

American International University-Bangladesh (AIUB)  
Department of Computer Science  
Faculty of Science &Technology (FST)  
Spring 23

Section: [J] Group: 8  
Software Quality Assurance and Testing

**“Developing a Test Plan for Metro Rail Ticket booking Systems”**

A Report submitted

By

|  |  |  |
| --- | --- | --- |
| SN | Student Name | Student ID |
| 1 | ABDUL AZIZ SAJIB | 20-42035-1 |
| 2 | MD. AREFUL HAQUE JONY | 19-41694-3 |
| 3 |  |  |
| 4 |  |  |

Under the supervision of **FARZANA BENTE ALAM**

Software Test Plan

For

Metro Rail Ticket booking Systems

Version 1.0 approved

**Checked By Industry Personnel**

**Name:**  Md Mia Sumon

**Designation:** Software Engineer Manager

**Company:** ZW Solutions Corp

**Sign:** Md Mia Sumon

**Email:** md.mia@thezwsolutions.com

**Date:** 25/04/2023

Table of Contents

[Revision History 3](#_Toc37271323)

[1. TEST PLAN IDENTIFIER: MR-TBS V1.0 3](#_Toc37271324)

[2. REFERENCES 3](#_Toc37271325)

[3. INTRODUCTION 3](#_Toc37271326)

[Background to the Problem 3](#_Toc37271327)

[Solution to the Problem 4](#_Toc37271328)

[4. REQUEIREMNT SPECIFICATION 4](#_Toc37271329)

[4.1 System Features 4](#_Toc37271330)

[4.2 System Quality Attributes 4](#_Toc37271331)

[4.3 System Interface 5](#_Toc37271332)

[4.4 Project Requirements 5](#_Toc37271333)

[5. FEATURES NOT TO BE TESTED 5](#_Toc37271334)

[6. TESTING APPROACH 5](#_Toc37271335)

[6.1 Testing Levels 5](#_Toc37271336)

[6.2 Test Tools 6](#_Toc37271337)

[6.3 Meetings 6](#_Toc37271338)

[7. TEST CASES/TEST ITEMS 6](#_Toc37271339)

[8. ITEM PASS/FAIL CRITERIA 7](#_Toc37271340)

[9. TEST DELIVERABLES 7](#_Toc37271341)

[10. STAFFING AND TRAINING NEEDS 8](#_Toc37271342)

[11. RESPONSIBILITIES 8](#_Toc37271343)

[12. TESTING SCHEDULE 8](#_Toc37271344)

[13. PLANNING RISKS AND CONTINGENCIES 9](#_Toc37271345)

[14. APROVALS 9](#_Toc37271346)

# TEST PLAN IDENTIFIER: MR-TBS V1.0

# REFERENCES

1. **One-step testing**

<https://www.onestoptesting.com/test-plan/references.asp>

1. coleyconsulting

<https://www.coleyconsulting.co.uk/references.htm>

1. Ibm

<https://www.ibm.com/docs/en/elms/elm/6.0.1?topic=sections-test-plan-template-reference>

1. The software must meet the following requirements.
2. The software will support both touch screen monitor and keyboard interface.
3. The program will be able to display a list of arriving trains, their destinations, arrival and departure timings, fare, and estimated travel time.
4. The software will allow for the purchasing of several tickets at the same time.
5. The program will allow you to limit the number of tickets you purchase at the same time. Only the administrator will have access to this privilege control.
6. The software will support ticket cancellation prior to final transaction confirmation.
7. The program will allow the administrator to cancel purchased tickets.
8. Credit card transactions and validation will be supported by the software.

# INTRODUCTION

## Background to the Problem

The need for a Metro Rail Ticket Booking System arises due to the increasing demand for public transportation in urban areas. Metro Rail has become a popular mode of transportation for people traveling within cities due to its efficiency, speed, and convenience. However, the traditional method of purchasing tickets at the counter can be time-consuming and inconvenient, especially during rush hours.

Moreover, the traditional method of purchasing tickets at the counter may not be feasible for passengers who do not live near a Metro Rail station. These passengers may need to make a special trip to the station just to purchase tickets. This can be a significant inconvenience for passengers and may discourage them from using Metro Rail as a mode of transportation.

To address these issues, a Metro Rail Ticket Booking System is necessary to provide passengers with a convenient and efficient way of booking tickets for Metro Rail. With a ticket booking system in place, passengers can book their tickets from the comfort of their homes, avoiding the long queues at the ticket counter. Additionally, the system can provide real-time information on the availability of tickets, making it easier for passengers to plan their travel in advance.

## 

## Solution to the Problem

The solution to the problem of long queues and inconvenience in purchasing tickets for Metro Rail is the development of a Metro Rail Ticket Booking System. The system will provide a platform for passengers to book tickets online, eliminating the need for them to visit the station to purchase tickets. This solution will simplify the process of ticket booking for Metro Rail and encourage more passengers to use the service.

The Metro Rail Ticket Booking System will be designed with a user-friendly interface that will allow passengers to easily search for available trains, select their preferred travel dates and times, and book their tickets online. The system will also provide a secure payment gateway for passengers to make payments online, ensuring the safety and security of their personal and financial information.

