**Assignment 2**

A blue and white logo

Description automatically generated

**Department of Computer Science**

**Submitted by:**

Ch Shamir Hussain SP20-BSE-033

Nouman Muzaffar SP20-BSE-045

Rana Umair Nazar SP20-BSE-083

**Section:** A

**Course Title:** Software Testing.

**Course Code:** CSC432

**Submitted to:** *Ma’am Sana Rizwan*

**Date:** 10/5/2023

# Introduction

## Test Objectives

The system integration test of "Tourist's Terra" should validate that:

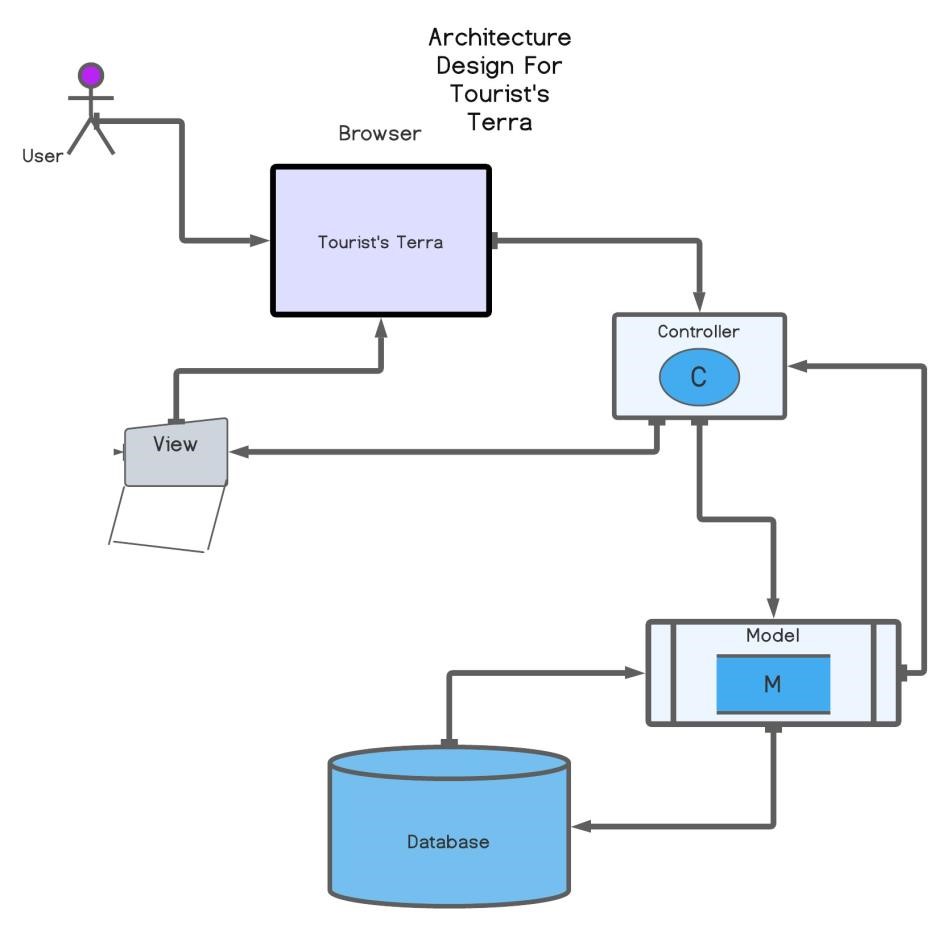
* All core features and functions of the application are working correctly.
* Users can create and manage their travel plans effectively.
* The social connectivity features, allowing users to connect with other travelers, function as expected.
* Service providers can register their accommodations and transportation services.
* Users can explore, book, and review accommodations.
* Users can explore, book, and review transportation services.
* The application complies with security controls to prevent unauthorized access.
* User interactions and communication within the application are seamless.
* The application provides a user-friendly interface for easy navigation.
* Customization features for users and service providers work as specified.
* Financial transactions, if any, are processed accurately and securely.

## Scope of Testing:

The scope of system integration testing for "Tourist's Terra" encompasses a comprehensive evaluation of the core functionalities and integrations within the web application. This testing will focus on critical areas, including user registration and login, travel plan creation and management, social interactions, accommodation and transportation services, user customization, and financial transactions where applicable. Additionally, the testing process will thoroughly assess the user experience and interaction aspects to ensure a seamless and user-friendly interface. Security controls, designed to prevent unauthorized access and safeguard user data, will be rigorously tested.

## System Overview:

"Tourist's Terra" is an integrated, company-wide web application designed to cater to the diverse travel needs of users. It serves as a centralized platform for travellers across various locations. The application connects users with a wide range of travel-related functionalities and services, such as creating and managing travel plans, connecting with fellow travellers, exploring, booking, and reviewing accommodations and transportation services, customizing their travel experiences, and potentially facilitating financial transactions. The system operates in a networked environment, making it accessible to users in various geographical locations. "Tourist's Terra" not only enhances the user experience but also streamlines travel-related tasks, offering users a convenient and efficient platform to fulfil their travel requirements. The application is designed to be user-friendly and efficient, with a focus on performance and usability. Users can access its tools and services, either free of charge or through cost-effective means, depending on their travel needs. "Tourist's Terra" aims to simplify and enhance the travel experience for users across the Pakistan.



