SOFTWARE REQUIREMENTS SPECIFICATION

FOR

VIDEO RENTAL SYSTEM

(VERSION 1.0)

TEAM:

PRANAYA CHOWDARY - 22CS10070

HARSHIT JAIN - 22CS10030

SHASHWAT SINGH RANKA - 22CS10071

CSE DEPT, IIT KHARAGPUR

DATE:

2 APRIL 2024

Table of Contents

Table of Contents 2
Introduction 3
- Purpose 3
- Scope of Project 3
- Document Conventions 4
- References 4

Overall Description 5
- Product Perspective 5
- Product Functions 5
- User Classes and Characteristics 6
- Operating Environment 6
- Design and Implementation Constraints 6

External Interface Requirements 7
- User Interfaces 7
- Hardware Interfaces 7
- Software Interfaces 7
- Communication Interfaces 8

System Features 9
- Secure Login 9
- Recommender Model 10
- Customer Features 11
- Staff Features 17
- Manager Features 21

Class Diagram 26
Non Functional Requirements 27
- Performance 27
- Security 27
- Reliability 27
- Usability 28
- Scalability 28
- Compatibility 28
- Maintainability 28
- Data Integrity 29
- Compliance 29
- Business Rules 29

Introduction

Purpose

This Software Requirements Specification (SRS) document outlines the functional and nonfunctional requirements for the Video Rental System (VRS). It aims to provide a comprehensive understanding of the system's capabilities and functionalities to stakeholders involved and developers.

The document is organised into sections covering different aspects of the system, starting with an overview of the purpose and scope, followed by detailed functional and non-functional requirements.

Scope of Project

The Video Rental System (VRS) is an online platform designed to facilitate movie renting and purchasing for customers while providing inventory management features for staff. The system enables customers to browse through a diverse collection of movies, view details such as genres, cast, and plot summaries, and rent or purchase desired titles, for selected duration.

Staff members manage inventory, monitor customer transactions, and ensure timely returns. Additionally, the system includes features to support the manager's role, such as overseeing inventory, performing audits on sold and rented movies, and managing the system's overall functionality.

The system aims to enhance user experience, streamline operations, and maximize customer satisfaction.

Document Conventions

Term	Convention
Software Requirements Specification (SRS)	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Database	Collection of all the information monitored by this system.
Video Rental System (VRS)	The software referred to in this SRS Document.
Inventory	The stock of each movie in the database.
Credentials	The login details for each user i.e. their selected username and password, which are then utilised for authentication.
Audit	Accessing statistical data to make informed business decisions.
Button	A clickable entity on a page with some associated action.
System	A web server for handling HTTP requests and an application server for executing application logic and managing interactions with databases or external services.
Staff Member	An employee of the company with low level clearance to perform day to day tasks.
Manager	An employee of the company with high level clearance to perform more involved and complicated tasks.
User	A customer of the business for whom the VRS has been constructed.

References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

Overall Description

Product Perspective

The Video Rental System (VRS) is a standalone product designed to provide an online platform for renting and purchasing movies. It is not part of a product family but rather a self-contained system. VRS serves as a replacement for traditional brick-and-mortar video rental stores, offering customers a convenient way to access a wide range of movies from the comfort of their own homes. The system interfaces with external payment gateways for transaction processing and may integrate with external databases for movie information.

Product Functions

- User authentication for Customers, Staff, and the Manager.
- Movie browsing by genre and search functionality.
- Adding movies to the cart for renting or purchasing.
- Connecting with an external payment gateway for final order placement.
- Generating invoices for movies that are to be purchased/rented.
- Inventory management for Staff and the Manager.
- Customer profile management, including order history.
- Notification system for staff regarding inventory status.
- Recruitment of Staff, auditing orders, and updating movie details for the Manager.
- Recommendation system based on customers’ order history.

User Classes and Characteristics

- **Customers**: Regular users of the system who browse, rent, and purchase movies. They may vary in terms of movie preferences and frequency of use.
- **Staff**: Employees responsible for managing inventory, processing transactions, and monitoring customer activities. They get access to administrative functions.
- **Manager**: Oversees the entire system, including inventory management, staff activities, and system functionality. The manager has higher-level privileges compared to staff members.

Operating Environment

The VRS system is designed to operate in a web-based environment. It is platform-independent and can be accessed through popular web browsers such as Chrome, Firefox, and Safari. The system requires a stable internet connection for both users and staff members. Additionally, VRS interfaces with external payment gateways for transaction processing.

