

Sri Lanka Institute of Information Technology



Information Technology Project (IT2080)

Batch ID - Y2.S2.WD.IT.04

Group ID - 92

Pearlora – Travel Agency Management System

Activity 02

	Registration Number	Name with Initials	Email	Contact Number
1.	IT23273580	Prasanth A	it23273580@my.sliit.lk	0779080621
2.	IT23210424	Thivyan J	it23210424@my.sliit.lk	0711050671
3.	IT23210974	Kesigan M	it23210974@my.sliit.lk	0740422764
4.	IT23321236	Nafy F A	it23321236@my.sliit.lk	0722203811
5.	IT23158436	Priyatharshini A H	it23158436@my.sliit.lk	0702455393

Contents

1.	Identify the stakeholders in your system and analyze them using an onion diagram.....	5
	Stakeholder	5
	Onion diagram	6
2.	List the Functional Requirements for the direct system users (in the innermost layer of the Onion diagram or main stakeholders).	7
	Customer	7
	Destination Manager	8
	Hotel Reservation Manager	8
	Transportation Manager	9
	Finance Manager	9
	Event Management	10
3.	List the related NFRs and analyze them user wise.....	10
	Customer	10
	Destination Manager	10
	Hotel manager	11
	Transportation Manager	11
	Finance Manager	11
	Event Manager	11
4.	State the Technical requirements for the system.....	12
5.	Model the requirements using a use case diagram	13
	Whole System	13
	Destination Management Use Case Diagram	14
	Hotel Management Use Case Diagram	15
	Transportation Management Use Case Diagram	16
	Financial Management Use Case Diagram	17
	Event Management Use Case Diagram	18
6.	use case descriptions for 5 main use cases in the diagram	19
	Financial Management Use Case Scenario	19
	Event Management Use Case Scenario	21
	Destination Management Use Case Scenario	22
	Hotel Management Use Case Scenario	23
	Transportation Management Use Case Scenario	24
7.	Develop suitable diagrams to show visual presentation of data flow, the process Flow and Data Connections to support the above (eg: system diagram, Flow chart, DFD)	25
	Destination Management Flow chart	25
	Hotel Management Flow chart	26
	Transportation Management Flow chart	27

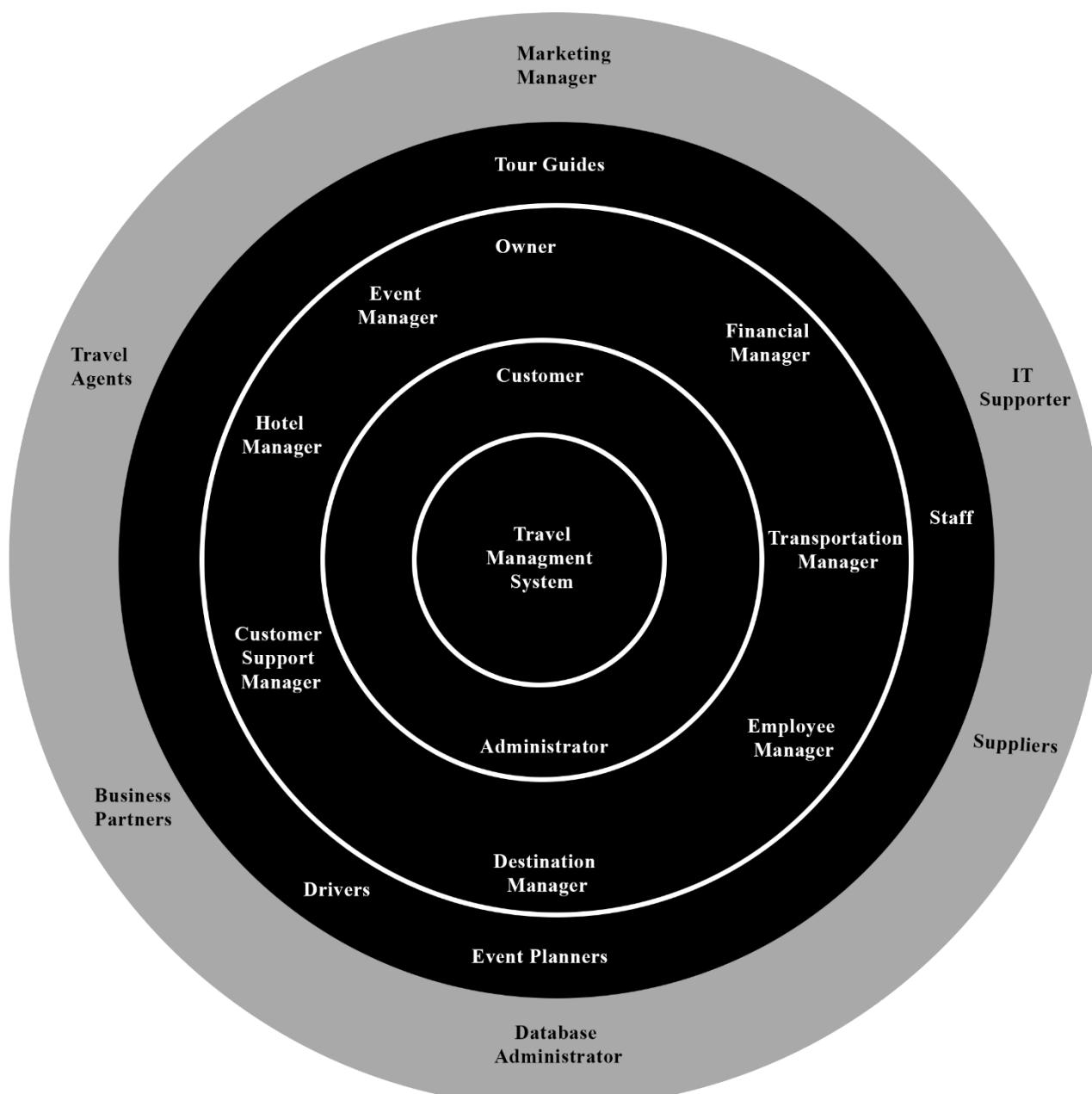
Financial Management Flow chart	28
Event Management Flow chart	29
Data Flow Diagram	30
System Diagram	30
8. A suitable plan to develop the project as a team	32
Project Plan	32

1. Identify the stakeholders in your system and analyze them using an onion diagram.

Stakeholder

- Customer (Traveler)
- Administrator
- Owner
- Destination reservation manager
- Hotel reservation manager
- Transportation reservation manager
- Financial manager
- Event reservation manager
- Employee manager
- Customer support manager
- Staff
- Tour Guide
- Event planners
- Drivers
- Marketing manager
- Travel Agents
- Suppliers
- Business partners
- IT supporter
- Database Administrator

Onion diagram



2. List the Functional Requirements for the direct system users (in the innermost layer of the Onion diagram or main stakeholders).

Functional Requirements

Customer

User Registration & Authentication

- Register, log in, and log out securely.
- Email and OTP verification.

Booking & Reservations

- Search and book hotels, transport, and events.
- View real-time availability.
- Modify or cancel bookings.
- Payments & Transactions

Secure payments via PayPal, Stripe, or local gateways.

- Instant payment confirmation via email/SMS.
- View payment history and invoices.
- Reviews & Feedback

Rate and review hotels, destinations, and transport services.

- Report issues or request help.
- Profile & Preferences Management

Modify personal information and travel preferences.

- Save favorite hotels, destinations, and transport services.
- Notifications & Alerts

Receive booking confirmations and payment status.

- Receive travel notifications (delays, transport availability).

Destination Manager

- Allow Destination Managers to add, update, delete, and view destination details, including attractions, activities, and facilities.
- Allow Destination Managers to allocate resources such as guides, transport, and permits for specific destinations.
- Allow Destination Managers to manage and approve destination visit requests from clients or event organizers.
- Allow Destination Managers to track visitor statistics, entry fees, and overall destination performance.
- Allow Destination Managers to generate reports on destination popularity, revenue, and visitor feedback.
- Allow Destination Managers to integrate with third-party tourism services for enhanced customer experiences.
- Allow Destination Managers to manage emergency response plans for safety incidents at destinations.
- Allow Destination Managers to monitor and ensure compliance with local regulations and environmental guidelines.
- Allow Destination Managers to update and promote seasonal or special event packages for destination.

