

NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY,

BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA



Database Management Systems (22CSG44)

on

“Movie Management System”

Submitted in partial fulfilment of the requirement for the award of Degree of

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

Mohan Acharya

1NT22CS116

Pradeep Badu

1NT22CS133

Lucky Ansari

1NT22CS104

Under the Guidance of

Shruthi Shetty J

Assistant Professor, Dept. of CS&E, NMIT



Department of Computer Science and Engineering
(Accredited by NBA Tier-1)

2023-2024

NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM
, APPROVED BY AICTE & GOVT.OF KARNATAKA)

Department of Computer Science and Engineering (Accredited by NBA Tier-1)



CERTIFICATE

This is to certify that the Project Report on **Movie Management System** is an authentic work carried out by Mohan Acharya (1NT22CS116), Pradeep Badu (1NT22CS133), Lucky Ansari and (1NT22CS104) bonafide students of **Nitte Meenakshi Institute of Technology**, Bangalore in partial fulfilment for the award of the degree of **Bachelor of Engineering** in COMPUTER SCIENCE AND ENGINEERING of Visvesvaraya Technological University, Belagavi during the academic year **2023-2024**. It is certified that all corrections and suggestions indicated during the internal assessment have been incorporated in the report.

Course Faculty

Shruthi Shetty J
Assistant Professor, Dept.
CSE, NMIT Bangalore

Signature of the HOD

Dr. Meenakshi Sundaram
Head of Dept. CSE,
NMIT Bangalore

DECLARATION

We hereby declare that.

- (i) The project work is our original work.
- (ii) This Project work has not been submitted for the award of any degree or examination at any other university/College/Institute.
- (iii) This Project Work does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other people.
- (iv) This Project Work does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
 - a) their words have been re-written, but the general information attributed to them has been referenced.
 - b) where their exact words have been used, their writing has been placed inside quotation marks, and referenced.
- (v) This Project Work does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

NAME	USN	Signature
Mohan Acharya	1NT22CS116	
Pradeep Badu	1NT22CS133	
Lucky Ansari	1NT22CS104	

Date:

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success. We express my sincere gratitude to our Principal **Dr. H. C. Nagaraj**, Nitte Meenakshi Institute of Technology for providing facilities. We wish to thank our HoD, **Dr. Meenakshi Sundaram**, for the excellent environment created to further educational growth in our college. We also thank him for the invaluable guidance provided which has helped in the creation of a better project.

We hereby like to thank our subject faculty **Mrs. Shruthi Shetty J** Assistant Professor, Department of Computer Science & Engineering for her periodic inspection, time to time evaluation and helping to bring the project to the present form.

We also thank all our friends, teaching, and non-teaching staff at NMIT, Bangalore, for all the direct and indirect help provided in the completion of the project.

NAME	USN	SIGNATURE
Mohan Acharya	1NT22CS116	
Pradeep Badu	1NT22CS133	
Lucky Ansari	1NT22CS104	

Date:

ABSTRACT

The Movie Management System (MMS) is an innovative solution designed to address the challenges of managing extensive movie databases. With the exponential growth of digital content in the entertainment industry, there is a pressing need for efficient data management systems. The MMS leverages SQLite for its backend database and Tkinter for its graphical user interface (GUI), providing a robust, user-friendly platform for administrators and users alike.

This report provides a comprehensive overview of the MMS project, detailing its background, purpose, motivation, and objectives. It also includes an extensive literature survey that reviews existing systems and technologies, highlighting the reasons for selecting SQLite and Tkinter for this project.

The system requirements specifications are outlined, covering both hardware and software needs to ensure optimal performance. The implementation chapter describes the Agile methodology adopted for the project, emphasizing iterative development and continuous improvement. The pseudo-code section provides a high-level overview of the core functionalities implemented in the system.

Extensive testing demonstrated the system's ability to perform CRUD operations efficiently, handle large datasets without performance issues, and offer an intuitive user experience. User feedback has been overwhelmingly positive, indicating high satisfaction with the system's ease of use and functionality.

The conclusion underscores the project's success in achieving its objectives and suggests potential future enhancements, such as user authentication, advanced search filters, and API integration. This project exemplifies the effective use of simple yet powerful technologies to create a reliable and scalable Movie Management System.