Design and Implementation Constraints

- The system must comply with relevant data protection regulations to ensure the security and privacy of user data.
- Hardware limitations, such as server capacity and network bandwidth, may impact system performance, particularly during peak usage times.
- Integration with external databases for movie information may require adherence to specific data formats and protocols.

External Interface Requirements

User Interfaces

The user interface for the Video Rental System (VRS) will be implemented as a web application following the MERN stack architecture. The following are the logical characteristics of each interface between the software product and the users:

- **Customer Interface**: Customers will interact with the system through a user-friendly web interface. They will navigate through various pages to browse movies, add them to the cart, and complete transactions.
- **Staff Interface**: Staff members will access administrative functionalities through a separate interface. They will have access to features such as inventory management, order processing, and customer support.
- **Manager Interface**: The manager will have a specialised interface with additional privileges for overseeing the entire system, including inventory management, staff recruitment, and system configuration.

Hardware Interfaces

The VRS system is designed to operate on standard computing hardware and devices with internet connectivity. Users can access the system using desktops, laptops, tablets, and smartphones. There are no specific hardware dependencies or interfaces required for the operation of the software.

Software Interfaces

- **Database**: The system interacts with a MongoDB database to store movie information, user data, and transaction records.
- **Operating System**: VRS is platform-independent and can run on any operating system that supports web browsers compatible with the MERN stack (MongoDB, Express.js, React, Node.js).

- **External APIs**: The system integrates with an external payment gateway for processing transactions securely.

Communication Interfaces

- **HTTP/HTTPS**: Communication between the client-side and server-side components of the application will occur via HTTP/HTTPS protocols.
- **Email**: The system utilises the users’ email information for communication purposes, allowing the staff to access users’ contact details as well.
- **Web Browser**: Users will access the system using web browsers such as Chrome, Firefox, Safari, etc. The system is designed to be compatible with modern web standards and responsive across different browser versions.

System Features

Secure Login

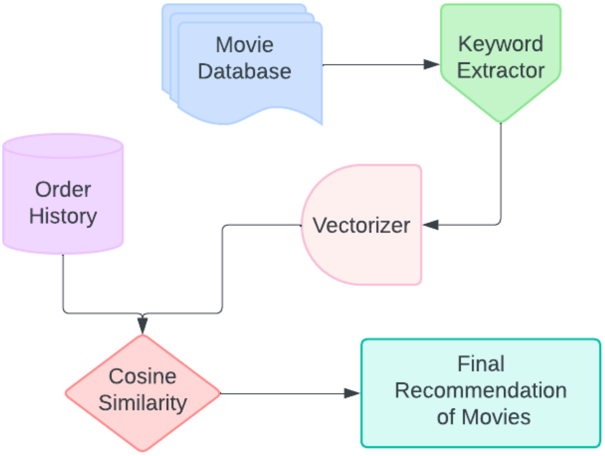
Our application implements a secure authentication system using the MERN stack and JSON Web Tokens (JWT). Users input their credentials on a login interface, and upon successful authentication, are directed to the appropriate interface based on their role.

In the MERN stack authentication process, JWT plays a vital role:

1. **Token Generation**: After successful authentication, the server generates a JWT token containing encoded user information, such as their identity and permissions.
2. **Token Signing**: The server signs the JWT token using a secret key, ensuring its integrity and authenticity.
3. **Token Transmission**: The signed JWT token is sent back to the client, typically as part of the response to the login request. The client securely stores this token, often in local storage or a cookie.
4. **Token Verification**: For subsequent authenticated requests, the client sends the JWT token along with the request. The server verifies the token's signature using the secret key to ensure its validity.
5. **Token Decoding**: Upon successful verification, the server decodes the JWT token to extract user information, enabling authentication and authorization of the requested actions.

Utilising JWT tokens in our MERN stack application ensures secure and stateless communication between client and server, enhancing security and scalability. MongoDB securely manages user login details and account information.