## Definitions/Acronym:

### Definition

|  |  |
| --- | --- |
| **Build** | "Tourist's Terra" consists of independently testable and functional software components, allowing for individual testing and integration with other builds as part of the development process. |
| **Critical Processing Unit** | Within "Tourist's Terra," there are critical processing units that play a pivotal role in ensuring the proper functioning of the system, and their failure could have a significant impact. |
| **Model Office** | A model office environment may be employed to validate the implementation, operation, and training aspects of "Tourist's Terra" in a simulated office setting, ensuring the system's effectiveness. |
| **Prototype** | "Tourist's Terra" may include prototypes that offer a working model of the software, providing a preview of its look and feel, even though not all features and functions are fully supported. |
| **Regression Testing** | Testing for "Tourist's Terra" includes regression testing to verify that unaltered parts of the software continue to operate as expected after changes have been made. |
| **Requirement** | "Tourist's Terra" is built to meet various user, business, and technical requirements, ensuring that it delivers the functionalities and capabilities expected by its users. |
| **Static Testing** | Before execution on a computer, static testing techniques, such as document reviews, are applied to ensure the accuracy and correctness of various aspects of "Tourist's Terra." |
| **System Integration Testing** | "Tourist's Terra" undergoes system integration testing to validate its internal and external system interfaces, ensuring that it operates as a coherent and fully functional system, meeting all specified requirements. |
| **Test Tool** | Various testing tools are utilized during the development and quality assurance of "Tourist's Terra" to assist in the testing process and to enhance the efficiency and effectiveness of testing procedures. |
| **Unit Testing** | "Tourist's Terra" features unit testing at the module level, where individual software units or modules are tested in isolation to ensure their correctness and functionality. |
| **Unit-to-Unit Testing** | A level of testing in "Tourist's Terra" that validates the integration between groups of related modules or units to ensure seamless collaboration and functionality between these components |

### Acronyms:

|  |  |
| --- | --- |
| **APIs** | Application Programmable Interfaces |
| **SaaS** | Software as a Service |
| **UI** | User Interface |
| **MERN** | MongoDB, Express, React, Node.js |
| **NPM** | Node Package Manager |
| **SDK** | Software Development Kit |

# Approach:

## Assumptions/Constraints:

### Assumptions:

* Each component or module within "Tourist's Terra," such as the Travel Planning Module, Accommodation Booking Module, and Social Connectivity Module, will undergo thorough individual testing before integration.
* By a target date of October 10, 2023, the first iterations of each major module will be ready for initial integration testing.
* The integration of these modules within the "Tourist's Terra" environment will be seamless post-unit testing, maintaining a smooth workflow.
* The data interfaces between different modules, like Accommodation Booking and Transportation Booking, will be consistent, ensuring effective communication and interoperability.

### Constraints:

* The unique functionalities and complexities of different modules might introduce unforeseen challenges during integration testing, potentially requiring additional time for issue resolutions.
* In some cases, there may be overlap or redundancy in functionalities between modules, such as Customization and Personalization Module and User Registration Module, which could lead to integration issues.
* Given the complexity of the system and the various functionalities of each module, certain integration scenarios might be missed in the initial round of testing, necessitating further iterations.
* Time constraints may limit the exhaustive testing of edge cases, especially when modules interact in complex workflows.
* External dependencies, such as third-party APIs or datasets used by certain modules, could introduce delays or inconsistencies during testing phases.

## Coverage:

* A comprehensive matrix that outlines all testable requirements associated with tools like BG Remover, Transcript AI, Brandite, and Quote Sense, alongside their corresponding test cases.
* A detailed matrix that captures all business processes the toolkit caters to, complemented by the respective business-oriented test scenarios and cases.

### Software Components:

Each software component and module within "Tourist's Terra," spanning Travel Planning Module, Accommodation Booking Module, Social Connectivity Module, User Registration Module, and any additional features, will undergo comprehensive testing to ensure their reliability and functionality.

### Requirement:

All user requirements, as documented in the "Tourist's Terra Requirements Specification Document," will undergo thorough testing. This process ensures that every module aligns perfectly with the intended functionality and meets users' expectations, such as efficient trip planning and seamless accommodation booking.

### Business Processes:

* **Travel Planning:** Verification of the system's ability to facilitate efficient travel planning, enabling users to create itineraries, explore travel destinations, and collaborate with other travelers seamlessly.
* **Accommodation Booking:** Testing the system's efficiency in handling accommodation reservations, from property selection to the confirmation of bookings, ensuring a smooth user experience.
* **Payment Processing:** Validation of secure and efficient payment processing for premium features and services, including subscription management and accurate invoicing.
* **Social Connectivity:** Ensuring that the social connectivity features, such as user interactions, group travel planning, and the sharing of travel experiences through blogs, function without issues.

## Test Tools

* **Selenium**: An open-source tool primarily used for automating web applications, making it ideal for web application testing and ensuring the functionality of the web-based features in "Tourist's Terra."
* **JIRA**: While primarily known as a project management tool, its integration, JIRA Test Management, can be effectively utilized for comprehensive test management, test case tracking, and issue management.
* **LoadRunner**: This performance testing tool by Micro Focus is valuable for evaluating system behavior under load, making it essential for load testing "Tourist's Terra" to ensure it can handle a significant number of users without performance issues.
* **Postman**: An interactive tool used for API testing, which is crucial for "Tourist's Terra" to validate the proper functioning of APIs integrated into the system, ensuring efficient data exchange and communication with third-party services.
* **TestRail**: A comprehensive test case management tool that can help in organizing, managing, and tracking test cases and test results, providing a systematic approach to testing "Tourist's Terra."
* **TestLink**: An open-source test management tool that aids in organizing test cases, plans, and specifications, ensuring efficient test case management for "Tourist's Terra."

## Test Types:

* **Unit Testing:** This phase involves testing each individual component or tool of "Tourist's Terra" in isolation to ensure that it functions correctly on its own. This ensures that each feature and functionality operates as intended.
* **Integration Testing**: After successful unit testing, the focus shifts to integration testing. During this phase, the various components and tools will be tested to ensure they work seamlessly together. It verifies that the interactions between different parts of "Tourist's Terra" are harmonious and error-free.
* **Usability Testing:** This crucial testing type evaluates the overall user-friendliness and user experience provided by "Tourist's Terra." It ensures that both technical and non-technical users can effectively navigate and utilize the toolkit, emphasizing a user-centric approach.
* **API Testing:** Given the reliance on multiple APIs for different functionalities within "Tourist's Terra," API testing is vital. This type of testing ensures that the toolkit effectively communicates with external services, sending and receiving data as expected.
* **Performance Testing:** To measure the responsiveness of "Tourist's Terra," especially when employing resource-intensive AI functionalities, performance testing is conducted. This ensures that response times meet benchmarks and align with user expectations for a seamless experience.
* **Security Testing:** This testing type is dedicated to evaluating the security measures of "Tourist's Terra." It identifies and addresses potential security breaches, unauthorized access, and vulnerabilities that could compromise the system's integrity and user data.
* **Cost-Efficiency Testing:** Given the financial perspective, this testing ensures that "Tourist's Terra" optimally utilizes resources. It focuses on cost-effective resource utilization, particularly when engaging third-party services or APIs, to ensure efficient cost management.

