**MSE800 Assessment II**

Hengpan He

Wen Liang

Arnold Aristotle Tayag

**Introduction**

As the tourism industry grows rapidly, online tour booking has become essential for tourists and operators. To address this, we are developing a Tour Booking Management System using Agile methodology. This system will help users quickly search and book tours while providing operators with tools to manage their offerings. The frontend will use HTML, CSS, and JavaScript, while Node.js and Python will handle backend logic. One–week sprints allow flexibility in meeting user requirements.

The goal of this project is to create a user-friendly platform for tourists to find tour information, book, pay, cancel reservations, and leave reviews. Operators and administrators will have tools to manage tours, user accounts, generate reports, and monitor system performance to ensure efficiency.

The project focuses on New Zealand tours, including browsing information, online booking, reviews, and monitoring. The development will be divided into three releases, gradually adding features at each stage.

The main goal is to deliver a comprehensive and user-friendly online booking platform to meet both tourists' needs and enhance operators' efficiency.

**Project Overview**

The Tour Booking Management System will allow users to search, book, and manage tours while providing tour operators with tools to create and manage tour offerings. The system will include various features that enhance user experience and operational efficiency.

**Key Features:**

* **Tour Browsing:** View detailed information (description, price, schedule, ratings).
* **Manage Online Booking:** Secure booking, email confirmations, real-time status updates.
* **Payment Processing:** Integration with payment gateways to facilitate secure online transactions.
* **User Registration and Profiles:** Users can create accounts to manage bookings, save favorite tours, and receive personalized recommendations
* **Tour Review:** Users can leave reviews and rate their experiences, helping future customers make informed decisions.
* **Admin Interface:** Manage user accounts, generate reports, and monitor system performance.
* **System Monitoring & Reporting:** Centralized performance and error log tracking.
* **Customer Support:** Email support for inquiries.
* **System Integration:** API for third-party integrations.

**Benefits:**

* User-Friendly Interface: Intuitive design ensures users can easily navigate the system and find suitable tours
* Streamlined Operations: Helps tour operators manage bookings and customer interactions efficiently
* Cultural Sensitivity: By including tours that focus on local Māori culture and history, the system can respect and promote indigenous perspectives
* Increased Visibility: Operators can showcase their tours to a wider audience, increasing bookings and revenue

**Potential Extensions:**

* Mobile Application: A companion mobile app for users to book and manage tours on the go
* Social Sharing Features: Allow users to share their experiences on social media to promote the tours and attract more customers
* Multilingual Support: The system can support multiple languages, catering to international users
* Calendar Integration: A calendar feature that shows available dates for each tour, allowing users to see availability in real time

**Project Plan**

**System Environment:**

* Backend: Node.js + Python
* Frontend: HTML, CSS, JavaScript (React/Vue)
* Database: SQLite
* Hosting: Local server or cloud-based
* Tools: Git for version control, Jira for project management

**Project Roles:**

* Product Owner / Business Users / Stakeholders / Sponsor: Provides requirements and feedback on the product's functionality.
* Agile Facilitator: Oversees project progress, facilitates meetings, and ensures agile practices are followed.
* Project Team: Responsible for designing, developing, and testing the system.

