**A PROPOSED OFFERING OF A WEB-BASED TRAVEL BOOKING SYSTEM FOR RARE EXPLORER TRAVEL&TOURS AT DATAMEX COLLEGE OF SAINT ADELINE VALENZUELA BRANCH**

A Project Proposal Presented to the

Faculty of Datamex College of Saint Adeline, Inc.

In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science in Information Technology

By:

Alelojo,Red Hernie

Wong,Chrisdelyn S.

September 2025

**REQUIREMENT SPECIFICATION**

**INTRODUCTION**

The project of this document is to provide the specifications for an online web-based travel booking system for Rare Explorer Travel & Tours. The system is being built to replace and modernize an existing slower manual booking process, and will ultimately provide customers with a faster and more accessible travel service browsing, selection, and booking option - while also supporting the company in managing bookings, customer records, and travel packages through an easy-to-use admin panel.The document will aim to provide developers, designers, and project stakeholders a useful guide during developments of the overall system.

In addition, the system should have responsive capabilities to improve overall user experience, and will provide an easy-to-use and clean user interface. As a web-based system, it has the additional benefit of being accessed at anytime and from anywhere there is an internet connection.The system will provide customers with an easier option for travel booking and support the operation of the business by opening up a much larger audience, and smooth transactions handled without any hassle.

**Overview**

The all-new travel booking system is designed to take the old manual procedure and turn it into something safe and modern. It will be an online platform used by both customers & staff.

For customers, the system will allow them to create an account, log in securely, and see all of the various travel options they have available. They will be able to search for tours, look at details of what the destination offers, look at destination history, costs, and itineraries inside the booking. The customer will be able to book directly inside the website, paying online, and receiving confirmation, receipts, and any e-tickets/itineraries immediately on that page. Each user will also have access to their own dashboard to see all bookings made.

For Staff, there will be a separate admin dashboard, that will allow the staff/admin to govern the entire system. Staff will be able to update the travel package description,track payments,confirm bookings,respond to customers inquires.

In addition, the system will generate reports and summaries to allow the company to remain organized and work more efficiently. The system will work on any device that has internet access, including desktops, laptops, tablets, and smart-phones. This makes it very flexible; customers can book while on the go, and staff can manage bookings anywhere they are.

**Scope of the Requirement Specification**

The application intends to simplify the user experience for travel management by creating a helpful and clear application for travelers to acquire tickets. The system will have its own relational database from which the reservation system will function. The database server will support hundreds of major cities all over the globe. Most importantly, we hope to provide a pleasant experience for the user, while putting the best possible prices in front of them. It should serve the developers, and will be the basis of validation for the end system delivered. Any changes to the requirements later on will be made with approval from the client. It is the responsibility of the developer to make requests for clarification, if needed.

**FUNCTIONAL REQUIREMENTS**

Functional requirements are the actions, tasks, and functions that the system must be able to do. They describe what the system should be able to offer its end users and admins (e.g. log in, book trips, make payments, receive notifications, generate reports, etc.). In short, they describe the primary features that make the system function

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REQUIREMENT ID | REQUIREMENT DESCRIPTION | PRIORITY | DEPENDENCIES | ACCEPTANCE |
| FR-001 | Users must be able to search for tours by destination,date and price | High | None | Search results must display relevant tours based on input filters |
| FR-002 | Users must be able to create an account and log in. | High | None | Users can successfully register, log in, and access their profile. |
| FR-003 | Users must be able to book a tour and receive confirmation | High | FR-002 (User Login) | Booking is stored in the system and a confirmation email is sent to the user. |
| FR-004 | Admins must be able to add,update and delete tours from the system | High | Admin Access Module | Admin dashboard allow full CRUD functionality on tours. |
| FR-005 | The system must generate an invoice after a successful booking. | Medium | FR-003 (Booking Functionality) | Invoice is view able/download after booking and contains all required details. |
| FR-006 | Users must be able to view and manage their bookings from their account. | Medium | FR-003,FR-002 | Users can see current, past, and canceled bookings and update/cancel future bookings. |
| FR-007 | The system should send automated email notifications for booking updates and reminders. | Medium | FR-003,Email Service Integration | Emails are triggered correctly for booking, cancellation, or updates. |
| FR-008 | The system should allow users to make secure online payments. | High | Payment Gateway Integration | Payment is processed securely and confirmation is shown. |
| FR-009 | Admins must be able to generate reports on bookings, revenue, and customer data. | Low | Admin Module,Booking Data | Reports can be downloaded in PDF/Excel format from the admin dashboard. |
| FR-010 | The system should send automated email notifications for booking updates and reminders. | Medium | FR-001 (Search Feature) | Users can view tours with discounts using filter options. |