## Test Data:

**User Input Data (USERINPUT):** This database encompasses simulated user inputs for various features and functionalities of "Tourist's Terra." It is a mixture of data sourced from earlier versions of travel applications and test scenarios meticulously crafted for comprehensive testing.

**Search for Travel Destination (User Input) - USERINPUT\_DESTINATION**

**Example:**

{"userID": "12345", "destination": "Lahore, Punjab", "departureDate": "2023-05-15", "returnDate": "2023-05-20", "budget": "Rs15000"}

This simulates a user with ID 12345 searching for a trip to Lahore, Punjab, with specific travel dates and a budget of Rupees 15000.

**Booking Accommodation (User Input) - USERINPUT\_ACCOMMODATION**

**Example:**

{"userID": "67890", "destination": "Islamabad, Pakistan", "checkInDate": "2023-05-15", "checkOutDate": "2023-05-20", "guests": 2, "budget": "Rs10000"}

This represents a user with ID 67890 booking accommodation in Islamabad, Pakistan, specifying check-in and check-out dates, number of guests, and a budget of Rupees 10000.

**Itinerary Planning (User Input) - USERINPUT\_ITINERARY**

**Example:**

{"userID": "54321", "destination": "Naran Kaghan", "Neelam Valley", "activities": ["Camping", "Boating"]}

This illustrates a user with ID 54321 planning an itinerary for their trip to Neelam valley, Naran listing sightseeing locations and selected activities.

**Processed Output Data (PROCESSEDOUTPUT):** This database holds data generated after processing user inputs through various features of "Tourist's Terra." It is crucial for validating if the toolkit delivers the expected outcomes based on the provided inputs.

**Search for Travel Destination (Processed Output) PROCESSEDOUTPUT\_DESTINATION**

**Example:**

{"userID": "12345", "recommendations": ["Saif- ul-malook lake Visit”], "bestDeal": "Pakistan North Explorer Package"}

This data reflects the results after a user with ID 12345 searches for a travel destination in Saif- ul-malook lake, Kaghan. It includes destination recommendations and the best deal found.

**Itinerary Planning (Processed Output) - PROCESSEDOUTPUT\_ITINERARY**

**Example:**

{"userID": "54321", "itinerary": ["Day 1: Karachi sea view and Clifton ", "Day 2: Jinnah Tomb"]}

This represents the generated itinerary for a user with ID 54321 after planning their trip to Karachi, Pakistan, listing daily activities and sights to see.

**Third-party API Responses (APIRESPONSES):** This contains a selection of typical responses from third-party APIs and SDKs that "Tourist's Terra" integrates with. These responses help ensure that the toolkit can handle diverse data responses from external integrations, covering different scenarios and edge cases.

**Third-party Integrations:** Here, we test the integration of "Tourist's Terra" with trusted third-party services. This ensures a wide array of payment options, cost optimization, and efficient financial transactions, as well as thorough analysis to identify potential savings in the system.

**3. Plan**

**3.1 Test Team:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title** | **Level of involvement** | **Responsibilities** |
| **Shamir Hussain** | Test Engineer | Full-time | Test case design and execution |
| **Umair Nazar** | Test Analyst | Full-time | Test environment setup and test execution |
| **Nouman Muzaffar** | QA Analyst | Full-time | Test data creation and analysis |
| **Marukh Saleem** | Supervisor | Part-time | Oversight and guidance |

**3.2 Team Reviews:**

The following reviews will be conducted by the entire test team and the supervisor.

* Test plan review
* Test case review
* Test progress review
* Post-test review

**3.3 Major Task and Deliverables:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Start Date** | **End Date** | **Deliverable(s)** |
| Requirements Analysis | 20/03/2023 | 20/03/2023 | Requirement Traceability Matrix |
| Test Case Design | 20/03/2023 | 20/03/2023 | Test cases and scenarios documented |
| Test Environment Setup | 20/03/2023 | 20/03/2023 | Functional test environment ready for data population |
| Test Data Creation | 20/03/2023 | 20/03/2023 | Datasets and test data prepared |
| System Integration Testing | 20/03/2023 | 20/03/2023 | Testing of integrated modules |
| User Acceptance Testing | 20/03/2023 | 20/03/2023 | Confirmation of system readiness for deployment |

**3.4 Environmental Need:**

**3.4.1 Test Environment:**

* Hardware: PCs/Laptops with specified configurations for testing
* Network: Local or development network setup
* Software: Necessary application, databases, and test tools

**3.4.2 Test Lab:**

* Testing tools and software environments
* Whiteboard for collaboration

**3.5 Training:**

Training for the team members who might require additional knowledge in testing techniques or tools.

**4. Features to be tested:**

**4.1 Platform Core**

***User Registration and Authentication***

* Validate sign-in/sign-up functionality.
* Test social login.
* Verify password change and reset.
* Check user authorization for specific features.

***User Profile Management***

* Test user profile updates and information changes.
* Validate verification from normal user to verified user and vice versa.
* Test blocking or suspending user accounts by the admin.

***User-Generated Content Management***

* Ensure the system can manage posts, comments, and reviews.
* Verify user-reported content management (e.g., deletion of inappropriate content).

***Admin Management***

* Validate admin login functionality and password change.
* Test the management of guidelines and policies.
* Verify the handling of user information, user verification, and number of registered users.
* Test the system's ability to block, suspend, or manage user accounts.