Moreover, the system will provide real-time information on the availability of tickets and the status of bookings, making it easier for passengers to plan their travel in advance. This will ensure that passengers are informed about the availability of tickets and can make necessary changes to their travel plans, reducing the likelihood of last-minute cancellations and no-shows.

The development of the Metro Rail Ticket Booking System will not only simplify the process of ticket booking for passengers but will also reduce the workload of ticketing personnel. The automation of the ticket booking process will allow personnel to focus on other important tasks, such as customer service and safety, improving the overall passenger experience.

The development of a Metro Rail Ticket Booking System is the solution to the problem of long queues and inconvenience in purchasing tickets for Metro Rail. The system will provide passengers with a convenient and efficient way of booking tickets, reducing the workload of ticketing personnel and improving the overall passenger experience.

# REQUEIREMNT SPECIFICATION

## System Features

**User Registration:** The system will allow users to register their account to access the booking system. Users will be required to provide basic information such as name, email address, phone number, and password.

**Train Availability:** The system will display the availability of trains for the selected date and time. The user can search for trains based on their preferred departure and arrival stations.

**Ticket Booking:** The user can select the desired train and book their ticket online. The system will generate a unique booking reference number, which the user can use to access their booking details.

**Payment Gateway:** The system will integrate a secure payment gateway for users to make online payments for their tickets. The payment gateway will accept various payment modes such as credit/debit cards, net banking, and mobile wallets.

**Booking Confirmation**: Once the payment is successful, the system will generate a booking confirmation message that will be sent to the user's email address and phone number. The message will contain details such as the booking reference number, train details, and seat number.

**Cancellation and Refund:** The system will allow users to cancel their booking and initiate a refund. The refund policy will be based on the terms and conditions of the Metro Rail operator.

**User Dashboard:** The system will have a user dashboard that will allow users to view their booking history, check the status of their booking, and update their profile information.

**Admin Dashboard**: The system will have an admin dashboard that will allow the Metro Rail operator to manage the booking system. The dashboard will allow the operator to add new trains, modify train schedules, view booking statistics, and generate reports.

**Real-Time Updates:** The system will provide real-time updates to users regarding train delays, cancellations, or rescheduling.

**Helpdesk Support:** The system will have a dedicated helpdesk support that will provide assistance to users in case of any issues or queries related to the booking process.

## System Quality Attributes

**Usability:** The system should have a user-friendly interface that is easy to use and understand. It should be intuitive and responsive to user actions.

**Reliability**: The system should be reliable and available 24/7. The users should be able to access the system without any downtime or technical issues.

**Security:** The system should have robust security features to protect user data and prevent unauthorized access. The system should comply with data protection and privacy regulations.

**Performance:** The system should be able to handle a large volume of user traffic and transactions without any performance degradation. The system should provide a fast and seamless user experience.

**Scalability:** The system should be scalable to accommodate future growth in user demand. The system should be able to handle a sudden increase in user traffic without any issues.

**Maintainability:** The system should be easy to maintain and update. The system should have proper documentation and be modular to allow for easy modification and integration of new features.

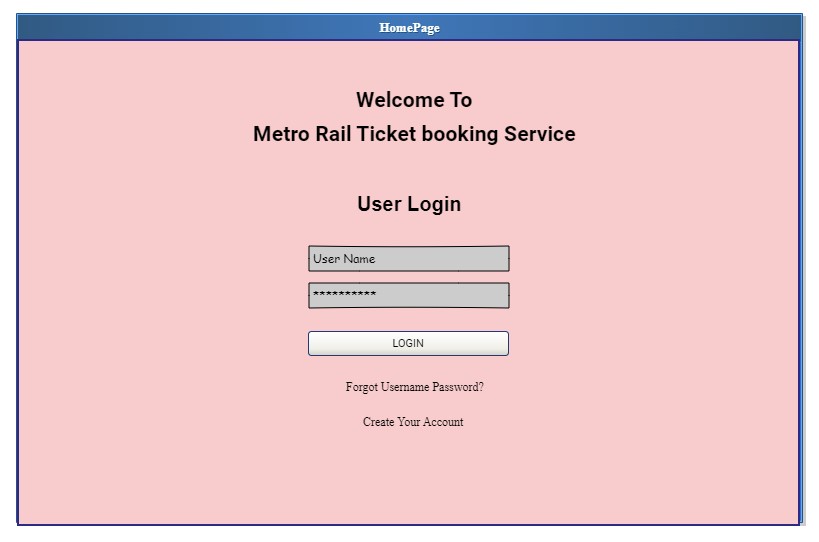
**Interoperability:** The system should be interoperable with other systems and services used by the Metro Rail operator. It should be able to exchange data and communicate seamlessly with other systems.

**Availability:** The system should be available to users at all times. The system should have a disaster recovery plan in place to ensure that the system can be restored quickly in case of any unexpected downtime or system failure.

