FLIGHT FINDER PLATFORM - PROJECT REPORT

1. INTRODUCTION

1.1 Project Overview

The Flight Booking System is a comprehensive web application designed to streamline and modernize the process of booking flights. Developed using modern web technologies, this system provides a user-friendly interface for customers to search, select, and book flights, while also offering administrative tools for managing flight schedules, bookings, and related data. This system caters to a variety of users, including individual travellers, travel agencies, and airline staff, offering a flexible and efficient solution for managing air travel.

In today's fast-paced world, efficient and reliable flight booking systems are crucial for the travel industry. Many existing systems suffer from complex interfaces, lack real-time updates, and offer limited customization options. The Flight Booking System addresses these issues by providing an intuitive and responsive platform that simplifies the booking process and enhances the overall user experience.

The system supports a wide range of features, including flight searching based on various criteria (destination, date, etc.), detailed flight information display, passenger management, and secure booking confirmation. Administrators have full control over managing flight schedules, availability, pricing, and user roles. The platform also includes features for handling booking modifications and cancellations.

What sets this Flight Booking System apart is its focus on a seamless booking experience, robust data management, and secure transactions. The system aims to integrate essential functionalities, providing an end-to-end solution for flight reservation management.

Beyond basic booking, the system empowers administrators with tools to manage and analyse booking data, schedule flights, and ensure smooth operations. These features are essential for airlines and travel agencies aiming to optimize their services, improve customer satisfaction, and achieve sustainable growth in the competitive travel industry.

1.2 Purpose

The primary objective of the Flight Booking System is to offer an intuitive, reliable, and scalable digital solution for managing flight bookings. It aims to provide a platform where customers can easily search and book flights, and where administrators can efficiently manage flight schedules, passenger information, and booking details, all without requiring extensive technical expertise. This project was initiated to address the increasing demand for efficient, accessible, and secure online flight booking solutions across the travel industry.

For instance, a customer planning a trip often faces challenges in comparing flight options, managing booking details, and making changes to their itinerary. Traditional booking methods can be time-consuming and cumbersome, involving multiple steps and potential errors. The Flight Booking System solves this problem by offering an all-in-one platform where travellers can easily search for flights, view detailed information, and complete their booking in a few simple steps.

Similarly, a travel agency tasked with managing multiple bookings may struggle to keep track of flight schedules, passenger details, and booking confirmations. The Flight Booking System enables agencies to manage bookings efficiently, access real-time flight information, and generate reports, ensuring accuracy and reducing the risk of errors.

Even for airline staff, managing flight schedules, availability, and pricing can be a complex and time-consuming process. This system provides tools to streamline these operations, automate tasks, and improve overall efficiency.

The platform's purpose extends beyond simply facilitating flight bookings. By integrating user-friendly interfaces, efficient search functionalities, and administrative management tools, the Flight Booking System helps travellers plan their trips, assists agencies in managing bookings, and enables airlines to optimize their operations. Ultimately, it creates a connected ecosystem that simplifies air travel for everyone involved.

2. IDEATION PHASE

2.1 Problem Statement

In today’s increasingly connected world, the ability to efficiently manage travel has become more of a necessity than a luxury. Whether for business trips, vacations, or visiting loved ones, travellers and travel providers alike rely heavily on digital platforms to streamline the booking process. However, many existing flight booking solutions fall short when it comes to offering a user-centric, secure, and fully integrated travel management ecosystem.

For airlines and travel agencies, creating and managing flight schedules and bookings often becomes a tedious and fragmented process. Many platforms limit customization, lack intuitive booking management tools, or require a steep technical learning curve — which discourages travel professionals from fully embracing digital solutions. Additionally, the absence of automated systems for managing passenger information, handling booking modifications, and generating reports makes it difficult for them to scale their operations and maintain service quality.

For customers, the booking experience is often disrupted by complex interfaces, lack of real-time updates, and disconnected information. In the absence of clear flight options, secure payment gateways, and efficient communication channels, travellers tend to experience frustration, resulting in lost bookings and decreased satisfaction.

Administrators, too, face unique hurdles. Managing user accounts, moderating bookings, handling secure payments, and monitoring system performance across a growing user base are essential but complex tasks. Most platforms require separate plugins, manual interventions, or even third-party services to accomplish what should ideally be part of a unified system.

Security is another major concern for both travel providers and travellers. Flight booking platforms often involve sensitive personal information, including passport details, payment data, and travel itineraries. Without built-in security features like encrypted transactions, secure authentication protocols, and role-based access control, users are left exposed to data breaches and fraud.

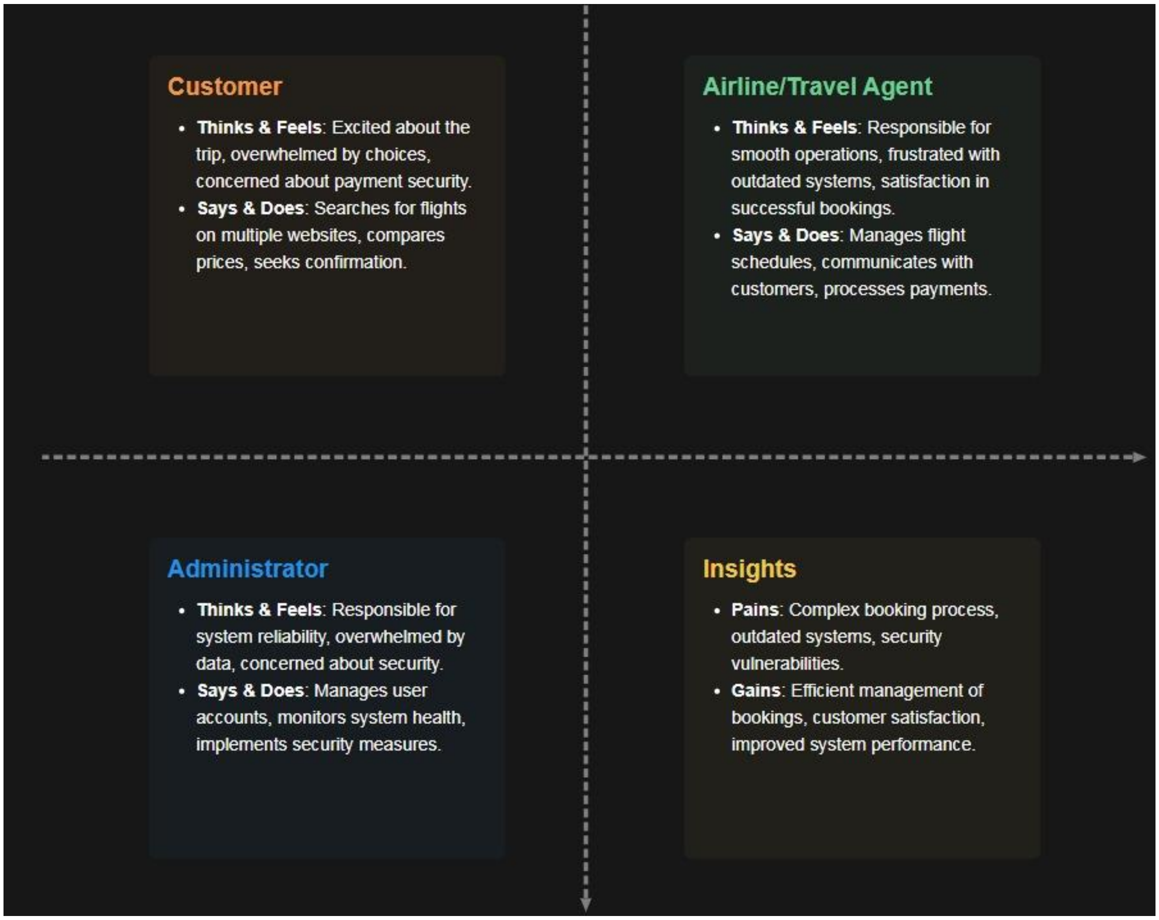
Given this context, there is a clear and urgent need for a flight booking platform that can bridge these gaps through an integrated, user-friendly, and secure solution. The ideal system should empower travel providers to focus on service delivery, allow customers to book flights with confidence, and equip administrators with tools for seamless operation — all while maintaining robust data privacy, real-time reporting, and customizable workflows.

The Flight Booking System was developed specifically to address these challenges, providing a single, streamlined ecosystem that enhances the booking, management, and administrative experience for modern digital travel.

Problem Statement Table:

I am (Role)	I’m trying to	But	Because	Which makes me feel
Customer	Book a flight	The booking process is complex	Interfaces are confusing and information is disconnected	Frustrated and dissatisfied
Airline/Travel Agent	Manage flight schedules and bookings	Tools lack customization	Systems are fragmented and require manual updates	Inefficient and overwhelmed
Traveler	Get clear flight options and secure booking	Tools lack customization	Payment gateways aren't always secure and communication is poor	Anxious and uncertain
Administrator	Manage user accounts and system performance	No unified dashboards	Monitoring tools are costly and require extra plugins	Inefficient and stressed

2.2 Empathy Map



Customer

- **Thinks & Feels**: Excited about the trip, overwhelmed by choices, concerned about payment security.
- **Says & Does**: Searches for flights on multiple websites, compares prices, seeks confirmation.

Airline/Travel Agent

- **Thinks & Feels**: Responsible for smooth operations, frustrated with outdated systems, satisfaction in successful bookings.
- **Says & Does**: Manages flight schedules, communicates with customers, processes payments.

Administrator

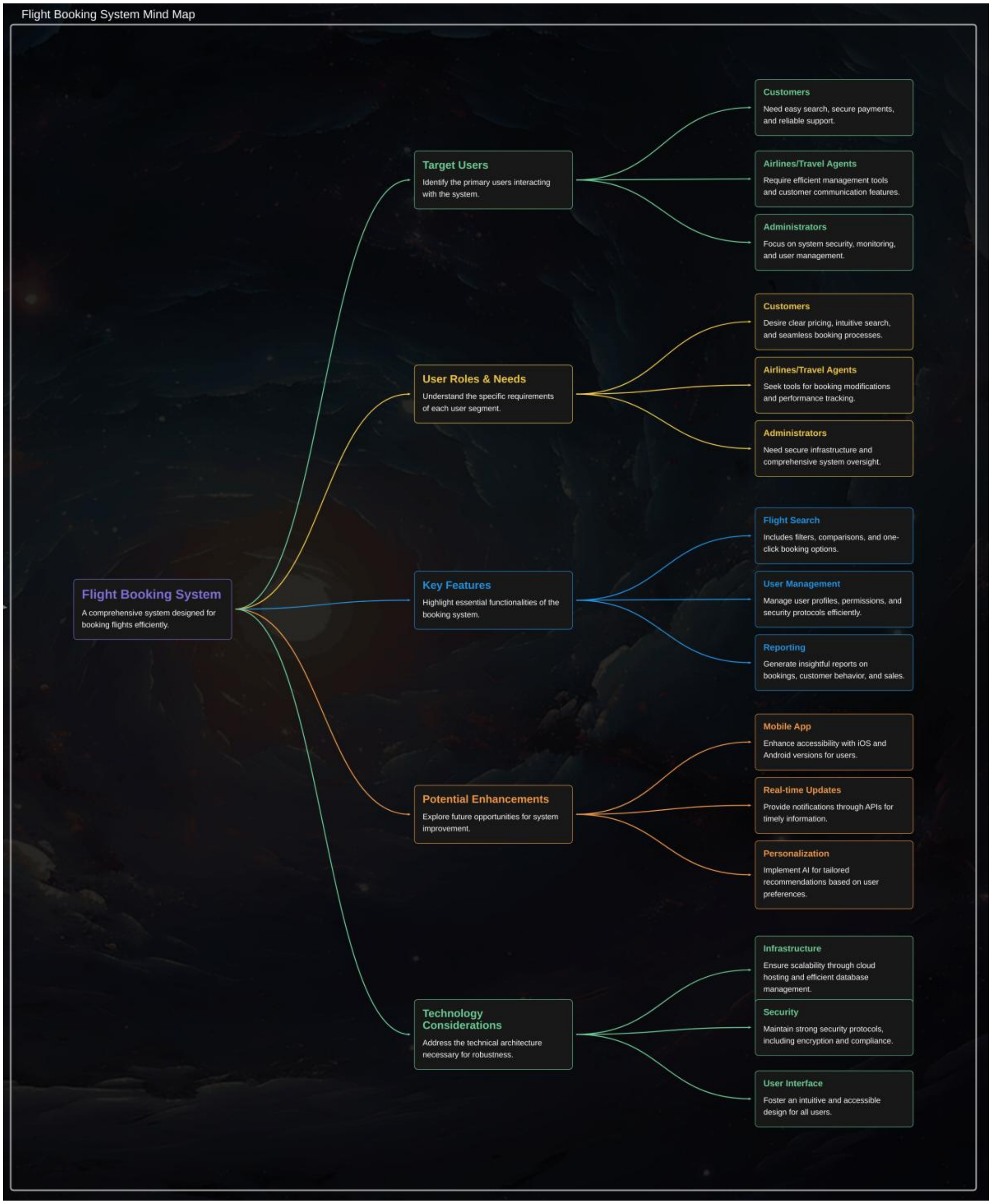
- **Thinks & Feels**: Responsible for system reliability, overwhelmed by data, concerned about security.
- **Says & Does**: Manages user accounts, monitors system health, implements security measures.

Insights

- **Pains**: Complex booking process, outdated systems, security vulnerabilities.
- **Gains**: Efficient management of bookings, customer satisfaction, improved system performance.

2.3 Brainstorming

Step-1: Team Gathering, Collaboration and Select the Problem Statement and Brainstorm, Idea Listing and Grouping



Flight Booking System Mind Map

Flight Booking System
A comprehensive system designed for booking flights efficiently.

Target Users
Identify the primary users interacting with the system.

- **Customers** Need easy search, secure payments, and reliable support.
- **Airlines/Travel Agents** Require efficient management tools and customer communication features.
- **Administrators** Focus on system security, monitoring, and user management.

User Roles & Needs
Understand the specific requirements of each user segment.

- **Customers** Desire clear pricing, intuitive search, and seamless booking processes.
- **Airlines/Travel Agents** Seek tools for booking modifications and performance tracking.
- **Administrators** Need secure infrastructure and comprehensive system oversight.

Key Features
Highlight essential functionalities of the booking system.

- **Flight Search** Includes filters, comparisons, and one-click booking options.
- **User Management** Manage user profiles, permissions, and security protocols efficiently.
- **Reporting** Generate insightful reports on bookings, customer behavior, and sales.

Potential Enhancements
Explore future opportunities for system improvement.

- **Mobile App** Enhance accessibility with iOS and Android versions for users.
- **Real-time Updates** Provide notifications through APIs for timely information.
- **Personalization** Implement AI for tailored recommendations based on user preferences.

Technology Considerations
Address the technical architecture necessary for robustness.

- **Infrastructure** Ensure scalability through cloud hosting and efficient database management.
- **Security** Maintain strong security protocols, including encryption and compliance.
- **User Interface** Foster an intuitive and accessible design for all users.

Step-2: Idea Prioritization

Our team conducted a structured brainstorming session to improve the usability, security, and efficiency of the Flight Booking System — a comprehensive tool designed for searching, booking, and managing flights. The session followed a three-step design thinking process: defining the problem, generating ideas, and prioritizing solutions.

We began by framing our core problem statement as:

"How might we make it easier, more secure, and more efficient for customers to book flights and for airlines/travel agents to manage flight operations through the Flight Booking System?"

With this guiding question, each of the four team members proposed three unique ideas based on their own experiences and research, resulting in a total of 12 key suggestions. From these contributions, four major themes naturally emerged during discussion:

- **Booking Experience & Usability**: Ideas focused on simplifying the flight search and booking process through an intuitive interface, advanced search filters, and a streamlined booking flow for customers.

- **Data & Security**: Given that flight booking platforms handle sensitive user information and financial transactions, ensuring strong security was identified as a top priority. Suggestions included secure payment gateway integration, SSL encryption, and robust data protection measures.

- **System Integration & Efficiency**: To improve operational efficiency and enhance the user experience, the team discussed integrating third-party services such as airline APIs for real-time flight data, secure payment gateways, and communication tools, along with comprehensive reporting and analytics for administrators.

- **Personalization & Customization**: Ideas were shared around giving users control over their booking experience through personalized recommendations, customizable notifications, and loyalty program integration.

After the brainstorming session, the team grouped similar concepts and assessed them based on their feasibility, technical complexity, and expected user impact. High-impact and high-feasibility features were selected as immediate development goals:

- Intuitive flight search and booking process

- Secure payment gateway integration

- Real-time flight updates and notifications

Meanwhile, more advanced or resource-intensive ideas — such as AI-driven personalized recommendations, integration with other travel services, and advanced reporting dashboards — were acknowledged as valuable but deprioritized for future iterations, once the core platform had stabilized.

This brainstorming exercise helped the team crystallize both short-term objectives and long-term ambitions for the Flight Booking System. It established a user-centered development roadmap that balances modern travel demands with technical scalability and system security.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey map

STAGES	**ENTICE (Become Aware)**	**ENTER (Explore App)**	**ENGAGE (Book Flight)**	**EXIT (Post-Booking)**	**EXTEND (Pre-Flight & Beyond)**
STEPS	- Discover via social media ads, app store search, referrals - Download the app	- Sign up/log in, browse flights - Use search filters, view flight details	- Select flight, enter passenger details - Complete booking, receive confirmation	- Manage booking (if needed) - Receive pre-flight reminders	- Check-in online, manage itinerary - Share travel experience, book again
PEOPLE	- Friends, family (for recommendations) - Online influencers, travel bloggers	- Customer support chatbot/agents	- Airline staff, payment gateway	- Customer support, airline agents	- Friends, family, social media connections

INTERACTIONS	- App store page, social media ads	- App interface, search filters	- Flight listings, passenger forms, payment options	- Booking confirmation screen, email notifications	- App (check-in, itinerary), social media platforms
THINGS	- App icon, screenshots, user reviews	- App interface, search filters, flight listings	- Flight details, passenger information fields, payment gateway	- Booking confirmation, e-ticket, itinerary	- Online check-in, boarding pass, travel updates, social media sharing features
GOALS & MOTIVATION	- Find a flight that meets travel needs - Get inspired for a trip	- Explore flight options and prices - Understand booking process	- Secure a flight booking - Receive booking confirmation	- Prepare for the trip - Share experience, plan future trips	
POSITIVE MOMENTS	- Attractive app icon and design - Easy-to-use search filters	- Smooth sign-up/login process - Clear flight information - Helpful chatbot support	- Seamless booking process - Secure payment process - Instant booking confirmation	- Easy access to booking details - Timely pre-flight reminders	- Convenient online check-in - Positive travel experience sharing
NEGATIVE MOMENTS	- Unclear app description	- Complicated sign-	- Complex booking process -	- Difficulty in changing	- Complicated check-in,

	- Poor app store reviews	up/login - Hidden fees, limited options - Unclear instructions	Payment failures, lack of security - No booking confirmation	booking - Lack of updates or support	poor communication - Negative experience sharing
AREAS OF OPPORTUNITY	- Personalized flight recommendations - AI-powered chatbot for initial support	- Streamlined search and filtering - Virtual tours of flight/services	- Simplified booking process - Secure and transparent transactions	- Proactive booking management tools - Integration with travel services	- Enhanced post-trip engagement - Loyalty programs and personalized offers

3.2 Solution Requirement

Functional Requirements Table:

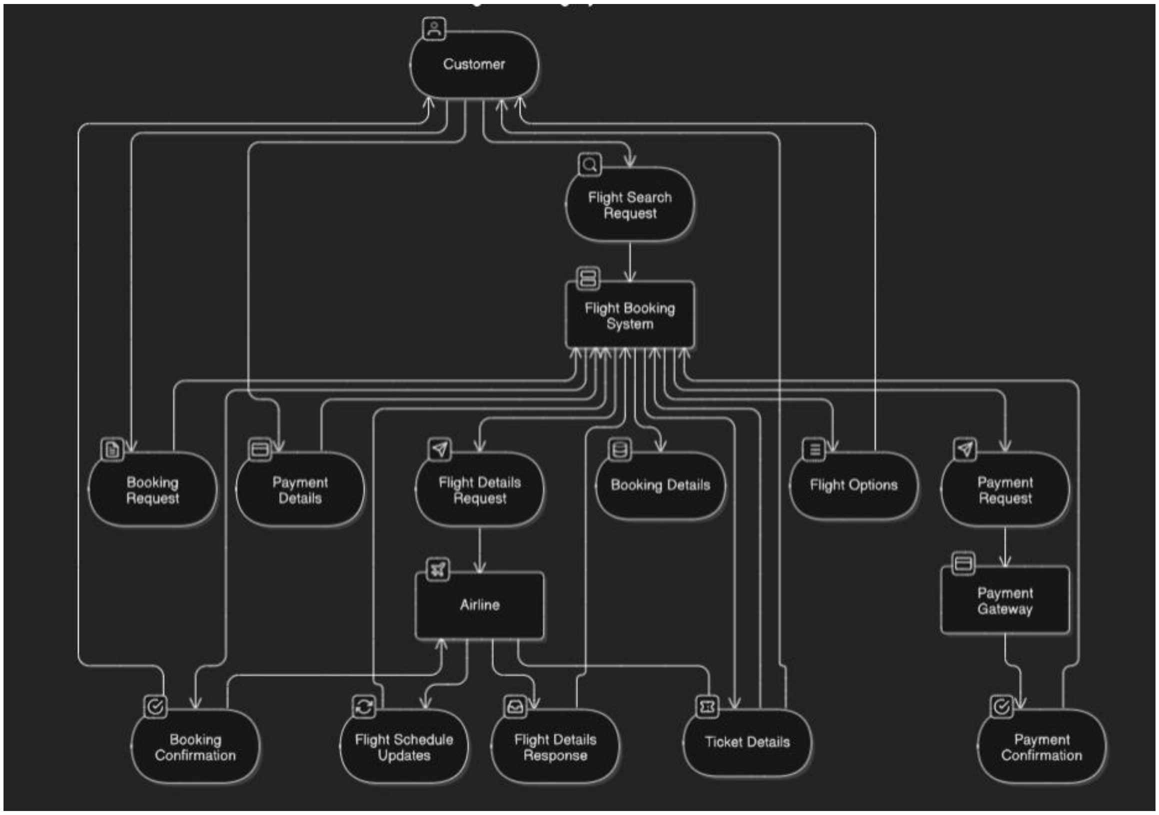
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
1	User Registration	Registration via Email Registration
2	Flight Search and Booking	Search Flights by Criteria (Date, Destination, Passengers) Display Flight Options (Airlines, Prices, Times) Select Flights and Add Passengers Secure Booking Confirmation

3	Booking Management	View Booking History Modify/Cancel Bookings Manage Passenger Information
4	Payment Processing	Secure Payment Gateway Integration (Credit Card, etc.) Payment Confirmation and Receipt Generation Handle Payment Refunds
5	Airline Integration	Retrieve Flight Schedules and Availability Update Flight Status
6	User Profile Management	View/Edit User Profile Manage Travel Documents (e.g., Passport) Store Frequent Flyer Information
7	Notifications and Communication	Send Booking Confirmations Send Flight Updates/Reminders (e.g., Delays) Provide Customer Support Communication Channels
8	Admin Dashboard	User Role Management (Admin, Customer, Airline Staff) System Health Monitoring Manage Flight Schedules and Pricing
9	Reporting and Analytics	Generate Booking Reports Track Popular Routes and Travel Times Revenue Reports for Airlines

Non-Functional Requirements Table

FR No.	Non-Functional Requirement	Description
1	Usability	User-friendly UI for all user roles (Customers, Airline Staff, Admins) Responsive design for web and mobile platforms Consistent and intuitive booking flow
2	Security	SSL encryption for secure communication Secure storage of user data (including payment information) Protection against common web vulnerabilities (e.g., SQL injection, XSS)
3	Reliability	System should ensure no data loss during booking process High availability for booking and flight status updates
4	Performance	Fast page load times, optimized for high traffic volumes Efficient flight search responses Minimal latency for booking confirmations
5	Scalability	Handle a large number of concurrent users and bookings Support a growing database of flights, airlines, and users
6	Availability	24/7 availability with minimal downtime (e.g., 99.9% uptime)
7	Compliance	Comply with relevant data privacy regulations (e.g., GDPR) Ensure secure payment processing (e.g., PCI DSS)

3.3 Data Flow Diagram



3.4 Technology Stack

Technical Requirements

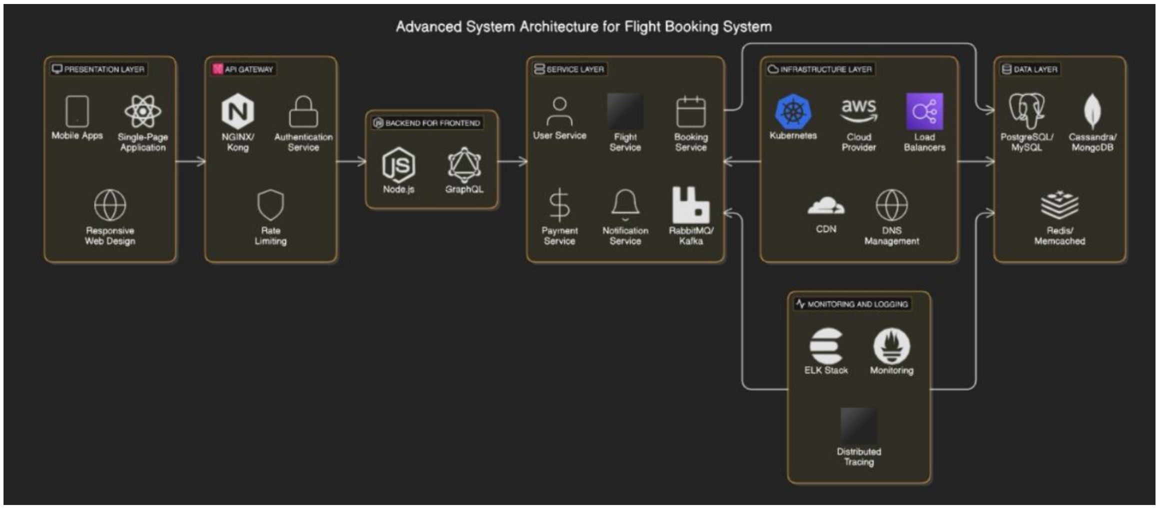


Table -1 : Technical Architecture

S.No	Component	Description	Technology
1	User Interface	Frontend interface for customers and airline staff to interact with the flight booking system.	HTML, CSS, JavaScript, React.js, Material UI
2	Application Logic - 1	Handles flight search, booking management, and user profile management.	React.js, Redux Toolkit
3	Application Logic - 2	Manages user authentication, session handling, and communication with external APIs.	Node.js, Express.js, JWT Authentication
4	Application Logic - 3	Generates booking reports, flight analytics, and system performance metrics.	Chart.js, React-Chartjs-2, Node.js

5	Database	Stores user accounts, flight schedules, booking details, and payment information.	Mongodb
6	Cloud Database	Cloud-hosted relational database for scalability and distributed storage.	Mongodb (Cloud)
7	File Storage	Stores uploaded documents, images, and other files related to bookings and user profiles.	AWS S3/Cloudinary Storage
8	External API - 1	Handles transactional email notifications (booking confirmations, itinerary updates, etc.).	Nodemailer, SMTP
9	External API - 2	Integrates with payment gateway for secure transaction processing.	Stripe API, Webhooks
10	Infrastructure	Cloud-hosted deployment of backend services with containerization and secure scaling.	Docker, Kubernetes, AWS/Azure/GCP

Table -2: Application Characteristics

S.No	**Characteristics**	**Description**	**Technology Used**

1	Open-Source Frameworks	The platform uses open-source, community-backed frameworks for creating responsive UIs and robust backend services.	React.js / Node.js / Express.js / Mongodb
2	Security Implementations	Security is ensured through HTTPS, JWT-based authentication, and secure payment handling.	SSL, HTTPS, JWT Tokens, Stripe API, bcrypt hashing
3	Scalable Architecture	Designed using a microservices-friendly and modular architecture for easy scaling and maintenance.	Node.js (Backend), React.js (Frontend), Mongodb (Database), Docker, Kubernetes
4	Availability	Cloud deployment with distributed architecture, auto-scaling, and load balancers ensure high uptime and fault tolerance.	AWS, Docker, Kubernetes
5	Performance	Optimized for fast response times using caching and efficient database querying.	Redis (Caching), Mongodb Indexing

4. PROJECT DESIGN

4.1 Problem Solution Fit

The Problem–Solution Fit Canvas allowed us to align our Flight Booking System's core features with the evolving needs of modern travellers, airlines, and travel agents. Through continuous user feedback and industry research, we identified that traditional flight booking tools often suffer from:

- Limited flexibility and personalization for customers.
- Complex and rigid management systems for airlines/travel agents.
- Lack of real-time insights and progress tracking for administrators.
- Fragmented payment and booking workflows, which reduce the booking experience's fluidity.

By carefully mapping out the user journey, pain points, and expectations of each stakeholder — customers, airlines/travel agents, and administrators — we refined our solution to directly address these challenges.

Our platform enables:

- Airlines/Travel Agents to efficiently manage flight schedules, bookings, and customer data, with flexible tools and real-time updates.
- Customers to easily search and compare flights, book securely, and manage their travel plans, with personalized options and clear communication.
- Administrators to oversee system health, manage user roles, and ensure smooth payment and reporting workflows.

This approach ensured that our solution is not only technically robust but also user-centred, scalable, and ready for real-world adoption in the travel industry.

4.2 Proposed Solution

S.No	**Parameter**	**Description**
1	Problem Statement	Many travellers and travel providers face difficulties accessing flexible, affordable, and efficient flight booking options. Existing platforms often lack personalization, smooth booking processes, and intuitive management tools — limiting both customer satisfaction and operational efficiency.

2	Idea / Solution Description	The Flight Booking System is a full-featured, user-friendly web application that empowers customers to easily search, compare, and book flights, while enabling airlines and travel agents to efficiently manage schedules, bookings, and customer data.
3	Novelty / Uniqueness	Unlike conventional flight booking platforms, this system combines a user-friendly interface, secure payment processing, and efficient management tools, built on a scalable architecture, allowing for seamless booking experiences and streamlined operations.
4	Social Impact / Customer Satisfaction	The platform enhances accessibility to convenient and affordable travel for customers, while empowering airlines and travel agents to optimize their operations and reach a wider audience — increasing both reach and satisfaction.
5	Business Model (Revenue Model)	The solution can adopt a commission-based model, charging a fee per booking to airlines or travel agents. Additional revenue streams can include premium features, API access for partners, and targeted advertising.
6	Scalability of the Solution	The platform’s cloud-native, modular architecture supports smooth scaling for both traffic spikes and user growth. Leveraging containerization (Docker) and cloud databases allows efficient handling of a large volume of flight data and concurrent user access.

4.3 Solution Architecture

The Flight Booking System is a dynamic online solution built using a modern architecture. Its architecture is designed to effectively bridge the gap between travel needs and technology by providing a secure, interactive, and scalable platform for digital flight booking.

1. Finding the Best Tech Solution for Travel Challenges

The Flight Booking System addresses common challenges in the travel industry, such as:

- The need for intuitive booking tools for customers
- Secure handling of customer data and payment processing
- Seamless integration with airline systems for enhanced functionality

The architecture was selected for its flexibility, performance, and robust ecosystem:

- Mongodb offers structured storage for flight schedules, user data, and booking details.
- Node.js and Express.js efficiently handle backend logic and API services.
- React.js provides a responsive, dynamic interface for both travellers and airline staff.

2. Describing the Software Architecture to Stakeholders

The Flight Booking System follows a multi-layered architecture:

- Presentation Layer (React.js): Handles all UI components including booking interfaces and admin panels.
- Application Layer (Express.js + Node.js): Manages booking logic, authentication, and payment processing.
- Data Layer (Mongodb): Stores flight content, user data, and analytics.

Additional platform features include:

- Real-time flight updates and notifications
- Customizable booking options and itinerary management
- Integrated analytics for performance monitoring

3. Defining Features, Development Phases, and Requirements

Key Features:

- Intuitive flight search and booking process
- Secure user authentication and payment integration

- Integration with airline APIs for real-time data
- Comprehensive booking management and reporting
- Admin dashboards for system management

Development Phases:

- Phase 1 – Core Booking Platform UI (React.js)
- Phase 2 – Backend Services & Database Architecture (Express)
- Phase 3 – Airline API and Payment Integration
- Phase 4 – Notification and Reporting Features
- Phase 5 – Deployment & Performance Optimization

Requirements:

- Responsive design for all device types
- Role-based access (customers, airline staff, admins)
- High availability with scalable infrastructure

4. Providing Specifications for Managing and Delivering the Solution

The Flight Booking System implements industry best practices for software delivery:

- Version Control: Git/GitHub for team collaboration
- CI/CD: Automated testing and deployment pipelines
- Containerization: Docker for consistent environments
- Cloud Deployment: AWS for global availability
- Monitoring: Performance tracking and error logging

5. PROJECT PLANNING & SCHEDULING

5.1 Project planning

Product Backlog and Sprint Schedule for E- learning Platform

1. Agile Methodology Implementation

The platform follows Scrum-based Agile development to ensure iterative delivery of functional components:

- Sprint Cycles: 2-week sprints with prioritized backlogs
- User Stories: Epics broken into atomic tasks (e.g., "As a customer, I want to view available flights")
- Story Points: Fibonacci-based estimation (1-5 scale) reflecting complexity
- Definition of Done: Completion criteria include passing tests and stakeholder review

2. Technical Architecture Theory

Technology Stack Rationale:

- Component-Based UI (React): Enables reusable components for flight search, booking, and user management.
- RESTful Services (Express/Node): Stateless API design supports horizontal scaling for high traffic.
- Relational Database (Mongodb): Structured storage for flight schedules, booking details, and user data.
- JWT Authentication: Secure token-based sessions with role-based claims for customers, airline staff, and admins.

Key Principles Applied:

- Self-service booking via intuitive UI and clear navigation
- Real-time updates through flight status notifications and booking confirmations
- Modular design of services for independent scaling and deployment

Assessment Design:

- Formative: Embedded validation in booking forms and search filters
- Summative: Booking confirmation and itinerary generation upon successful booking
- Predictive: Analytics to identify popular routes, booking patterns, and potential issues

3. Quality Assurance Framework

Testing Pyramid Implementation:

- Unit Tests: Jest for React components, Mocha for API routes
- Integration Tests: Supertest for API endpoints, testing interactions between services
- E2E Tests: Cypress for critical user journeys like flight search, booking, and payment

Security Protocols:

- CSRF protection for forms
- Secure payment data handling via Stripe API
- Regular dependency audits using npm audit

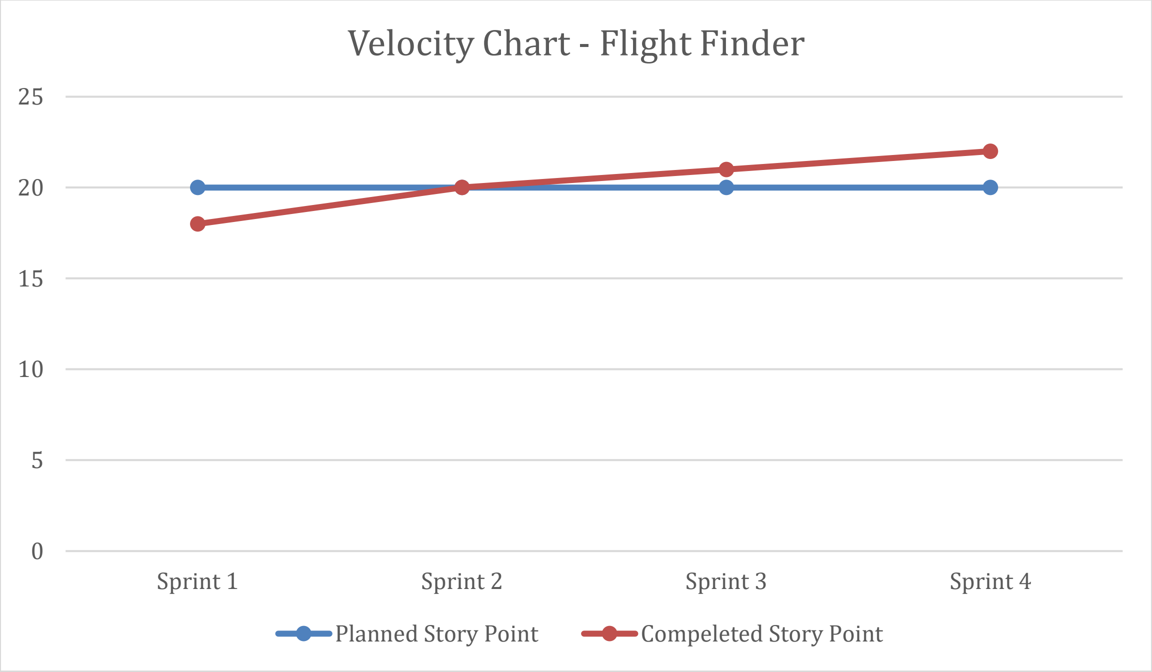
Project Tracker, Velocity & Burndown Chart

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed	Velocity
Sprint 1	20	2 weeks	2025-02-18	2025-03-03	18	18
Sprint 2	22	2 weeks	2025-03-04	2025-03-17	20	20
Sprint 3	22	2 weeks	2025-03-18	2025-03-31	21	21
Sprint 4	22	2 weeks	2025-04-01	2025-04-13	22	22
Total	**86**	**8 weeks**			**81**	

Velocity Chart (Summary)

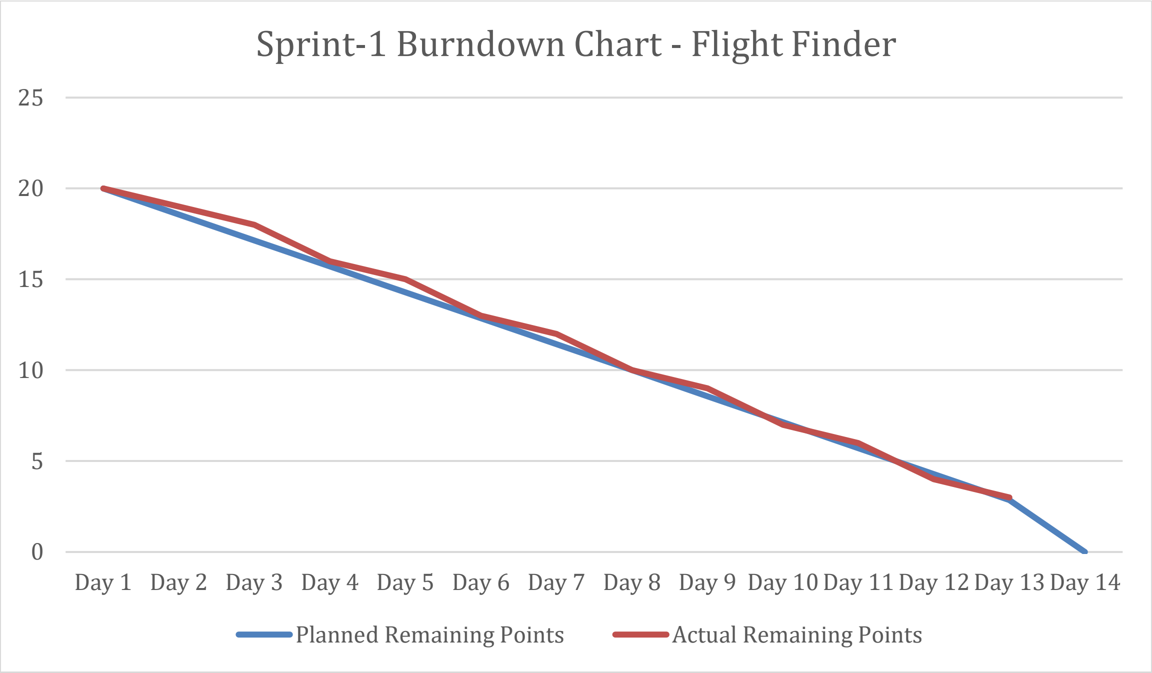
Sprint	Story Points Completed
Sprint-1	18
Sprint-2	20
Sprint-3	21
Sprint-4	22

Average Velocity=(18+20+21+22)/4=20.25 Story Points per Sprint



Burndown Chart Data

Day	Planned Remaining Points	Actual Remaining Points
Day 1	20	20
Day 2	18.57	19
Day 3	17.14	18
Day 4	15.71	16
Day 5	14.29	15
Day 6	12.86	13
Day 7	11.43	12
Day 8	10.00	10
Day 9	8.57	9
Day 10	7.14	7
Day 11	5.71	6
Day 12	4.29	4
Day 13	2.86	3
Day 14	0	0



Agile Sprint Breakdown for E-learning Platform

Epic: User authentication and Design

Task	**Story**	**Story Points**	**Complexity**
Design flight search page	**USN-1**	**3**	**Medium**
Implement flight search logic	**USN-2**	**5**	**Medium**
Implement user authentication	**USN-3**	**3**	**Medium**
Develop flight booking page	**USN-4**	**5**	**Medium**

Total Story Points (Sprint 1): 16

Sprint 2 (5 Days)

Epic: Booking and User Management Features & Security

Task	**Story**	**Story Points**	**Complexity**
Implement passenger details form	**USN-5**	**3**	**Medium**
Implement secure payment processing	**USN-6**	**5**	**High**
Implement user role management	**USN-7**	**3**	**Medium**
Implement system health monitoring	**USN-8**	**2**	**Medium**

Total Story Points (Sprint 2): 13

Velocity Calculation

Using the data from the Flight Booking System Project Tracker table:

- Sprint 1 Story Points: 18
- Sprint 2 Story Points: 20
- Sprint 3 Story Points: 21
- Sprint 4 Story Points: 22
- Number of Sprints: 4

Calculation:

Velocity = (Sprint 1 Story Points + Sprint 2 Story Points + Sprint 3 Story Points + Sprint 4 Story Points) / Number of Sprints

Velocity = (18 + 20 + 21 + 22) / 4

Velocity = 81 / 4

Velocity = 20.25 Story Points per Sprint

Result:

The team's average velocity is 20.25 Story Points per Sprint.