

# TEST PLAN DOCUMENT

FOR

## VIDEO RENTAL SYSTEM ( VERSION 1.2 )

---

TEAM:

**PRANAYA CHOWDARY - 22CS10070**

**HARSHIT JAIN - 22CS10030**

**SHASHWAT SINGH RANKA - 22CS10071**

**CSE DEPT, IIT KHARAGPUR**

DATE:

**9 APRIL 2024**

---

## Table of Contents

<b>Table of Contents</b>	<b>2</b>
<b>Test Plan Identifier</b>	<b>3</b>
<b>References</b>	<b>3</b>
<b>Introduction</b>	<b>3</b>
<b>Software Risk Issues</b>	<b>3</b>
<b>Features to be Tested</b>	<b>4</b>
<b>Approach</b>	<b>7</b>
<b>Item Pass/Fail Criteria</b>	<b>8</b>
<b>Suspension Criteria</b>	<b>9</b>
<b>Test Deliverables</b>	<b>10</b>
<b>Staffing Requirements</b>	<b>11</b>
<b>Responsibilities</b>	<b>12</b>
<b>Schedule</b>	<b>13</b>
<b>Planning Risks and Contingencies</b>	<b>14</b>

## Test Plan Identifier

- Project: Video Rental System (VRS)
- Test Plan Version: 1.0

## References

- Software Requirements Specification (SRS) document v1.0
- IEEE 829 Standard for Software Test Documentation

## Introduction

This Test Plan provides a comprehensive overview of the testing approach, strategies, and resources needed to ensure the successful implementation of the Video Rental System (VRS) software project. It serves as a roadmap for the testing efforts throughout the project lifecycle, guiding the testing team in executing various testing activities effectively and efficiently.

The scope of this Test Plan encompasses all aspects of testing for the VRS project, including functional testing, integration testing, performance testing, security testing, and usability testing. It defines the testing objectives, identifies the test items, and outlines the responsibilities of the testing team members.

Throughout the testing process, the Test Plan will be closely aligned with the Software Project Plan and other relevant project documentation, such as the Software Requirements Specification (SRS) document. This alignment ensures that testing activities are consistent with project goals and requirements.

## Software Risk Issues

The integration of payment gateways and external databases is crucial for the functioning of the VRS. Delays or issues with the delivery of these third-party products may impact the project schedule and functionality.

## Features to be Tested

### User Registration and Login:

Description: Verify that users can successfully register for an account and log in to the system.

Risk Level: H (High)

Justification: Registration and login are critical functionalities that directly impact user access to the system. Any issues in this area could prevent users from accessing the platform, leading to dissatisfaction and potential loss of customers.

### Movie Browsing and Selection:

Description: Ensure users can browse through the available movie catalogue and select desired movies for rental or purchase.

Risk Level: M (Medium)

Justification: While movie browsing and selection are essential user activities, issues in this area may not directly prevent access to the system. However, any difficulties or limitations in browsing and selecting movies could impact user experience and satisfaction.

### Order Placement and Payment Processing:

Description: Test the functionality for users to place orders for renting or purchasing movies and process payments securely.

Risk Level: H (High)

Justification: Order placement and payment processing are critical for the core functionality of the system. Any failures or errors in these processes could result in financial loss, customer frustration, and damage to the system's reputation.

### **Inventory Management (Staff):**

Description: Validate that staff members can effectively manage the inventory by adding, updating, and removing movies as needed.

Risk Level: H (High)

Justification: Inventory management is crucial for maintaining the availability and accuracy of movie listings. Any issues in this area could lead to discrepancies in the catalogue, affecting user experience and business operations.

### **Recruiting Staff:**

Description: Managers should be able to effectively recruit staff members for various roles within the organisation. This involves tasks such as posting job listings, reviewing applications, conducting interviews, and making hiring decisions.

Risk Level: M (Medium)

Justification: While recruiting staff is crucial for maintaining a skilled and efficient workforce, failures or issues in this area may not directly impact user access or functionality. However, it can impact the organisation's ability to meet staffing needs and maintain productivity.

### **Editing Movie Details:**

Description: Managers need the ability to edit movie details within the system, including updating information such as titles, descriptions, release dates, genres, and cast/crew details. This ensures that the movie database remains accurate and up-to-date.

Risk Level: M (Medium)

Justification: Editing movie details is important for maintaining the quality and relevance of the content available to users. While issues in this area may not directly impact user access, they can lead to inconsistencies or inaccuracies in the movie database, affecting user experience and trust in the system.

### **Auditing Revenue:**

Description: Managers must be able to audit revenue generated by the system, including ticket sales, subscriptions, merchandise sales, and advertising revenue. This involves reviewing financial records, conducting analyses, and ensuring compliance with accounting standards.

Risk Level: M (Medium)

Justification: Auditing revenue is critical for financial management and decision-making within the organisation. While failures or issues in this area may not directly impact user access or functionality, they can have significant implications for the organisation's financial health, regulatory compliance, and strategic planning.

### **Integration with External Systems:**

Description: Test integration with external systems such as payment gateways and databases to ensure seamless operation and data exchange.

Risk Level: H (High)

Justification: Integration with external systems is critical for the system's functionality and performance. Any disruptions or failures in these integrations could result in transaction errors, data loss, and system downtime, impacting user experience and business operations.

### **Recommendation System:**

Description: Test the recommendation system's functionality based on data science algorithms to suggest movies to users.

Risk Level: L (Low)

Justification: The recommendation system is a key feature that enhances user experience and engagement. Any issues in recommendation accuracy or performance could affect user satisfaction and platform usability.

### **Notification System:**

Description: Test the notification system to alert users and staff when stock is low or movies are overdue.

Risk Level: M (Medium)

Justification: The notification system plays a crucial role in user engagement and inventory management. Failures in delivering timely notifications could lead to missed opportunities or disruptions in operations.

## **Approach**

### **6.1 Testing Levels:**

The testing for the Attendance system will encompass Unit, System/Integration, and Acceptance test levels.

#### **Unit Testing:**

Developers will conduct unit testing and collectively approve the tests. All unit test results will be shared with the testing team for reference.

#### **System/Integration Testing:**

Conducted by the test manager and development team with mutual assistance. No specific test tools are required. Programs will enter this phase after critical defects have been resolved. A program may proceed with up to two major defects, provided they do not hinder testing. System/Integration testing aims to ensure seamless integration and functionality across all components.

#### **Acceptance Testing:**

End users will perform acceptance testing with guidance from the test manager and developers. End users will be provided with acceptance test cases in advance. Programs will enter this phase after critical and major defects have been addressed. A program may have

one major defect, if manageable, but all critical and major defects must be resolved before final acceptance.

Acceptance testing ensures that the software meets user requirements and expectations.

## **6.2 Defect Reporting:**

### **During Unit Testing:**

The development team will identify and report defects in the codebase.

Defects will be documented with details on origin, severity, proposed solutions, and estimated resolution time.

Information will be shared with the testing team and project manager for further analysis and resolution.

### **During System/Integration and Acceptance Testing:**

Focus on identifying defects not caught during Unit Testing or arising from module interactions.

Special attention to defects that should have been detected during Unit Testing.

Goal is to ensure software meets quality standards and is free from critical defects impacting user experience and data integrity.

## **Item Pass/Fail Criteria**

The completion criteria for this test plan will be determined based on the level of testing being conducted.

### **Unit Test Level:**

All test cases completed.

A specified percentage of cases completed with a percentage containing some number of minor defects.

Code coverage tool indicates all code covered.



### **System/Integration Test Level:**

All lower level plans completed.

A specified number of plans completed without errors and a percentage with minor defects.

### **Acceptance Test Level:**

All critical and major defects resolved before final acceptance.

A specified percentage of test cases passed without errors.

### **Defect Evaluation:**

Number and severity of defects located will be documented and evaluated.

Comparison with the total number of defects may not be feasible due to undetected defects.

Defects that may cause a failure, yet acceptable to remain in the application, will be identified.

Failures resulting from defects will be assessed based on user impact, such as system crashes or incorrect functionality.

## **Suspension Criteria**

### **Suspension Criteria:**

Testing will be suspended if the number or severity of defects reaches a point where further testing provides no additional value.

If a truly fatal error is encountered during testing, all subsequent testing will be suspended to prevent the generation of false defect reports.

### **Acceptable Defect Level:**

The acceptable level of defects that will allow testing to proceed will be defined based on the severity and impact of the defects.

Minor defects that do not significantly impact system functionality may be tolerated if they do not impede the testing process.

Critical defects that pose a risk to system stability or user experience will require resolution before testing can resume.

### **Resumption Requirements:**

Testing may resume once the identified defects have been addressed and verified by the development team.

A re-evaluation of the testing environment and test cases may be necessary to ensure that testing can proceed effectively after suspension.

Any issues or lessons learned from the suspension period will be documented and incorporated into the testing process to prevent similar occurrences in the future.

## **Test Deliverables**

As part of this test plan, the following deliverables will be provided:

Test Plan Document: Comprehensive document outlining the testing approach, strategy, and test items.

Test Cases: Detailed descriptions of test scenarios, procedures, and expected outcomes.

Test Design Specifications: Documentation outlining the design and structure of test cases and procedures.

Tools and Their Outputs: Any testing tools used during the testing process along with their outputs and reports.

**Simulators:** If applicable, simulators used for emulating specific environments or conditions during testing.

**Static and Dynamic Generators:** Tools or scripts used for generating test data or simulating dynamic scenarios.

**Error Logs and Execution Logs:** Logs documenting errors encountered during testing and execution logs detailing test execution activities.

**Problem Reports and Corrective Actions:** Reports documenting identified defects and actions taken to resolve them.

It's important to note that the software itself is not considered a test deliverable as it falls under the test items and is delivered by the development team.

## Staffing Requirements

Staffing and training requirements for the successful execution of the test plan are as follows:

**Application/System Training:** All testing team members will undergo comprehensive training on the application/system to be tested. This training will include understanding the functionalities, workflows, and user interfaces of the application/system.

**Test Tool Training:** Training will be provided for any test tools to be used during the testing process. This includes familiarisation with the tools' features, functionalities, and usage procedures.

**Responsibility Assignment:** Responsibilities for testing and training will be assigned based on the sections outlined in the test plan. Testing responsibilities will be assigned to specific team members based on their expertise and domain knowledge. Similarly, training responsibilities will be assigned to ensure that all team members are adequately trained on the application/system and test tools.

By addressing staffing and training needs in alignment with the test plan sections, the testing team can ensure that all members are adequately prepared to execute their roles and responsibilities effectively.

## **Responsibilities**

Various responsibilities must be assigned to ensure the successful execution of the test plan:

**Setting Risks:** The project manager or testing lead is responsible for identifying and assessing risks associated with the testing process. This includes evaluating potential impacts on project timelines, resources, and deliverables.

**Selecting Features:** The testing lead, in collaboration with stakeholders, determines which features are to be tested and which are not. This decision is based on factors such as criticality, complexity, and priority.

**Overall Strategy:** The testing lead develops the overall testing strategy for this level of the plan, considering factors such as test objectives, scope, and resources available.

**Ensuring Readiness:** The testing lead ensures that all required elements for testing are in place, including test environments, resources, and documentation.

**Resolution of Conflicts:** The project manager or testing lead resolves scheduling conflicts, especially if testing is to be conducted on the production system. This may involve coordinating with other project stakeholders to prioritise testing activities.

**Training Provision:** The training coordinator or designated team member is responsible for providing the required training to testing team members. This includes training on the application/system, test tools, and testing processes.

**Critical Decisions:** The project manager, testing lead, or designated decision-maker is responsible for making critical go/no-go decisions for items not covered in the test plans. This may include decisions related to the readiness of the system for deployment or the resolution of critical defects.

By clearly defining responsibilities, the testing process can be effectively managed, ensuring that all necessary tasks are completed and decisions are made in a timely manner.

## **Schedule**

The schedule for testing should be based on realistic and validated estimates to ensure effective planning and execution. It is crucial to align the testing schedule with the overall project plan, as testing is an integral part of the development process.

It is important to recognize that testing is often the first area to face cuts when project timelines become tight. However, compromising on testing quality can lead to subpar outcomes and negatively impact the overall success of the project. Therefore, it is essential to prioritise testing activities and allocate sufficient time and resources for thorough testing.

Any slippage in the schedule should be addressed proactively. The impact of schedule slippage on testing and overall project delivery should be communicated transparently to stakeholders. By discussing potential effects in advance, stakeholders may be more understanding and willing to accommodate necessary adjustments to ensure a better-tested application.

Identifying relevant milestones and their relationship to the development process is essential for tracking progress and identifying potential slippage. Test dates should be directly tied to related development activity dates to prevent the test team from being perceived as the cause of delays. This ensures that testing activities are synchronised with development milestones and progress.

By employing dependent or relative dating, testing activities commence immediately following the completion of related development activities. This approach ensures that testing is seamlessly integrated into the development process and helps maintain alignment with project timelines.

## Planning Risks and Contingencies

Risks to testing and project success include:

**Resource Shortages:** Lack of skilled personnel may delay testing. Contingencies include resource reallocation or hiring.

**Resource Availability:** Delays in essential resources like hardware or software may impede testing. Contingencies involve alternative procurement or schedule adjustments.

**Late Deliveries:** Late delivery of software or tools can disrupt testing timelines. Contingencies include renegotiation of delivery schedules or prioritisation of critical tasks.

**Training Delays:** Delayed training on applications or tools may affect team readiness. Contingencies include accelerated training or alternative resources.

**Requirement Changes:** Changes in project requirements may disrupt testing. Contingencies involve impact assessments, plan adjustments, and effective communication.

Mitigation actions include schedule adjustments, resource allocation, overtime work, scope changes, resource optimization, and consideration of extreme measures as a last resort.