TABLE OF CONTENTS

1.CHAPTER 1: INTRODUCTION

- 1.1 Background
 - 1.1.1 Brief history of Technology/concept
 - 1.1.2 Applications
- 1.2 Research motivation and Problem statement
 - 1.2.1 Research Motivation
 - 1.2.2 Statement of the Problem
- 1.3 Research objectives and contributions
 - 1.3.1 Primary objectives
 - 1.3.2 Main contributions
- 1.4 Summary

2.CHAPTER 2: LITERATURE SURVEY

- 2.1 Introduction
- 2.2 Study of Tools/Technology
- 2.3 Summary

3.CHAPTER 3: SYSTEM REQUIREMENTS SPECIFICATIONS

- 3.1 General Description
 - 3.1.1 Product Perspective
- 3.2 System Requirements
 - 3.2.1 Hardware Requirements
 - 3.2.2 Software Requirements
- 3.3 Summary

4.CHAPTER 4: IMPLEMENTATION

4.1 Methodology

4.2 Description of Process

4.3 Pseudo-code

5. CHAPTER 5: LIST OF TABLES

6. CHAPTER 6: RESULTS

7.CHAPTER 7: CONCLUSIONS

8.REFERENCES

CHAPTER 1: INTRODUCTION

1.1 Background

The Movie Management System (MMS) has become a crucial tool in the entertainment industry for managing vast amounts of data related to movies, directors, actors, and other associated information. With the rapid growth of digital content, the need for efficient data management solutions has become increasingly important. MMS helps in organizing and managing movie databases, providing an interface for users to easily access and manipulate the data.

1.2 Purpose

The purpose of this document is to define the software requirements for the Movie Management System. It aims to provide a detailed description of the system's features, interfaces, and requirements to guide the development and testing processes. The system is designed to facilitate efficient data storage, retrieval, and management, ensuring a user-friendly experience.

1.3 Motivation

The motivation behind developing the Movie Management System is to address the challenges faced in managing large movie databases manually. Manual data management is prone to errors, time-consuming, and inefficient. An automated system can significantly improve the accuracy, efficiency, and accessibility of movie data, benefiting both the administrators and end-users.

1.4 Problem Statement

Managing extensive movie databases manually is prone to errors and inefficiencies. There is a need for an automated system that can handle data effectively. The system should provide functionalities for adding, updating, deleting, and retrieving movie information efficiently. It should also offer a user-friendly interface to facilitate easy interaction with the database.

1.5 Research Objective

The primary objective of this project is to design and implement a Movie Management System that leverages SQLite for the backend and Tkinter for the frontend. The system should be capable of efficiently managing movie data, providing functionalities for data manipulation, and offering a user-friendly interface for users. Specific objectives include:

- Developing a robust database schema for storing movie information.
- Implementing functionalities for data manipulation (CRUD operations).
- Designing an intuitive graphical user interface using Tkinter.
- Ensuring the system is scalable and maintainable.

CHAPTER 2: LITERATURE SURVEY

2.1 Introduction

This chapter presents a review of the literature related to Movie Management Systems and the technologies used in their development. It includes a study of existing systems, tools, and technologies that have been utilized in similar projects.

2.2 Study of Tools/Technology

The development of Movie Management Systems involves various tools and technologies. Some of the key technologies reviewed for this project include:

SQLite: A lightweight, disk-based database that doesn't require a separate server process. It is widely used for embedded database systems.

Tkinter: A standard GUI library for Python, used for creating graphical user interfaces. Tkinter provides a fast and easy way to create desktop applications.

Python: A high-level programming language known for its simplicity and readability. Python is used for writing the backend logic and integrating the database with the user interface.

Other Relevant Technologies: Various other technologies and tools such as SQLAlchemy for ORM, Flask for web applications, and REST APIs for data communication were also reviewed.

2.3 Summary

The literature survey highlighted the importance of using robust and scalable technologies for developing Movie Management Systems. SQLite and Tkinter were identified as suitable choices for this project due to their simplicity, ease of use, and extensive documentation. Python, being a versatile programming language, was chosen for its ability to integrate with these technologies seamlessly.

CHAPTER 3: SYSTEM REQUIREMENTS SPECIFICATIONS

3.1 General Description

The Movie Management System is designed to manage a comprehensive database of movies, including information about directors, actors, genres, and ratings. The system provides functionalities for adding, updating, deleting, and retrieving movie information. It aims to facilitate efficient data management and provide a user-friendly interface for interacting with the database.

3.2 Product Perspective

The Movie Management System is a standalone desktop application that interacts with a SQLite database. It is designed to be used by administrators who manage movie data and end-users who wish to browse and search for movies. The system includes a graphical user interface built with Tkinter, providing an intuitive experience for users.

