Assessment II

MSE800 – Professional Software Engineering

Group Members:

Hengpan He

Wen Liang

Arnold Aristotle Tayag

Date:

31 October 2024

Introduction

As the tourism industry grows rapidly, an online tour booking system has become essential for tourists and operators. To address this, we are developing a Tour Booking Management System. 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. Agile methodology will be used and a duration of one–week sprint for development is adopted to 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. A minimum viable product (MVP) will be available in the first release and 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' business 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

Development 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
- GitHub repository: https://github.com/aamtayag/mse800-assessment2.git

Development Approach:

- Agile methodology (Scrum + XP)
- 1-week sprint
- Relative estimation (using Fibonacci): 1, 2, 3, 5, 8, 13

Stakeholders:

- Product Owner / Business Users (Tourists/Customers/Tour Operators/Tour Owners) / 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

The development team structure and the integrated roles in the team is a key element to fulfill the final target. In our team, almost every critical role in the process of software development is assigned to each of our team members. Here is some detailed information about these roles.

The development team adopted a typical software development team structure, with a clear division of labor to ensure that the project has professional support in all aspects of architectural design, front-end and back-end development, testing and management. Core team members include project managers, architects, front-end development engineers and back-end development engineers, with each role taking on key responsibilities at different stages of the project to ensure steady progress.

Architect

Team Member: Arnold Aristotle Tayag

Responsible for the overall design of the system architecture, providing recommendations on the technical framework, structure and scalability of the project. The architect will take into account the requirements of system performance, scalability, and disaster recovery capability in the design to ensure that the technical implementation meets the business requirements.

Responsibility：

- Develop and maintain the technical architecture and core design of the system
- Guide technology selection and choose the right technology stack and tools for the project
- Ensure the system has good scalability and efficient data processing capabilities

- Collaborate with front and back-end teams to ensure proper implementation of architectural design

Frontend Developer

Team Member: Wen Liang

Responsible for the design and implementation of the user interface to ensure a smooth and intuitive user interaction experience in the system. They present the system functionality to the user in the best possible way according to the designer's visual requirements.

Responsibility：

- Implement user interface design and interaction features to ensure compatibility on various devices
- Collaborate with the back-end team to integrate APIs and ensure seamless interaction of data with the interface
- Optimize front-end performance to improve system loading speed and response time
- Focus on user experience details to optimise interface elements and interaction logic

Backend Developer

Team Member: Hengpan He

Responsible for the implementation of the system's data processing and business logic, creating and maintaining API interfaces, which are the core support for the normal operation of system functions.

Responsibility：

- Implement core business logic and handle complex data operations and processes
- Design and implement system database and API interfaces to ensure smooth data interaction between front and back end
- Configure system security to protect user data and prevent security breaches
- Ensure the stability of the back-end system through unit testing and integration testing.

Project Manager

Team Member: Arnold Aristotle Tayag

Key coordinator and bridge of communication for project time planning, resource management and schedule control to ensure all teams work together efficiently.

Responsibility :

- Manage project schedules to ensure phases are completed on time
- Maintain open communication to ensure unobstructed flow of information between the client, development team and test team
- Identify and manage project risks and develop prevention strategies
- Organize regular project status meetings to support the team and resolve issues

Project Kick-off Meeting

This Project Kickoff Meeting is designed to provide a formal communication platform for the core participants of the project - both the client team and the project development team. The meeting will provide a detailed overview of the project requirements and ensure that the project team has a clear understanding of the client's expectations, laying a solid foundation for the subsequent project planning and development work. The project aims to build a tour e-commerce system that provides users with a convenient online tour product booking experience and helps tour booking companies efficiently manage bookings and product information.

Meeting Objectives

- Define the main functions of the system and the operating privileges of key user roles
- Establish communication process and feedback mechanism to ensure timely problem solving during the development process
- Identify project risks, analyze potential challenges and discuss initial response strategies.

Key Participants:

- Client representative: Arnold Aristotle Tayag
- Development team Representative: Wen Lian
- Project Manager: Hengpan He

With a thorough conversation and discussion about the project for several times, some valuable outcomes are concluded finally. Here are the achievements that we accomplished:

In the aspects of demand:

- Confirm the process of publishing basic information (e.g., name, description, price, etc.) about a tour product and the authority to modify it
- Discuss the key steps in the booking process, including the process design for booking confirmation and cancellation
- Clarify the notification mechanism for users and administrators to ensure transparency and real-time system operation.

Feedback mechanics:

- Discuss how to achieve efficient communication and ensure two-way feedback between the client team and the development team. Specifically include:
  - Weekly project update meetings to report on the week's progress and the work plan for the following week
  - An assessment of the time point to the next milestone
  - Task tracking and issue management using the project management tool Jira

Project risks identification:

Identify technical or business challenges that may be encountered in the project, conduct risk analyses and initially discuss response strategies. Key considerations include:

- User booking cancellation timeliness issues and system responsiveness during peak periods
- System stability, including Service Level Agreement (SLA) related elements
- Horizontal expansion capability of the system and disaster recovery requirements
- Objectives and requirements of the system's core data storage and backup strategy

Summary

Through this project kick-off meeting, the project team should have a comprehensive understanding of the client's business requirements and project objectives, and initially define the system's functional structure and development schedule and establish a smooth communication mechanism. The results of the meeting will provide an important foundation for the requirements analysis and architecture design phase.

Scope of Work

The system focuses on providing tour booking services exclusively for tours within New Zealand. It will cover the following key features:

1. **Tour Browsing**: Displays detailed information such as tour descriptions, prices, itineraries, and ratings.
2. **Online Booking**: Supports email confirmation, payment processing, and tracking of booking statuses.
3. **Tour Review**: Allows users to provide feedback and ratings after the tour.
4. **User Management**: Enables administrators to manage user accounts and permissions.
5. **System Integration**: Offers APIs for integrating with third-party payment services and tour information providers.

Out of Scope

The following items are beyond the scope of this project:

1. Offline travel arrangements or services outside of New Zealand.
2. **Travel insurance management**: The system will not handle or process insurance requests or claims.
3. **Multi-currency support**: The platform will only process payments in New Zealand dollars (NZD) without exchange rate calculations.

System Design

System Architecture

1. **Frontend Technologies:** The frontend will be developed using HTML, CSS, and JavaScript, ensuring a responsive and user-friendly interface for tourists and administrators.
2. **Backend Technologies:** The backend logic will be handled using a combination of Python and Node.js to efficiently manage business logic, API calls, and data processing.
3. **Database:** The system will use SQLite as the primary database to store data such as tour details, user information, bookings, and feedback.

Module Breakdown

1. **Tour Browsing Module:** Loads available tours from the database and displays relevant information to the users based on their preferences.
2. **Online Booking Module:** Processes bookings and payments, updates booking status, and handles email confirmations.
3. **Admin Interface:** Provides system administrators with the ability to manage users, process booking requests, and generate reports.
4. **Tour Review Module:** Collects and displays feedback from users, including ratings and comments.
5. **System Integration Module:** Integrates with third-party services through APIs, such as payment gateways.

System Interfaces

1. **Tour Listing Page:** Displays all available tours and allows users to filter based on preferences like destination, date, and tour type.
2. **User/Admin Login Page:** Provides separate login interfaces for tourists and administrators.
3. **Booking Page:** Facilitates the booking process by collecting traveler details, confirming availability, and processing payments.

Data Flow and Communication

1. **Frontend and Backend Communication:** The frontend will communicate with the backend using REST APIs to ensure smooth data exchange. Node.js will handle API requests, while Python will manage the business logic and data processing.
2. **Database Operations:** SQLite will store all necessary data, including user accounts, tours, bookings, and reviews. The backend will manage data queries and ensure synchronization between modules.

3. **Scalability Considerations:** Although SQLite is a lightweight database suitable for the initial development phase, the system architecture allows for future migration to more robust databases (e.g., MySQL or PostgreSQL) if needed.

Product Backlog Prioritization & Refinement

After many meetups and discussions with various stakeholders, below features were identified as critical to meet the business requirements:

Breakdown:

- 19 user stories, 113 story points
- Release 1: 10 user stories, 50 story points, 4 sprints
- Release 2: 7 user stories, 42 story points, 4 sprints
- Release 3: 2 user stories, 21 story points, 2 sprints

User Story #	Functionality	User Story	Acceptance Criteria	Prioritization
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
2	For Tourists, Customers	As a tourist, I want to view detailed information about each	Each tour page includes a description, itinerary,	Release 1

		tour, so that I can make informed decisions	duration, price, and reviews	
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
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
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
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
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

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 2
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
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
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
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

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
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
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
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
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

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
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
   - Price
   - schedule (place, time, activities)

2. Online booking
   - send booking email confirmation
   - booking cancellation
   - online secure payment
   - booking status: booked, paid, cancelled, modify

3. Admin Interface

- manage user accounts

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

  activities: customer + user creation, change privilege

4. Customer Support
   - send email for clarifications

Release 2:

1. Tour browsing
   - show computed average rating
   - tour management

2. Tour review
   - 1 rating from 1 to 5
   - customer comments/recommendations

3. Admin Interface
   - report generation

4. System Monitoring & Reporting
   - system performance
   - monitor error logs
   - booking report and analytics

Release 3:

1. System integration
   - API for third-party integrations

Sprint Plan for Release 1

User Story #	Estimation (Fibonacci)	Sprint Number
1	2	1

2	2	1
3	5	1
4	5	2
5	5	2
7	5	2
12	4	3
13	5	3
17	13	4
18	4	4

Sprint Plan for Release 2

User Story #	Estimation (Fibonacci)	Sprint Number
6	3	1
8	5	1
9	8	2
10	5	2
11	5	3
14	8	3
16	8	4

Sprint Plan for Release 3

User Story #	Estimation (Fibonacci)	Sprint Number
15	8	1
19	13	2

Database Schema:

Tables:

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

DB Scripts:

```
create database TourBooking.db;


CREATE TABLE users (
ID integer primary key autoincrement,
        USERNAME char(50) not null,
        EMAIL char(50) not null,
        PASSWORD char(50) not null,
        ROLE char(10) not null
);


insert into users values ('admin', 'admin@mail.com', 'admin123', 1);


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()));

CREATE TABLE booking_status(
        ID int primary key not null,
        STATUS_NAME char(50) not null,
        STATUS_DESC char(50) not null
);

CREATE TABLE bookings (
        ID integer primary key autoincrement,
        TOUR char(50) not null,
        NAME char(50) not null,
        EMAIL char(50) not null,
        DATE char(50) not null,
        STATUS int not null
);


create table payments (
        ID int primary key not null,
        BOOKING_ID integer not null,
        AMOUNT real not null,
        ENTERED_BY char(50) not null,
        ENTRY_DATE text not null
);

create table reviews (
        ID int primary key not null,
        BOOKING_ID integer not null,
        RATING integer not null,
        DESCRPTION char(50) not null,
        ENTERED_BY char(50) not null,
        ENTRY_DATE text not null
);
```

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 allocated is based on a mid–sized tour agency

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

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:

Initial Development & Setup Costs: $35,000 - $78,000

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

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

Project Sign–Off

Project Title **:** Tour Booking Management System
Project ID **: Alpha-1xB.1257**
Date **:** 31 October 2024
Project Manager **:** Wen Liang
Client/Stakeholder : Arnold Aristotle Tayag

Project Overview

This project, i.e. Tour Booking Management System, is intended to streamline and automate the processes involved in booking tours, its management, tracking, and reporting. The system envisions to enhance user experience while also improving the efficiency in managing bookings and providing real-time management and administration for system users.

Deliverables Checklist

The following project deliverables have been completed, reviewed, and approved:

Deliverable	Status	Comments
Requirement Specification	Completed	_________
System Design Document	Completed	_________
Database Design	Completed	_________
Application Development	Completed	_________
System Testing	Completed	_________
User Acceptance Testing (UAT)	Completed	_________
Training & Documentation	Completed	_________
Go-Live/Deployment	Completed	_________

Acceptance Criteria

All project requirements, deliverables, and acceptance criteria have been reviewed and meet the standards agreed upon in the original project plan, including:

- Functional and non-functional requirements
- Performance and scalability requirements
- Security and compliance requirements
- Usability and user interface requirements

Sign-Off

By affixing your signature below, both parties agree that the project meets all user specifications and requirements and is considered officially complete. Any addendum, modifications, enhancements, or new functionalities added will be treated as new requirements and will be the subject of a new project proposal.

Role	Name	Signature	Date
Project Manager	Wen Liang	_________	31 October 2024
Client/Stakeholder	Arnold Aristotle Tayag	_________	31 October 2024
Technical Lead	Hengpan Han	_________	31 October 2024