*Table 1:Functional Requirement*

**Non-Funtional Requirements**

Non-functional requirements denote how the system should perform its task instead of the purpose of the task. These could cover things like speed, security, reliability, and ease of use.

· **Performance** – The system should load tour listings, booking pages, and user profiles in less than 3 seconds, even when handling hundreds or thousands of active bookings. Reports and invoices should generate within a few seconds without delay.

· **Usability** – The system should have a simple and user-friendly interface with clear menus for Home, Tour Packages, Bookings, Payments, Profile, Support, and Admin Dashboard. Users should be able to navigate the platform easily without needing technical help.

· **Reliability** – The system should operate smoothly during business hours without frequent crashes or downtime. Data such as bookings and payment records should be automatically backed up to prevent loss.

· **Security** – Users must log in using secure username and passwords. Different user roles (like customers, staff, and admins) should only have access to the features they are authorized to use. Sensitive information, including personal and payment details, should be protected through encryption.

· **Scalability** – The system should be able to handle more users, bookings, and tours as the company grows, without slowing down or requiring major system changes.

· **Maintainability** – The system should be built in a way that makes it easy to update, fix bugs, or add new features in the future. Clear technical documentation should be available to support ongoing development and maintenance.

**USE CASE**

Use cases show the different ways users and administrators interact with the system. Use cases outline step-by-step specific tasks for users, such as logging in, booking a trip, making a payment, or managing travel packages. Use cases describe what the system will do from the perspective of the users.

**UC-001: Search for Tour Packages**

· **Description**: The user searches for available tour packages by entering criteria such as destination, travel dates, and budget. The system displays a list of matching packages.

· **Actors**: User (Customer)

· **Preconditions**: User is on the website or logged into the platform.

· **Post-conditions:** The system displays relevant tour packages based on the search filters.

·  **Alternate Flows:** The system displays a no tour available.

**UC-002: Book a Tour**

· **Description**: The user selects a tour package, enters traveler details, chooses a date, and proceeds to confirm the booking.

· **Actors**: User (Customer)

· **Preconditions**: User must be logged in and have selected a tour package.

· **Post-conditions:** Booking is recorded in the system and a confirmation email is sent to the user.

· **Alternate Flows**: Tour becomes unavailable during booking: User is notified and asked to select another tour or date.

**UC-003: Manage Bookings**

· **Description**: The user views, updates, or cancels their tour bookings from their account dashboard.

· **Actors**: User (Customer)

· **Preconditions**: User must be logged in and have existing bookings.

· **Post-conditions:** Changes are saved, and the system reflects updated booking information.

· **Alternate Flows**: Cancellation deadline passed: System prevents cancellation and notifies the user.

**UC-004: Admin Add’s New Tour Package**

· **Description**: Admin logs into the system and creates a new tour package by entering details such as name, price, itinerary, and availability.

· **Actors**: Admin

· **Preconditions**: Admin must be logged in with the proper permissions.

· **Post-conditions:** New tour is saved and available to users for browsing and booking.

· **Alternate Flows**: System shows a warning to avoid duplicates.

**DATA REQUIREMENTS**

Data requirements refer to the information that the system needs to store, process, and manage. This can include information about users, bookings, payments, and travel packages. It explains what type of data will be required, how it will be structured, and how it will be utilized in the system to ensure the smooth and accurate operation of the system.

· **Users** – Stores information about system users including ID, full name, email, password, phone number, role (e.g., customer, admin), and registration date.

· **Tour Packages** – Stores details of available tours such as tour ID, name, destination, description, start and end dates, price, availability, and tour type (e.g., group or private).

