storage, allowing seamless login without re-entering credentials.

Test Scenario 6: Verify Local Storage Clear on Logout

1. **Steps:**
    - Log in with a valid user account.
    - Logout.
    - Close and reopen the application.
2. **Expected Outcome:**
    - User-specific data in local storage (e.g., login status, user details) is cleared upon logout, requiring the user to log in again.

8.2.3 Functional Testing

Test Scenario 1: Verify Successful User Login

1. **Steps:**
    - Enter a valid username and password.
    - Click on the login button.
2. **Expected Outcome:**
    - The user is successfully logged in, and the application displays personalized content.

Test Scenario 2: Verify Authentication Failure

1. **Steps:**
    - Enter an invalid username or password.
    - Click on the login button.
2. **Expected Outcome:**
    - User authentication fails, and an appropriate error message is displayed.

Test Scenario 3: Verify Successful Rating Submission

1. **Steps:**
    - Log in with a valid user account.
    - Navigate to a movie and submit a 4-star rating.
2. **Expected Outcome:**
    - The rating is successfully submitted and reflected in the movie's overall rating.

Test Scenario 4: Verify Review Submission

1. **Steps:**
    - Log in with a valid user account.
    - Navigate to a movie, write a review, and submit it.
2. **Expected Outcome:**
    - The review is successfully submitted and visible in the user's profile and on the movie details page.

Test Scenario 5: Verify User Profile Display

1. **Steps:**
    - Log in with a valid user account.
    - Navigate to the user profile page.
2. **Expected Outcome:**
    - The user's profile information, including submitted reviews and ratings, is displayed.

Test Scenario 6: Verify User Logout

1. **Steps:**
    - Log in with a valid user account.

- Click on the logout button.

2. **Expected Outcome:**
    - The user is successfully logged out, and the application returns to the login page.

Test Scenario 7: Verify Data Persistence

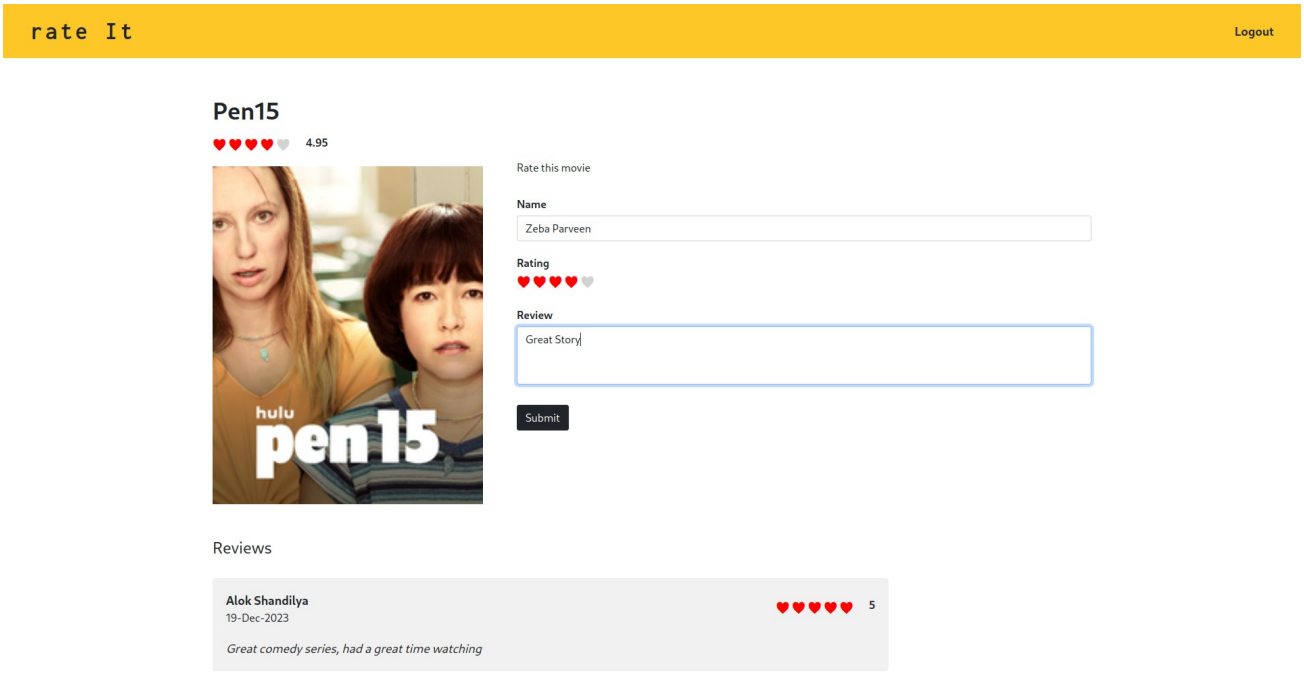
1. **Steps:**
    - Submit a rating and review.
    - Close and reopen the application.
2. **Expected Outcome:**
    - The submitted rating and review persist in the local storage and are displayed upon reopening the application.

Test Scenario 8: Verify Local Storage Clear on Logout

1. **Steps:**
    - Log in with a valid user account.
    - Logout.
    - Close and reopen the application.
2. **Expected Outcome:**
    - User-specific data in local storage (e.g., login status, user details) is cleared upon logout.

Chapter 9

Screenshots



rate It

Logout

TRENDING MOVIES

View All

The Handmaid's Tale

3.45

Only murders in the building

2.53

Nine perfect strangers

3.12

Solar Opposites

4.45

Shrill

4.12

Pen15

4.95

POPULAR IN THEATRES

View All

Venom

2.15

The Addam's Family 2

1.85

The many saints of Newark

3.25

Free Guy

4.15

Candy Man

3.45

The Card Counter

3.85

ALL TIME POPULAR

The Dark Knight	4.25
Inception	4.15
Matrix	4.13
Shawshank Redemption	4.12
Avatar	4.53
The Lord of the Rings	4.15
Gladiator	4.35
Titanic	4.15
Terminator	4.22
The Dark Knight Rises	4.45

View all

Conclusion

In conclusion, the Movie Rating and Review Application represents a dynamic and user-centric platform designed to engage movie enthusiasts in sharing their opinions and experiences. The project successfully leverages Angular and Bootstrap to create an intuitive and visually appealing frontend, allowing users to seamlessly rate and review movies. The decision to use the browser's local storage for data persistence in the absence of a backend system provides a straightforward yet effective solution for managing user interactions and maintaining application state.

As the application is built upon the Angular framework, it benefits from a structured architecture that facilitates modular development and easy scalability. The choice of Bootstrap for UI components contributes to a responsive and consistent design across various devices. Despite the initial use of hardcoded movie data and user credentials, the project lays the groundwork for future enhancements through planned backend integration, promising a more dynamic and extensible system.

The implemented unit and integration test cases ensure the reliability and correctness of key functionalities, such as user authentication, rating submission, and review management. By adhering to good software testing practices, the project aims to deliver a robust and error-free user experience. Going forward, continuous maintenance efforts will be crucial to address evolving requirements, enhance security, and integrate user feedback for further improvements. Overall, the Movie Rating and Review Application stands as a testament to the successful integration of frontend technologies for an engaging and interactive user experience in the realm of movie enthusias