Hotel Reservation Manager

- Allow Hotel Reservation Managers to add, update, delete, and view hotel room availability and pricing.
- Allow Hotel Reservation Managers to allocate rooms to clients based on booking requests.
- Allow Hotel Reservation Managers to manage check-in and check-out processes for guests.
- Allow Hotel Reservation Managers to handle special accommodation requests, such as accessibility needs or VIP services.
- Allow Hotel Reservation Managers to generate reports on room occupancy, revenue, and booking trends.
- Allow Hotel Reservation Managers to process and approve or reject room booking requests from clients or event organizers.
- Allow Hotel Reservation Managers to integrate with third-party hotel booking platforms for wider accessibility.
- Allow Hotel Reservation Managers to manage cancellation and refund policies for hotel bookings.
- Allow Hotel Reservation Managers to track and manage hotel staff assignments and housekeeping schedules.
- Allow Hotel Reservation Managers to collect and review guest feedback to improve hotel services.

Transportation Manager

- Allow Transportation Managers to add, update, delete, and view transportation details, including vehicle availability, schedules, and assigned drivers.
- Allow Transportation Managers to allocate vehicles and drivers to specific events based on requirements.
- Allow Transportation Managers to track real-time vehicle locations using IoT-based GPS tracking.
- Allow Transportation Managers to manage fuel consumption, maintenance schedules, and vehicle conditions.
- Allow Transportation Managers to generate reports on transportation costs, vehicle usage, and efficiency.
- Allow Transportation Managers to process and approve transportation requests from Event Organizers or clients.
- Allow Transportation Managers to integrate with third-party ride-hailing or rental services for additional transport options.
- Allow Transportation Managers to manage emergency response plans for vehicle breakdowns or delays.
- Allow Transportation Managers to monitor driver performance and ensure compliance with safety regulations.
- Allow Transportation Managers to cancel or reschedule transport services based on event change

Finance Manager

- Allow Finance Managers to record, update, delete, and view financial transactions related to event bookings, vendor payments, and revenue.
- Allow Finance Managers to allocate and track budgets for different events.
- Allow Finance Managers to generate reports on income, expenses, profits, and losses for each event or overall business operations.
- Allow Finance Managers to create, send, and manage invoices for event bookings and vendor services.
- Allow Finance Managers to process refunds for canceled or rescheduled events based on policies.
- Allow Finance Managers to track revenue sources, including ticket sales, sponsorships, and client payments.
- Allow Finance Managers to integrate payment gateways (e.g., Stripe, PayPal, local providers) for secure online transactions.
- Allow Finance Managers to monitor financial compliance, including tax deductions, invoices, and audits.
- Allow Finance Managers to approve or reject financial requests from Event Organizers or other stakeholders.
- Allow Finance Managers to manage payroll for event staff, including salary payments and reimbursements.

Event Management

- Allow Event Organizers to add, update, delete, and view event details.
- Allow Event Organizers to allocate resources (e.g., venues, staff, equipment) to events.
- Allow clients or team members to submit event proposals and resource requests. Allow Event Organizers to search for events by IDs, names, dates, or other criteria. Allow Event Organizers to generate reports on event details, budgets, and resource utilization.
- Allow clients to submit feedback at a post-event.
- Allow Event Organizers to cancel or reschedule events

3. List the related NFRs and analyze them user wise.

Non-Functional Requirements

Customer

- Performance: Pages must be loaded and booking processed in less than 2 seconds
- Usability: UI must be interactive and responsive for smooth navigation
- Availability: IT must be online 24x7 to support the customer to make a booking any time
- Security: Customer Credit card and contact details must be encrypted
- Scalability: The application must support larger traffic during heavy loads
- Compatibility: The program must be implemented on different platforms
- Reliability: The booking should provide 99.9% uptime to prevent loss of sales

Destination Manager

- Data Integrity: Destination data must be accurate and consistent across the system
- Maintainability: The system should allow easy changes of travel destination
- Performance: The system should be able to retrieve destinations data quickly
- Security: Destination data can be changed by the authorized users only
- Scalability: The platform should be capable of handling unlimited destinations

Hotel manager

- Performance: Hotel availability checks must be processed within less than 2 seconds
- Security: Hotel managers alone must be allowed to modify hotel information and room availability
- Availability: The booking system must be accessible 24/7 to prevent revenue loss
- Usability: The hotel management interface must be easy to use
- Data consistency: Hotel room information must be synchronized with the central database

Transportation Manager

- Reliability: The system must provide accurate vehicle locations updates
- Security: Access control must restrict unauthorized staff from updating transportation data
- Performance: The system must be updated in real-time with minimal latency
- Scalability: The system must be able to scale to accommodate additional vehicles over a given period of time

Finance Manager

- Security: Payment transaction must be encrypted
- Reliability: Transaction must be updated accurately to prevent loss of finance
- Performance: Payment processing should be completed within under 3 seconds
- Scalability: The system should support increasing transaction loads
- Compliance: The system should focus on tax regulations and financial reporting standards
- Documentation: The system will provide information about how to use it and all the indispensable information about the system.

Event Manager

- Availability: The event management module should be accessible 24/7.
- Performance: The system should handle high traffic during peak event registrations.
- Security: Only authorized users should have access to manage event details.
- Usability: The event management interface should be simple and efficient.
- Scalability: The platform should support multiple event listings with increasing user participation.

4. State the Technical requirements for the system

Technical Requirements

The Perlora travel agency system is built using the MERN stack (MongoDB, Express.js, React, Node.js) and integrates IoT for real-time vehicle tracking and financial transactions.

Front-end frameworks:

- **React App** It is used to build UI components that create the user interface of the single page web application.

Back-end languages:

- **Express.js**: The JavaScript runtime environment. It is used to run JavaScript on a machine rather than in a browser.
- **NodeJS**: framework layered on top of NodeJS, used to build the backend of a site using NodeJS functions and structures.

Database tool:

- **MongoDB** is A document-oriented, No-SQL database used to store the application data.