**4.2 Trip Planning & Accommodation**

***Trip Posting and Management***

* Validate the posting of trips and trip approvals.
* Verify search and filtering features for user-posted trips.

***Accommodation Listings & Booking***

* Ensure the functionality to view, review, and approve accommodation listings.
* Test booking accommodations, search, and filters.

**4.3 Social Connectivity**

***Friend & Group Management***

* Validate user-to-user interactions like sending/receiving friend requests.
* Test the creation, updating, and deletion of trekking groups.
* Verify following/hiring local guides and their ad management.

***Blog & Travel Insights***

* Validate blog post creation, updates, and deletion.
* Test user interaction with travel blogger content (like, follow, comment).

**4.4 Payment and User Experience**

***Payment Processing***

* Test multiple payment methods.
* Verify payment encryption and storage for future transactions.
* Test subscription services and notification systems for subscription expiry.

***System Administration & Ease of Use***

* Validate the admin dashboard functionalities.
* Test announcement distribution, user management, and financial transaction handling.
* Verify user navigation, error handling, and informative UI.

***Security and Reliability***

* Validate data encryption and secure authentication methods.
* Ensure protection against common security threats (XSS, SQL injections, etc.).
* Test platform availability, downtime, and failure rates.

**4.5. Test Scenarios**

***User Management***

* Validate user registration and login functionality.
* Test user profile updates, authorization, and verification processes.
* Verify user-generated content handling, including reports and moderation.

***Trip Planning & Accommodation***

* Test trip posting, approval, and management.
* Validate accommodation listings, booking functionalities, and search/filter features.

***Social Connectivity***

* Verify friend/group management and interactions.
* Test the interaction and management with blog/travel insights features.

***Payment and User Experience***

* Validate multiple payment methods and subscription services.
* Test the admin dashboard, announcement distribution, and system usability.
* Verify platform security and reliability.

**5. Features not to be tested**

**5.1 External Integrations**

***Map & Analytics SDKs***

* Pending integrations such as Google Maps and Google Analytics.
* Forthcoming live chat or ticket support.
* Excluded chatbot support features.
* Botic Tool integrations.

**5.2 Admin Dashboard**

***Real-time Data Counts and Historical Analytics***

* Verification of real-time active user count under varying network conditions.
* Analytics spanning extensive periods, including months or years.
* Custom reporting based on ad-hoc user queries.
* Testing settings intended for future releases but currently disabled.

**5.3 Financial Features**

***Currency Conversion and Unintegrated Payment Methods***

* Real-time currency conversion for international payments.
* Payment methods still in negotiation and not yet integrated.

**Testing Strategy**

|  |  |
| --- | --- |
| **Name of the Project or Application** Tourist’s Terra. | **Author** Terra’s Org. |
| **Type of computing Environment**  Web-based application. | **Types of the software** Frontend and Backend modules, Database. |
| **User Demographics** Primarily global tourists, travel enthusiasts, and adventurers. | **Assumptions** Users have basic internet access and familiarity with web-based applications. |
| **Purpose of Testing** Validate the functionality, performance, security, and user experience of Tourists Terra. | **Phases of Testing:**   * Requirement Analysis * Test Planning * Test Design * Test Execution * Defect Reporting and Tracking * Regression Testing * User Acceptance Testing |
| **Scope of Testing:**  ***Included:*** User modules, admin functionalities, accommodation features, user-generated content tools.  ***Excluded:*** External APIs, pending integrations, and finance features under negotiation. | **Critical Success Factors:**   * Error-free accommodation booking. * Secure and stable payment processing. * Reliable user-generated content and engagement. * Seamless user experience across devices and browsers |
| **Types of Testing**   * Functional Testing * Performance Testing * Security Testing * Compatibility Testing * Usability Testing | **Test Profiles**   * End Users (tourists) * Administrators * Different types of accommodations * Varying network speeds and devices |
| **Development Tools and Test Tools** ***Development Tools:*** Git, Visual Studio Code, Node.js, MongoDB  ***Test Tools:*** JIRA for defect tracking. | |
|  | |
| **Risks** ***Data Security:*** Risks associated with handling sensitive user data.  ***Performance Issues:*** Unforeseen traffic causing slow response times.  ***Browser Compatibility:*** Interface may not render uniformly across different browsers. | |
| **Others**   * Weekly status meetings to track testing progress. * Regular status reports to the project stakeholders. * Immediate communication for high-priority issues. | |

# Testing Procedures

## Test Case Execution

### Test Cases

Table 1 Admin Login

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-001 | | Test Case Description: Admin Login |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Ch. Shamir Hussain  Design Date: 6/3/2023 | | **Executed By**: Umair  **Execution Date**: 15-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The admin account exists in the system with valid login credentials. |  | **1** | **umairadmin@gmail.com** |
| 2 |  |  | **2** | **1234@234** |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Verify that the system allows admins to log in using their respective Terra accounts and passwords. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Results** | **Actual Result** | **Pass/Fail** |
| 1. Admin opens the Tourist's Terra application. 2. Admins navigate to the login page. 3. Admin enters the valid Terra account credentials in the respective fields. 4. Admin clicks on the login button. | 1. The application will open. 2. The application will open Login page. 3. The system authenticates the admin's credentials. 4. The system redirects the admin to the admin dashboard. | Pass  Pass  Pass  Pass | Pass  Pass  Pass  Pass |

Table 2 Forget Admin Password

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-002 | | Test Case Description: Forgot Admin Password |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Ch. Shamir Hussain  Design Date: 6/3/2023 | | **Executed By**: Umair  **Execution Date**: 15-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The admin account exists in the system. |  | **1** | **umairadmin@gmail.com** |
| 2 |  |  | **2** | **1234@234** |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Verify that the system guides the admin through the password recovery process and allows them to set a new password. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1. Admin opens the Tourist's Terra application. 2. Admin navigates to the login page. 3. Admin clicks on the "Forgot Password" link. 4. Admin follows the password recovery process (e.g., provide email, answer security questions). 5. Admin sets a new password for the admin account. 6. Admin Log in with the new password. | 1. The Application will open. 2. The application will open the Login page. 3. The Popup appears consisting of form. 4. The system guides the admin through the password recovery process. 5. The admin successfully sets a new password. 6. The system logs in the admin with the new password. | Pass  Pass  Pass  Pass  Pass  Pass | Pass    Pass  Pass  Pass  Pass  Pass |