3.3 System Requirements

3.3.1 Hardware Requirements

- Processor: Intel i3 or higher
- RAM: 4GB or higher
- Hard Disk: 500GB or higher
- Display: 1024x768 resolution or higher

3.3.2 Software Requirements

- Operating System: Windows 10, macOS, or Linux
- Python 3.x
- SQLite
- Tkinter
- Any text editor or IDE for development (e.g., Visual Studio Code, PyCharm)

3.4 Summary

The system requirements specifications provide a detailed overview of the hardware and software requirements for the Movie Management System. These specifications ensure that the system operates efficiently and meets the needs of both administrators and end-users.

CHAPTER 4: IMPLEMENTATION

4.1 Methodology

The implementation of the Movie Management System follows the Agile methodology. Agile development is characterized by iterative cycles, known as sprints, where each sprint results in a potentially shippable product increment. This methodology allows for flexibility, continuous feedback, and iterative improvements.

4.2 Description of Process

The implementation process involved the following steps:

Requirement Analysis: Identifying and documenting the requirements of the system.

System Design: Creating the database schema and designing the graphical user interface.

Development: Writing the code for the backend and frontend components.

Testing: Conducting unit tests, integration tests, and user acceptance tests to ensure the system functions correctly.

Deployment: Deploying the system for use by administrators and end-users.

Maintenance: Providing ongoing support and updates to address any issues and improve functionality.

4.3 Pseudo-code

The following pseudo-code outlines the core functionality of the Movie Management System:

1. Initialize the SQLite database connection.
2. Create tables for movies, directors, actors, and genres if they do not exist.
3. Define functions for CRUD operations:
 - AddMovie(title, director, actor, genre, rating)
 - UpdateMovie(movie_id, title, director, actor, genre, rating)
 - DeleteMovie(movie_id)
 - GetMovie(movie_id)
 - SearchMovies(query)
4. Design the graphical user interface using Tkinter:
 - Create main window with menu options (Add, Update, Delete, Search).
 - Add input fields and buttons for each operation.
 - Display search results in a list or table format.
5. Implement event handlers for user interactions (button clicks, form submissions).
6. Test the system to ensure all functionalities work as expected.

CHAPTER 5: LIST OF TABLES

- Table 3.1: Hardware Requirements
- Table 3.2: Software Requirements
- Table 4.1: Functions for CRUD Operations

CHAPTER 6: RESULTS

The Movie Management System was tested extensively to ensure it meets the specified requirements. The following results were observed:

- The system successfully performs CRUD operations on the movie database.
- The graphical user interface is intuitive and user-friendly, allowing users to easily interact with the database.
- Search functionality works efficiently, providing accurate and quick results based on user queries.
- The system handles large datasets effectively without performance degradation.
- User feedback indicated high satisfaction with the ease of use and functionality provided by the system.

CHAPTER 7: CONCLUSIONS

The Movie Management System project successfully achieved its objectives of creating an efficient and user-friendly application for managing movie data. By leveraging SQLite for the backend and Tkinter for the frontend, the system provides a robust solution for storing, retrieving, and manipulating movie information. The Agile methodology facilitated iterative development and continuous improvement, ensuring the final product met user needs and requirements.

The system's design and implementation highlight the effectiveness of using lightweight yet powerful technologies to address complex data management challenges. Users found the graphical user interface intuitive and easy to navigate, which contributed to the overall positive reception of the system. The successful integration of SQLite and Tkinter showcases the feasibility of combining these technologies to develop scalable and maintainable applications.

Future enhancements could include expanding the system to support additional features such as user authentication, advanced search filters, and integration with external APIs for fetching movie data. User authentication would enhance security, allowing only authorized users to access and modify the database. Advanced search filters would improve the user experience by providing more precise and tailored search results. Integration with external APIs could automate data fetching and updating, keeping the movie database current without manual intervention.

Overall, the project demonstrates the potential of combining simple yet powerful technologies to create effective data management solutions. The system's success underscores the importance of iterative development, user feedback, and continuous improvement in creating applications that meet user needs and expectations.

REFERENCES:

1. SQLite Documentation. (n.d.). Retrieved from <https://www.sqlite.org/docs.html>
2. Tkinter Documentation. (n.d.). Retrieved from <https://docs.python.org/3/library/tkinter.html>
3. Python Documentation. (n.d.). Retrieved from <https://docs.python.org/3/>
4. Agile Software Development. (n.d.). Retrieved from <https://agilemanifesto.org/>
5. Design Patterns: Elements of Reusable Object-Oriented Software by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.

Project Report Format Reference:

<https://vtu.ac.in/examination-guidelines/#:~:text=PROJECT%20REPORTS,Top%20and%20Bottom%20%E2%80%93%200.75%E2%80%B3.>

