Test Plan

for

Video Rental System

Prepared by**:** ***Team Binary Brains***

Akshat Pandey (22CS10005)

Abhinav Akarsh (22CS30004)

Pranav Jha (22CS30061)

March 28, 2024

Table of Contents

1. Test Plan Identifier

This is Test Plan 1 for the first version of the Video Rental System.

2. References

Software Requirements - Specifications for this project.

3. Introduction

This test plan aims to lay out a methodical framework for testing the functionalities described in the SRS document for this project. This is needed to ensure that the various components of this web application are working as expected in a wide variety of situations. This helps deliver a smooth, glitch-free user experience to the students, wardens, etc.

4. Test Items

- **User Interface Testing**: Ensuring seamless navigation, efficient search functionality, and user-friendly movie browsing and selection processes, along with effective cart management and profile customization options.
- **Functional Features Testing**: Validating user registration and login processes, robust movie renting and purchasing functionalities and effective inventory management capabilities.
- **Performance Testing**: Evaluating system response time, scalability under varying loads, and the ability to handle multiple concurrent users without degradation.
- **Security Testing**: Verifying the effectiveness of authentication mechanisms and stringent access control measures to safeguard user data and system integrity.
- **Compatibility Testing**: Ensuring seamless functionality across web browsers and devices, including mobile responsiveness, for optimal user experience on various screen sizes.
- **Usability Testing**: Assessing the intuitiveness of the user interface, ease of navigation, and clarity of instructions and prompts to enhance user satisfaction and engagement.
- **Data Integrity Testing**: Validating the accuracy and consistency of user data and properly handling data transactions to maintain data integrity and reliability.

5. Software Risk Issues

- **Security Vulnerabilities**: Potential risks of data breaches, unauthorised access, or cyberattacks leading to loss of sensitive information and user trust.
- **Incorrect user management and privilege separation**: Different classes of users have different roles to play in the web app, and certain classes are not privy to certain information and abilities. They should not be allowed to perform certain actions. For example, students should not be allowed to submit action-taken reports for complaints filed by other students. That is a task that should be performed only by the wardens. If students can log in as wardens, this could result in a catastrophic lack of accountability.
- **Performance Bottlenecks**: Risks of slow response times, system crashes, or scalability issues under heavy loads impacting user experience and system reliability.
- **Integration Challenges**: Risks related to compatibility issues, data synchronisation problems, or service disruptions when integrating with external systems or third-party APIs.
- **Requirements Volatility**: Risks arising from changes in project requirements or scope, leading to project delays, budget overruns, or rework.

6. Features To Be Tested

These are the features to be tested from the user's perspective. This primarily consists of user-facing features handled by a graphical web page allowing privileged access to the internal server operations.
Procedure: These features will be tested by following a sequence of instructions for navigating the web pages associated with the application, followed by a verification of the final internal state of the server-side application

6.1 User Registration and Authentication

- **Registration Process:** Verify that users can create accounts easily, providing necessary information such as email, password, name and phone no.
- **Authentication:** Ensure that users can securely log in to their accounts using valid credentials, with appropriate error messages for invalid login attempts.
- **Password Management:** Test features such as password reset or recovery to ensure users can securely manage their account passwords.

6.2 Movie Browsing and Selection

- **Genre Browsing:** Verify that users can browse various movie genres to find movies of interest.
- **Movie Details:** Ensure that users can view detailed information about each movie, including title, description, release date, duration, rating, and cast.

6.3 Cart Management

- **Adding and Removing Items:** Verify that users can add movies to their cart and remove or update quantities of items already added.
- **Price Calculation:** Ensure that the system accurately calculates the total cost of items in the cart, including any applicable taxes or fees.
- **Checkout Process:** Test the checkout process, including entering payment information and confirming the order.

6.4 Order Processing

- **Order Submission**: Verify that users can successfully submit orders for rental or purchase with appropriate validation and error handling.
- **Payment Processing**: Test payment processing functionality and successful completion of transactions.
- **Invoice Generation**: Ensure that invoices are generated accurately and include relevant details such as order items, prices, taxes, and payment information.

6.5 Inventory Management

- **Stock Updates**: Test the system's ability to update inventory stock levels accurately based on user rentals, purchases, and returns.
- **Movie Rental Status**: Verify that movies are marked as *returned* or *not returned* for rental or purchase based on current stock levels.
- **Backorder Handling**: Ensure that the system can handle backorders or out-of-stock items appropriately, with notifications or alternative options provided to users

6.6 Search Functionality

- **Keyword Search**: Test the system's ability to search for movies based on keywords, titles, actors, directors, or other criteria, returning relevant results.
- **Search Result Accuracy**: Ensure that search results are accurate and up-to-date and include relevant movie details to help users make informed decisions.

6.7 User Profile Management

- **Profile Updates**: Test the functionality for users to update their profile information, including contact details.
- **Order History**: Verify that users can view their order history, including past rentals, purchases, and transactions, with relevant details such as dates, prices, and order statuses.

6.8 Compatibility and Usability

- **Cross-Browser Testing**: Test the system's compatibility with popular web browsers (e.g., Chrome, Firefox, Safari, Edge) to ensure consistent functionality and appearance across different platforms.
- **Mobile Responsiveness**: Verify that the system is responsive and displays correctly on various devices, including smartphones, tablets, and desktops, with optimised layouts and navigation for smaller screens.

7. Features Not To Be Tested

The payment portal will not be tested as it is a dummy portal. Since this is a one-off project, all other features must be tested.

8. Approach (Strategy)

8.1 General Interface Testing

- The tester will open the web application on various browsers to test for browser compatibility issues.
- They will click on all the links on the web page to ensure no part of the user interface contains broken links.
- They will try accessing web pages that should not be accessible to them to verify if the authentication system is bug-free.

8.2 Testing of Various Specific Subsystems

- The tester will open up the relevant web pages for the subsystem considered and follow the instructions in the test cases. These instructions will include which button to click, etc, followed by the expected system response.

- If there's a mismatch between the actual system response and expectations, a bug report will be filed, following which the developers will try to isolate the cause and fix it.

8.3 Testing Client Compatibility

- Apart from testing browser compatibility, the tester should also use various devices to ensure the application works smoothly across operating systems and hardware platforms.
- Internet bandwidth tests need to be performed to see if the web app is functional in low-bandwidth situations since it might be possible that poor network conditions prevail. The application must work even during such situations.
- The tester should also check if the web pages are rendered correctly for different configurations involving fonts, browser settings, JavaScript permissions, cookie permissions, etc.

8.4 What Metrics Will Testers Collect?

- Web page loading times.
- Response when under load.
- Errors involving bugs in the system.
- Differences across various client configurations (such as devices).
- Potential user interface complications such as convoluted menus and submenus.

9. Item Pass Fail Criteria

- All unit tests involving specific isolated subsystems must work successfully. No test can fail since any test failure implies that the subsystem is not properly accessible to the end user. A release version of this application cannot tolerate such issues.
- All client configurations must be tested, and if the application is not working on niche platforms (such as the older versions of Internet Explorer), they should be documented. The test will succeed if the application runs under reasonably modern systems. It DOES NOT have to work on older systems that are no longer widely used. This information will be relayed to the end-user, who will be asked to modify their configuration.

10. Suspension Criteria and Resumption Requirements

The tests associated with a subsystem should be suspended in only one case, i.e. when the main web page associated with the specific subsystem does not load in a client. This means other test cases related to that subsystem cannot even proceed. This is the only case where suspension should occur. The tests need not be suspended otherwise since the isolation of subsystems is a core principle. Test failures in specific subsystems can be used to fix errors about only those subsystems, thus guaranteeing that this failure is completely contained within a specific unit. This also tells us how various components interact, and suspending the tests means we won't be able to gather sufficient information. Resumption occurs once we fix the errors associated with the test failures that prevent the main pages from loading.

11. Test Deliverables

- Test plan
- Test cases
- Bug reports
- Error logs

12. Schedule

The testing will co-occur with the development. The final date is 2 April 2024.