Recommender Model



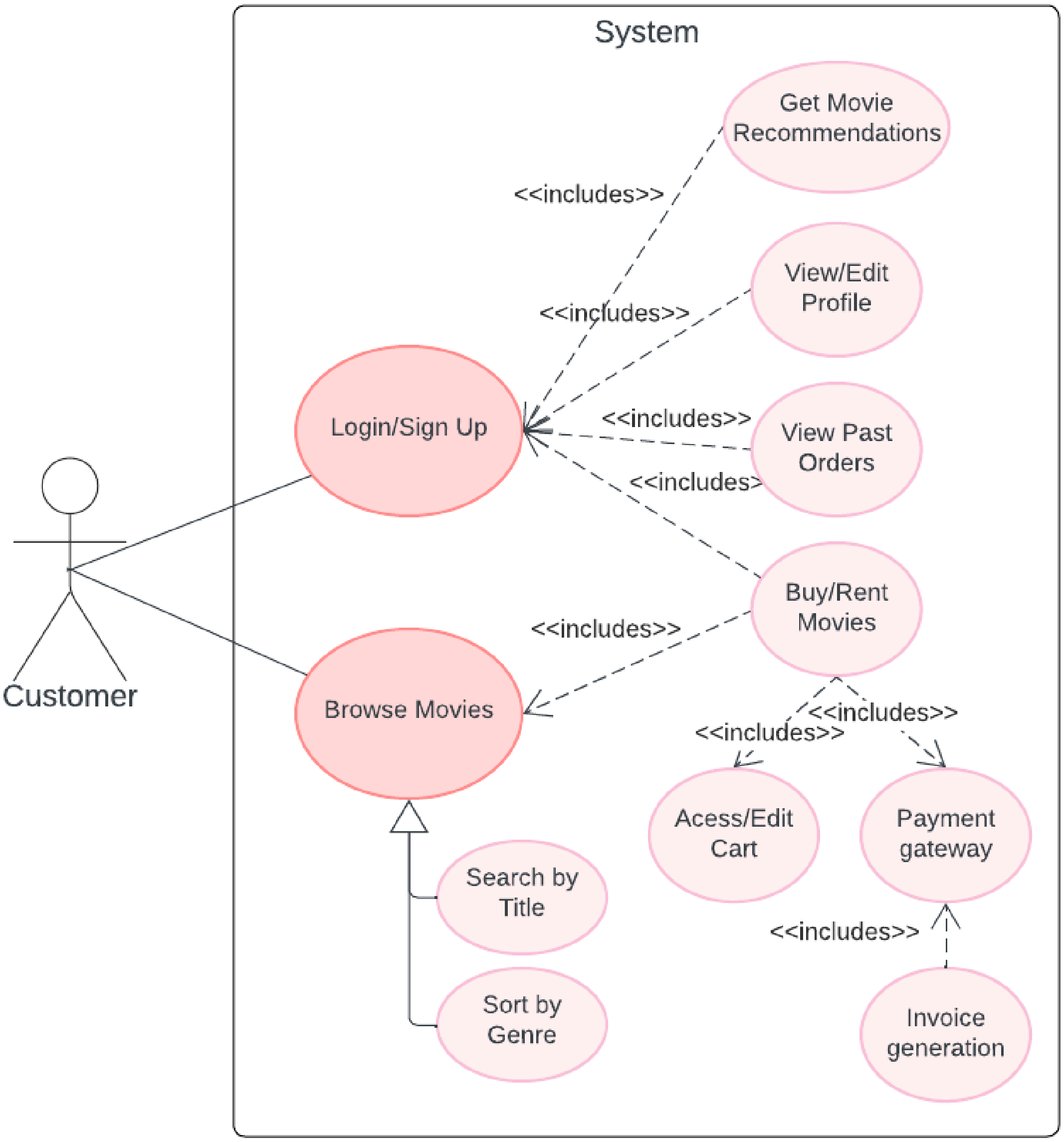
The movie recommender system leverages Rapid Automatic Keyword Extraction (RAKE) for identifying significant keywords and phrases from movie plots. These extracted keywords encapsulate key themes and plot points essential for understanding each film's narrative.

Next, the system utilises TF-IDF (Term Frequency-Inverse Document Frequency) vectorization to transform these keywords into numerical vectors. TF-IDF assigns weights to each term based on its frequency in the plot and rarity across all plots, capturing the importance of keywords while mitigating the impact of common terms.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

Once vectorized, the system calculates cosine similarity, a widely used metric, to quantify the similarity between the vectorized movie plots and user preferences. This similarity calculation enables the system to recommend movies with plot structures and themes closely aligned with the user's historical preferences, resulting in personalised and relevant movie suggestions.

Customer Features



Use Case: Sign Up

Description: New users can create an account by signing up on the platform.

Step-by-Step Description:

a. User accesses the signup/registration page of the platform.
b. User enters required information such as name, email, password, and other necessary details.
c. User clicks on the "Sign Up" or "Create Account" button.
d. System checks for any errors in the provided information (e.g., valid email format, password strength).
e. If there are no errors, the system creates a new account for the user.
f. If there are errors, the system displays error messages prompting the user to correct the information.
g. Once the account is successfully created, the user can log in using the credentials provided during signup.