Table 3 Edit Guidelines and Policies

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-003 | | Test Case Description: Edit Guidelines and Policies |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Ch. Shamir Hussain  Design Date: 6/3/2023 | | **Executed By**: Umair  **Execution Date**: 15-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The admin is logged into the system and has access to the Guidelines and Policies section. |  | **1** | **Alphanumeric Text** |
| 2 |  |  | **2** |  |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Validate the system's ability to allow the admin to edit the Guidelines and Policies. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1. Admin accesses the admin dashboard. 2. Admin navigates to the Guidelines and Policies section. 3. Admin clicks on the edit button. 4. Admin makes the necessary edits or modifications to the Guidelines and Policies. 5. Admin clicks on the save the changes button. | 1. The admin dashboard will open. 2. The application will open the Guidelines and policies page. 3. The application should allow the admin to edit the Guidelines and Policies. 4. The application will display the changes. 5. The System will successfully save the changes and display message “successfully edited”. | Pass  Pass  Pass  Pass | Pass  Pass  Pass  Pass |

Table 4 Update U ser Information

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-004 | | Test Case Description: Update User Information |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Ch. Shamir Hussain  Design Date: 6/3/2023 | | **Executed By**: Umair  **Execution Date**: 15-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The admin is logged into the system and has access to user management functionalities. |  | **1** | **Alphanumeric Text** |
| 2 |  |  | **2** |  |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Verify if the system allows the admin to update user information. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Result** | **Pass/Fail** | **Comment** |
| 1. Admin accesses the admin dashboard. 2. Admin navigates to the user management section. 3. Admin clicks on Users. 4. Admin selects a user whose information needs to be updated. 5. Admin makes the necessary changes to the user's information. 6. Admin clicks on the Save changes button. | 1. The application will open the admin Dashboard. 2. The application will open the User management section. 3. The application will open the User Section. 4. The application will show the selected user. 5. The changes will display on the screen. 6. The system will update the user's information successfully and display a message “Successfully Updated”. | Pass  Pass  Pass  Pass | Pass  Pass  Pass  Pass |

Table 5 Verify Simple User

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-006 | | Test Case Description: Verify simple user |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Ch. Shamir Hussain  Design Date: 6/3/2023 | | **Executed By**: Umair  **Execution Date**: 15-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The admin is logged into the system and has access to user management functionalities. |  | **1** |  |
| 2 |  |  | **2** |  |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Validate the system's ability to allow the admin to change a normal user to a verified user. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1. Admin accesses the admin dashboard. 2. Admin navigates to the user management section. 3. Admin clicks on the ‘Request for verification’ option. 4. Admin selects a normal user to be verified. 5. Admin changes the user's status to "verified." | 1. The admin Dashboard will open. 2. The system will display the User management page. 3. The system will display the ‘Request for verification’ page. 4. The User will select and display on the screen. 5. The system should update the user's status to "verified." | Pass  Pass  Pass  Pass | Pass  Pass  Pass  Pass |

Table 6 Search User

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-007 | | Test Case Description: Search Users |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Ch. Shamir Hussain  Design Date: 6/3/2023 | | **Executed By**: Umair  **Execution Date**: 15-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The admin is logged into the system and has access to user management functionalities. |  | **1** | **Alphanumeric Text (Ali, Ahmad1)** |
| 2 |  |  | **2** |  |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Validate the system's ability to allow the admin to search for users based on specific criteria such as name, email, or location. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1. Admin accesses the admin dashboard. 2. Admin navigates to the user management section. 3. Admin locates the search field or search functionality. 4. Admin enters the specific criteria, such as name, email, or location, to search for users. 5. Admin initiate the search action/ clicks on the search button. | 1. The Dashboard will open. 2. The system will open/display the User Management Section. 3. The search field will activate. 4. The searched keyword will display on the screen. 5. The system should display a list of users matching the entered location. | Pass  Pass  Pass  Pass | Pass  Pass  Pass  Pass |

Table 7 Customer Support

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-019 | | Test Case Description: Customer Support - Respond to User Queries |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Ch. Shamir Hussain  Design Date: 6/3/2023 | | **Executed By**: Umair  **Execution Date**: 15-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The admin is logged into the system and has access to user management functionalities. |  | **1** |  |
| 2 |  |  | **2** |  |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Validate if the system allows the admin to respond back to the queries of the users. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Results** | **Actual Result** | **Pass/Fail** |
| 1. Admin clicks on the admin console. 2. Admin navigates to the customer support interface. 3. Admin locates the user query that requires a response. 4. Admin crafts a response to the user query. 5. Admin sends the response to the user by clicking send button. | 1. The console will open. 2. The Customer support interface opens. 3. The system opens the User Query. 4. The response is written and shown to the Admin. 5. The system transfers the response and display a message “Successfully Responded”. | Pass  Pass  Pass  Pass | Pass  Pass  Pass  Pass |

Table 8 Upload Post

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-031 | | Test Case Description: Upload Post of feeds |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Nouman  Design Date: 6/3/2023 | | **Executed By**: Shamir  **Execution Date**: 17-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The user must have registered account. |  | **1** | **Image,urls** |
| 2 | The user must be logged in. |  | **2** |  |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Verify that the system allows registered users to upload post along with associated details and descriptions. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Results** | **Actual Result** | **Pass/Fail** |
| 1. The user navigates to the home page. 2. The user clicks on the upload post’s section 3. The user selects the photo/video from gallery and click on post button. | 1. The system will display homepage. 2. System will display phone gallery. 3. The system will upload your selected photo/video to your user profile. | Pass  Pass  Pass | Pass  Pass  Pass |