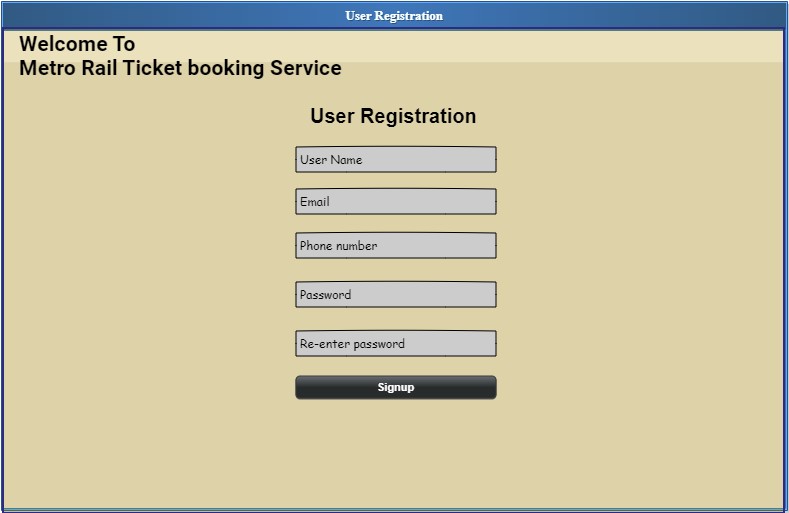
**Accessibility:** The system should be accessible to users with disabilities. The system should comply with accessibility guidelines such as WCAG 2.1 to ensure that users with disabilities can access and use the system.

**User Support:** The system should provide adequate user support to users in case of any issues or queries related to the booking process. The user support should be available through various channels such as email, phone, and live chat.

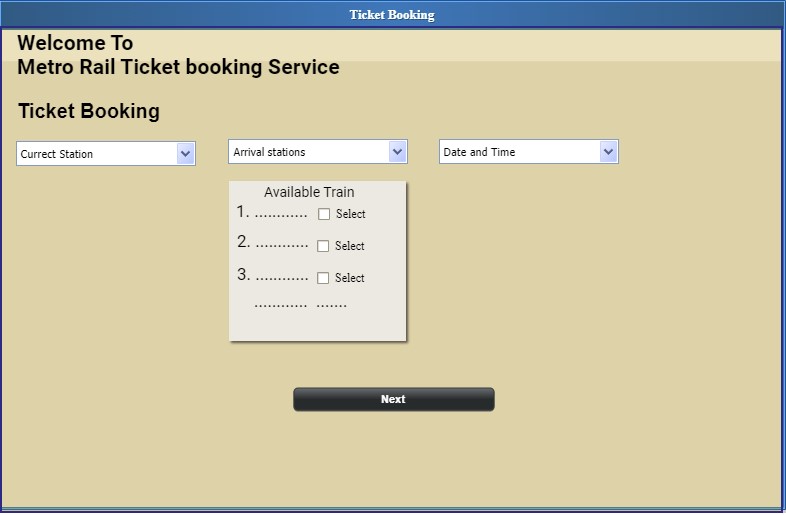
## System Interface



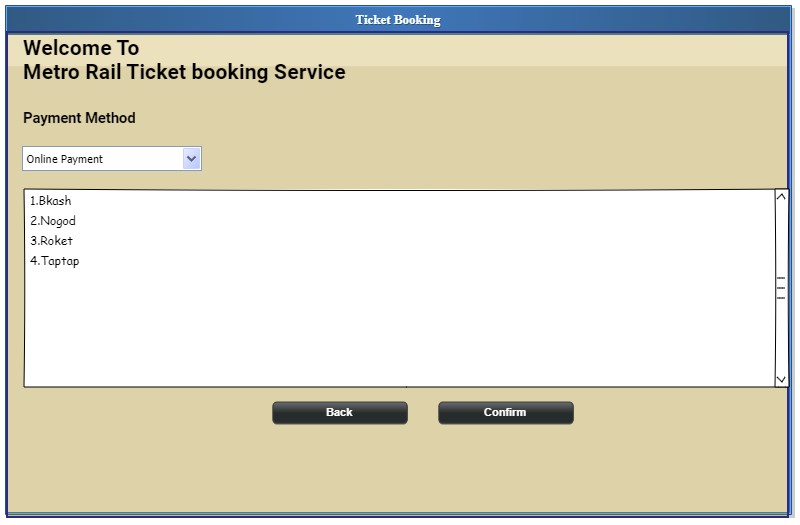
**Login Interface**



**User Registration**



**Ticket Booking**



## Project Requirements

**Recommended Requirements for Hardware**

1. A 64-bit operating system like Windows 10 or later.
2. Intel Core i5 or Intel Core i7 (the 11th generation).
3. 16 GB of RAM or more.
4. 1 TB of free disk space for installation, plus extra space for temporary files during test runs.
5. 1280 × 1024 or higher display resolution.
6. Mouse, keyboard or other pointing device.
7. Sound card w/speakers
8. Web cam and microphone.
9. Digital Printer.

**Recommended Requirements for Software**

**User Registration:** The system will require users to register on the platform before they can make any bookings. The registration process will require users to provide their name, email address, phone number, and password.

**Train Search:** The system will allow users to search for trains based on their preferred departure and arrival stations, date, and time.

**Seat Selection:** The system will allow users to select their preferred seat for the selected train. The system will display a seat layout for the selected train, and users can choose their seat based on availability.

**Payment Gateway:** The system will have an online payment gateway that will allow users to make payments for their tickets using various payment options such as credit/debit cards, net banking, and mobile wallets.

**Booking Confirmation:** The system will provide users with a booking confirmation that will include the booking reference number, train details, seat number, and fare. The confirmation message will be sent to the user's email address and phone number.

**User Dashboard:** The system will provide users with a dashboard that will allow them to view their booking history, check the status of their booking, and update their profile information.

**Admin Dashboard:** The system will provide the Metro Rail operator with a dashboard that will allow them to manage the booking system. The dashboard will allow the operator to add new trains, modify train schedules, view booking statistics, and generate reports.

**Real-Time Updates:** The system will provide real-time updates to users regarding train delays, cancellations, or rescheduling. Users will be notified via email, SMS, or through the system interface.

**Helpdesk Support:** The system will provide users with a dedicated helpdesk support that will provide assistance to users in case of any issues or queries related to the booking process. The helpdesk support will be available through various channels such as email, phone, and live chat.

**Security:** The system will have robust security features to protect user data and prevent unauthorized access. The system will comply with data protection and privacy regulations.

**Performance:** The system will be able to handle a large volume of user traffic and transactions without any performance degradation. The system will provide a fast and seamless user experience.

**Scalability:** The system will be scalable to accommodate future growth in user demand. The system will be able to handle a sudden increase in user traffic without any issues.

**Accessibility:** The system interface will be designed to comply with accessibility guidelines such as WCAG 2.1 to ensure that users with disabilities can access and use the system.

**System Maintenance:** The system will be easy to maintain and update. The system will have proper documentation and be modular to allow for easy modification and integration of new features.

**System Integration:** The system will be interoperable with other systems and services used by the Metro Rail operator. It will be able to exchange data and communicate seamlessly with other systems.

**Cost and Profit Analysis**

**COCOMO MODEL:**

In COCOMO model we know,

Effort =  a (KLOC)^b  (person-month)

Development Time = c (Effort)^d    (month)

Here,

Line of code ( LOC) = 30000

Kloc = 30

a = 2.4

b = 1.05

c = 2.5

d = 0.38

Effort   = a (Kloc)^b

            = 2.4(30)^1.05

            = 85.34(peron month)

Person Months, PM =  c(Effort)^d month

                                =2.5(85)^0.35 month

                                =14.26 month

                                = 11.83 month

Development Time, DM: (2.4\*11.8368) =6.626 =7 Months =1232 Working

Developing salary in hour = 600 taka

Total devolving costing    = 600 \* 1232

                                          = 73, 9200

Rent expense:

Room per month = 8000 Taka

Total in 12 months = 96000 Taka

                      Other cost for Room with internet = 3000 \* 12 = 36,000

                      Total cost = 99600 taka

Project Manager Salary:

Per month= 30,000 Taka

Total in 12 months = (30000\*12) =360,000 Taka

Accountant Salary:

Per month salary =8,000 Taka

Total in 12 Months = (8000\*12) =84,000 Taka

Total marketing cost: 500000 taka

Hardware expense estimation: 100000Taka

Requirement Analysis: 100000 taka

Others cost =1000000

**Total cost**

Total cost = 739200 + 99,600 + 100000 + 84,000 + 360000+ 500000 + 60000+ 1000000

                                 = 1,863,800

**Total Profit**

  20% of Total estimated expense =2,121,200\*20% = 372560 TAKA

|  |  |
| --- | --- |
| **Area** | **Cost (BDT)** |
| Developer Salary | 73,9200 |
| Rent expense | 99,600 |
| hardware expense estimation | 10,0000 |
| Requirement Analysis | 10,0000 |
| Maintenance | 60,000 |
| Project Manager Salary | 360,000 |
| Accountant Salary | 84,000 |
| Marketing Cost | 500000 |
| Total estimated expense | 1,862,800 |
| Profit | 372560 |

# FEATURES NOT TO BE TESTED

The following are examples of features that should not be tested as part of the Metro Rail Ticket Booking System:

**User's internet connectivity:** The system should not be tested for the user's internet connectivity or speed. This is outside the scope of the system as it is beyond the control of the system.

**User's personal device performance:** The performance of the user's personal device, such as a computer or mobile phone, should not be tested as it is not a part of the system being tested.

**Payment gateway processing:** The payment gateway is typically provided by a third-party service provider, and the system should not be tested for the processing of payments as it is beyond the scope of the system.

**Third-party service integration:** If the system integrates with third-party services, such as Google Maps or social media platforms, these services should not be tested as part of the Metro Rail Ticket Booking System.

**Operating system or browser compatibility:** The system should not be tested for compatibility with specific operating systems or web browsers as it is outside the scope of the system.

**Testing user behavior:** The system should not be tested for user behavior, such as whether users select the right train or choose the right seat. User behavior is unpredictable, and it is not feasible to test every possible scenario.

**Availability of trains:** The system should not be tested for the availability of trains. The availability of trains is determined by the Metro Rail operator, and it is beyond the control of the system being tested.

# TESTING APPROACH

## Testing Levels

The Metro Rail Ticket Booking System will undergo the following testing levels:

1. **Unit Testing:** This testing level involves testing individual modules or components of the system to ensure that they function correctly. The focus of this testing is to identify defects in the code and to ensure that each module works as intended.
2. **Integration Testing:** This testing level involves testing the interactions between different modules or components of the system to ensure that they work together seamlessly. The focus of this testing is to identify defects in the integration points between modules and to ensure that the system as a whole functions correctly.
3. **System Testing:** This testing level involves testing the complete system to ensure that it meets the specified requirements. The focus of this testing is to verify that the system functions correctly as a whole and that it meets the user's needs.
4. **Acceptance Testing:** This testing level involves testing the system with the end-users to ensure that it meets their requirements and expectations. The focus of this testing is to verify that the system satisfies the user's needs and that it is user-friendly and easy to use.
5. **Performance Testing:** This testing level involves testing the system's performance under different loads and stress conditions to ensure that it can handle the expected user traffic without any performance degradation. The focus of this testing is to ensure that the system is scalable and can handle future growth in user demand.
6. **Security Testing:** This testing level involves testing the system's security features to ensure that user data is protected, and the system is secure from unauthorized access or attacks. The focus of this testing is to identify security vulnerabilities and to ensure that the system complies with data protection and privacy regulations.

## Test Tools

We have done manual testing and selenium.

## Meetings

Meet with the clients and learn about their needs. Discuss briefly the key aspects of their criteria. Inform them about the Test Plan, go over the timetable, and set a deadline for submission. Create a test plan based on the client requirements and system features. This will serve as the road map for conducting the test from beginning to end. The required testing team may require a number of qualified experts. Some testers may require additional training on a specific testing process for this.

# TEST CASES/TEST ITEMS

7.1 **Test Case: Test Case: User Login**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Metro Rail Ticket booking Systems | | | Test Designed by: Abdul Aziz Sajib | | |
| Test Case ID: FR\_1 | | | Test Designed date: 19 Apr 23 | | |
| Test Priority (Low, Medium, High) : HIGH | | | Test Executed by: Abdul Aziz Sajib | | |
| Module Name: User Login | | | Test Execution date: 19 Apr 23 | | |
| Test Title: verify login with valid username and password | | |  | | |
| Description: Test website login page | | |  | | |
| Precondition (If any): User must have valid username and password | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) | |
| 1. **Navigate to the login page.** 2. **Enter valid login credentials.** 3. **Click on the "Login" button.** 4. **Verify that the user is redirected to the dashboard page.** | Username: SAJB01  Password: 6336 | The user should be able to log in to the system successfully. | | As expected, | Pass | |
| Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Metro Rail Ticket booking Systems | | | Test Designed by: Abdul Aziz Sajib | | |
| Test Case ID: FR\_1 | | | Test Designed date: 20 Apr 23 | | |
| Test Priority (Low, Medium, High) : HIGH | | | Test Executed by: Abdul Aziz Sajib | | |
| Module Name: Search for Trains | | | Test Execution date: 20 Apr 23 | | |
| Test Title: verifying searching for train | | |  | | |
| Description: To ensure that users can search for available trains. | | |  | | |
| Precondition (If any): N/A | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) | |
| 1. Navigate to the search trains page. 2. Enter the "From" and "To" destinations. 3. Select the date and time of the journey. 4. Click on the "Search" button. 5. Verify that the available trains are displayed. | From: Mirpur  Destination : Banani | The available trains should be displayed based on the search criteria. | | As expected, | Pass | |
| Post Condition: N/A | | | | | |

7.2 **Test Case: Search for Trains**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Metro Rail Ticket booking Systems | | | Test Designed by: AREFUL HAQUE JONY | | |
| Test Case ID: FR\_1 | | | Test Designed date: | | |
| Test Priority (Low, Medium, High) : HIGH | | | Test Executed by: | | |
| Module Name: Search for Trains | | | Test Execution date: | | |
| Test Title: verifying searching for train | | |  | | |
| Description: To ensure that users can book a ticket successfully. | | |  | | |
| Precondition (If any): N/A | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) | |
| 1. Navigate to the booking page. 2. Select the train and the class of the ticket. 3. Enter the passenger details. 4. Click on the "Book Ticket" button. 5. Verify that the ticket is booked successfully. | From: Mirpur  Destination : Banani | The available trains should be displayed based on the search criteria. | | As expected, | Pass | |
| Post Condition: N/A | | | | | |

7.3 **Test Case: Book a Ticket**

**7.4 Test Case: Cancel a Ticket**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Metro Rail Ticket booking Systems | | | Test Designed by: AREFUL HAQUE JONY | | |
| Test Case ID: FR\_1 | | | Test Designed date: 22 Apr 23 | | |
| Test Priority (Low, Medium, High) : Medium | | | Test Executed by: AREFUL HAQUE JONY | | |
| Module Name: Cancel a Ticket | | | Test Execution date: 22 Apr 23 | | |
| Test Title: verifying Cancel a Ticket for train | | |  | | |
| Description: To ensure that users can cancel a ticket successfully. | | |  | | |
| Precondition (If any): N/A | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) | |
| 1. Navigate to the cancellation page. 2. Enter the ticket details. 3. Click on the "Cancel Ticket" button. 4. Verify that the ticket is cancelled successfully. | From: Mirpur  Destination : Banani | The ticket should be cancelled successfully, and the user should receive a confirmation message. | | As expected, | Pass | |
| Post Condition: N/A | | | | | |

**7.5 Test Case: Payment Gateway Integration**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Metro Rail Ticket booking Systems | | | Test Designed by: Abdul Aziz Sajib | | |
| Test Case ID: FR\_1 | | | Test Designed date: 25 Apr 23 | | |
| Test Priority (Low, Medium, High) : Medium | | | Test Executed by: Abdul Aziz Sajib | | |
| Module Name: Payment Gateway Integration | | | Test Execution date: 25 Apr 23 | | |
| Test Title: verifying Payment Gateway Integration a Ticket for train | | |  | | |
| Description To ensure that the payment gateway is integrated correctly. | | |  | | |
| Precondition (If any): N/A | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) | |
| 1. Navigate to the payment page. 2. Enter the payment details. 3. Click on the "Pay" button. 4. Verify that the payment is processed successfully. | From: Mirpur  Destination : Banani | The payment should be processed successfully, and the user should receive a confirmation message. | | As expected, | Pass | |
| Post Condition: N/A | | | | | |

# ITEM PASS/FAIL CRITERIA

**User Login:**

Pass: User can log in to the system successfully and is redirected to the dashboard page.

Fail: User cannot log in to the system, or is not redirected to the dashboard page.

**Search for Trains:**

Pass: Available trains are displayed based on the search criteria.

Fail: No trains are displayed, or the displayed trains do not match the search criteria.

**Book a Ticket:**

Pass: User can book a ticket successfully, and the user receives a confirmation message.

Fail: User cannot book a ticket, or the booking process fails, or the user does not receive a confirmation message.

**Cancel a Ticket:**

Pass: User can cancel a ticket successfully, and the user receives a confirmation message.

Fail: User cannot cancel a ticket, or the cancellation process fails, or the user does not receive a confirmation message.

**Payment Gateway Integration:**

Pass: The payment gateway is integrated correctly, and the payment is processed successfully.

Fail: The payment gateway is not integrated correctly, or the payment fails, or the user does not receive a confirmation message.

**System Performance:**

Pass: The system performs well under heavy load, and the performance metrics meet the defined criteria.

Fail: The system crashes, or the performance metrics do not meet the defined criteria.

# TEST DELIVERABLES

**Test Plan**: This document outlines the testing approach, test objectives, scope, and schedule for the testing phase. It includes information about the testing types, tools, and resources needed to perform the testing.

**Test Cases:** These are detailed step-by-step instructions for testing the various functionalities of the system. The test cases should cover all possible scenarios and edge cases to ensure comprehensive testing.

**Test Scripts:** These are automated scripts that can be used to execute the test cases automatically. Test scripts can save time and effort and help ensure that tests are executed consistently.

**Test Reports:** These are documents that summarize the testing results, including the test cases executed, defects found, and the status of the testing phase. Test reports help stakeholders understand the testing progress and make informed decisions.

**Defect Reports**: These are documents that describe the defects found during testing, including their severity, priority, and steps to reproduce. Defect reports help the development team prioritize and fix the defects.

**Traceability Matrix**: This document links the requirements with the test cases to ensure that all requirements are covered by the tests. The traceability matrix helps ensure comprehensive testing and helps identify any gaps in the testing coverage.

**User Acceptance Test Plan:** This document outlines the testing approach and criteria for the user acceptance testing phase. It includes information about the acceptance testing types, schedule, and resources needed to perform the testing.

# STAFFING AND TRAINING NEEDS

**Testing Team:** The testing team should include experienced software testers who have a good understanding of the testing methodologies and techniques. The team should be large enough to ensure comprehensive testing of the system.

**Test Manager:** The test manager is responsible for planning, coordinating, and managing the testing activities. The test manager should have experience in managing testing projects and a good understanding of the testing process.

**Test Automation Engineer:** The test automation engineer is responsible for creating and maintaining the test automation scripts. The engineer should have experience in test automation frameworks and scripting languages.

**Training:** The testing team may need training on the testing tools, technologies, and methodologies used in the project. The training may be provided by external trainers or in-house trainers.

**Infrastructure:** The testing team will need access to the required infrastructure, including testing environments, test data, and testing tools.

**Documentation:** The testing team will need to create and maintain the test documentation, including test plans, test cases, and test reports. The team should have good documentation skills and attention to detail.

**Communication:** The testing team should have good communication skills to interact with the development team, project managers, and other stakeholders. They should be able to report defects effectively and provide feedback to improve the system's quality.

# RESPONSIBILITIES

**Testing Team:** The testing team is responsible for planning, executing, and reporting on the testing activities. They should create and maintain the test plan, test cases, and test scripts, execute the tests, report defects, and provide feedback on the system's quality.

**Test Manager:** The test manager is responsible for overseeing the testing activities and ensuring that the testing objectives are met. They should manage the testing team, coordinate with the development team, and report on the testing progress and results.

**Development Team:** The development team is responsible for fixing defects found during testing, providing feedback on the testing results, and improving the system's quality.

**Project Manager:** The project manager is responsible for managing the overall project, including the testing phase. They should ensure that the testing objectives are aligned with the project goals and provide the necessary resources and support to the testing team.

**Business Analyst:** The business analyst is responsible for ensuring that the system requirements are properly documented and communicated to the testing team. They should also review the test plans and test cases to ensure that they cover all the requirements.

**End Users:** The end users are responsible for participating in the user acceptance testing phase, providing feedback on the system's usability and functionality, and verifying that the system meets their expectations.

**Quality Assurance:** The quality assurance team is responsible for ensuring that the testing process adheres to the established standards and procedures. They should review the test plans, test cases, and test reports and provide feedback to improve the testing process.

# TESTING SCHEDULE

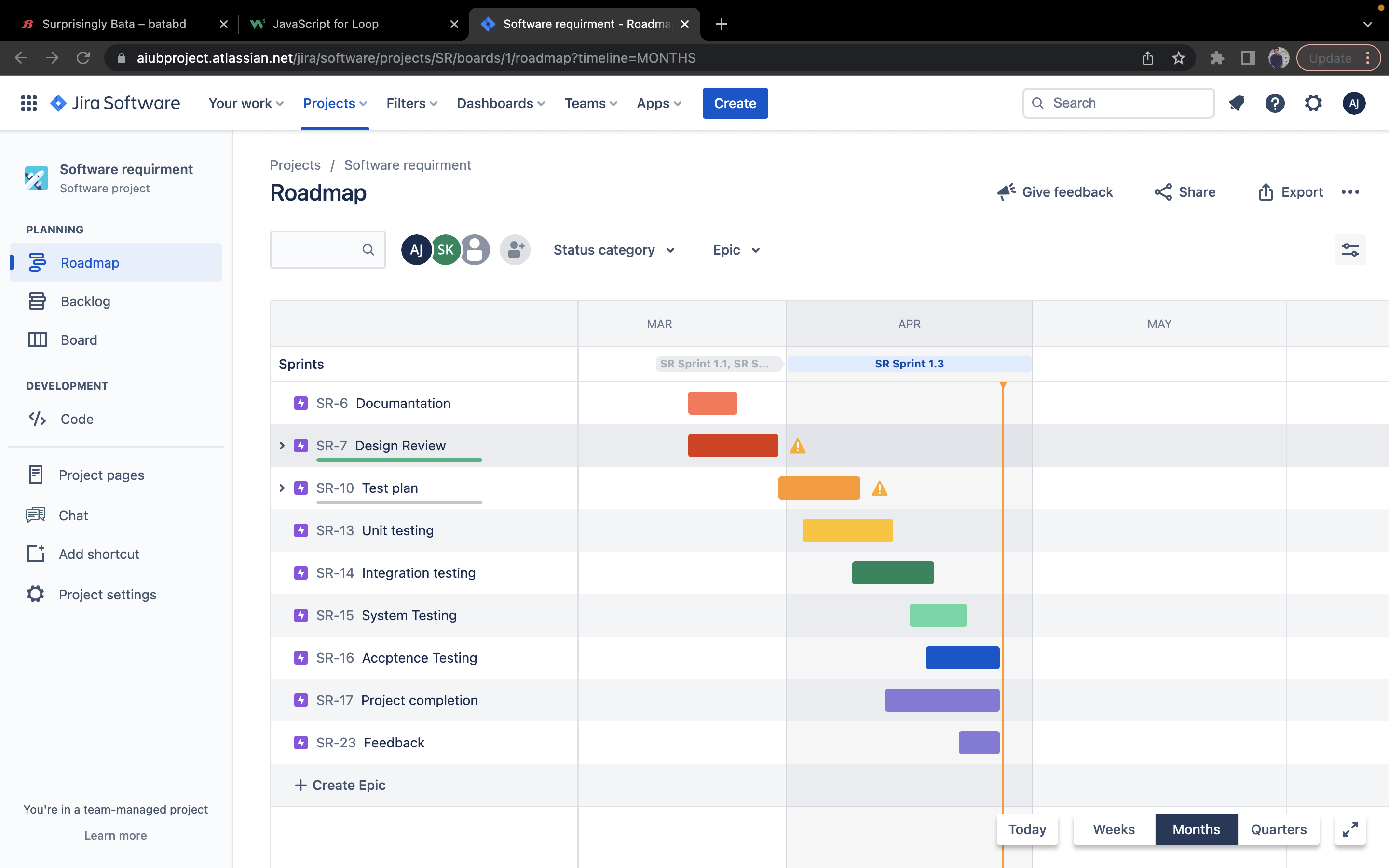
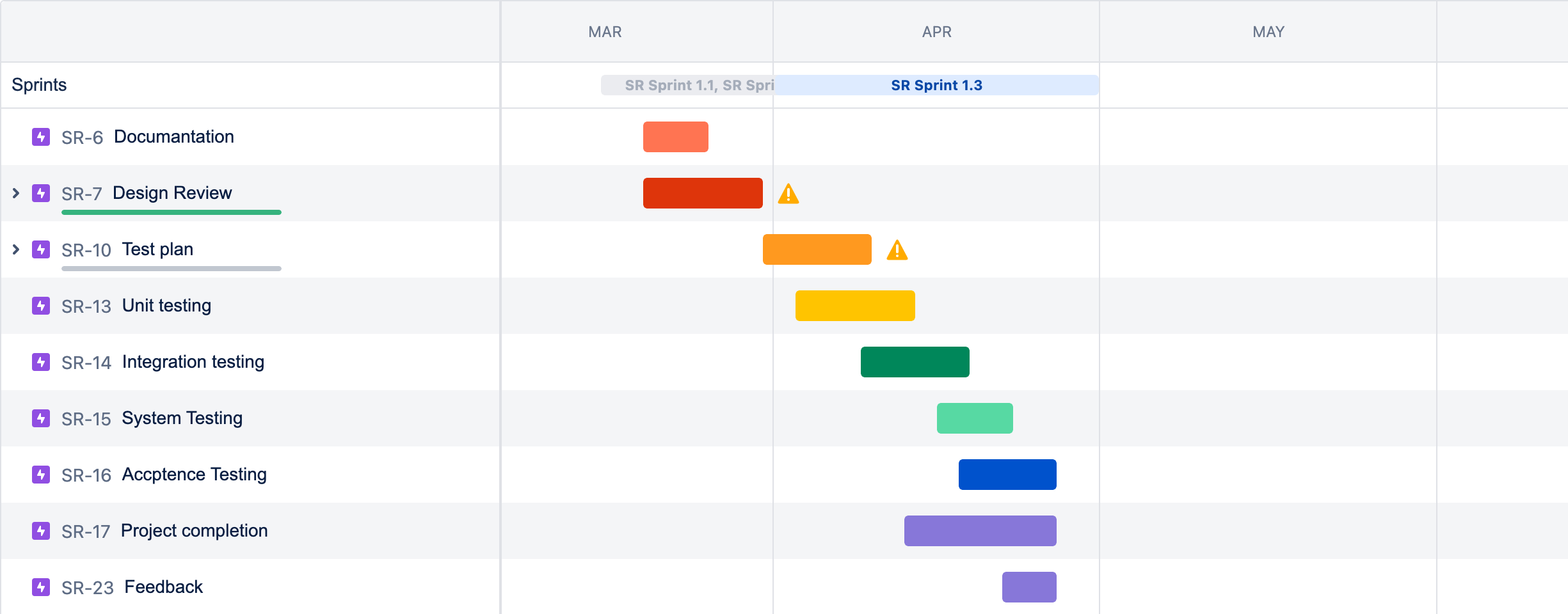


Figure: Jira Tools

# PLANNING RISKS AND CONTINGENCIES

Planning for risks and contingencies is an essential part of any project, including testing the Metro Rail Ticket Booking System. Here are some potential risks and contingencies that could be considered during the planning phase:

**Resource Availability:** One risk could be a shortage of resources, including testing tools, hardware, or personnel. A contingency for this risk could be to identify backup resources, such as outsourcing testing or using cloud-based testing tools.

**System Integration:** Another risk could be that the system does not integrate well with other systems, such as payment gateways or backend databases. A contingency for this risk could be to perform extensive integration testing and have a backup plan for managing transactions if the integration fails.

**Security Breaches:** There is a risk that the system could be subject to a security breach or hacking attempt. A contingency for this risk could be to conduct regular security testing and have a plan in place for managing security incidents, such as activating a security response team.

**Scope Creep:** Another risk could be that the project scope could expand beyond the original requirements, leading to delays and cost overruns. A contingency for this risk could be to have a clear change management process in place and regular review sessions to ensure that the project stays on track.

**User Acceptance:** A risk could be that the end-users may not accept the system or find it challenging to use. A contingency for this risk could be to involve end-users in the testing process, perform extensive user acceptance testing, and have a plan in place for addressing user feedback.

**Defects:** A risk could be that the system contains significant defects that could impact the project's success. A contingency for this risk could be to have a process in place for managing defects, such as prioritizing high-severity defects and resolving them promptly.

# APROVALS

|  |  |
| --- | --- |
| Project Sponsor | Md Mia Sumon |
| Development Management | Abdul Aziz Sajib |
| EDI Project Manager | Ariful Haque Jony |
| RS Test Manager | Sakibur rahman |
| RS Development Team Manager | Jahidul hauqe |
| Reassigned Sales | Mohammad rofiq |
| Order Entry EDI Team Manager | salehin |