SOFTWARE ENGINEERING

Laboratory 2/3

Subject: Specification of functional and non-functional requirements

1. Overview of the Project

The project focuses on creating an online platform where users can easily rent movies and TV shows. The platform will offer functionalities for user account management, browsing a wide range of content, making one-time rentals or subscribing to a monthly plan, and paying securely online. It will also feature a recommendation system based on user preferences and viewing history, as well as seamless streaming on multiple devices, including desktops, tablets, and smartphones.

2. Functional Requirements

1. User Management:

- Users must be able to register, log in, and log out securely.
- Users should have the option to recover their accounts in case of forgotten credentials.
- Each user will have an individual account with a personal history of rentals and preferences.

2. Catalog Browsing:

- Users will be able to search and filter movies and TV shows by genres, ratings, release year, and other relevant criteria.
- The catalog should include a wide variety of content, allowing users to explore different genres and discover new titles.

3. Rental Options:

- The platform will provide two types of rental options:
  - **One-time rentals**: Users can rent individual titles for a specified period.
  - **Subscription service**: Users can subscribe to a monthly plan with unlimited access to the entire content library.

4. Recommendations:

- An intelligent recommendation system will suggest movies and TV shows to users based on their viewing history, preferences, and similar user behavior.
- Recommendations will be displayed on the homepage or personalized pages of users.

5. Payment System:

- Users must have the option to make secure online payments for both one-time rentals and subscription services.
- Payment options should include common methods like credit cards, debit cards, and online payment platforms (e.g., PayPal).

6. Streaming:

- The platform must deliver seamless video playback with high-quality streaming.
- Streaming must be compatible with multiple devices (e.g., computers, tablets, smartphones) and should adapt to various screen sizes and internet connection speeds.

---

3. Non-Functional Requirements

1. Performance:

- The system must ensure fast response times even under heavy load, especially during peak usage times (e.g., weekends, evenings).
- Streaming should be smooth with minimal buffering, and content should be available for immediate playback.

2. Scalability:

- The system should be scalable to accommodate an increasing number of users as the platform grows.
- As the catalog of films and TV shows expands, the platform should be able to handle the additional data without compromising performance.

3. Security:

- All user data (including payment information) should be protected by robust encryption and stored securely.
- Transactions should be processed in compliance with industry standards for security (e.g., PCI-DSS for payment processing).

4. Cross-Platform Compatibility:

- The platform must be compatible with various operating systems (e.g., Windows, macOS, Android, iOS) and devices (e.g., desktop computers, smartphones, tablets).

- The user interface (UI) should adapt to different screen sizes and ensure a smooth experience across devices.

5. Reliability:

- The platform should be highly reliable with minimal downtime.
- Backups and failover systems should be in place to prevent data loss and ensure service availability at all times.

---

4. Compliance Requirements

- The platform will adhere to privacy and data protection regulations (e.g., GDPR) to ensure that users' personal data is handled appropriately.
- The system will meet industry standards for accessibility, making sure it is usable for people with disabilities.

---

5. Requirements Engineering Process

The requirements engineering process will follow these steps:

1. **Identification of Requirements:** Gather the key functionalities and non-functional needs from stakeholders (including end-users and business clients).
2. **Analysis and Negotiations:** Collaborate with stakeholders to refine and prioritize requirements.
3. **Specification of Requirements:** Write detailed requirements documentation, including functional and non-functional aspects, ensuring clarity and completeness.
4. **System Modeling:** If applicable, create system models (e.g., data flow diagrams, UML diagrams) to represent the system’s architecture and interactions.
5. **Requirements Validation:** Regularly validate the requirements with stakeholders to ensure they are correct, complete, and aligned with the project’s goals.

---

6. Quality Criteria for Requirements

Good requirements must be:

- **Correct**: Clearly specify what the system must do.
- **Complete**: Cover all necessary aspects of the system.
- **Consistent**: Avoid contradictions between requirements.
- **Unambiguous**: Written in a way that leaves no room for multiple interpretations.
- **Verifiable**: Able to be tested and verified during the development process.

- **Modifiable**: Easy to update and maintain as the project evolves.
- **Traceable**: Each requirement should have a clear origin and be linked to its source.
- **Understandable**: Clear and easy for both developers and users to understand.

---

7. IEEE Standards for Requirements Specification

The project will follow the following IEEE standards:

- **IEEE 610**: Standard Glossary of Software Engineering Terminology.
- **IEEE 830**: Software Requirements Specification (SRS).
- **IEEE 1233**: Guide for the Development of Software Requirements Specifications.
- **IEEE 1362**: Standard for Software Design Descriptions.
- **IEEE 12207**: Software Life Cycle Processes.