Table 9 Delete Post

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-033 | | Test Case Description: Delete Post of feeds |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Nouman  Design Date: 6/3/2023 | | **Executed By**: Shamir  **Execution Date**: 17-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The user must be logged into the system and have a post to delete. |  | **1** |  |
| 2 |  |  | **2** |  |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Summary:** | Verify that the system allows registered users to delete post(s) from his/her profile. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Results** | **Actual Result** | **Pass/Fail** |
| 1. User navigates to their profile where they can view their posts. 2. User selects the post they want to delete and click on delete button located on post. 3. The user confirms the deletion. | 1. Profile screen will appears. 2. System displays a confirmation message asking the user if they want to delete the post. 3. The system will delete the post from user’s profile. | Pass  Pass  Pass | Pass  Pass  Pass |

Table 10 Add Bank Account Details

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-037 | | Test Case Description: Add Bank Account Details |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Nouman  Design Date: 6/3/2023 | | **Executed By**: Shamir  **Execution Date**: 17-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The user must be logged into the system. |  | **1** | **Alphanumeric Text** |
| 2 |  |  | **2** |  |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Summary:** | Verify that the system allows seller users to add their bank account details. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Results** | **Actual Result** | **Pass/Fail** |
| 1. The verified seller users navigate to their seller account/profile. 2. If verified seller user selects individual option. 3. Verified seller users enter their bank account details. 4. The verified seller users submits the bank account details form. | 1. The system displays the verified seller user to their account page. The system displays two options:  * Individual * Corporate business  1. The system displays a form for the seller user to enter the account holder name, IBAN, branch number, bank name, and bank code. 2. The system validates the entered information (e.g., checks for valid IBAN format). | Pass  Pass  Pass | Pass  Pass  Pass |

Table 11 Payment

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-040 | | Test Case Description: Payment for Purchases |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Nouman  Design Date: 6/3/2023 | | **Executed By**: Shamir  **Execution Date**: 17-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The user has items in their shopping cart. |  | **1** |  |
| 2 | The user has selected the payment method |  | **2** |  |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Summary:** | Verify that the system allows Registered Users to make payment for purchased product. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Results** | **Actual Result** | **Pass/Fail** |
| 1. The registered user clicks on the "Payment" button to initiate the payment process. 2. If registered users, select the bank account payment method. 3. If registered users, select the jazz cash/ easy paisa payment method. 4. The user confirms the payment details by clicking the “proceed to payment”. | 1. System displays the available payment methods. 2. The system shall display screen to enter bank account details. 3. The system shall display screen to enter jazz cash/ easy paisa number. 4. The system processes the payment using the selected payment method. | Pass  Pass  Pass  Pass | Pass  Pass  Pass  Pass |

Table 12 Book Accommodation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case id: TC-043 | | | Test Case Description: Book Accommodation | | | |
| Reviewed by: Sana Rizwan | | | **Version: 1.3** | | | |
| Test Case (Pass/Fail/Not Executed): Pass | | | | | | |
| Designed By: Umair  Design Date: 6/3/2023 | | | **Executed By**: Noman  **Execution Date**: 17-10-23 | | | |
| QA Tester’s Log : | |  | | | | |
| S # | **Prerequisites:** | | |  | **S #** | **Test Data** |
| 1 | The registered user has performed a search for hotels/houses/resorts and has selected a specific accommodation option. | | |  | **1** | **Alphanumeric Text** |
| 2 |  | | |  | **2** |  |
| 3 |  | | |  | **3** |  |
| 4 |  | | |  | **4** |  |

|  |  |
| --- | --- |
| **Summary:** | Verify that the system allows registered users to book the accommodation. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Results** | **Actual Result** | **Pass/Fail** |
| 1. The user has selected a specific hotel/house/resort from the search results. 2. The user selects the desired room type(s) and specifies the number of rooms required. 3. The user chooses the dates of stay by entering the check-in and check-out dates. 4. The user proceeds the booking process by providing payment details through different payment methods. | 1. The system displays the result of the selected hotel/house/resort on the screen. 2. The system will save the information of user selection. 3. The system will save the information of user selection. 4. The system validates the payment information and processes the payment transaction securely. | Pass  Pass  Pass  Pass | Pass  Pass  Pass  Pass |

Table 13 Search by Search bar

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-045 | | Test Case Description: Search by Using Search Bar/Icon. |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Umair  Design Date: 6/3/2023 | | **Executed By**: Noman  **Execution Date**: 17-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The User Must be logged In to Tourist Terra Application. |  | **1** | **Alphanumeric Text** |
| 2 | The user must be on required page. |  | **2** |  |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Verify that the user can search local guide, city, trekking group, transportation, blog by clicking on the search icon. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Results** | **Actual Result** | **Pass/Fail** |
| 1. Successfully login to application. 2. User navigates the search bar on the page. 3. User enters the specific credentials which they want to search in the search bar press the search icon. | 1. Application opens the home page for user. 2. The System successfully navigates User to search bar. 3. The system will successfully display the searched result. | Pass  Pass  Pass  Pass | Pass  Pass  Pass  Pass |

Table 14 Create Trekking group.

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-46 | | Test Case Description: Create Trekking group. |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Umair  Design Date: 6/3/2023 | | **Executed By**: Noman  **Execution Date**: 17-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The User Must be logged In as verified user on Tourist Terra Application. |  | **1** |  |
| 2 | User must be on the home page. |  | **2** |  |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Verify that the verified user can create trekking groups by clicking on the create group button. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Results** | **Actual Result** | **Pass/Fail** |
| 1. The user clicks on the "Create New Group" button to initiate the group creation process. 2. The user fills in the required details for the trekking group and submits the request to create the trekking group by clicking the submit button | 1. The application presents a form for creating a new trekking group. 2. The application successfully creates the new trekking group after analysing the provided details. | Pass  Pass | Pass  Pass |