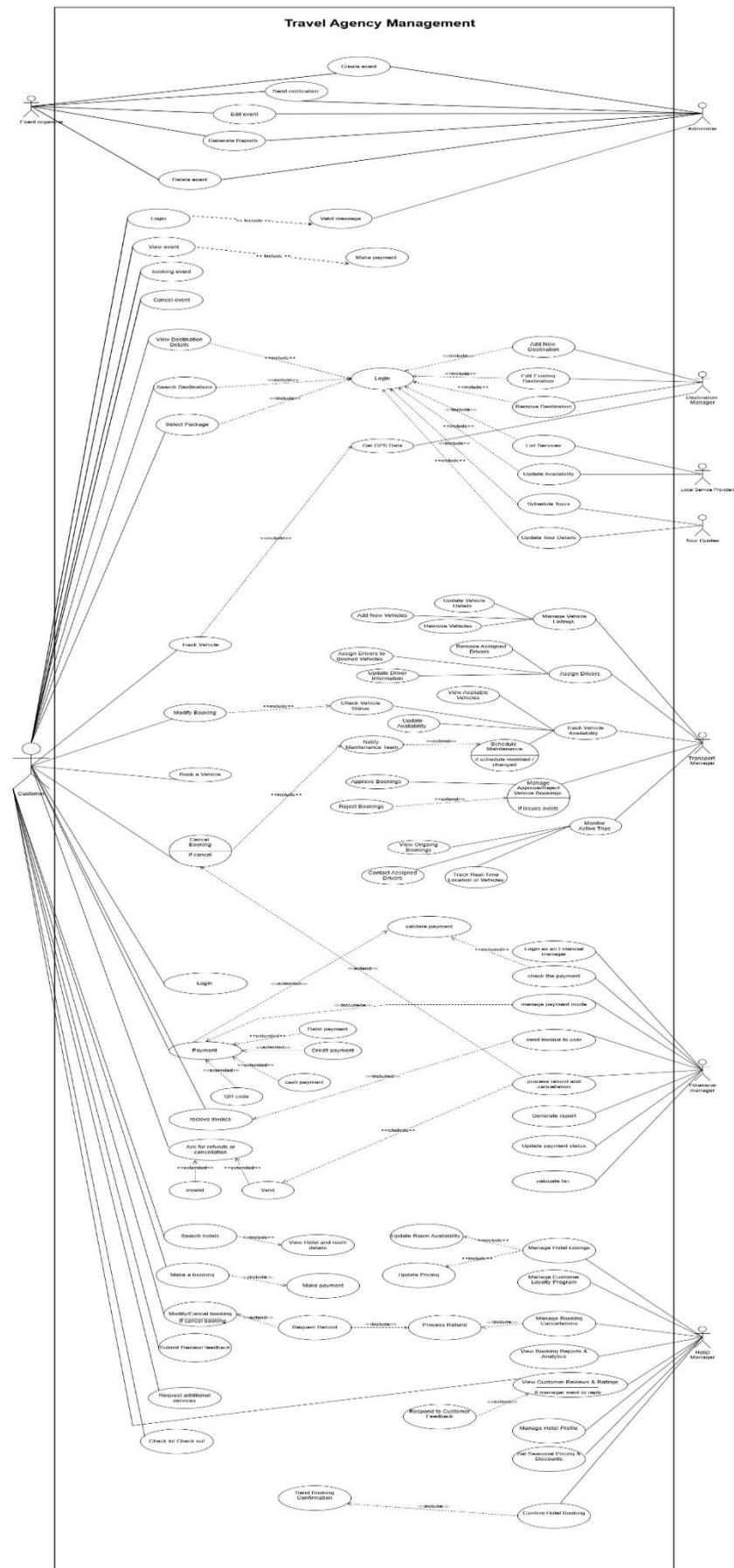
Backend testing tools: **Postman API**

Project management tool: **GitHub**

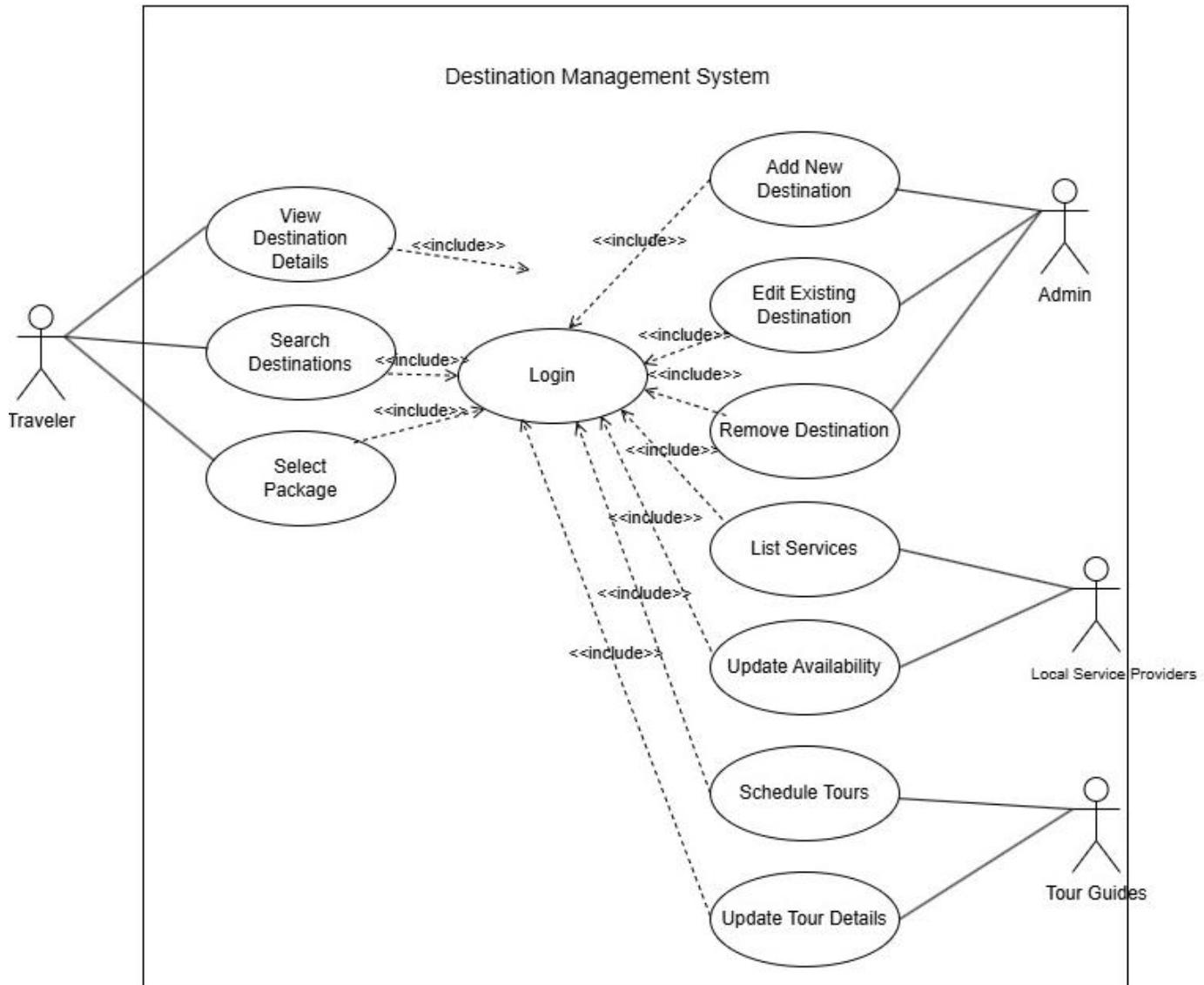
IDE: **Visual studio code**

5. Model the requirements using a use case diagram

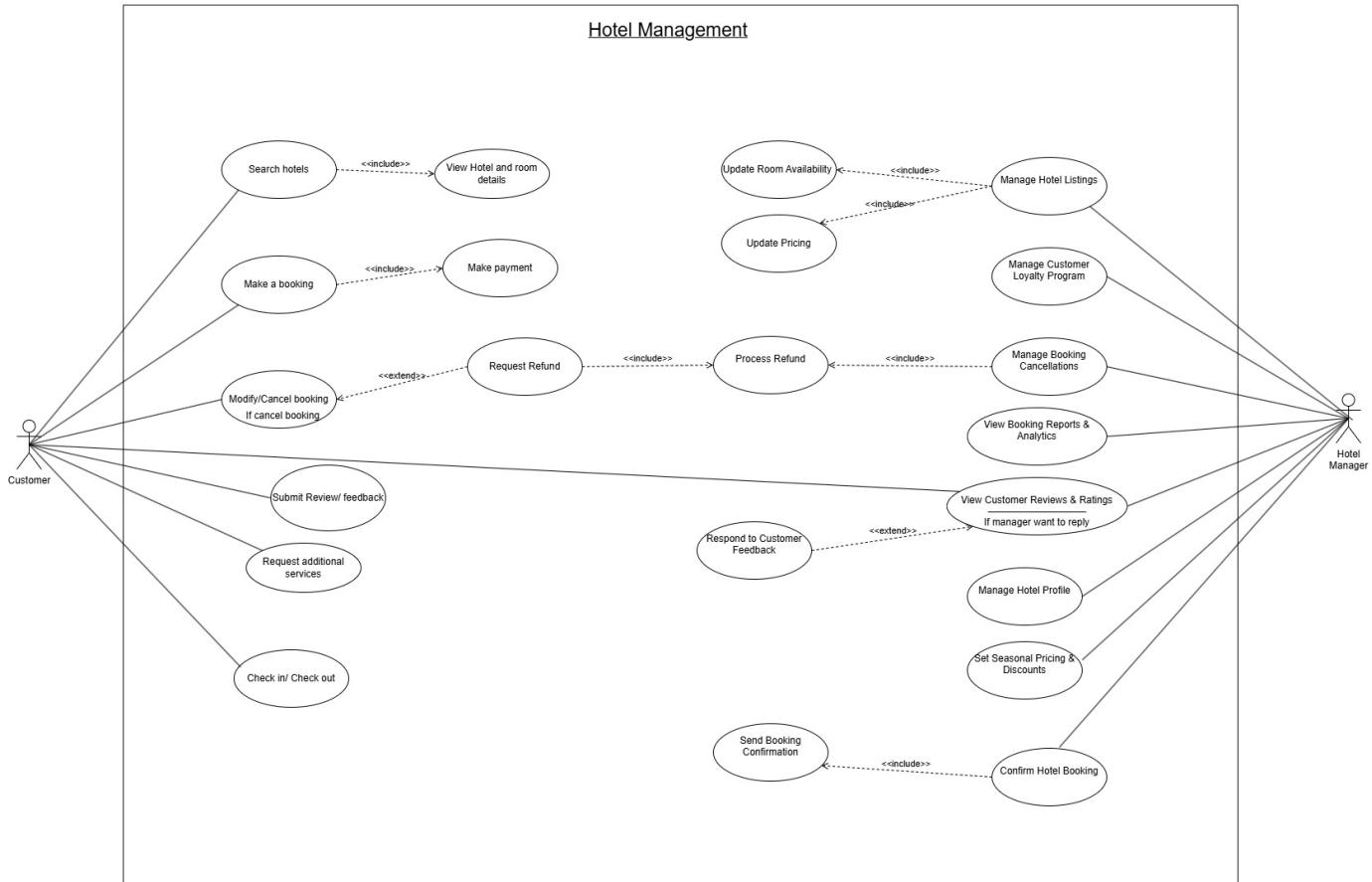
Whole System



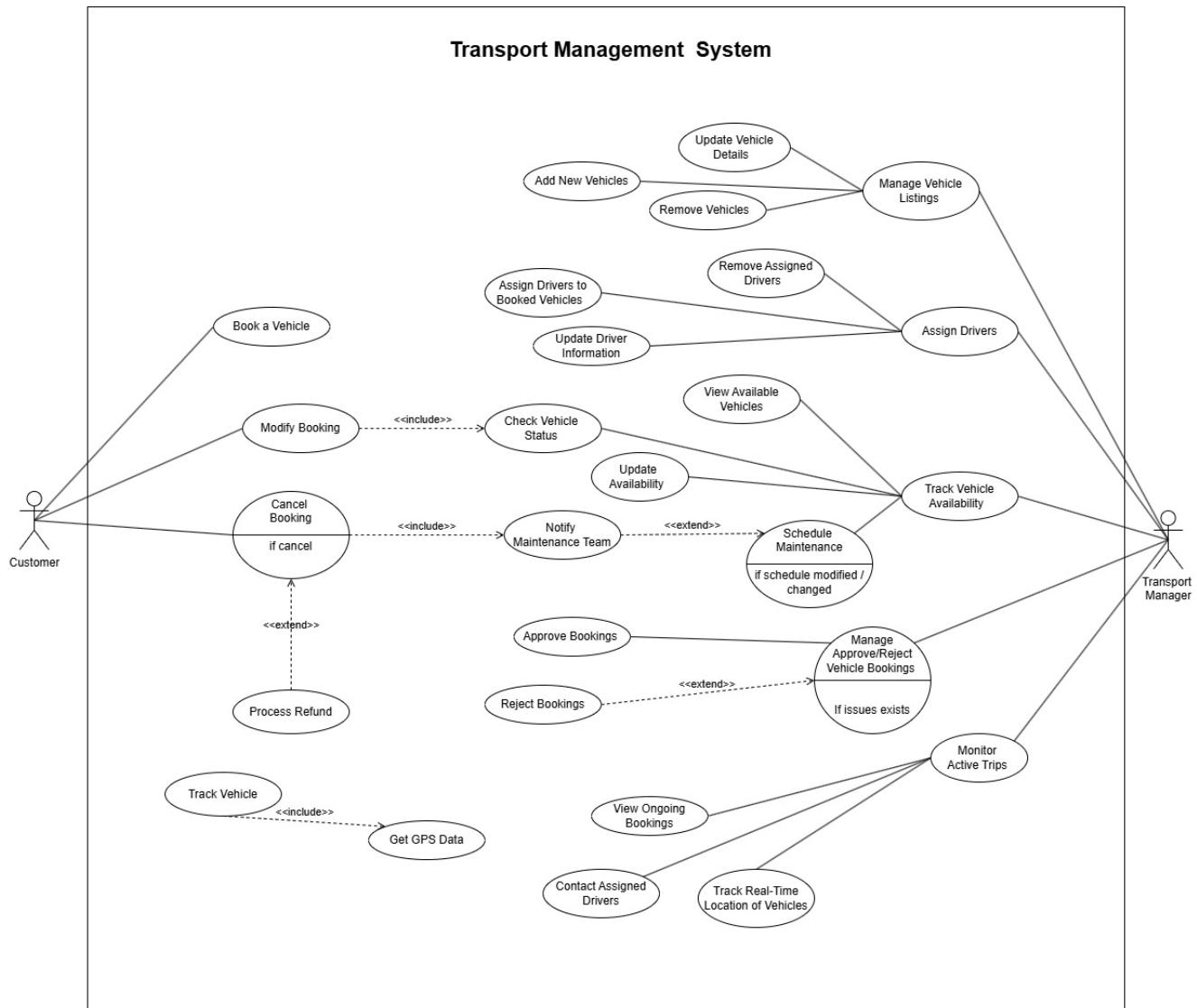
Destination Management Use Case Diagram



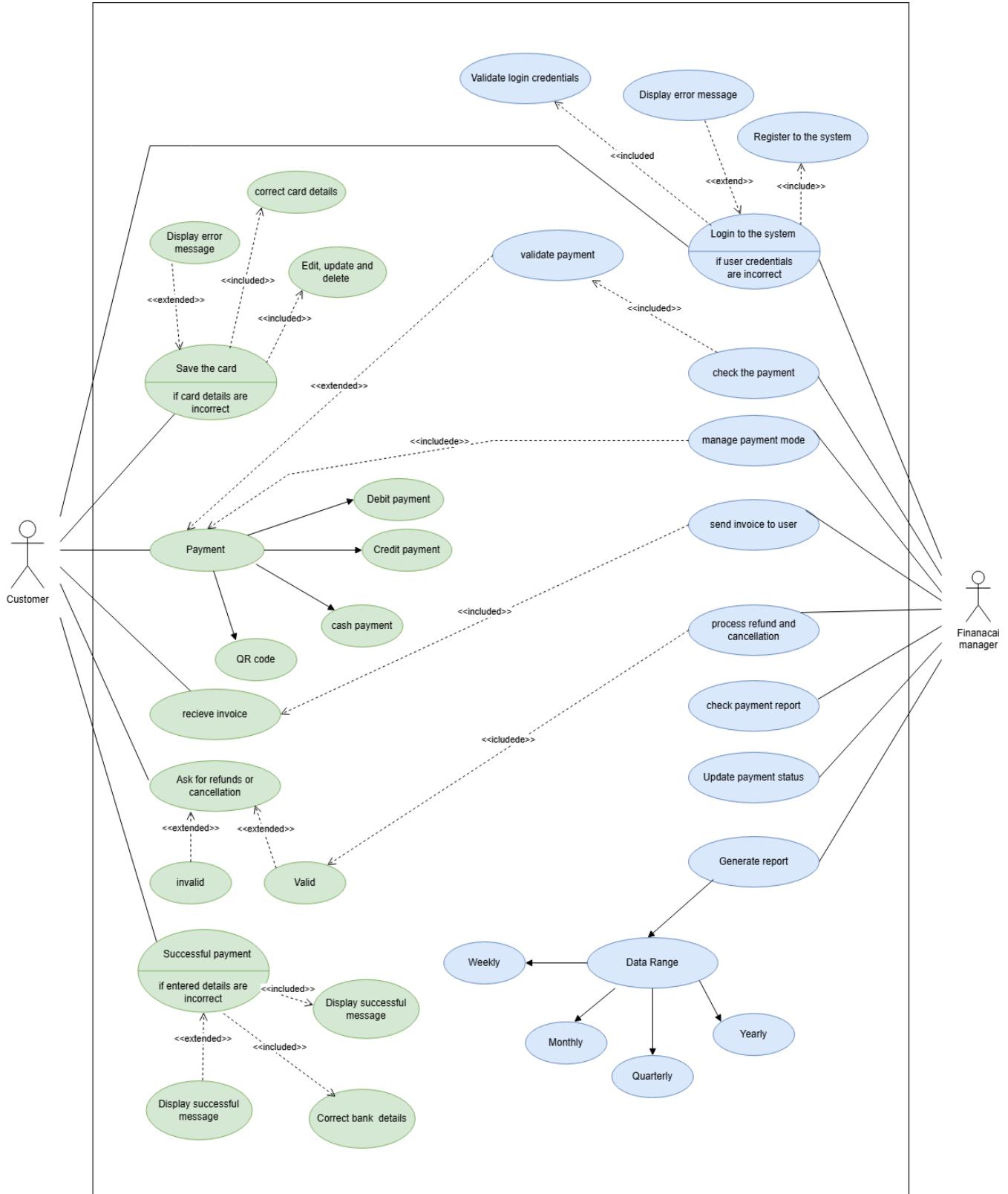
Hotel Management Use Case Diagram



Transportation Management Use Case Diagram



Financial Management Use Case Diagram



Event Management Use Case Diagram



6. use case descriptions for 5 main use cases in the diagram

Financial Management Use Case Scenario

Number	IT23321236	
Use Case ID	UC004	
Use Case Name	Generates Financial Report	
Summary	Handles user's package payments transactions and all the financial reports in the system	
Priority	01	
Pre-condition	Customers and the financial manager must register to the system	
Post-condition	Customers can get their payment report and manager can check all payment reports.	
Primary actor	Customer and Admin (Financial Manager)	
Trigger	<p>If a customer is registered with the system, they must keep their payments details.</p> <p>If a financial report is generated in the system, it should be stored in the payment page.</p>	
Main scenario	Step	Action
	01	The user selects a tour package or service.
	02	The user proceeds to the checkout page
	03	The system will display available payment options (credit/ debit card, cash payment or QR payment)
	04	The user selects a preferred payment method
	05	The system redirects the user to the perspective payment gateway
	06	If the user selects the cash payments, they should go to the travel agency and pay with cash
	07	If the user selects the card payment, they can go to the travel agency and pay with a card or they can use an online card payment method.
	08	When users save their card in the system, they can edit, update and delete
	09	If the user selects OR payment option he /she can pay with the QR code
	10	However, after the user selects payment type and method, they want to click on the 'submit' button.
	11	The system verifies the payment with the provider.
	12	If the payment is successful, the system generates a payment receipt.
	13	The system updates the booking status as "Confirmed."
	14	The system sends a confirmation email/SMS with the booking details.
	15	The system logs the transaction for financial records