· **Bookings** – Stores records of user bookings including booking ID, user ID, tour ID, booking date, number of travelers, total cost, and booking status (e.g., confirmed, canceled).

· **Payments** – Stores user payments including payment ID, booking ID, payment date, payment method, amount paid, transaction reference, and payment status.

· **Invoices** – Stores billing information related to bookings, including invoice ID, booking ID, invoice date, amount, tax, total amount due, and payment status.

· **Reviews** – Stores user feedback on tour packages including review ID, user ID, tour ID, rating (1–5), comment, and review date.

· **Documents** – Stores uploaded files such as booking confirmations, payment receipts, travel permits, or ID scans, linked to users, bookings, or payments.

**Relationship**

* A user can make many bookings and write many reviews.
* A tour package can have many bookings and many reviews.
* Each booking generates one invoice and one payment.
* Each invoice can be linked to multiple payments (in case of installment payments).
* Documents can be attached to users, bookings, payments, or invoices.

**ASSUMPTIONS AND CONSTRAINTS**

Assumptions are the things we expect to be true when we are creating the system. Assumptions are things we believe will happen, but they are not guaranteed. For example, we have the assumption that users will have an internet connection and a device to interact with the system.

Constraints are limits or constraints on the system which may introduce restrictions. Constraints may be based on time, budget, technology, or resources.

**Assumption**

* Users will have internet and use common web browsers like Chrome or Firefox.
* Users will enter correct personal and payment info when booking.
* The system will use a payment service that accepts credit cards and popular online payments.
* Emails will be sent to confirm bookings and send updates.
* Staff managing the system know how to use it properly.
* The number of users and bookings will grow gradually, not explode suddenly.

**Constraints**

* The system must follow privacy laws to protect user data.
* The system can only use the server space and internet speed that we have.
* Payments depend on the payment service being online and working well.
* At first, the system will only work in English and with a few currencies.
* Updates and maintenance should happen when few people are using the system.
* The system works best on current browsers, but might not work well on very old ones.

**GLOSSARY**

The glossary is a collection of significant terms and words used in the project, which includes simple explanations. It provides readers with a method of explaining the technical or less common words in a simple way. The glossary breaks down and clarifies concepts of the study, making it easier to process and follow. The glossary also provides brief definitions as a reference for students, teachers and other future readers, who may not know some terminology shared throughout the system.

· **User**: A person who uses the travel booking system, including customers, staff, and administrators.

· **Tour Package**: A set of travel services offered together, including destination, itinerary, dates, and price.

· **Booking**: A confirmed reservation made by a user for a specific tour package.

· **Payment**: The transfer of money from a user to Rare Explorer Travel & Tours to confirm a booking.

· **Invoice**: A document detailing the charges for a booking, including costs and taxes.

· **Review**: Feedback or rating given by a user based on their experience with a tour package.

· **Document**: Digital files linked to bookings or users, such as receipts, permits, or identification.

· **Admin**: A user with special permissions to manage tours, bookings, payments, and users.

· **Customer**: A user who browses tours and makes bookings.

· **Scalability**: The system’s ability to handle growing numbers of users and bookings without slowing down.

· **Security**: Measures to protect user data and payment information from unauthorized access.

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description of Changes** | **Authors()** |
| Aug 28, 2025 | **1.0** | First draft of the requirements prepared. | Project Team |
| Sept. 5, 2025 | **1.1** | Added more details to requirements and the format. | Project Team |
| Sept.08,2025 | **1.2** | Fix the format and add a info | Project Team |

Table 2 .Revision Table

**APPENDIX**

*OneClick IT Consultancy. (2020, August 8). Travel booking software: A closer look at types, features, and advancements*

*[https://www.oneclickitsolution.com/blog/travel-booking-software-a-closer-look-at-types-features-and-advancements](https://www.oneclickitsolution.com/blog/travel-booking-software-a-closer-look-at-types-features-and-advancements" \t "_blank))*

*CourseHero. (n.d.). Software requirement specification for travel application*

*<https://www.coursehero.com/file/66160597/Software-requirement-Specification-for-Travel-applicationdocx/>*