Use Case: Login

Description: Existing users can log in to their accounts to access the platform's features.

Step-by-Step Description:

Before this use case can be initiated, the user must have accessed the Sign Up use case of the platform.

a. User accesses the login page of the platform.
b. User enters their username/email and password and clicks on the "Login" button.
c. System verifies the login credentials.
d. If the credentials are correct, the system grants access to the user's account.

e. If the credentials are incorrect, the system displays an error message.

Use Case: Categorical Sort of Movies

Description: Users can sort movies by genre.

Step-by-Step Description:

a. User accesses the movie sorting feature on the platform.
b. User selects a genre.
c. System sorts and displays movies according to the selected genre.

Use Case: Search for Movies

Description: The user can search for movies based on title.

Step-by-Step Description:

a. The user accesses the movie search feature on the platform.
b. The user enters a search query.
c. System displays a list of movies with titles matching the search criteria.
d. User selects a movie from the search results and is redirected to a detailed movie page.

Use Case: Add Movie to Cart

Description: After selecting a movie, users can add it to their cart for purchase.

Step-by-Step Description:
Before this use case can be initiated, the user must have accessed the Login use case of the platform.

a. The user selects a movie to buy/rent and accesses its description page.

b. User selects the "Add to Cart” feature and is then prompted with the options to buy/rent and select the quantity.
c. System adds the selected movie to the user's shopping cart.

Use Case: View Profile & Orders

Description: Users can view their profile information, including personal details and order history.

Step-by-Step Description:
Before this use case can be initiated, the user must have accessed the Login use case of the platform.

a. User accesses the profile section of the platform.
b. System displays the user's profile information and previous due and returned orders.

Use Case: Edit Profile

Description: Users can edit and update their profile information as needed.

Step-by-Step Description:
Before this use case can be initiated, the user must have accessed the View Profile & Orders use case of the platform.

a. User accesses the profile editing feature on the platform.
b. User makes changes to the desired profile information.
c. System saves the updated profile information.

Use Case: View & Edit Cart

Description: Users can view and edit the movies currently in their shopping cart before proceeding to place an order.

Step-by-By-Step Description:
Before this use case can be initiated, the user must have accessed the Login use case of the platform.

a. User accesses the shopping cart section of the platform.
b. System displays the movies currently in the users cart.
c. The user can then edit the quantity and duration of each movie in the cart, provided the Add Movie to Cart use case was accessed before.

Use Case: Place Order

Description: Users can proceed to checkout and place an order for the movies in their cart.

Step-By-Step Description:
Before this use case can be initiated, the user must have accessed the View and Edit Cart use case of the platform.

a. The user accesses the place order feature of the platform.
b. The user is then directed to the payment gateway.
c. Upon successful payment, the user is redirected to an invoice confirming their order, and their shopping cart is emptied.
d. Upon faulty use of the payment gateway, the user is redirected to the shopping cart.

Use Case: Get Recommendations

Description:

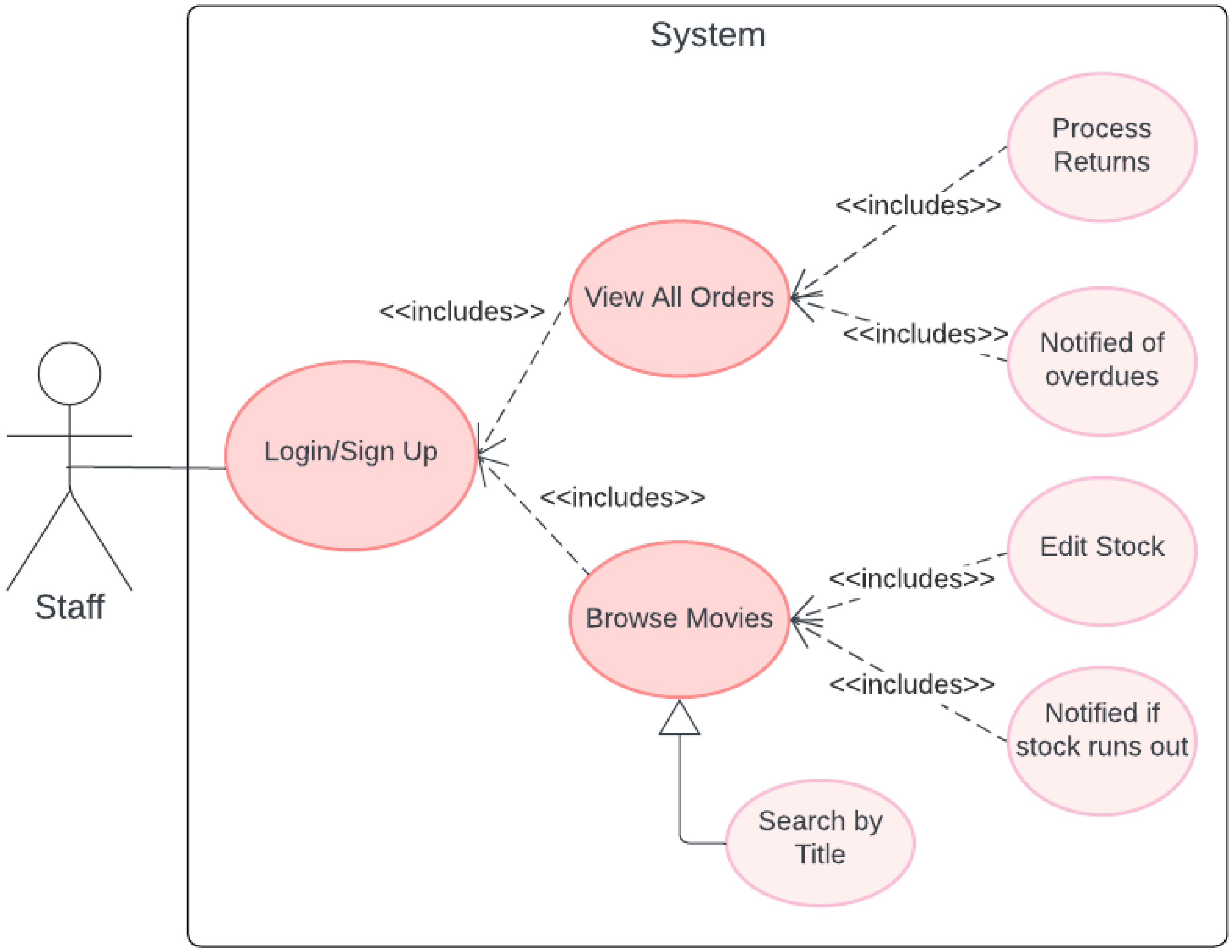
Users can receive personalised movie recommendations based on their viewing history and preferences.

Step-By-Step Description:

Before this use case can be initiated, the user must have accessed the Login and Place Order use case of the platform

a. User accesses the recommendations section of the platform.
b. System analyses user data, including viewing history, and preferences.
c. System generates and displays personalised movie recommendations for the user.

Staff Features



Use Case: Staff Login

Description: Staff members can log in to the platform using their credentials to access administrative functionalities.

Step-by-Step Description:

Before this use case can be initiated, the manager must have recruited the staff member by accessing the Recruit Staff use case.

a. Staff member navigates to the login page of the platform.
b. Staff member enters their username and password.
c. Staff member clicks on the "Login" button.
d. The system verifies the provided credentials against the database.
e. If the credentials are valid, the system grants access to the staff dashboard.
f. If the credentials are invalid, the system displays an error message and prompts the user to enter valid credentials.

Use Case: Manage Stock of Movies

Description: Staff members can view and edit the stock of movies available in the inventory.

Step-by-Step Description:

Before this use case can be initiated, the staff member must have accessed the Staff Login use case.

a. Staff member accesses the inventory management section of the staff dashboard.
b. Staff member views the list of available movies in the inventory, along with their quantities.

c. Staff member selects a movie to edit its stock.
d. Staff member updates the quantity of the selected movie.
e. Staff member saves the changes, and the system updates the inventory accordingly.

Use Case: View Due Orders and Overdue Orders

Description: Staff members can view orders that are due for return and access the customer details.

Step-by-Step Description:
Before this use case can be initiated, the staff member must have accessed the Staff Login use case.

a. Staff member accesses the "All Orders" section of the staff dashboard.
b. Staff member views the list of orders categorised as due, returned, or bought, along with details such as order ID, customer information, and return dates.

Use Case: Mark Movies as Returned

Description: Staff members can mark movies as returned when customers return rented items.