	16	The second primary actor financial manger handles the whole payment page.
	17	Financial Manger logs into the system using Admin credentials.
	18	The system verifies credentials and grant access.
	19	The financial manager navigates to the “Financial management” section of the system.
	20	Financial manager Selects “Generate Financial Report”
	21	The system prompts the Admin to enter the report parameters: <ul style="list-style-type: none"> • Data Range (weekly, monthly, quarterly, yearly or custom) • Type of Report Revenue report, Expense Report, Profit and loss statement, Transaction summary)
	22	The admin confirms the sections and submits a request.
	23	The system fetches the financial data from the database based on the selected parameters.
	24	The system processes and generates the financial report in the selected format (PDF, Excel or Web view)
	25	The Admin reviews the generated report and can update, print, or delete the reports but they never can edit reports.
	26	The Admin can Download or print the report if needed
Extensions	Step	Branching Action
	4a	If the user cancels the payment process : the system cancels the transaction and redirects the user back to the checkout page.
	11a	If the payments fail due to incorrect details or insufficient funds, the system will display an error message
	17a	If login fails, the system displays an error message “Incorrect password Try again “
	21a	No financial data available <ul style="list-style-type: none"> • If no financial records exist for the selected period, the system displays a message “No data available for the selected range” • Admin can choose a different range or exit
	25a	If the system encounters an error, an error message is displayed: “Unable to generate the report. Please try again later”

Event Management Use Case Scenario

Number	IT23158436	
Use Case ID	UC005	
Use Case Name	Event managing	
Summary	Event organizers manage event details, update information, and ensure all event arrangements are executed as planned.	
Priority	5	
Pre-condition	The event organizer must be logged into the system and An event must already be booked.	
Post-condition	Event details are updated and notifications are sent to relevant parties.	
Primary actor	Event organizer	
Trigger	Event organizer initiates the process to manage event details using the website.	
Main scenario	Step	Action
	01	Event organizer logs into the system using Admin credentials.
	02	Event organizer selects the booked event to view or update details.
	03	Event organizer updates event details such as guest list, menu preferences, or additional services
	04	System saves the updated details and sends notifications to relevant parties
	05	Event organizer can communicate with clients and vendors through the system's messaging feature
	06	system logs all changes and communications for future reference
Extensions	Step	Branching Action
	1a	If enters invalid login credentials, the system displays an error message and returns to the login screen.
	4a	If the event organizer needs to cancel the event, the system allows cancellation and notifies all relevant parties. If the user booking event is confirm notifies conformation message for user
	5a	If there are any disputes or issues, the system administrator can intervene and facilitate resolution.
NRF		Scalability, Performance, Security, Usability, Compatibility
TRF		MERN Stack(Mongo DB, Express js, React js, Node js.) Data Handling

Destination Management Use Case Scenario

Use Case Name	Manage Destination Information	
Actor:	Admin (Travel Agency Staff)	
Goal	To efficiently add, update, or remove destination details, ensuring travelers have access to accurate and up-to-date information.	
Overview:	The Admin manages the repository of travel destinations featured on the website. This includes creating new destination entries, editing existing information, and removing outdated or irrelevant destinations.	
Pre-conditions:	<input type="checkbox"/> Admin is authenticated and has the necessary permissions to manage destination content. <input type="checkbox"/> The content management system (CMS) is operational.	
Post-condition	<input type="checkbox"/> Travelers can view the updated list of destinations with accurate details. <input type="checkbox"/> Changes are reflected immediately on the website	
Basic Path:	Step	Action
	01	Logging to the System
	02	Access Destination Management
	03	Add New Destination
	04	Edit Existing Destination
	05	Remove Destination
Alternative Path	Step	Action
	1a	If login fails, the system displays an error message “Incorrect password Try again
NFR		The interface should be user-friendly, allowing Admins to manage destinations without technical assistance.
		Changes made by the Admin should reflect on the website within 5 seconds
TRS		MERN Stack(Mongo DB, Express js, React js, Node js.) • Data Handling

Hotel Management Use Case Scenario

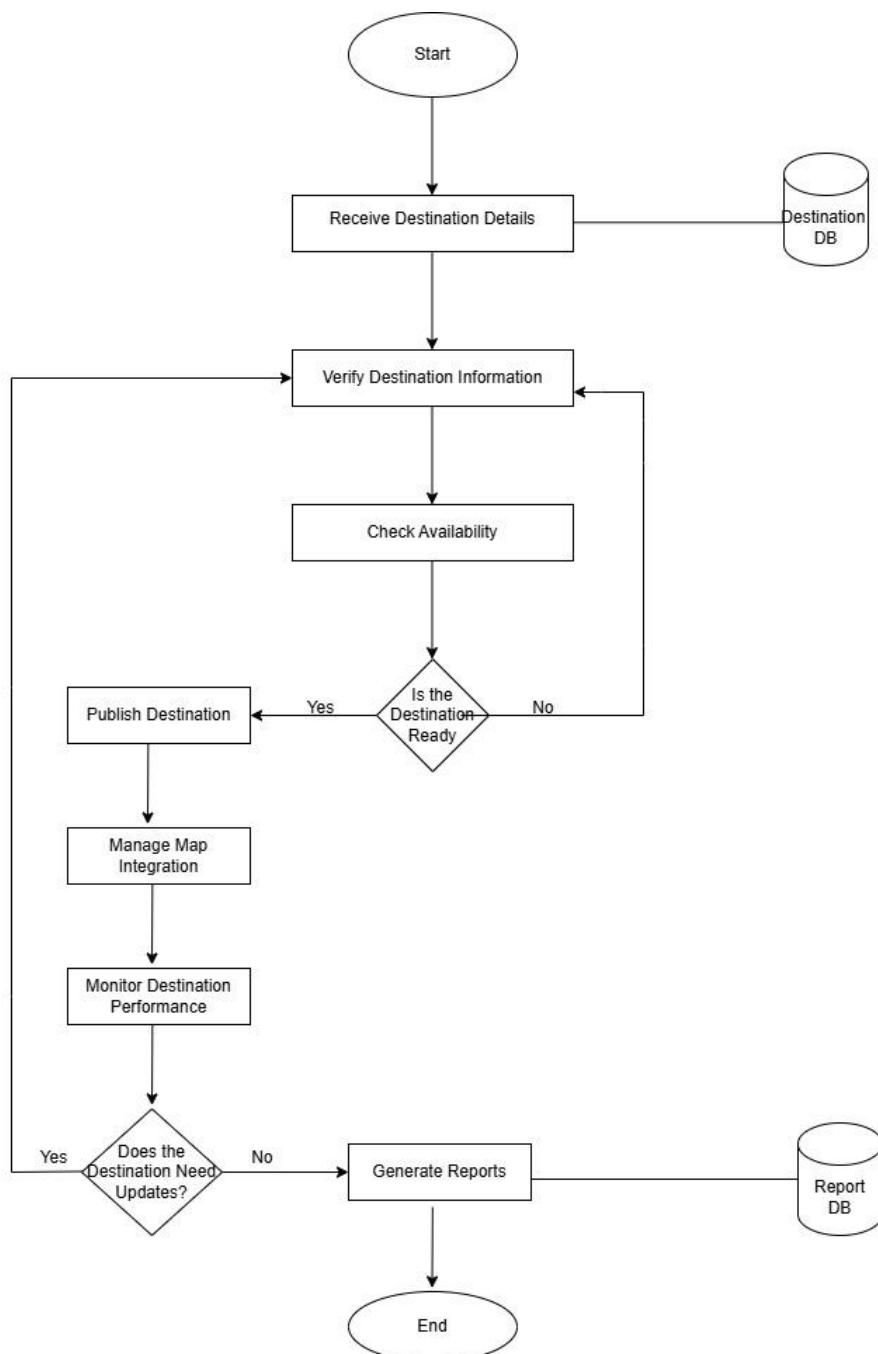
Use case name	Hotel Profile Management	
Primary actor	Hotel manager	
Secondary actor	None	
Goal	To allow the hotel manager to update and manage the hotel's profile, including name, description, images, contact details, and policies.	
Overview	The hotel manager logs into the system and navigates to the hotel profile management section to update hotel details. The system validates and saves the changes, ensuring the profile is updated successfully.	
Pre-Conditions	The hotel manager must be logged into the system as an admin.	
Post-Conditions	The updated hotel profile is successfully saved in the system.	
Basic Path	Step	Action
	1	The hotel manager logs into the hotel management system.
	2	The hotel manager navigates to the "Hotel Profile Management" section.
	3	The hotel manager selects the "Edit Hotel Profile" option.
	4	The system displays the current hotel details (name, description, images, policies, etc.).
	5	The hotel manager updates the required fields, such as hotel name, description, images, contact info, policies, and amenities.
	6	The hotel manager submits the updated details.
	7	The system validates the inputs and checks for errors.
	8	The system updates the hotel profile and displays a success message.
	9	The system logs the changes for record-keeping.
Alternative Path	Step	Action
	1.1	If Login Failed , The system displays a "Login failed" message and prompts the hotel manager to retry.
	7.1	If invalid/missing data is entered , the system displays an error message.
	8.1	If the image upload fails, the system notifies the hotel manager and allows retry.
NFR	<ul style="list-style-type: none"> • Usability • Maintainability • Performance & Scalability 	
TRS	<ul style="list-style-type: none"> • Database: MongoDB • Backend: Express.js , Node.js • Frontend: React.js • Data Handling 	

Transportation Management Use Case Scenario

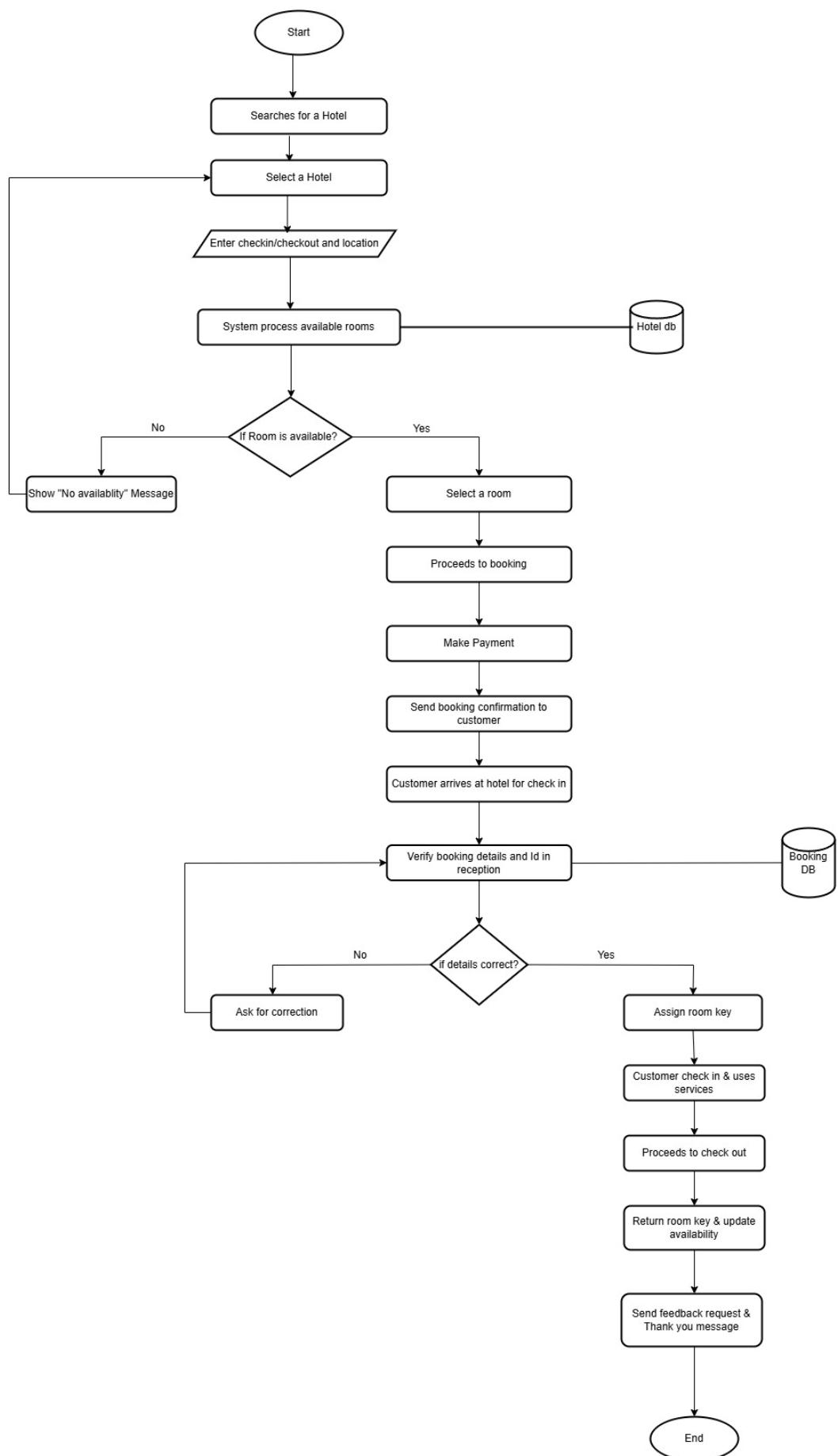
Name	Manage Vehicle Bookings	
Primary Actor(s)	Transportation Manager	
Summary	The Transportation Manager reviews, approves, or rejects vehicle booking requests.	
Priority	High	
Pre-conditions	<ul style="list-style-type: none"> - The user must be logged in as a Transportation Manager. - Vehicles must be available for booking. 	
Post-conditions	<ul style="list-style-type: none"> - If approved, the vehicle is assigned to the customer. - If rejected, the customer is notified with an alternative or cancellation message. 	
Basic Path	Step	Action
	1	The Transportation Manager logs into the system.
	2	Navigates to the "Manage Vehicle Bookings" section.
	3	Views the list of pending vehicle bookings.
	4	Selects a booking request to review details.
	5	If a vehicle is available, the manager approves the booking.
	6	The system assigns the vehicle and notifies the customer.
	7	The process is completed successfully.
Alternative path	Step	Branching actions
	1	Vehicle Unavailable: The manager rejects the request, and the system notifies the customer with alternative options.
	2	Customer Cancels: The booking request is removed from the system.
	3	System Error: The manager gets an error message and retries the operation.
NFRs	<ol style="list-style-type: none"> 1. Performance 2. Usability 3. Performance 4. Usability 5. Scalability 6. Reliability 7. Speed 	
TRs	<ol style="list-style-type: none"> 1. Web-based application 2. Back-end – NodeJs, ExpressJs 3. Front-End – React, Tailwind Css 4. Database – Mongoose 5. Tools – Postman, VS Code, MongooseDbCompass 	

7. Develop suitable diagrams to show visual presentation of data flow, the process Flow and Data Connections to support the above (eg: system diagram, Flow chart, DFD)

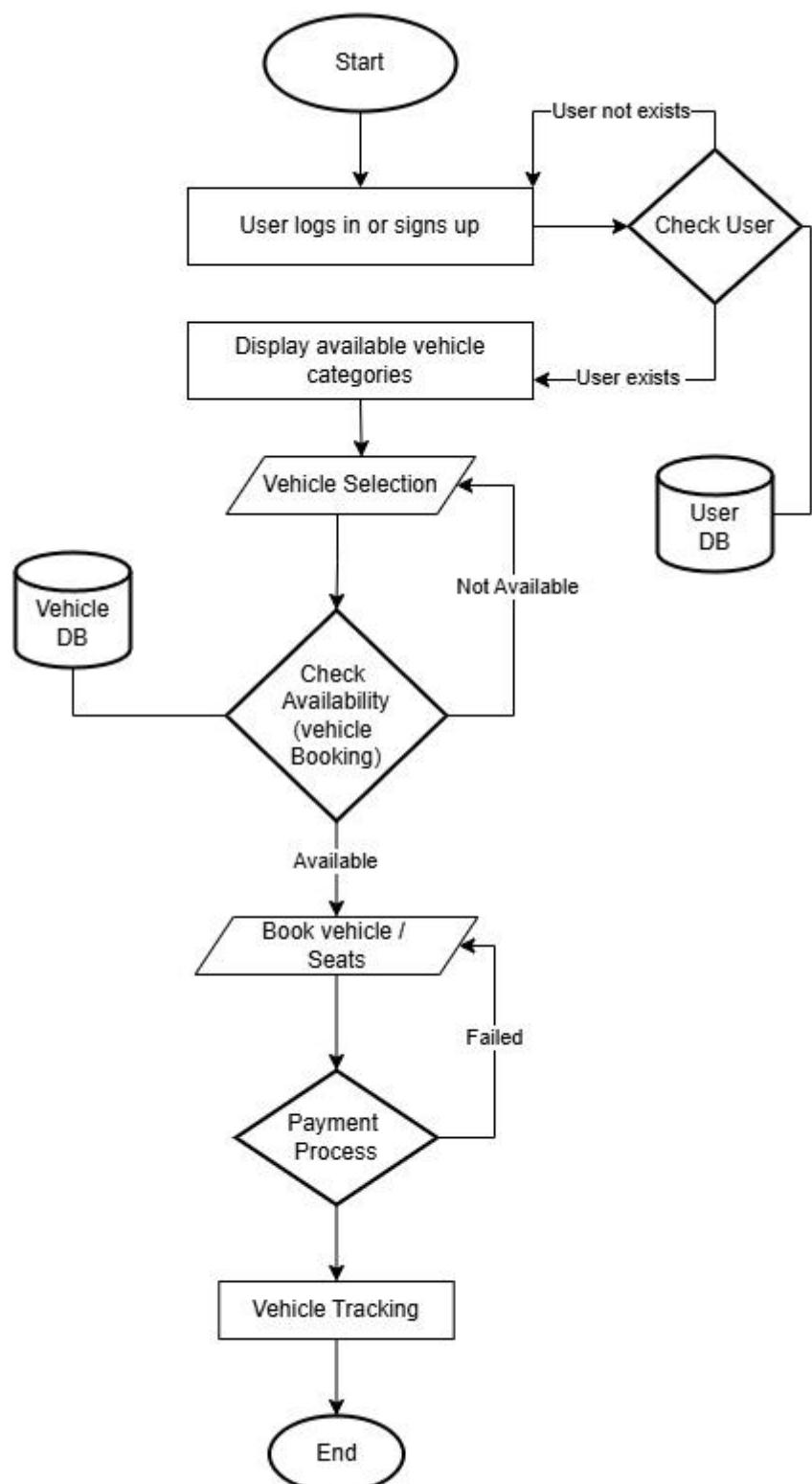
Destination Management Flow chart



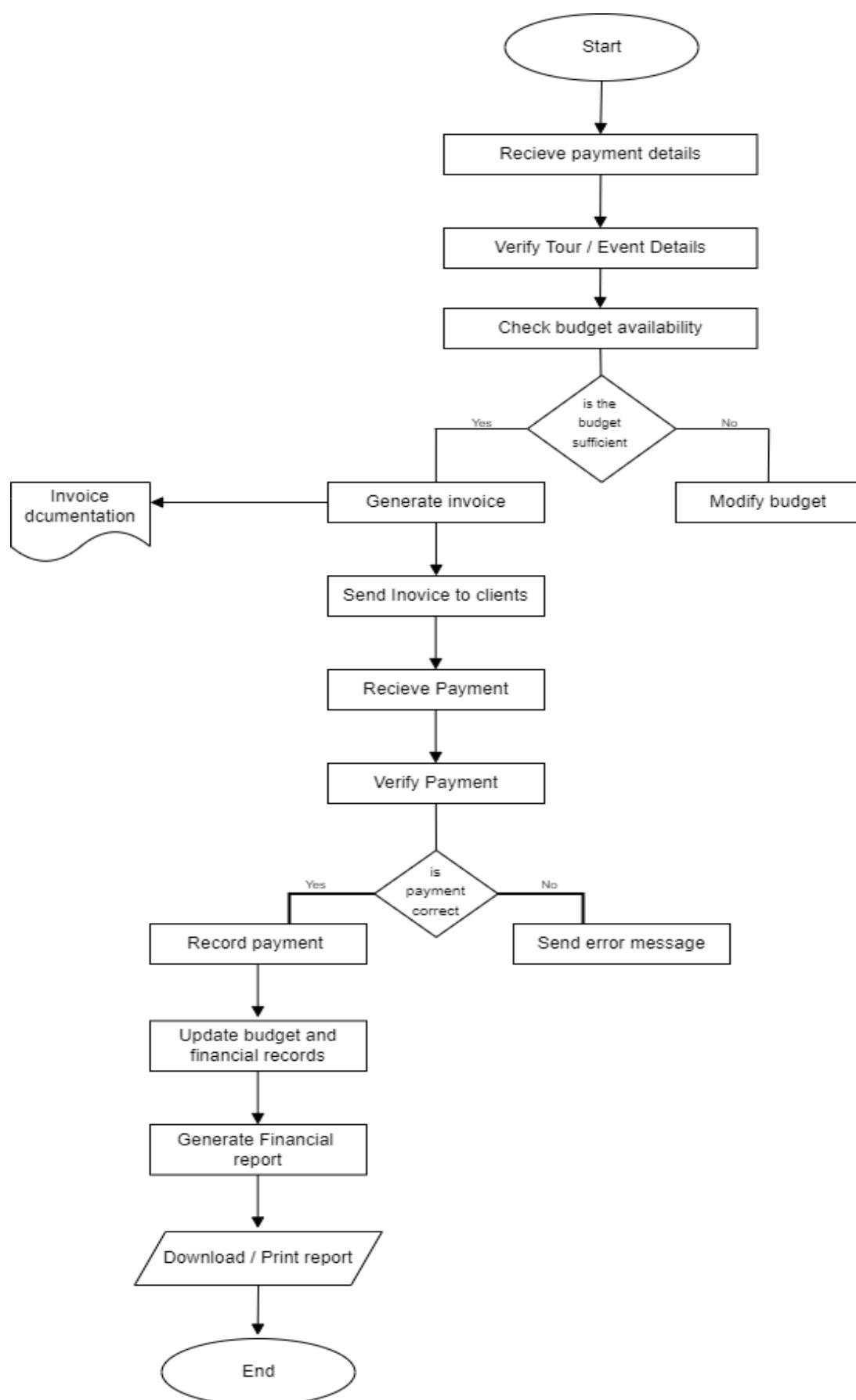
Hotel Management Flow chart



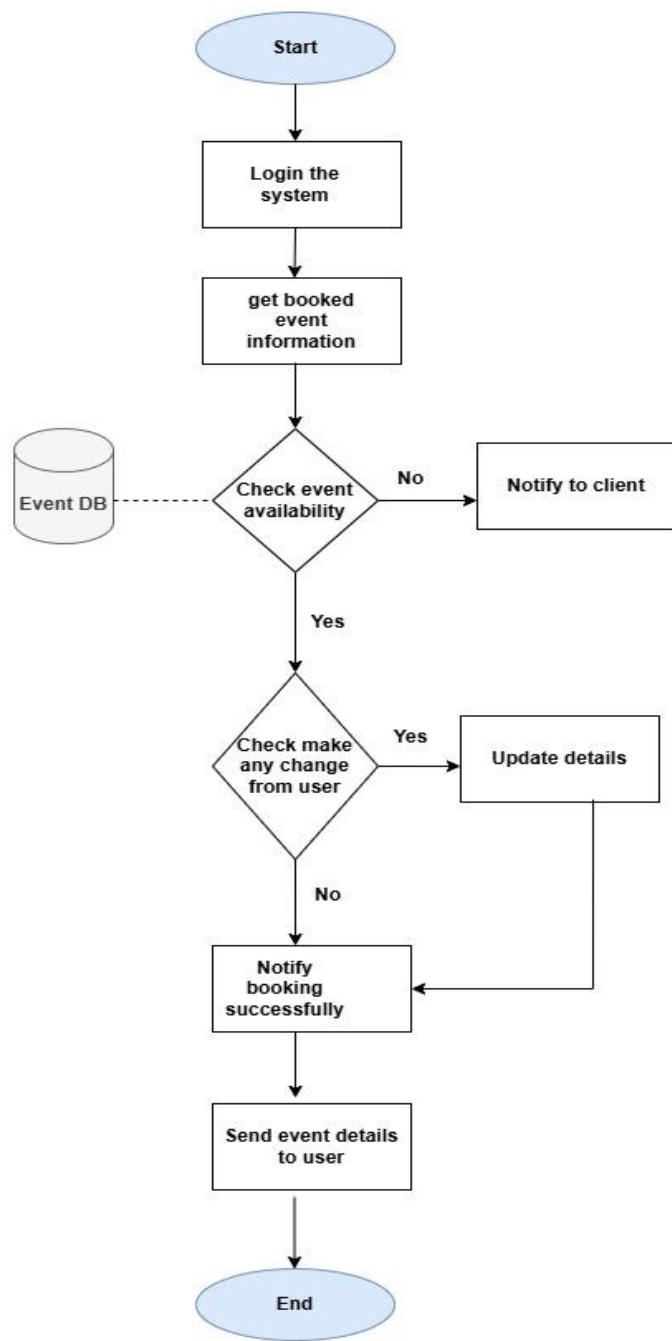
Transportation Management Flow chart



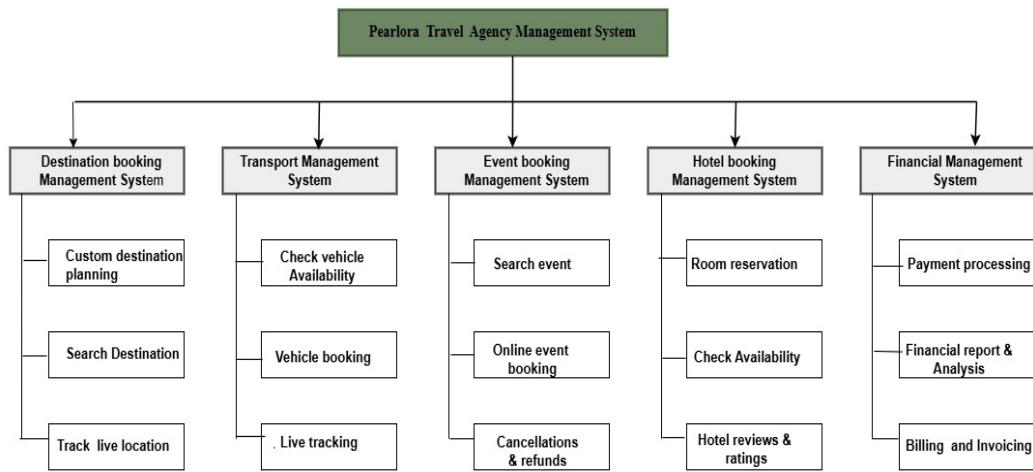
Financial Management Flow chart



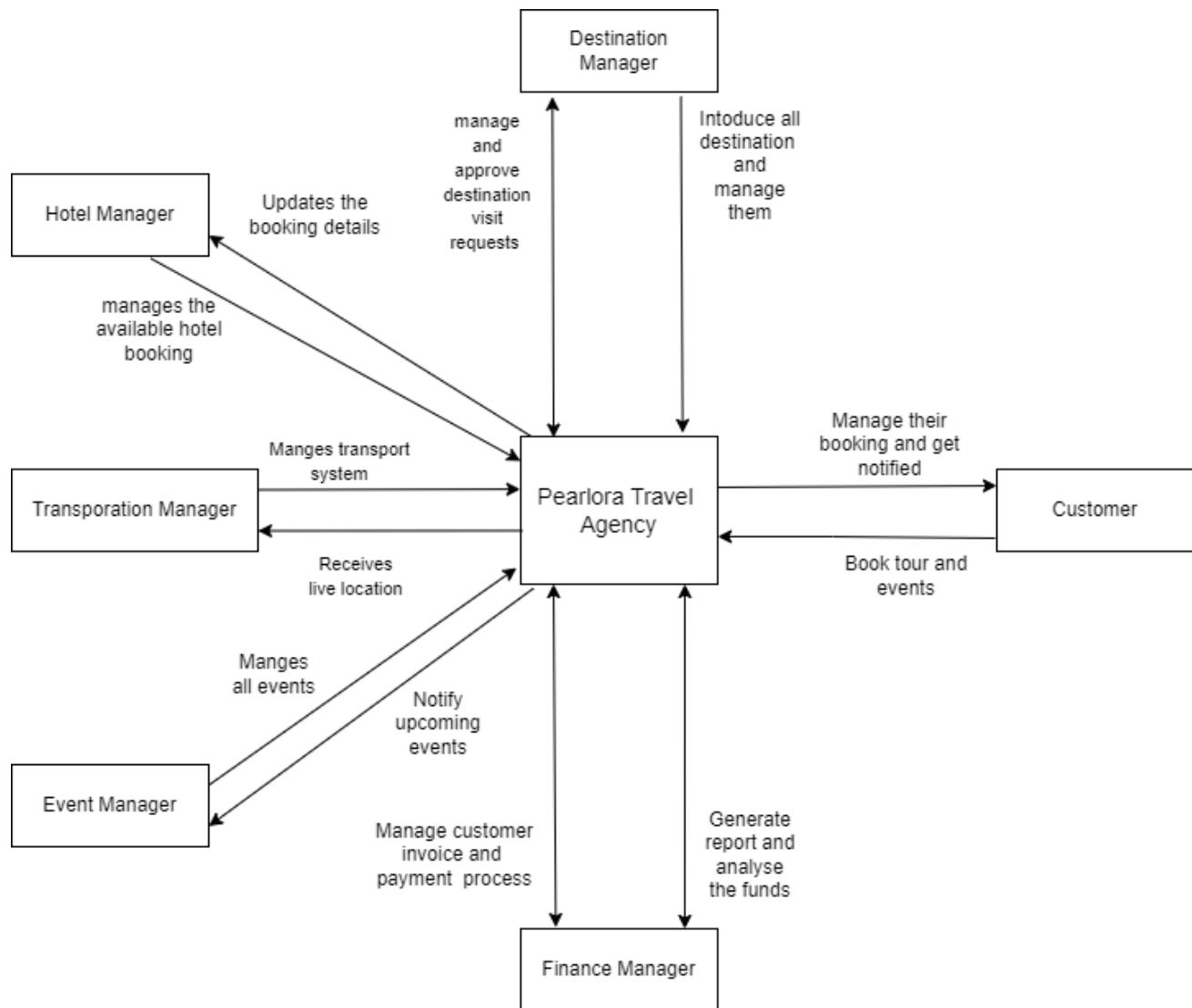
Event Management Flow chart

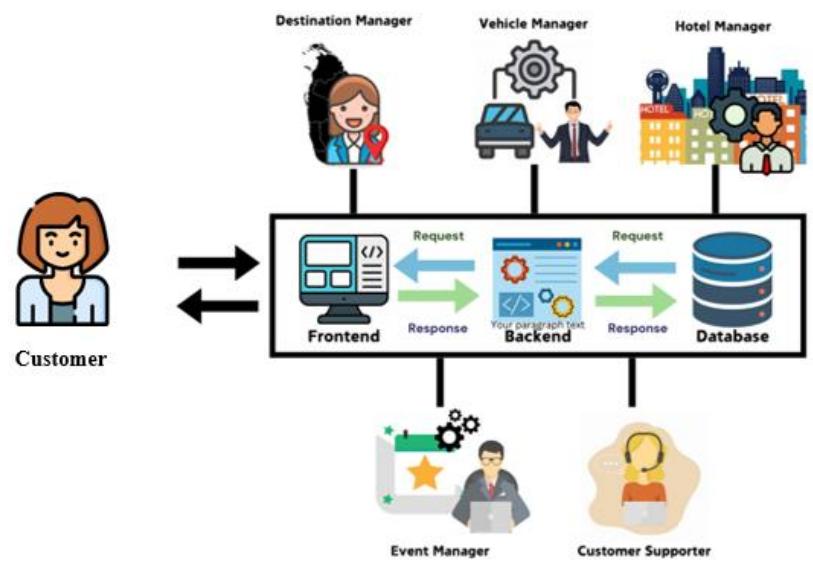


Data Flow Diagram



System Diagram





8. A suitable plan to develop the project as a team

Project Plan