Table 15 Avail Service of Service provider

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-48 | | Test Case Description: Avail services of service providers. |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Umair  Design Date: 6/3/2023 | | **Executed By**: Noman  **Execution Date**: 17-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The User Must be logged In to Tourist Terra Application. |  | **1** | **Alphanumeric Text** |
| 2 | The details of service providers must be populated in the Application |  | **2** |  |
| 3 | The User must be on desired post detail page. |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Verify that the user can avail services (Hire, Schedule) of service providers by using the Tourist Terra. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Results** | **Actual Result** | **Pass/Fail** |
| 1. The user clicks on the "Hire or Schedule" button to initiate the service availing process. 2. The user enters the required details and submits the form. 3. The user confirms their decision to avail service. 4. The service provider receives the hiring/scheduling request and can review the details and accept or reject the request accordingly. | 1. The application prompts the user to enter relevant details, such as the desired date, duration, and specific requirements for the availing services. 2. After Analysing the request, the application prompts the user to confirm their decision to avail service. 3. The application sends the request to the respective service provider. 4. The application notifies the user of the service provider’s response in inbox. | Pass  Pass  Pass  Pass | Pass  Pass  Pass  Pass |

Table 16 Post Update Ad\_Post

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-49 | | Test Case Description: Post, Update Ad. |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Umair  Design Date: 6/3/2023 | | **Executed By**: Noman  **Execution Date**: 17-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The User Must be logged In to Tourist Terra Application. |  | **1** | **Image ,URLs ,Alphanumeric Text** |
| 2 | The User must be verified as service provider on application. |  | **2** |  |
| 3 |  |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Verify that the verified service provider can post, update the ad on tourist Terra. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Results** | **Actual Result** | **Pass/Fail** |
| 1. The service provider selects the option to post an ad. 2. The service provider enters the details of the ad, including service description, pricing, availability, contact information, and any other relevant information and submit the filled-out post.   -----Update-----   1. After updating the post, the user clicks on the "update ad" button to initiate the ad upgradation process.      1. The verified user confirms the update by clicking on the confirm update button. | 1. The application presents the ad posting form to the service provider. 2. The application validates the provided information for completeness and correctness. If the information is complete and correct, the application posts the ad.   -----Update-----   1. The application verifies the upgradation and displays a message asking user to confirm upgradation of the ad. 2. The application updates the ad from the application. | Pass  Pass  Pass  Pass | Pass  Pass  Pass  Pass |

Table 17 Add Reviews, Add Rating, Report ad post

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-50 | | Test Case Description: Add Review, Add Rating and Report the posted ad. |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Umair  Design Date: 6/3/2023 | | **Executed By**: Noman  **Execution Date**: 17-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The User Must be logged In to Tourist Terra Application. |  | **1** | **Alphanumeric Text** |
| 2 | The User must be registered on application.  . |  | **2** |  |
| 3 | User must be on the desired post detail page. |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Verify that the user can add review**,** add rating and report the posted ad on tourist Terra. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Results** | **Actual Result** | **Pass/Fail** |
| 1. The user clicks the add review button to post review on the ad. 2. The user provides the service using details.   -----Add Rating-----   1. The user clicks the add rating button to post rating on the ad. 2. The user provides the service using.   -----Report-----   1. The user clicks the report button to report the ad. 2. The user provides the service using. | 1. The application ask user to provide service using details. 2. The application validates the provided details for completeness and correctness. If the detail is complete and correct, the application adds the review on the ad.   ----- Add Rating -----   1. The application ask user to provide service using details. 2. The application validates the provided details for completeness and correctness. If the detail is complete and correct, the application adds the rating on the ad.   ----- Report -----   1. The application ask user to provide service using. 2. The application validates the provided details for completeness and correctness. If the detail is complete and correct, the application reports the ad. | Pass  Pass  Pass  Pass | Pass  Pass  Pass  Pass |

Table 99 Become Service Provider

|  |  |  |
| --- | --- | --- |
| Test Case id: TC-51 | | Test Case Description: Become service provider. |
| Reviewed by: Sana Rizwan | | **Version: 1.3** |
| Test Case (Pass/Fail/Not Executed): Pass | | |
| Designed By: Umair  Design Date: 6/3/2023 | | **Executed By**: Noman  **Execution Date**: 17-10-23 |
| QA Tester’s Log : |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S # | Prerequisites: |  | S # | Test Data |
| 1 | The User Must be logged In to Tourist Terra Application. |  | **1** |  |
| 2 | The User must be registered on application.  . |  | **2** |  |
| 3 | User must be on the desired post detail page. |  | **3** |  |
| 4 |  |  | **4** |  |

|  |  |
| --- | --- |
| **Test Scenario:** | Verify that the user can become a travel service provider by using the Tourist Terra. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow of events** | **Expected Results** | **Actual Result** | **Pass/Fail** |
| 1. The user clicks on the required option to become service provider "i.e., Become guide, Become Transport provider, Become blogger etc" button. 2. The user enters their personal information, including password, email, and any other required details and submits the filled form. | 1. The application presents the Verified service provider form to the user. 2. The application validates the provided information for completeness and correctness and if the information is complete and correct it verifies the user and generate a unique code for user and notify the user. | Pass  Pass | Pass  Pass |

### Order of Testing

The order of testing will be done by the build order. So with each build following order will be followed:

#### Build 1

* Admin login
* Admin Police and Guidelines
* User Info

#### Build 2

* Verify User
* Search User
* Customer Support

#### Build 3

* Post in feed
* Deleted Post from Feed
* Payment Module

#### Build 4

* Accommodation Booking
* Add, delete, Update Reviews
* Become Service Provider

## Pass/Fail Criteria

### 6.2.1Build 1:

#### Admin Login:

Pass: Admin can successfully log in using valid credentials.

Fail: Admin login fails, or valid credentials are rejected.

#### 6.2.1.2Admin Policies and Guidelines:

Pass: Admin can access, view, and manage policies and guidelines.

Fail: Admin cannot access or manage policies and guidelines.

#### 6.2.1.3User Info:

Pass: Admin can access and view user information.

Fail: Admin cannot access or view user information.

### Build 2:

#### 6.2.2.1 Verify User:

Pass: The system successfully verifies user accounts.

