

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

**EVENT BOOKING**

A Software Quality and Testing Project Submitted

By

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Semester: Summer\_24\_25** | | | | **Section: B** | **Group No:3** |
| **SL** | **SN** | **Student Name** | **Student ID** | Individual  Contribution (in %) | Total Marks: 50 |
| Earned Marks: |
| **A** | 10 | MAIASHA SULTANA | 21-45492-3 | 98% |  |
| **B** | 11 | MD REZWAN NABI | 21-45752-3 | 95% |  |
| **C** | 12 | SADIA AFROSE | 21-45820-3 | 93% |  |
| **D** | 13 | RADIA YESMIN OISHI | 21-45908-3 | 90% |  |

The project will be Evaluated for the following Course Outcomes

|  |  |  |
| --- | --- | --- |
| **EVALUATION CRITERIA** | **Total Marks (50)** | |
|  | |
| Revision History, Test Plan Identifier, Reference Materials, Problem Background, Solutions | [10 Marks] |  |
| Requirements Specification (System feature, Quality Attributes, System Interface, Project Requirements) | [10 Marks] |  |
| Item Not to be tested, Testing approach (Testing levels, tools, meetings), Test cases | [10 Marks] |  |
| Item pass/fail criteria, Test deliverables, Staffing and Training, Responsibilities, Scheduling, Risk | [10 Marks] |  |
| Approval, Format, Submission, and Defense | [10 Marks] |  |

Software Test Plan

for

<EVENT BOOKING>

Version 7.0 approved

Prepared by <Maiasha Sultana>

<Md Rezwan Nabi>

<Sadia Afrose>

<Radia Yesmin Oishi>

<American International University-Bangladesh (AIUB)>

<16 September 2025>

Table of Contents

[Revision History 3](#_Toc208957754)

[1. TEST PLAN IDENTIFIER: AT-TP01.3 4](#_Toc208957755)

[2. REFERENCE MATERIALS 4](#_Toc208957756)

[3. INTRODUCTION 4](#_Toc208957757)

[3.1 Background to the Problem 4](#_Toc208957758)

[3.2 Solution to the Problem 5](#_Toc208957759)

[4. REQUEIREMNT SPECIFICATION 6](#_Toc208957760)

[4.1 System Features 6](#_Toc208957761)

[4.2 System Quality Attributes 12](#_Toc208957762)

[4.3 System Interface 14](#_Toc208957763)

[4.4 Project Requirements 22](#_Toc208957764)

[5. FEATURES NOT TO BE TESTED 23](#_Toc208957765)

[6. TESTING APPROACH 23](#_Toc208957766)

[6.1 Testing Levels 23](#_Toc208957767)

[6.2 Test Tools 24](#_Toc208957768)

[6.3 Meetings (Weekly) 25](#_Toc208957769)

[7. TEST CASES/TEST ITEMS 26](#_Toc208957770)

[8. ITEM PASS/FAIL CRITERIA 42](#_Toc208957771)

[9. TEST DELIVERABLES 43](#_Toc208957772)

[10. STAFFING AND TRAINING NEEDS 43](#_Toc208957773)

[11. RESPONSIBILITIES 44](#_Toc208957774)

[12. TESTING SCHEDULE 44](#_Toc208957775)

[13. PLANNING RISKS AND CONTINGENCIES 45](#_Toc208957776)

[14. APROVALS 46](#_Toc208957777)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Updated by** | **Update Comments** |
| 0.1 | 2025.09.13 | Maiasha Sultana | System Interface |
| 0.2 | 2025.09.13 | Md Rezwan Nabi | System Interface |
| 0.3 | 2025.09.14 | Sadia Afrose | UI/UX design |
| 0.4 | 2025.09.15 | Maiasha Sultana | System Interface |
| 0.5 | 2025.09.16 | Radia Yesmin Oishi | Gantt Chart |
| 0.6 | 2025.09.16 | Sadia Afrose | Pass/fail criteria |
| 0.7 | 2025.09.16 | Md Rezwan Nabi | Test Case |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# TEST PLAN IDENTIFIER: AT-TP01.3

# REFERENCE MATERIALS

1. Karim, M., & Sultana, R. (2023). *College event management system survey*. **International Research Journal of Modernization in Engineering, Technology and Science, 5**(11), 258–263. <https://www.irjmets.com/uploadedfiles/paper/issue_11_november_2023/45827/final/fin_irjmets1699242577.pdf>
2. Silvers, J. (2017). Event management research: The state of the art. **Event Management, 21**(1), 1–12. <https://doi.org/10.3727/152599517X14855497412707>
3. Bramhe, M. V., Waghmare, A., Awate, B., Rao, K., Kadu, A., & Dangre, T. (2024). *Online event management system: A critical review of research findings and methodologies*. **International Journal of Innovations in Engineering and Science, 9**(5), 11–13. <https://doi.org/10.46335/IJIES.2024.9.5.3>
4. Ganiyu, M., Egwuche, O. S., & Adekunle, K. O. (2024). *Online events booking and reservation system*. **ResearchGate Project**. Retrieved from ResearchGate.
5. Web Technology project- Event Booking.

# INTRODUCTION

## 3.1 Background to the Problem

The event booking sector tackles the difficulty of effectively managing the reservation of venues, tickets, or services for occasions that can range from intimate gatherings to large conferences and performances. Usually, the issues in this area arise from the challenges of coordinating several parties’ attendees, organizers, venues, and service providers across various systems and platforms. This leads to problems like double bookings, unavailability of real-time information, poor communication, and complications in processing payments and cancellations.

Addressing this issue is extremely important because inefficient event booking can cause major financial losses, unhappy customers, and wasted resources. For those organizing events, having a smooth booking system is necessary to maximize attendance and revenue while reducing logistical mistakes. Attendees, on the other hand, greatly benefit from dependable and easy-to-use booking experiences to aid in their planning and participation. With the growing dependence on online platforms and the variety of event types, creating a common, efficient, and transparent booking process helps everyone involved by improving operational efficiency, enhancing user satisfaction, and enabling growth in the rapidly changing events industry. This mutual understanding helps ensure that all stakeholders are aligned with the project's objectives and recognize why the solution is essential in this context.

## 3.2 Solution to the Problem

To tackle the issues faced in the event booking sector, any proposed software and solutions need to concentrate on simplifying workflows, improving communication, and enhancing the user experience while effectively achieving business goals.

Proposed Solutions and Their Appropriateness

* Centralized Event Booking System
* Integration of Secure Online Payments
* Incorporated Communication Tools
* Data Analysis and Reporting
* Customization and Adaptability
* User-friendly Design.

This solution is particularly well-suited as it directly addresses the underlying issues of inefficiency, miscommunication, and fragmentation in event booking. It allows for growth and flexibility regardless of event size or type, aligning with business aims of boosting revenue, enhancing user experience, and reducing operational mistakes. Additionally, this solution is practical with modern technology, and the readily available development tools and cloud resources make it cost-effective to implement and manage.

The software in question is a comprehensive system for Event Booking aimed at making the booking process easier and more efficient for venues and ticketing. Its main goal is to create a single platform where event organizers, venue managers, and participants can effectively handle all aspects such as event information, registrations, payments, and communications.

**Objectives and goals of the software:**

* Prevent double bookings and scheduling issues
* Boost revenue from bookings by enhancing user experience
* Ensure effective and timely communication among all parties involved
* Support decision-making based on data
* Accommodate a variety of events of different types and sizes for wider usability

Numerous software applications and research studies reveal the advantages and capabilities of event booking systems, showcasing both their essential features and some of their drawbacks.

1. Investigations into technologies such as Spring Boot, Angular, and MySQL highlight the importance of improving payment gateway integration, providing advanced analytics, and developing unified communication tools to fix issues found in older systems.
2. Commercial platforms like Cvent, Eventbrite, Bizzabo and Whova deliver thorough event management services, offering functionalities like registration, ticket sales, real-time analytics, mobile applications, and marketing resources.
3. Meanwhile, tools such as Planning Pod, Amilia, and Event leaf concentrate on automating the management of venues, catering services, billing processes, and engaging attendees.
4. Platforms that prioritize user experience, such as Event Bookings and Time.ly, concentrate on making ticket sales and venue scheduling easier.

These available solutions demonstrate the effectiveness of integrated event booking software and show a range of customizable features that can be adapted to meet various project requirements. Many of these platforms emphasize cloud-based, scalable designs that facilitate remote control and teamwork, aligning with today’s business goals of efficiency and accessibility.

# REQUEIREMNT SPECIFICATION

## System Features

**1. System Login**   
**Functional Requirements**

1.1 The software allows users to login with their username and password.

1.2 If the username and/or password are entered incorrectly more than three times, the system shall generate a random verification code for reattempting login.

1.3 If the login attempt exceeds five times, the system shall block the user account for one hour.

**Priority Level:** High  
**Precondition:** user have valid user id and password.

**Cross reference:** Registration, User Profile

**2. Registration and Verification**

2.1 The system shall allow new users to register by providing details such as full name, email address, phone number, and password.

2.2 The system shall validate input fields (e.g., email format, phone number format, password strength).

2.3 The system shall send a verification email or SMS with a one-time code (OTP) or activation link to confirm account creation.

2.4 The system shall allow users to verify their account by entering the OTP or clicking the verification link.

2.5 The system shall prevent login until the user account has been successfully verified.

2.6 The system shall prevent duplicate registration using the same email or phone number.

2.7 If the user fails to verify the account within a specified time (e.g., 24 hours), the system shall deactivate the pending registration.

**Priority Level:** High  
**Precondition:** User must not have an existing account.

**Cross reference:** Login, User Profile, Home page

**3. Home page**

3.1 The system shall display a homepage that serves as the main entry point for all users (registered users, admin).

3.2 The homepage shall display upcoming events, featured events, and popular categories.

3.3 The homepage shall include a service allowing users to search events by name or category.

3.4 The homepage shall provide navigation menus to access key features such as Logout, Registration, Event Categories, My Bookings, and Contact Us.

3.5 The system shall display promotional banners, discounts, or announcements on the homepage.

3.6 For logged-in users, the homepage shall display personal recommendations based on past bookings or preferences.

3.7 The system shall ensure that all homepage content is responsive and accessible on different devices (desktop, tablet, mobile).

**Priority Level:** High  
**Precondition:** User must have an account.

**Cross reference:** User profile, Dashboard, Services, Contact, Notification, Logout

**4. User Profile**

4.1 The system shall allow each registered user to create and manage their personal profile.

4.2 The profile shall store user information such as full name, email address, phone number, gender, date of birth, and profile picture.

4.3 The system shall allow users to update or edit their profile details any time, except for email (require re-verification).

4.4 The system shall allow users to change their password securely through profile settings.

4.5 The system shall display booking history and upcoming events in the user’s profile section.

4.6 The profile shall allow users to manage saved payment methods.

4.7 The system shall ensure sensitive information (e.g., password, payment details) is encrypted and not directly visible.

4.8 The system shall allow users to deactivate or delete their profile/account if requested.

**Priority Level:** High  
**Precondition:** User must have an existing account.

**Cross reference:** Dashboard, Events, logout

**5. Service**

5.1 The software allows users to select an event.  
 5.2 The system shall verify ticket availability before confirming the booking.  
 5.3 The system shall allow users to select seat categories (VIP, General, etc.).  
 5.4 A booking confirmation shall be generated with a unique booking ID.

**Priority Level:** High  
 **Precondition:** User must be logged in.  
 **Cross reference:** Ticket, Payment, Check- In

**6. Dashboard**

6.1 The system shall display a personalized dashboard for logged-in users.

6.2 The dashboard shall show a summary of the user’s recent activities, including

upcoming booked events, past booking history, and payment status.

6.3 The dashboard shall provide quick links to important features such as “Venue,”

Events,” “Payments,” and “Attendee List.”

**Priority Level:** High  
 **Precondition:** User is logged in.  
 **Cross reference:** Calander, Venue, Events, Home page, Payment, Attendee, Ticket

**7. Notifications**

7.1 The system shall send notifications to users regarding booking confirmations, payment status, event reminders, and cancellations.

7.2 The system should support multiple notification channels (e.g., email, SMS, in-app alerts).

7.3 The system shall display a notification panel within the user dashboard showing recent updates.

7.4 The system allows users to configure notification preferences (e.g., email only, SMS only, both).

7.5 The system shall ensure timely delivery of critical notifications (e.g., schedule changes, cancellations).

7.6 The system shall mark notifications as read/unread for better user tracking.

**Priority Level**: High  
**Precondition**: User has an active account or booking.  
**Cross reference**: Events, services, Payment, Home page, Venue

#### 8. Contact & Support

8.1 The system shall provide a “Contact Us” section accessible from the home page and dashboard.

8.2 The system allows users to submit queries or complaints via a contact form (name, email, subject, message).

8.3 The system should provide direct support options, such as email address, phone number, or live chat (if enabled).

8.4 The system shall generate a unique ticket ID for each support query for tracking purposes.

8.5 The system shall allow administrators/support staff to respond to user queries and update ticket status.

8.6 The system shall provide a FAQ/help section for quick answers to common issues.

**Priority Level**:Medium **Precondition**:User visits the contact section. **Cross reference**:Home page

**9. Calendar**

9.1 The system shall allow users to view upcoming events in calendar format

9.2 The system shall allow organizers to add or edit events on the calendar.

9.3 The system shall provide notifications/reminders for scheduled events.

**Priority Level:** Medium

**Precondition:** User must be logged in

**Cross Reference:** Dashboard

**10. Attendee**

10.1The system shall allow attendees to register for an event.

10.2The system shall allow viewing attendee lists for organizers.

**Priority Level:** Medium

**Precondition:** User must be logged in

**Cross Reference:** Dashboard

**11. Ticket**

11.1The system shall allow users to purchase tickets

11.2 The system shall allow users to download or receive tickets via email.

**Priority Level:** High

**Precondition:** User must be logged in

**Cross Reference:** Services, Dashboard

**12. Venue**

12.1The system shall allow organizers to add venue details

12.2 The system shall allow mapping of venues to specific events.

**Priority Level:** High

**Precondition:** User must be logged in

**Cross Reference:** Dashboard

**13. Events**

13.1The system shall allow users to view event details**.**

13.2 The system shall link events with venue and ticketing.

**Priority Level:** High

**Precondition:** User must be logged in

**Cross Reference:** Events, Dashboard, Ticket, Venue

**14. Check In**

14.1The system shall validate ticket authenticity at the venue

14.2 The system shall update attendee status to Checked in.

**Priority Level:** Medium

**Precondition:** User must have a valid purchased ticket.

**Cross Reference:** Venue, Ticket, Dashboard

**15. Payment**

15.1The system shall allow users to select a payment method

15.2 The system shall generate an invoice/receipt after successful payment.

**Priority Level:** High

**Precondition:** User must have selected a ticket or service.

**Cross Reference:** Dashboard, Ticket, Services

**16. Logout**

16.1The system shall allow users to log out securely.

16.2 The system shall redirect the user to the homepage or login screen

**Priority Level:** Medium

**Precondition:** User must have selected a ticket or service.

**Cross Reference:** User profile, Home page

## System Quality Attributes

**QA1 - Usability:** A first-time user shall be able to search, select, and book an event ticket in an average of five and a maximum of eight minutes.

**Priority Level:** High

**Precondition:** User has access to the system via desktop.

**Cross Reference:** QA2, QA5

**QA2 – Performance:** The system shall process booking confirmations within 3 seconds for 96% of transactions.

**Priority Level:** High

**Precondition:** Stable internet connection.

**Cross Reference:** QA1, QA4

**QA3 – Reliability:** The system shall have an uptime of 99.5% during peak event seasons.

**Priority Level:** High

**Precondition:** Hosting server and database are operational.

**Cross Reference:** QA2, QA6

**QA4 – Security:** All payment transactions shall be encrypted using SSL/TLS, and sensitive

data shall not be stored in plain text.

**Priority Level:** High

**Precondition:** User proceeds to checkout/payment gateway.

**Cross Reference:** QA2, QA5

**QA5 – Scalability:** The system shall support up to 10,000 concurrent users without

performance degradation.

**Priority Level:** Medium

**Precondition:** Cloud infrastructure is available.

**Cross Reference:** QA2, QA3

**QA6 – Availability:** The system shall be accessible 24/7, with a maximum downtime of 2

hours per month for maintenance.

**Priority Level:** Medium

**Precondition:** Scheduled maintenance notification provided to users.

**Cross Reference:** QA3, QA5

**QA7 – Maintainability:** System updates (e.g., UI changes, bug fixes) shall be deployed

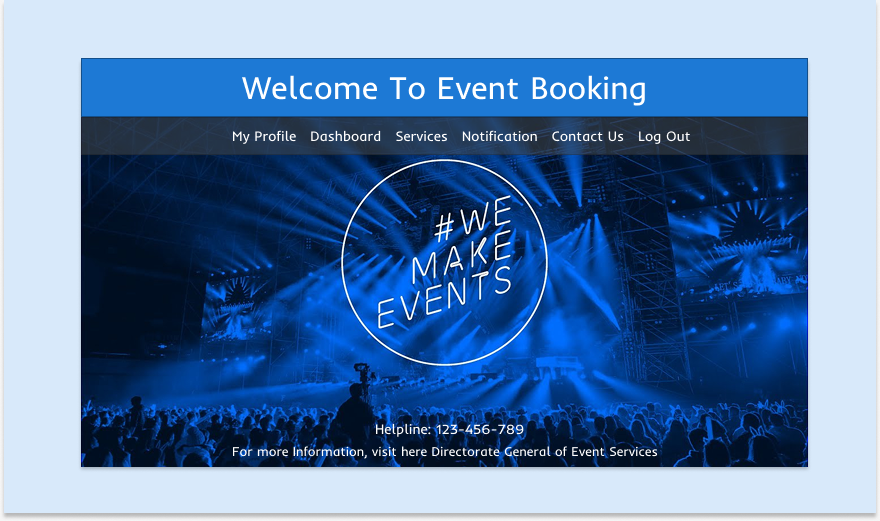
with zero downtime in less than 30 minutes.

**Priority Level:** Medium

**Precondition:** Development and staging environment available.

**Cross Reference:** QA3, QA6

## System Interface

A screenshot of a login form

AI-generated content may be incorrect.

A registration form with a logo

AI-generated content may be incorrect.

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a login screen

AI-generated content may be incorrect.

A screenshot of a welcome page

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a website

AI-generated content may be incorrect.

A screenshot of a website

AI-generated content may be incorrect.

A screenshot of a web page

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a ticket

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a checkout

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Project Requirements

**Effort Estimation Using COCOMO Model**

* **Project Type:** Semi-detached (moderate complexity with mixed flexible requirements
* **SLOC (Source Lines of code):** 17,000 (estimated for web admin panel)
* **Coefficient (Effort Factor):** 3.0
* **P (Project Complexity):** 1.12
* **T (Time Exponent):** 0.35

**Calculations:**

1. **Effort (PM)** in person-months:

PM= 3.0\*(17,000/1000) ^1.12

= 71.65 person-months

1. **Development Time (DM)** in months:

**DM= 2.50\*(71.65) ^0.35**

= 11.15 months

1. **Staffing (ST)** required:

ST= PM/DM

= 71.65/11.15

= 6.43 ≈ 7 members

**Projects Constraints:**

1. **Timeline:**

COCOMO Output: 11.15 months ~ 12 months

Resolution: Scope reduction to core features

1. **Team Size:**

COCOMO Output: 7 members

Mitigation: Use low code tools, focus on Minimum Viable Product

1. **Technical Constraints:**
   1. HIPAA/GDPR compliance (SRS 2.5)
   2. Real-time GPS accuracy ≥ 95% (SRS 3.1)
   3. Cloud infrastructure costs ≤ $200/month

# FEATURES NOT TO BE TESTED

* Hardware integration (e.g., physical ticket printers, POS devices)
* Third-party event promotion APIs (external marketing integrations)
* Cross-browser performance for outdated browsers (e.g., Internet Explorer 11)
* Manual/physical booking or registration (non-digital processes)
* Check-In cannot be tested in lab conditions; only partial simulation is possible. Final validation requires on-site UAT.”

# TESTING APPROACH

## Testing Levels

Here’s how we’ll validate the Event Booking System at every testing level:

1. **Unit Testing**

* **Objective:** Verify each module of the event booking application works correctly in isolation.
* **Examples for event booking:**
  + Validate event creation form (title required, date format, ticket price > 0).
  + Check seat availability calculation logic.
  + Ensure user registration validates email and password rules.

1. **Integration Testing**

* **Objective:** Ensure interaction between modules is smooth.
* **Examples for event booking:**
  + Booking module writes ticket data correctly into database.
  + Payment gateway API returns success → booking confirmed → email notification sent.
  + Ticket cancellation updates both user dashboard and admin event capacity.
  + Login system correctly retrieves booking history.

1. **System Testing**

* **Objective:** Validate the entire Event Booking System as a complete product.
* **Examples for event booking:**
  + End-to-end workflow: Search event → Select tickets → Payment → Confirmation.
  + Stress test: 1000 concurrent users booking the same event.
  + Security test: unauthorized access to booking records should be blocked.
  + Cross-platform testing: Mobile (Android/iOS), different browsers.

1. **Acceptance Testing**

* **Objective:** Validate system against business requirements and user expectations.
* **Examples:**
  + Event organizers verify they can create, update, and cancel events easily.
  + Users verify booking and refund process is simple and error-free.
  + Stakeholders confirm business rules (e.g., max seat limit, early-bird discounts) work as expected.
  + Accessibility tested (e.g., readable fonts, assistive tools).

## Test Tools

* **Selenium**: Features like login, registration, and booking so we don’t test manually every time.
* **Figma** – to design the UI/UX and visualize how the system screens look before development.
* **Canva** – to create our meeting plan and Gantt chart for showing project schedule clearly.
* **JMeter** – to test system performance and check how it works with many users booking at the same time."

## Meetings (Weekly)

**13 August 2025**

|  |  |
| --- | --- |
| Agenda | Attendees |
| * Topic selection * Weekly meeting time fixing * Project template overview * Questions & Discussion | * Md Rezwan Nabi * Sadia Afrose * Maiasha Sultana * Radia Yesmin Oishi |

**18 August 2025**

|  |  |
| --- | --- |
| Agenda | Attendees |
| * Task distribution among team members has been done * Finalize UI design * Create meeting in week * Questions & Discussion | * Md Rezwan Nabi * Sadia Afrose * Maiasha Sultana * Radia Yesmin Oishi |

**26 August 2025**

|  |  |
| --- | --- |
| Agenda | Attendees |
| * Checking the previous task * share UI/UX design * Questions & Discussion * Discuss test case and test plan * Define testing level | * Md Rezwan Nabi * Sadia Afrose * Maiasha Sultana * Radia Yesmin Oishi |

**1 September 2025**

|  |  |
| --- | --- |
| Agenda | Attendees |
| * Checking test cases * Test deliverables * Questions & Discussion * Individual’s previous weeks task check | * Md Rezwan Nabi * Sadia Afrose * Maiasha Sultana * Radia Yesmin Oishi |

**11 September 2025**

|  |  |
| --- | --- |
| Agenda | Attendees |
| * Review all test cases, the test plan * Checking full documents * Complete reference materials * Questions & Discussion | * Md Rezwan Nabi * Sadia Afrose * Maiasha Sultana * Radia Yesmin Oishi |

# TEST CASES/TEST ITEMS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Md Rezwan Nabi | | |
| Test Case ID: FR\_1 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Login Session for website | | | Test Execution date: | | |
| Test Title: Verify login with valid username and password | | |  | | |
| Description: Test website login page | | |  | | |
| Precondition (If any): User must have a registered account. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Go to the website 2. Enter username 3. Enter password 4. Click submit | Username: Rezwan  Password:  21-rezwan-3 | User should login into the application | |  |  |
| Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database. | | | | | |

Table 1: Test Case for **Login Session**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Maiasha Sultana | | |
| Test Case ID: FR\_2 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Registration Session for website | | | Test Execution date: | | |
| Test Title: Verify Registration with valid name, email, password and contact. | | |  | | |
| Description: Test website registration page | | |  | | |
| Precondition (If any): The user must not be an account | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Navigate to the “Registration” screen.  2.Enter valid user details  3.Click “Register Now” to  submit.  4. Verify that the system generates a user ID and sends a confirmation email.  5. Ensure the user can log in using the credentials. | Username: Maiasha  Password:  Boishakhi738  Confirm Password:  Boishakhi738 | Registration successful, confirmation email received, user can log in. | |  |  |
| Post Condition: User is successfully registered and can log in. | | | | | |

Table 2: Test Case for **Registration Session**

Table 3: Test Case for **Home page**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Radia Yesmin Oishi | | |
| Test Case ID: FR\_3 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Home page for website | | | Test Execution date: | | |
| Test Title: Verify role-based home visibility | | |  | | |
| Description: Verify role-specific home access for users (Owner, Admin, Customer). | | |  | | |
| Precondition (If any): User must be logged in with appropriate role. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Log in  2. Verify that the home displays relevant sections | Username: Oishi  Email: oishi7875@gmail.com  Password: RYO213 | Each user should see the relevant sections according to their role. | |  |  |
| Post Condition: Role-based access control is verified. | | | | | |

Table 4: Test Case for **Dashboard**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Md Rezwan Nabi | | |
| Test Case ID: FR\_4 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Dashboard for website | | | Test Execution date: | | |
| Test Title: Verify role-based Dashboard visibility | | |  | | |
| Description: Verify role-specific Dashboard access for users | | |  | | |
| Precondition (If any): User must be logged in with appropriate role. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Log in  2. Verify that the Dashboard displays relevant sections | Username: Rezwan  Email: rezwan852@gmail.com  Password: 21-rezwan-3 | Each user should see the relevant sections according to their role. | |  |  |
| Post Condition: Role-based access control is verified. | | | | | |

Table 5: Test Case for **My Profile**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Maiasha Sultana | | |
| Test Case ID: FR\_5 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: profile for website | | | Test Execution date: | | |
| Test Title: registered users to manage their personal information, booking history, and security settings. | | |  | | |
| Description: manages all user-related personal information and preferences within the Event Booking System. | | |  | | |
| Precondition (If any): User must be logged into the application to access profile features. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Login with valid credentials  2.Navigate to profile section | Username: Maiasha  Email: sultana90@gmail.com  Password: Boishakhi738 | User profile details (name, email, phone, bookings) are displayed correctly. | |  |  |
| Post Condition: Updated profile information is saved in the database and reflected immediately on the user’s profile page. | | | | | |

Table 6: Test Case for **Services**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Sadia Afrose | | |
| Test Case ID: FR\_6 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Service for website | | | Test Execution date: | | |
| Test Title: View Available Services | | |  | | |
| Description: users with additional features and functionalities beyond event booking. | | |  | | |
| Precondition (If any): User must be logged in | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Navigate to “Services” page. 2. Select event. 3. View available services. | Service name: Event Photography  ServiceID: SER20250913-001 | List of services displayed correctly. | |  |  |
| Post Condition: List of services is retrieved from the database and displayed correctly to the user. | | | | | |

Table 7: Test Case for **Notification**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Md Rezwan Nabi | | |
| Test Case ID: FR\_7 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Notification System | | | Test Execution date: | | |
| Test Title: Verify Notification System for Communication Between Roles | | |  | | |
| Description: Verify the email notification system for communication between roles. | | |  | | |
| Precondition (If any): User must be logged in and have access to the module. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Log in  2.Complete payment for a booking  3.Trigger payment success notification | Type: Event Reminder  Delivery channel: In app notification | Notification of successful payment is delivered to user with correct payment details. | |  |  |
| Post Condition: Notification is delivered successfully via the selected channel | | | | | |

Table 8: Test Case for **Calendar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Radia Yesmin Oishi | | |
| Test Case ID: FR\_8 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: | | |
| Module Name: Calendar for website | | | Test Execution date: | | |
| Test Title: Display Upcoming Events in Calendar | | |  | | |
| Description: users to view, manage, and track upcoming events. | | |  | | |
| Precondition (If any): User must be registered and logged in. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Log in  2.Open the calendar  3. Observe displayed events. | Event Date: 18/09/25  Event time: 12:00pm | All upcoming events are displayed correctly on corresponding dates | |  |  |
| Post Condition: Calendar reflects all added, updated, or deleted events accurately. | | | | | |

Table 9: Test Case for **Attendee**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Md Rezwan Nabi | | |
| Test Case ID: FR\_9 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: | | |
| Module Name: Attendee System | | | Test Execution date: | | |
| Test Title: View Attendee List (Admin/Organizer) | | |  | | |
| Description: manages all information and actions related to users who attend events | | |  | | |
| Precondition (If any): User must be registered and logged in. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Log in as organizer/admin  2.Navigate to event attendee list | Username: Rezwan  Email: rezwan852@gmail.com  Password: 21-rezwan-3 | All registered attendees are displayed correctly with details | |  |  |
| Post Condition: Attendee registration is recorded in the system and linked to the corresponding event. | | | | | |

Table 10: Test Case for **Ticket**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Sadia Afrose | | |
| Test Case ID: FR\_10 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Ticket for website | | | Test Execution date: | | |
| Test Title: Select Ticket Type for Event | | |  | | |
| Description: all aspects of ticket creation, purchase, distribution, and validation for events. | | |  | | |
| Precondition (If any): Event must exist and have available tickets | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Log in  2.Navigate to event page  3.View ticket types  4.Select ticket | Ticket type: VIP  Price: $99 | Correct ticket options displayed and selectable for purchase. | |  |  |
| Post Condition: Ticket purchases are recorded and linked to the user and event. | | | | | |

Table 11: Test Case for **Venue**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Maiasha Sultana | | |
| Test Case ID: FR\_11 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Venue for website | | | Test Execution date: | | |
| Test Title: View Venue Details. | | |  | | |
| Description: All information related to event locations. | | |  | | |
| Precondition (If any): User or organizer must be registered and logged in. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Log in  2.Navigate to venue list  3.Click on venue | Username: Maiasha  Email: sultana90@gmail.com  Password: Boishakhi738 | All venue information (name, address, capacity, amenities, availability) is displayed correctly. | |  |  |
| Post Condition: Venue information is stored correctly in the database. | | | | | |

Table 12: Test Case for **Events**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Sadia Afrose | | |
| Test Case ID: FR\_12 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Events for website | | | Test Execution date: | | |
| Test Title: View Event Details (User) | | |  | | |
| Description: manages the creation, updating, scheduling, and viewing of events | | |  | | |
| Precondition (If any): Venues must exist and be available. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| |  | | --- | |  |   1.Log in as user  2.Navigate the event list  3.Click on event   |  | | --- | |  | | Username: Sadia  Email: sadia.afrose70@gmail.com  Password: Sadii | All event information displayed accurately (name, date, venue, tickets, description). | |  |  |
| Post Condition: Events details are stored correctly in the system | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Maiasha Sultana | | |
| Test Case ID: FR\_13 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: payment module | | | Test Execution date: | | |
| Test Title: Make payment | | |  | | |
| Description: manages all financial transactions related to event bookings. | | |  | | |
| Precondition (If any): User must have selected tickets or services for purchase | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Log in  2. Select tickets/services.  3.Select payment method  4. Submit payment | Username: Maiasha  Email: sultana90@gmail.com  Password: Boishakhi738 | Payment processed successfully; | |  |  |
| Post Condition: Payment transactions are recorded in the system and linked to user, event, and ticket/service. | | | | | |

Table 13: Test Case for **Payment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Sadia Afrose | | |
| Test Case ID: FR\_14 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Verification Module | | | Test Execution date: | | |
| Test Title: Verify user account via Email and OTP | | |  | | |
| Description: A newly registered user can verify their account successfully using a verification email link and OTP code. | | |  | | |
| Precondition (If any): A verification email with a link and OTP is sent to the user’s registered email | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| |  | | --- | |  |   1. User opens the verification email from the system  2. User clicks on the verification link provided  3. System redirects to the verification page.  4. User enters the OTP received in the email.  5. User clicks on the “Verify” button. | Username: Sadia  Email: sadia.afrose70@gmail.com  Password: Sadii0519 | The system validates the OTP | |  |  |
| Post Condition: User can now log in and access booking services. | | | | | |

Table 14: Test Case for **Verification**

\

Table 15: Test Case for **Log out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Event Booking | | | Test Designed by: Radia Yesmin Oishi | | |
| Test Case ID: FR\_15 | | | Test Designed date: 29 August 2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Log out from website | | | Test Execution date: | | |
| Test Title: Verify User Log Out Functionality | | |  | | |
| Description: Verify that users can successfully log out from the system | | |  | | |
| Precondition (If any): User must be logged in. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Navigate to the "Home" page/dashboard.  2.Locate the "Logout" option (typically in the profile or settings section).  3.Click on the "Logout" button.  4.Verify that the user is logged out and redirected to the login screen or landing page. | Username: Oishi  Email: oishi7875@gmail.com  Password: RYO213 | Attendee status updated to “Checked-In”; confirmation displayed. | |  |  |
| Post Condition: User is logged out and returns to the login page. | | | | | |

# ITEM PASS/FAIL CRITERIA

|  |  |  |
| --- | --- | --- |
| Test Case | Pass Criteria | Fail Criteria |
| Login Session | System grants access when user enters valid credentials. | Access is denied incorrectly, or valid credentials fail to log in. |
| Registration | Accounts are created and verification emails sent when valid data is entered. | Invalid data accepted, no account created, or verification email not sent. |
| Home Page | Loads within 3 seconds and displays events correctly. | Page fails to load, load slowly (>3s), or events not displayed properly. |
| Dashboard | Personalized view loads correctly with user-specific data. | Incorrect/no data shown, or dashboard fails to load. |
| User Profile | Profile updates are saved and displayed correctly. | Updates not saved, displayed incorrectly, or errors occur. |
| Services | Services load, availability is checked, and booking is possible. | Services don’t load, incorrect availability shown, or booking fails. |
| Notification | User receives accurate Email/SMS/In-App notification within expected time. | Notification not sent, delayed, or incorrect content delivered. |
| Calendar | Events scheduled, recurring events saved, and reminders triggered on time. | Events not saved, recurring setup fails, or reminders not sent. |
| Attendee | Attendees added, displayed, and managed properly. | Attendees not added, displayed incorrectly, or can’t be managed. |
| Ticket | Tickets generated and valid E-ticket issued after payment. | Ticket not generated, invalid, or not sent to user. |
| Venue | Venue details saved, displayed, and linked with events. | Venue not saved, not displayed, or not linked to events. |
| Events | Events created, updated, and displayed correctly in system. | Event creation/update fails, or events not visible. |
| Payment | Valid payment processed, confirmation sent, and transaction recorded. | Payment fails, confirmation not sent, or transaction not recorded. |
| Verification | Email/OTP/ID validated successfully and account verified. | Invalid Email/OTP/ID accepted, or verification fails. |
| Log Out | Session ends and user redirected to login page. | Session not ended, or user not redirected. |

# TEST DELIVERABLES

The materials we will deliver:

* Test plan
* Test cases/results
* SRS document
* Defect reports
* Final summary report
* Future plan for the project

# STAFFING AND TRAINING NEEDS

| **Audience** | **Training Content** |
| --- | --- |
| **Developers** | Training on chosen tech stack (e.g., React, Node.js, MySQL/PostgreSQL, APIs). |
| **QA Engineers** | Training on testing tools (e.g., Selenium, JUnit, Postman) and bug tracking systems. |
| **Admins/Organizers** | Training in event creation, ticketing, notifications, and attendee management. |
| **Customer Support** | Training on handling user queries, troubleshooting login/payment/verification issues. |
| **End Users** | Simple user guide/training videos on booking, payment, and check-in process. |
| **Security Staff** | Training in data privacy, fraud prevention, and compliance (GDPR/PCI DSS if needed). |

# RESPONSIBILITIES

|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| Designer | Designing the system interface. |
| Test case analyzer | Analyzing the test cases, setting meeting agendas. |
| Test case creator | Creating test cases by analyzing the system interface. |
| Requirements coordinator | Deciding all the requirements and planning the testing approach and project risks. |

# TESTING SCHEDULE

A screenshot of a computer screen

AI-generated content may be incorrect.

A chart with colorful squares

AI-generated content may be incorrect.

# PLANNING RISKS AND CONTINGENCIES

Table 16: Risk Mitigation Plan for testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S/N | Risk Description | Probability | Impact | Mitigation Plan |
| 1 | Unrealistic time estimate | 40% | Delay project by 2 weeks | Use multiple estimation techniques (PERT, expert judgment), add buffer time. |
| 2 | Server downtime or crash | 30% | Service unavailability | Cloud hosting, load balancing, failover servers, regular monitoring. |
| 3 | Payment gateway failure | 25% | Users unable to complete bookings | Integrate multiple payment gateways, retry options, manual fallback method. |
| 4 | Data loss or corruption | 20% | Loss of booking/ticket/user data | Daily backups, database replication, recovery plan in place. |
| 5 | Security breach (hacking/fraud) | 15% | Data theft, loss of trust | Encryption, MFA, penetration testing, real-time security monitoring. |
| 6 | High user load (peak time booking) | 35% | System slowdown, failed bookings | Load testing, caching, auto-scaling infrastructure. |
| 7 | Staff turnover/shortage | 20% | Project delays, knowledge gap | Cross-training, proper documentation, maintain backup resource pool. |
| 8 | Miscommunication between teams | 25% | Incorrect implementation, delays | Daily stand-ups, agile practices, central documentation system. |
| 9 | Budget overrun | 30% | Reduced features, financial strain | Regular budget tracking, prioritize MVP features, stakeholder alignment. |
| 10 | Scope creep | 25% | Project complexity and cost increase | Formal change control process, strict stakeholder approval for new features. |
| 11 | Low user adoption | 20% | Poor ROI, underutilized system | Marketing campaigns, user-friendly design, end-user training sessions. |
| 12 | Event cancellation & refunds | 15% | Financial and reputation loss | Automated refund system, clear cancellation policies. |

# APROVALS

This section documents the individuals who have reviewed and approved the **Event Booking System** specification. Approval of this document indicates agreement on the stated requirements, risks, and plans.

|  |  |  |
| --- | --- | --- |
| **Role** | **Signature** | **Date** |
| Project Manager | ------------------------------ | 16/09/2025 |
| Development Team Lead | ------------------------------ | 15/09/2025 |
| QA/Test Lead | ------------------------------ | 14/09/2025 |
| Client Representative | ------------------------------ | 17/09/2025 |

**Text Format**

* Style: Times New Roman
* Size: 12
* Line and Paragraph Spacing: 1.00
* Alignment: Justify