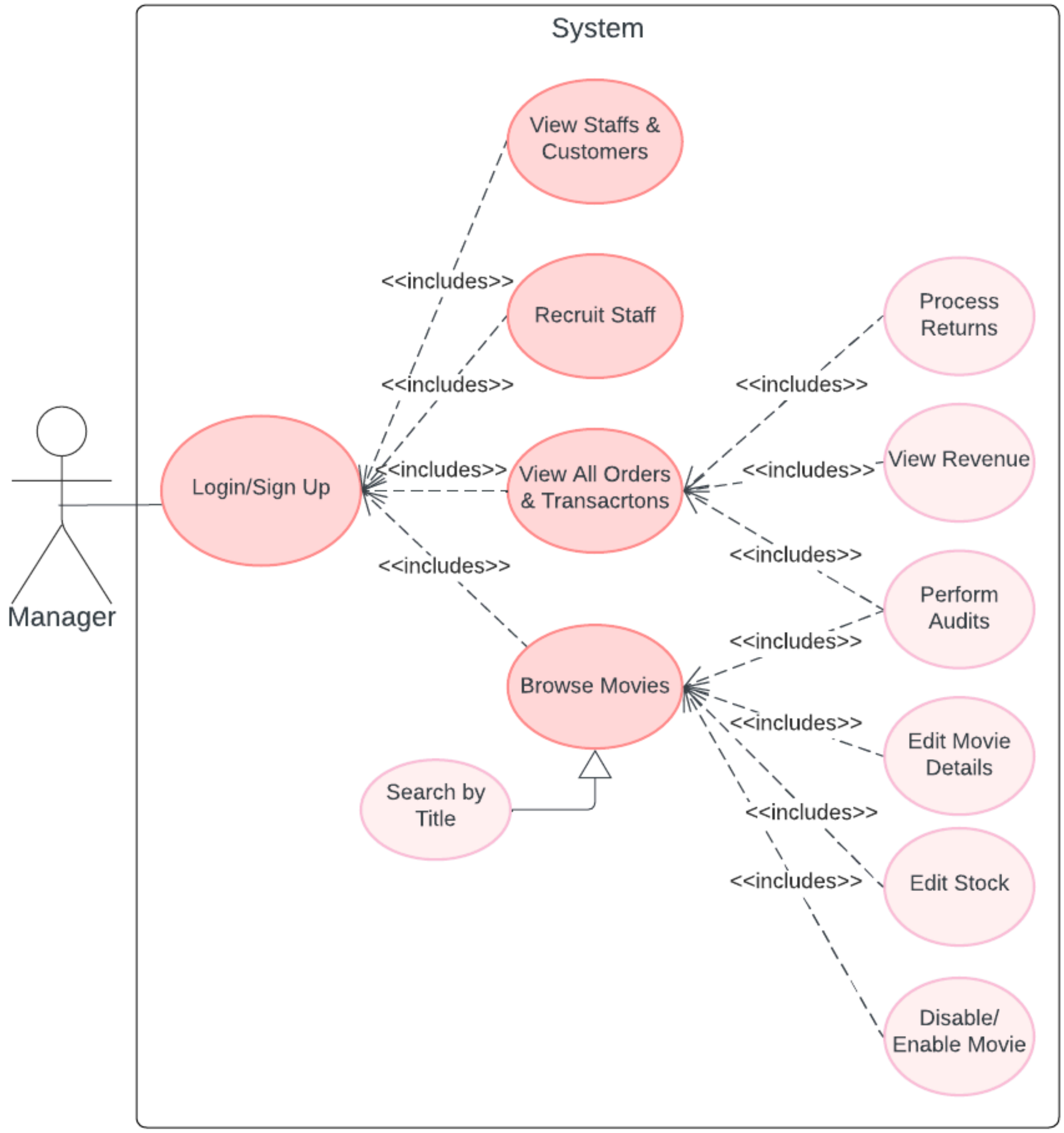
Step-by-Step Description:

Before this use case can be initiated, the staff member must have accessed the Staff Login use case.

a. Staff member accesses the “All Orders" page, which has a Due Orders section.
b. Staff member selects the order corresponding to the returned movie.

c. Staff member updates the order status to indicate that the movie has been returned.
d. The system updates the inventory to reflect the returned movie and adjusts the order status accordingly.

Manager Features



Use Case: Manager Login

Description: The manager can log in to the platform using their credentials to access administrative functionalities.

Step-by-Step Description:

a. Manager navigates to the login page of the platform.
b. Manager enters their username and password, and clicks on the “Login button”.
c. The system verifies the provided credentials against the database.
d. If the credentials are valid, the system grants access to the manager dashboard.
e. If the credentials are invalid, the system displays an error message and prompts the manager to enter valid credentials.

Use Case: Recruit Staff

Description: The manager can recruit new staff members to assist in managing the platform.

Step-by-Step Description:

Before this use case can be initiated, the manager must have accessed the Manager Login use case.

a. Manager accesses the recruit new staff section of the manager dashboard.
b. Manager fills in the required information for the new staff member, such as name, email, contact, and login details.
c. Manager clicks on the "Recruit Staff" button.
d. The system creates a new staff account with the provided information.
e. The new staff member can log in with their credentials on the staff side.

Use Case: Edit Movie Details

Description: The manager can edit details of existing movies in the inventory, such as genre, stock, and price.

Step-by-Step Description:

Before this use case can be initiated, the manager must have accessed the Manager Login use case.

a. Manager accesses the movie management section of the manager dashboard.
b. Manager chooses the movie to be edited from the list of available movies and chooses the edit movie option.
c. Manager updates the movie details as needed, including genre, stock and price.
d. Manager saves the changes, and the system updates the movie details in the inventory.

Use Case: Add Movies

Description: The manager can add new movies to the inventory for customers to rent or purchase.

Step-by-Step Description:

Before this use case can be initiated, the manager must have accessed the Manager Login use case.

a. Manager accesses the add new movies section of the manager dashboard.
b. Manager fills in the required information for the new movie, such as title, genre, cast, price, rating, and stock.
c. Manager uploads any associated media files, such as movie posters.

d. Manager saves the new movie entry, and the system adds it to the inventory.

Use Case: Disable and Enable Movies

Description: The manager can disable or enable movies in the inventory based on availability or other criteria.

Step-by-Step Description:

Before this use case can be initiated, the manager must have accessed the Manager Login use case.

a. Manager accesses the movie management section of the manager dashboard.
b. Manager chooses the movie to be disabled or enabled from the list.
c. Manager toggles the status of the selected movie to disable or enable.
d. The system updates the movie status accordingly, making it unavailable or available for customers to rent or purchase.

Use Case: View Due Orders and Returns by Staff

Description: The manager can view due orders, and track returns by staff members.

Step by Step Description:

Before this use case can be initiated, the manager must have accessed the Manager Login use case.

a. Manager accesses the all orders and returns section of the manager dashboard.
b. The system displays a list of due orders, including details such as order ID, customer information, return dates, and associated staff members (if any).

c. Manager also has access to a log of returned movies, showing which staff member processed each return and when it occurred.
d. Manager can analyse the data to monitor staff performance, track order fulfilment, and ensure timely returns.

