Letterboxd: A Film Socia Media

Maria del Pilar Pradilla Cely
Alejandro Nuñez Barrera

Eng. Carlos Andres Sierra

Universidad Distrital Francisco José de Caldas
Computer Engineering
April 9, 2024
Bogotá, Colombia

Laboratorio de Automatización, Sistemas Embebidos y Robótica LASER
Carrera 8 No 40 - 62, Piso 7, Bogotá - Colombia
PBX 57 (1) 323 9300 Ext. 1701

Índice

Abstract

The engineering project of the course focuses on developing a user-friendly movie application akin to Letterboxd. This platform aims to streamline the movie-watching experience by providing users with features such as personalized recommendations, watchlist creation, and community engagement. Leveraging modern software development methodologies and robust backend systems, our project seeks to enhance the digital entertainment landscape by delivering a high-performance and intuitive movie platform for users worldwide.

Keywords

Application, Engineering, Films, Users, Methodologies.

1. Introduction

The advent of digital technologies has revolutionized the way we consume media, particularly in the realm of movies. With the proliferation of streaming services and the vast array of film choices available, users often face the challenge of navigating through a sea of options to find content that aligns with their interests. In response to this need, our engineering project aims to address the gap in the market by developing a comprehensive movie application inspired by platforms like Letterboxd.

Our project seeks to provide users with a centralized hub for discovering, exploring, and sharing their favorite films. By leveraging modern software development methodologies and advanced database management techniques, we endeavor to create a user-friendly platform that offers personalized recommendations, seamless browsing experiences, and robust community engagement features.

The application will be designed in Python, with some tools for backend and frontend, like Django. For the database, there will be used PosgreSQL and SQLAlchemy in order to implements databases. The objetive of the project is use the knowledge learnt among the advanced programming classes to develop an useful application taking as reference one existing one, like Letterboxd. Furthermore, another objective is to apply different concepts regarding object-oriented design, RESTFul APIs, and Model-Template-Controller.

Throughout this document, we will delve into the process of development and design of the code underpinning our project. We will explore in depth the various stages of development, from initial conception to final implementation, providing a detailed insight into design decisions, development methodologies, and technologies employed. The approach taken to structure the code, optimize performance, and ensure scalability of the system will be examined, with the aim of offering a comprehensive understanding of the architecture and functioning of the developed software. Additionally, challenges encountered during the development process and the solutions implemented to overcome them will be highlighted, providing a view of the project from a technical perspective.

2. Methods and Materials

The project aimed to design and develop a comprehensive movie application akin to Letterboxd, employing modern software engineering principles and advanced programming techniques. The project followed an iterative and collaborative approach, incorporating elements of Agile methodologies to ensure flexibility and adaptability throughout the development process.

Target Users: The target population for the study comprised potential users of the movie application, including individuals interested in exploring, reviewing, and discussing films. Given the nature of the project, no specific sampling strategy was employed, as the focus was on designing a platform that caters to a broad audience of movie enthusiasts.

Procedures: The development process involved several key stages, including requirements gathering, system design, implementation, testing, and deployment. Requirements were elicited through the info on the web and user stories. System design encompassed the creation of diagrams, user interface mockups, and architectural diagrams to visualize the application's structure and functionality. Implementation utilized industry-standard programming languages and frameworks, with an emphasis on scalability, maintainability, and extensibility. Testing involved both manual and automated techniques to ensure the reliability and quality of the software. Deployment was conducted on a local host infrastructure in Apache to facilitate accessibility and scalability.

Tools and materials: The project utilized a range of tools and materials, including programming languages such as Python and HTML/CSS for backend and frontend development, respectively. Frameworks such as Django were employed to streamline development and enhance productivity. Version control was managed using Git, with hosting on platforms like GitHub for collaboration and code management. Additionally, third-party APIs and libraries were integrated to access movie data, user authentication, and social featuresm, using FastAPI.

3. Experiments and results

The development of the movie application commenced with a thorough analysis of user requirements and market trends. Extensive research was conducted to identify key features and functionalities essential for creating an engaging and user-friendly platform akin to Letterboxd. Based on this analysis, the project team devised a comprehensive development plan outlining the steps required to design, implement, and deploy the application.

Key aspects of the experiment include:

- Design and Architecture: The application's design and architecture were meticulously planned to ensure scalability, flexibility, and maintainability. Object Oriented Design were employed to create a robust and extensible codebase.
- Implementation: The application was implemented using industry-standard programming languages and frameworks, including Python, Django, and PostgreSQL. Features such as user registration, movie search, review posting, and social interaction were developed and integrated into the application.
- Testing and Quality Assurance: Rigorous testing procedures were conducted throughout the development process to identify and rectify any bugs or issues. Unit tests, integration tests, and user acceptance tests were employed to ensure the reliability and performance of the application.

The implementation of the movie application resulted in the creation of a feature-rich platform that provides users with a seamless and intuitive movie-watching experience. Key results obtained from the project include:

- Feature Completeness: The application successfully implemented a wide range of features, including user registration, movie browsing and searching, review posting and rating, social interaction, and personalized recommendations.
- Future Directions: While the current version of the application meets the basic requirements outlined in the project scope, there are opportunities for future enhancements and refinements. Potential areas for improvement include further optimization of performance, addition of new features, and expansion of platform compatibility.

Referencias

[1] Letterboxd. «About Letterboxd.» (2021), dirección: `https://letterboxd.com/about/`.

[2] Movie Database API. «Documentation.» (2021), dirección: `https://developer.themoviedb.org/reference/intro/getting-started`.

[3] E. R. Harold, *Python 3 Object-Oriented Programming*. Packt Publishing, 2015.