Team Structure & Governance

Project Kick-off Meeting

Scope of Work

System Design

**Product Backlog Prioritization & Refinement**

Product Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Seq# | Functionality | User Story | Acceptance Criteria | Prioritization |
| User Story 1 | For Tourists, Customers | As a tourist, I want to browse available tours, so that I can find options that interest me | Users can filter tours by category, location, and price | Release 1 |
| User Story 2 | For Tourists, Customers | As a tourist, I want to view detailed information about each tour, so that I can make informed decisions | Each tour page includes a description, itinerary, duration, price, and reviews | Release 1 |
| User Story 3 | For Tourists, Customers | As a tourist, I want to book a tour online, so that I can secure my spot easily | Users can select dates, number of participants, and complete payment through a secure gateway | Release 1 |
| User Story 4 | For Tourists, Customers | As a tourist, I want to receive confirmation emails after booking, so that I have all the details at hand | A confirmation email is sent immediately after booking, containing all relevant details | Release 1 |
| User Story 5 | For Tourists, Customers | As a tourist, I want to be able to cancel or modify my booking, so that I can manage my plans flexibly | Users can cancel or modify bookings within a specified time frame, with a clear refund policy | Release 1 |
| User Story 6 | For Tourists, Customers | As a tourist, I want to leave reviews for tours I’ve taken, so that I can share my experiences with others | Users can rate tours and provide written feedback after the tour is completed | Release 2 |
| User Story 7 | For Tourists, Customers | As a user, I want to log in and set my language preference (e.g., Māori) | Users can set preferred language to any language supported by the system | Release 1 |
| User Story 8 | For Tour Operators, Administrators | As a tour operator, I want to create and manage tour listings, so that I can keep my offerings up to date | Operators can add, edit, or delete tours and manage details like pricing and availability | Release 1 |
| User Story 9 | For Tour Operators, Administrators | As a tour operator, I want to view booking reports and analytics, so that I can assess the performance of my tours | Operators can access dashboards showing sales, cancellations, and customer feedback | Release 2 |
| User Story 10 | For Tour Operators, Administrators | As a tour operator, I want to manage customer inquiries, so that I can provide timely responses | Operators have a messaging system to communicate with customers regarding inquiries or concerns | Release 2 |
| User Story 11 | For Tour Operators, Administrators | As a tour operator, I want to set seasonal pricing and promotions, so that I can attract more customers | Operators can schedule discounts and special offers that apply to specific dates or tours | Release 2 |
| User Story 12 | For Tour Operators, Administrators | As a tour operator, I want to manage payment processing, so that I can receive payments securely | The system integrates with payment gateways, allowing for secure transactions and providing operators with transaction reports | Release 1 |
| User Story 13 | For Admins | As an admin, I want to manage user accounts, so that I can maintain the integrity of the platform | Admins can create, edit, and deactivate user accounts for both customers and operators | Release 1 |
| User Story 14 | For Admins | As an admin, I want to monitor system performance and error logs, so that I can ensure the system runs smoothly | Admins have access to performance metrics and can view logs for troubleshooting | Release 2 |
| User Story 15 | For Admins | As an admin, I want to enforce policies and regulations for tour operators, so that the platform maintains quality standards | Admins can set guidelines for tour operators and review compliance | Release 3 |
| User Story 16 | For Admins | As an admin, I want to generate reports on user activity and bookings, so that I can analyze system usage and revenue | Admins can generate customizable reports on various metrics | Release 2 |
| User Story 17 | For Developers | As a developer, I want to ensure the system is responsive, so that users can book tours on any device | The booking platform functions seamlessly on mobile, tablet, and desktop | Release 1 |
| User Story 18 | For Developers | As a developer, I want to implement secure payment processing, so that user financial information is protected | The payment system complies with industry standards for security (e.g., PCI DSS) | Release 1 |
| User Story 19 | For Developers | As a developer, I want to create an API for third-party integrations, so that the system can connect with other applications | The API provides endpoints for tour listings, bookings, and user management | Release 3 |

**Release Planning / Sprint Planning**

**Release Plan**

Release 1:

1. Tour browsing -

- description about the tour -1

- price -1

- schedule (place, time, activities) -1

2. Online booking

- send booking email confirmation -1

- booking cancellation -1

- online secure payment -1

- booking status: booked, paid, cancelled, modify -1

4. Admin Interface

- manage user accounts

user types: customer, system users (user, admin) -1

activities: customer + user creation, change privilege -1

6. Customer Support

- send email for clarifications -1

Release 2:

1. Tour browsing -

- show computed average rating -2

- tour management -2

3. Tour review

- 1 rating from 1 to 5 -2

- customer comments/recommendations -2

4. Admin Interface

- report generation -2

5. System Monitoring & Reporting

- system performance -2

- monitor error logs -2

- booking report and analytics -2

Release 3:

7. System integration

- API for third-party integrations -3

**Sprint Plan for Release 1**

|  |  |  |
| --- | --- | --- |
| User Story | Estimation | Sprint Number |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Costing/Budget**

Assumptions:

1. Sample budget outline (baseline only) for a Tour Booking Management System, broken down into development, infrastructure, and operational expenses

2. Budget assumes a small to mid-sized tour agency is building a custom system

3. Actual cost will vary based on system complexity, user volume, and additional features, such as AI-driven personalization or advanced reporting

1. Development Costs

a. Planning & Design

* Market Research & Feasibility Study: $1,500 - $3,000
* Requirement Gathering and Documentation: $1,500 - $3,000
* System Architecture Design: $1,500 - $3,000

b. Software Development

* Frontend Development (HTML, CSS, Javascript (React/Angular/Vue)): $5,000 - $10,000
* Backend Development (Node.js/Django): $10,000 - $20,000
* Database Setup (SQLite): $1,000 - $2,000
* System Integration: $5,000 - $10,000
* Payment Gateway Integration: $2,000 - $5,000

c. Testing & Quality Assurance

* Automated & Manual Testing: $3,000 - $6,000
* User Acceptance Testing (UAT): $1,000 - $3,000

2. Infrastructure & Licensing Costs

a. Hosting & Server Costs

* Cloud Hosting (AWS, Azure, or Google Cloud): $100 - $500/month
* Domain Name Registration: $10 - $30/year
* SSL Certificate: $50 - $200/year

b. Software Licensing & Subscriptions

* Booking & CRM Software Integration: $500 - $2,000/year
* Payment Gateway Fees (Stripe, PayPal, etc.): 2.9% + $0.30 per transaction
* Analytics Tools (Google Analytics): $10 - $100/month

3. Operational Costs

a. Content Creation & Marketing

* Graphic Design: $1,000 - $2,500
* SEO & Marketing: $300 - $1,000/month

b. Ongoing Maintenance & Updates

* System Maintenance & Bug Fixes: $1,000 - $2,500/month
* Feature Updates & Enhancements: $2,000 - $4,500/quarter

c. Staff Training & Support

* Training Sessions for Admin & Staff: $500 - $1,500
* Documentation Creation: $500 - $1,500

Total Estimated Cost (1st Year):

Initial Development & Setup Costs: $36,000 - $81,000

Ongoing Monthly Costs: $2,000 - $5,000

Total Yearly Operational Cost: $26,000 - $66,000

**Sign–Off**

**Reflection Report**

#################################

Database Tables

1. Users
2. Bookings
3. Payments
4. Reviews
5. Roles
6. The
7. The

Database creation scripts:

create database TourBooking.db;

create table users (

ID int primary key not null,

FNAME char(50) not null,

LNAME char(50) not null,

USERID char(50) not null,

PASSWORD char(50) not null,

ROLE char(10) not null,

ENTERED\_BY char(50) not null,

ENTRY\_DATE text not null

);

insert into users values (1, 'admin', 'admin', 'admin', 'admin@123', 1, 'admin', strftime('%d/%m/%Y', date()));

create table bookings (

ID int primary key not null,

...

ENTERED\_BY char(50) not null,

ENTRY\_DATE text not null

);

create table payments (

ID int primary key not null,

...

ENTERED\_BY char(50) not null,

ENTRY\_DATE text not null

);

create table reviews (

ID int primary key not null,

...

ENTERED\_BY char(50) not null,

ENTRY\_DATE text not null

);

create table roles (

ID int primary key not null,

ROLE\_NAME char(50) not null,

ROLE\_DESC char(50) not null,

ENTERED\_BY char(50) not null,

ENTRY\_DATE text not null

);

insert into roles values (1,'admin', 'admin', 'admin', strftime('%d/%m/%Y', date()));

insert into roles values (2,'user', 'ordinary user', 'admin', strftime('%d/%m/%Y', date()));

insert into roles values (3,'customer', 'customer', 'admin', strftime('%d/%m/%Y', date()));