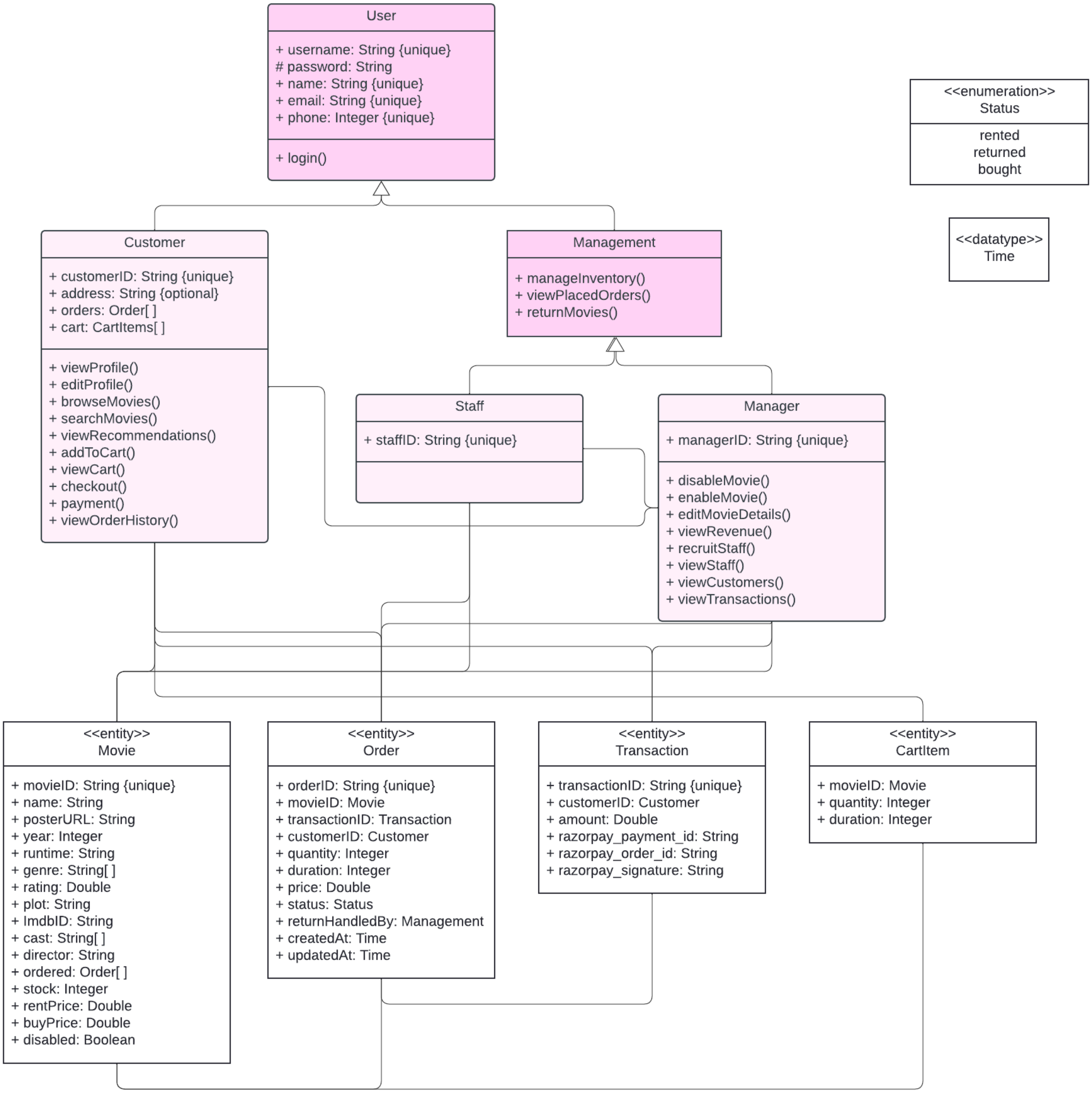
Use Case: Audit Movies

Description: The manager can audit movies to review the revenue generated by each movie over a specified period.

Step-by-Step Description:
Before this use case can be initiated, the manager must have accessed the Manager Login use case.

a. Manager can access the movie page for any specific movie and view details such as all the orders placed and the total revenue generated from that movie.
b. Manager may use the insights gained from the audit to make informed decisions regarding inventory management, pricing strategies, and promotional activities.

Class Diagram



Non Functional Requirements

Performance

- The system should respond to user interactions within 2 seconds under normal load conditions.
- Transaction processing time should not exceed 5 seconds.
- The system should be able to handle a minimum of 100 concurrent users without significant degradation in performance.

Security

- User passwords must be securely hashed and stored in the database.
- All communication between the client and server should be encrypted using HTTPS.
- Access to sensitive information and administrative functionalities should be restricted based on user roles and permissions.
- The system should implement measures to prevent SQL injection, cross-site scripting (XSS), and other common security vulnerabilities.

Reliability

- The system should have an uptime of at least 99.9%.
- Data backups should be performed regularly to prevent data loss in case of system failure.
- Error handling mechanisms should be in place to gracefully handle unexpected errors and exceptions.

Usability

- The user interface should be intuitive and easy to navigate, with consistent layout and design elements.
- Help resources and tooltips should be available to assist users in understanding system functionalities.

Scalability

- The system architecture should be designed to scale horizontally to accommodate increasing user traffic and data volume.
- Load balancing mechanisms should be implemented to distribute incoming requests evenly across multiple server instances.

Compatibility

- The system should be compatible with modern web browsers such as Chrome, Firefox, Safari, and Edge.
- The system should be responsive and accessible on a variety of devices, including desktops, laptops, tablets, and smartphones.

Maintainability

- The system code should follow best practices and coding standards to facilitate ease of maintenance and future enhancements.
- Documentation should be provided for system architecture, code structure, and deployment procedures to aid in system maintenance and troubleshooting.

Data Integrity

- The system should ensure the integrity of user data and transaction records through proper data validation and error checking mechanisms.
- Data backups and redundancy measures should be in place to prevent data loss due to hardware failure or system errors.

Compliance

- The system should comply with relevant data protection regulations such as GDPR and CCPA.
- Payment processing should adhere to PCI DSS standards to ensure the security of payment transactions and customer financial information.

Business Rules

The software will be free to use for all users and the source code will be publicly hosted for free use and modification.