Fail: User verification fails or is inaccurate.

#### Search User:

Pass: Admin can perform user searches with accurate results.

Fail: User search functionality is not working correctly, or results are inaccurate.

#### Customer Support:

Pass: Admin can access and provide customer support.

Fail: Admin cannot access or provide customer support.

### 6.2.3Build 3:

#### Post in Feed:

Pass: Users can post content in the feed, and it's displayed correctly.

Fail: Posting fails, or displayed content is incorrect.

#### Deleted Post from Feed:

Pass: Users can delete their posts, and the posts are removed from the feed.

Fail: Post deletion does not work or leaves remnants in the feed.

#### Payment Module:

Pass: The payment module correctly processes transactions and payments.

Fail: Payment processing fails or is inaccurate

### 6.2.4 Build 4:

#### Accommodation Booking:

Pass: Users can successfully book accommodations, and bookings are accurately recorded.

Fail: Booking functionality fails, or bookings are not recorded correctly.

#### Add, Delete, Update Reviews:

Pass: Users can add, delete, and update reviews, and the changes are reflected accurately.

Fail: Review operations do not work as intended, or changes are not reflected.

#### Become Service Provider:

Pass: Users can successfully become service providers with access to relevant features.

Fail: The process to become a service provider is not successful, or access to features is not granted as expected.

# Risk and Contingencies

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Category** | **Probability** | **Impact** | **RMMM** |
| Lack of expertise | *DE* | *30%* | *2* | *R01* |
| Changing Requirements | *PR* | *50%* | *3* | *R02* |
| Time Constraints | *PR* | *35%* | *3* | *R03* |
| Resource Limitation | *DE* | *35%* | *3* | *R04* |

# 7.1 RMMM Plan Risk 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk information sheet** | | | | | |
|  | Risk ID:  R01 | Date:  6/2/2023 | Probability:  30% | Impact:  2 |  |
| **Description:**  The lack of expertise refers to the insufficient knowledge or skills in a specific area that may hinder the successful completion of the project. | | | | | |
| **Refinement/context:**  This risk may arise due to the team's limited experience or unfamiliarity with certain technologies, methodologies, or subject matter related to the project. | | | | | |
| **Mitigation/Monitoring:**   * Collaborate with faculty members or experts in the relevant field to provide guidance and mentorship. Seek their advice and leverage their expertise to overcome knowledge gaps. * Engage in self-learning by exploring relevant resources such as books, online tutorials, or courses to acquire the necessary expertise. * Regularly seek feedback from your project supervisor or mentor to identify areas where additional knowledge or skills are required. | | | | | |
| **Management/contingency plan/trigger:**   * Establish a clear plan for knowledge acquisition and skill development. Define specific milestones and goals to measure progress. * If the team members find themselves unable to acquire the necessary expertise within the defined timeline, consider reallocating tasks or seeking external assistance from consultants or contractors with the required expertise. | | | | | |
| **Current status:**  6/9/2023: Mitigation steps initiated. | | | | | |
| Originator: Sana Rizwan. Assigned: Ch. Shamir Hussain | | | | | |

## 7.2 RMMM Plan Risk 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk information sheet** | | | | | |
|  | Risk ID:  R02 | Date: 2/5/2023 | Probability:  50% | Impact:  3 |  |
| **Description:**  The risk of changing requirements refers to the possibility that the project requirements may undergo modifications or additions during the course of the project, which can impact the project's scope, timeline, and resources. | | | | | |
| **Refinement/context:**  This risk may arise due to evolving business needs, market trends, or new insights gained during the project's execution. It can lead to rework, delays, or misalignment with stakeholder expectations. | | | | | |
| **Mitigation/Monitoring:**   * Conduct thorough discussions with your project supervisor and stakeholders to clearly define and document the initial project requirements. Ensure that all parties have a shared understanding of the scope and deliverables. * Maintain open and regular communication with your project supervisor and stakeholders to identify any potential changes or new requirements as early as possible. * Implement a change control process that requires formal documentation, impact assessment, and approval for any requested changes to the project requirements. | | | | | |
| **Management/contingency plan/trigger:** | | | | | |
| * Continuously track and monitor any requested changes or modifications to the project requirements. * Regularly review the progress and feedback from stakeholders to identify emerging trends or potential shifts in requirements. | | | | | |
| **Current status:**  6/9/2023: Mitigation steps initiated. | | | | | |
| Originator: Sana Rizwan. Assigned: Nouman Muzaffar. | | | | | |
|  | | | | | |

## RMMM Plan Risk 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk information sheet** | | | | | |
|  | Risk ID:  R06 | Date: 6/9/2023 | Probability:  25% | Impact:  3 |  |
| **Description:**  The risk of end users not adopting the system refers to the possibility that the intended users may not fully embrace or utilize the system as intended. This can result in reduced benefits, low user satisfaction, and potentially project failure. | | | | | |
| **Refinement/context:**  This risk may arise due to various factors such as resistance to change, lack of awareness about the system's benefits, or a mismatch between user expectations and the system's capabilities. | | | | | |
| **Mitigation/Monitoring:**   * Engage potential end users or the target audience early in the project to understand their needs, expectations, and potential challenges in adopting the system. * Conduct user research, surveys, or interviews to gather feedback and identify user requirements and preferences. * Involve users in the design and development process through usability testing, focus groups, or iterative feedback loops. | | | | | |
| **Management/contingency plan/trigger: :**   * Incorporate user feedback into the project development process to refine the system and align it with user requirements and expectations. * Provide comprehensive user documentation, training materials, and support resources to facilitate system adoption and ensure users have the necessary knowledge and skills. | | | | | |
| • If user adoption remains low or feedback indicates significant usability or acceptance issues, consider conducting additional user research and making further improvements to address the identified issues. | | | | | |
| **Current status:**  6/9/2023: Mitigation steps initiated. | | | | | |
| Originator: Sana Rizwan. Assigned: Rana Umair Nazar | | | | | |

# Work Breakdown Structure:

