Online Movie Rental Management Program

Composition of the project group:

- Jakub Juziuk

1. Introduction

The program will be a platform enabling users to easily rent movies online. The main functions of the system will include the ability to register and log in users, ensuring that each person has access to their individual account and rental history. Users will be able to browse a wide catalog of movies and series, filter them according to various categories (genres, ratings, new releases), and make quick rentals. The system will offer both a onetime rental option and a monthly subscription with unlimited access to movies within the payment framework. A built-in movie recommendation system will suggest titles based on the user's viewing history and preferences. Users will have the option to pay online, and the system will ensure high-quality movie playback on various devices (computer, tablet, smartphone).

2. System requirements specification

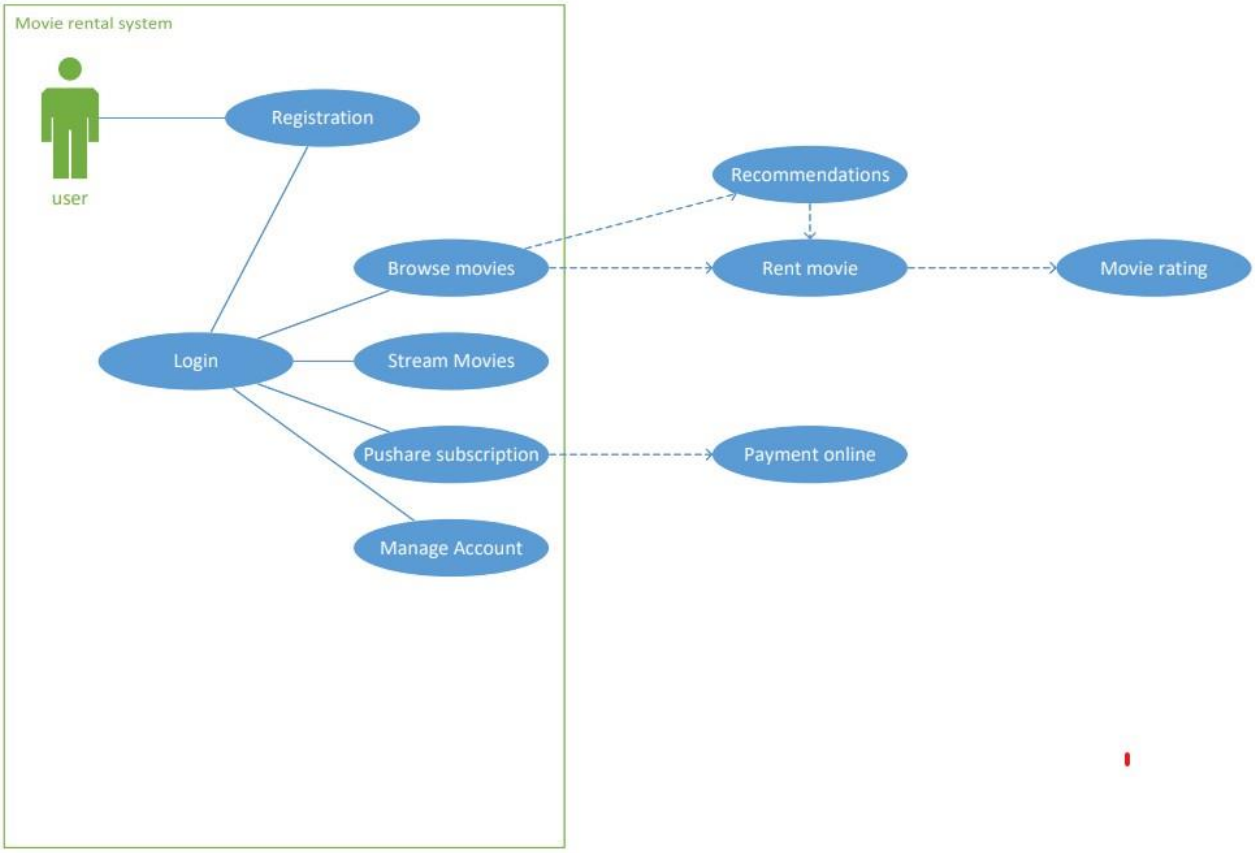
2.1. Functional requirements

- **User Management**: Allow users to register, log in, log out, and recover accounts.
- **Browse Catalog**: Enable users to search and filter movies/series by genre, rating, release year, etc.
- **Rental Options**: Offer one-time rentals and subscription-based access.
- **Recommendations**: Include an algorithm to recommend content based on user preferences and history.
- **Payment System**: Provide secure online payment options for rentals and subscriptions.
- **Streaming**: Deliver seamless playback of movies/series on multiple devices.

2.2. Non-functional requirements

- **Performance**: Ensure smooth streaming and fast response times under heavy load.
- **Scalability**: Support increasing numbers of users and expanding content catalog.
- **Security**: Protect user data and transactions with robust encryption.
- **Cross-Platform Compatibility**: Ensure the platform works on desktops, tablets, and smartphones
- **Reliability**: Guarantee high availability with minimal downtime.

2.3. Use cases



1.1. Use Case 1: User Registration

- **Use Case Name**: User Registration
- **Actor(s)**: User
- **Description**: The user creates a new account on the platform by providing their personal information (name, email, password).
- **Preconditions**: The user is not yet registered in the system.
- **Basic Flow :**
    1. User selects the "Register" option.
    2. User provides required information (name, email, password).
    3. The system validates the data and creates a new account.
    4. The user receives a confirmation of successful registration.

1.2. Use Case 2: User Login

- **Use Case Name**: User Login
- **Actor(s)**: User
- **Description**: The user logs into their account to access the platform.
- **Preconditions**: The user has a registered account.
- **Basic Flow** :
    1. User selects the "Login" option.

2. User enters email and password.
3. The system verifies the login credentials.
4. User gains access to their account.

1.3. Use Case 3: Browse Movies

- **Use Case Name**: Browse Movies
- **Actor(s)**: User
- **Description**: The user browses through the catalog of movies and TV shows, filtering them by various categories (genres, ratings, new releases).
- **Preconditions**: The user is logged into the system.
- **Basic Flow** :
    1. User selects the "Browse Movies" option.
    2. The system displays the movie catalog.
    3. User can apply filters (genre, rating, release year) to narrow down results.
    4. User selects a movie to view its details.

1.4. Use Case 4: Rent Movie

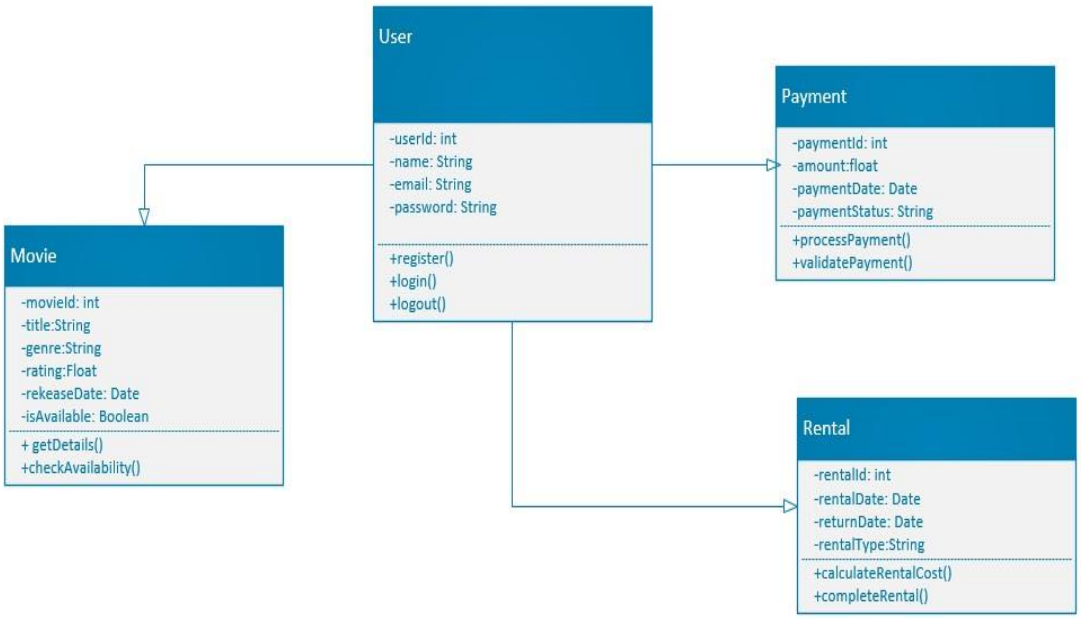
- **Use Case Name**: Rent Movie
- **Actor(s)**: User
- **Description**: The user rents a movie either on a one-time basis or through a subscription plan.
- **Preconditions**: The user is logged in and has selected a movie.
- **Basic Flow** :
    1. User selects a movie to rent.
    2. User chooses either the one-time rental or subscription option.
    3. The system processes the payment.
    4. The movie is available for viewing for a specific period.

1.5. Use Case 5: Purchase Subscription

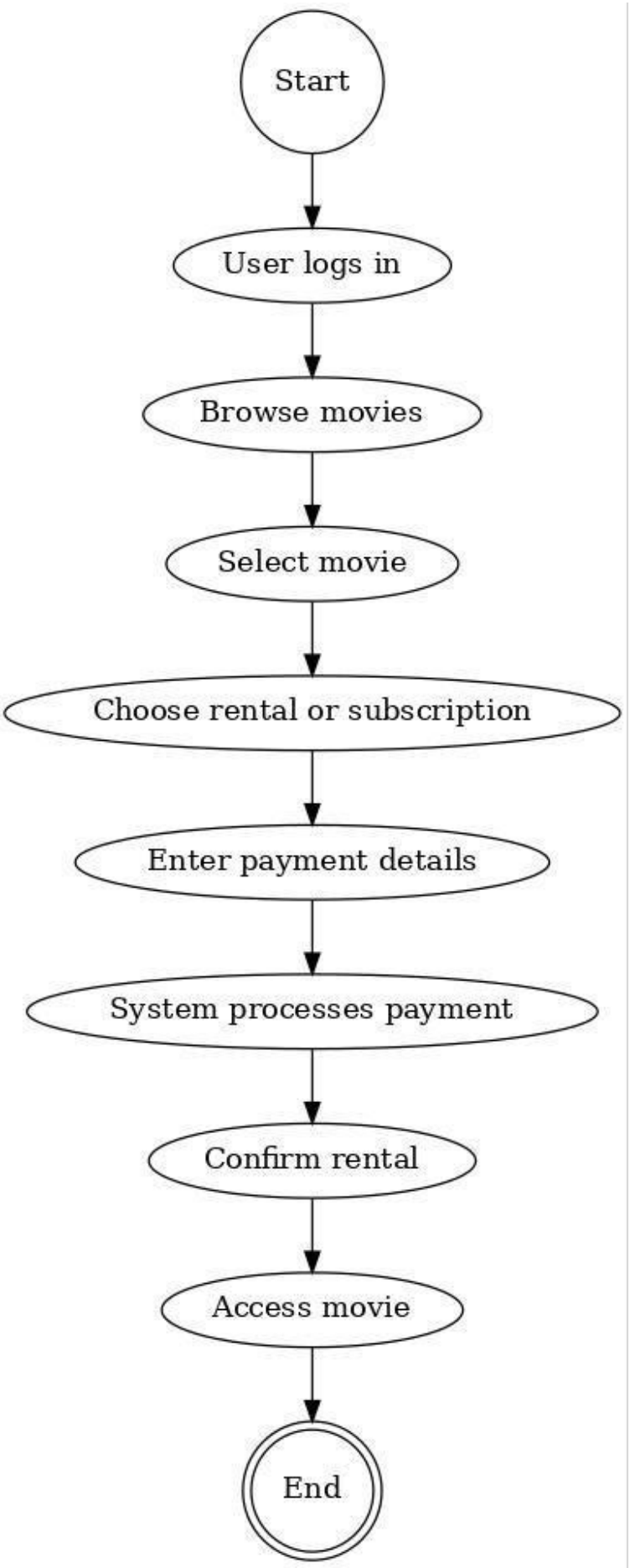
- **Use Case Name**: Purchase Subscription
- **Actor(s)**: User
- **Description**: The user purchases a subscription plan to gain unlimited access to the movie catalog.
- **Preconditions**: The user is logged in.
- **Basic Flow** :
    1. User selects the "Purchase Subscription" option.
    2. User chooses a subscription plan (e.g., monthly, yearly).
    3. User provides payment details.
    4. The system processes the payment and activates the subscription.

3. System model

3.1. Class diagram



3.2. Activity diagram


Start
User logs in
Browse movies
Select movie
Choose rental or subscription
Enter payment details
System processes payment
Confirm rental
Access movie
End

4. Graphic design

Key Interface Elements:

- **Homepage:**
  The homepage features a large banner highlighting selected or new releases, a prominently placed search bar, and quick access buttons for user login or registration. Below the banner, users can see curated collections such as “Top Rated,” “New Releases,” and “Trending Now.”

- **Movie Catalog Page:**
  This page displays a grid-based layout of movie and series thumbnails. Users can apply filters by genre, release year, or rating. Each item includes a title, cover image, and a quick-view button to see more details or proceed to rental.

- **Movie Detail Page:**
  When a user selects a movie, a detail page is shown with an expanded description, cast information, a trailer (if available), user reviews, and rental or subscription options.

- **User Dashboard:**
  After logging in, users can access their personalized dashboard. It displays recent activity, rental history, active subscriptions, recommended titles based on viewing habits, and account settings.

- **Responsive Design:**
  The entire interface is responsive and adapts to different screen sizes. Mobile users are presented with a simplified, touch-friendly layout, while desktop users benefit from a more extensive and visually rich experience.

Visual Style:

- **Color Scheme:**
  A dark theme is preferred to create a cinematic atmosphere and enhance focus on video content. Bright accent colors are used for buttons and key actions (e.g., rent, subscribe).

- **Typography:**
  The font selection is modern and legible, ensuring readability on both large and small screens. Headings are bold and clear, while body text is neutral and consistent.
- **Icons and Navigation:**
  Simple and minimalistic icons are used for navigation and action buttons to keep the interface uncluttered. Key interactions (e.g., filtering, playback) are placed where users naturally expect them.

5. System implementation

The system will be developed as a full-stack web application using modern and scalable technologies to ensure performance, maintainability, and a smooth user experience.

5.1. Architecture

The platform will follow a **client-server architecture** based on a RESTful API, with clear separation of concerns between frontend, backend, and database layers. The implementation will ensure modularity and support future feature expansion.

5.2. Technologies

- **Frontend**:
  - Developed using **React.js** or **Vue.js** for building dynamic, component-based interfaces.
  - Styled with **CSS (Tailwind or Bootstrap)** to ensure responsiveness and consistent UI design.
  - Axios (or Fetch API) for handling API calls to the backend.
- **Backend**:
  - Implemented in **Node.js with Express** or **Django (Python)** for managing business logic, authentication, and API endpoints.
  - Integration with a **recommendation engine module**, possibly using collaborative filtering or content-based filtering techniques.
- **Database**:
  - **PostgreSQL** or **MySQL** for storing structured data such as user accounts, movie catalog, rental history, and payments.
  - Optional: **Redis** or similar caching solution to optimize frequently accessed data (e.g., popular movies).
- **Authentication & Authorization**:
  - Handled using secure practices like **JWT (JSON Web Tokens)** or session-based authentication. ○ Role-based access control (e.g., admin vs. user) if needed.
- **Payment Integration**:
  - Integration with trusted third-party providers like **Stripe** or **PayPal** to process onetime rentals and subscriptions securely.
- **Streaming & Media Hosting**:
  - Video content will be hosted on a dedicated server or external storage (e.g., AWS S3).
  - Streaming will be enabled via **HLS** (HTTP Live Streaming) or **DASH** for adaptive bitrate delivery.
- **Deployment**:
  - Hosted on a cloud platform like **Heroku**, **Render**, or **AWS**.
  - CI/CD pipelines (e.g., GitHub Actions, GitLab CI) may be implemented for automated testing and deployment.

5.3. Development Workflow

The implementation will follow **Agile** or **Scrum** methodology with the team working in short development sprints. Key practices include:

- Version control with **Git** (hosted on GitHub or GitLab).
- Issue tracking and task management using **Trello**, **Jira**, or GitHub Projects.

- Regular code reviews and testing (unit, integration, and end-to-end).

5.4. Testing

To ensure quality and reliability, the system will include:

- **Unit tests** for critical backend logic and services.
- **Integration tests** for API endpoints.
- **End-to-end tests** simulating user actions using tools like Cypress or Selenium.
- Manual testing of the UI to verify responsiveness and usability on various devices.