IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Name: Henaka Kumara
Student Reference Number: 10898536

Module Code: PUSL2021	Module Name: Computing Group Project
Coursework Title: PUSL2021 Course Work Final Report	
Deadline Date: 20.04.2024	Member of staff responsible for coursework: Mr. CMB Attanayake
Programme: Airline reservation Management System	

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

Name(as on Dle)	Plymouth ID
10898536	Henaka Kumara
10898502	Shakya Jayathilaka
10908162	Horagala Piyumani
10749144	Kaluthanthiri Patabandi
10898438	Jayasundara Dasumi
10898467	Tharushi Galappaththi

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.
Signed on behalf of the group: chamod

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed :

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I *have used/not used translation software.

If used, please state name of software……

Overall mark _____% Assessors Initials _____ Date_____

Table of Contents

INTRODUCTION

Overview of the project

Air travel has enhance an essential component of modern maneuverability in contemporary's fast-moving people. The increasing recognition of air travel makes necessary the happening of reliable and persuasive orders for management label, departure arranging, and different joined processes. So that meet these goals, the airplane Check Administration Plan offers an robotic terrace that authorizes passengers and airlines to surely control airplane stipulations, label, and added joined movements. Most of us undertake any tasks light part of 24 hours that demand communicating accompanying databases. Skilled's a chance that dignitary will approach a table as some our projects, to a degree when making inn or air carrier conditions.This program determines you accompanying an survey of the stipulation administration process inside the air carrier paper administration plan. Accompanying the aid concerning this project, the whole air license administration order process is explained. It offers the talent to increase, eliminate, refine, and search air carrier card administration news.

Purpose of the project

The aircraft Ticket Management System project aims to offer an all-inclusive software solution that tackles the issues and demands that airlines and travelers encounter when handling aircraft bookings and tickets. Below are the key purpose of the project :-

1. **Streamlining the Booking Process:** By offering an easy-to-use interface for passengers to search for flights, choose seats, and make bookings quickly, the system seeks to streamline the process of booking flights for travelers.

2. **Effective Ticket Management:** It makes it easier to manage tickets that have been reserved in an efficient manner. This includes keeping track of reservation information, assigning seats, and easily processing changes or cancellations.

3. **Improving Customer Experience:** The system improves the overall experience for travelers, resulting in higher satisfaction and loyalty, by providing a user-friendly interface, a smooth booking procedure, and timely notifications.
4. **Streamlining Operations for Airlines:** Automated seat assignments, in-flight communication in real time, and data-driven insights for scheduling and resource optimization can all help airlines run more efficiently.

5. **Revenue Generation:** By providing dynamic pricing, marketing auxiliary services, and using analytics to spot trends and demand patterns, the system helps airlines to optimize revenue prospects.

6. **Ensuring Security and Compliance:** The system safeguards sensitive passenger data and guarantees safe transactions by putting strong security measures in place and according to industry laws (including GDPR and PCI DSS).

7. **Offering Reports and Analytics:** Airlines may make data-driven decisions by gaining insights into booking patterns, passenger demographics, revenue performance, and operational indicators using comprehensive reports and analytics.

Justification for the project

The rationale behind executing the Flight Ticket Management System project is to tackle multiple pressing requirements and obstacles encountered by airlines and travelers in the aviation sector. This is a thorough explanation of the project's rationale:

1. **Efficiency and Cost Reduction:** Conventional manual flight reservation and ticketing procedures need a lot of time and resources. The implementation of a comprehensive management system to automate these operations results in notable enhancements in efficiency, reduction of manual errors, and decreased operational expenses for airlines.

2. **Improved Customer Experience:** Travelers of today anticipate easy and convenient booking procedures. The Flight Ticket Management System offers tailored services, real-time updates, and an intuitive interface that improves the overall client experience, increasing satisfaction and loyalty.

3. **Optimized Operations:** To stay competitive, airlines must effectively manage their flight schedules, seat assignments, and resource allocations. Airlines can work more efficiently and profitably thanks to the management system's real-time flight management, seat optimization, and resource allocation technologies.
4. **Revenue Maximization:** Airlines can maximize revenue prospects through data-driven insights, ancillary service promotion, and dynamic pricing. Increased revenue production is achieved by the system's facilitation of the execution of targeted pricing strategies, identification of demand trends, and promotion of new services to passengers.

5. **Compliance and Security:** Airlines must make sure they are in compliance with standards like GDPR and PCI DSS because the aviation sector is subject to strict regulations governing data privacy and security. The management system protects passengers and airlines by enforcing regulatory compliance and putting strong security measures in place to secure passenger data.

6. **Decision Support and Analytics:** The management system's data-driven insights enable airlines to decide on pricing, schedule, and resource allocation with confidence. Airlines may enhance their operations and boost profitability by

utilizing analytics solutions, which offer insightful data on revenue performance, passenger demographics, and booking trends.

Scope of the project

This is a summary of the extent:

1. **User Management:** Passengers, airline employees, and administrators can all register and authenticate. Access control and administration of user profiles according to roles and permissions.

2. **Flight Search and Booking:** Users can look for available flights using different parameters like date, destination, and class. Easy-to-use booking interface that lets users choose flights, input passenger information, and make reservations.

3. **Ticket Management**: Information on the passenger, flight information, seat assignments, and ticket status (such as confirmed, pending, or canceled) are all stored and managed in relation to booked tickets. Features that let users see, edit, or cancel their reservations as needed.

4. **Payment Processing:** Integration with payment gateways to make safe flight reservation transactions possible. support for a variety of payment options, such as mobile wallets, credit/debit cards, and electronic financial transfers.
5. **Seat Allocation:** Seats can be assigned automatically or manually in accordance with user preferences, availability, and specific needs (such as extra legroom or wheelchair access).

6. **Check-in and Boarding:** Online check-in features that help travelers choose their seats, print boarding passes, and finish any required steps before getting to the airport. Boarding management tools to efficiently manage passenger lines and speed up the boarding process.

7. **Flight Management:** Administration tools that help airlines keep track of available flights, adjust timetables, and deal with unforeseen circumstances like cancellations or delays. Features for efficiently allocating resources, scheduling and canceling flights, and adding new ones.

Objectives

- To automate the management of airline tickets to some extent.
- To assist the airline system in increasing the effectiveness of its operations.
- An extra draw for their prospective clients.
- Additionally, it will demonstrate the management's attitude that they are aware of the newest technologies and prepared to use them.
- Make sure that travelers can make educated decisions by providing them with up-to-date information on flight availability, costs, and seat selections.
- Make sure the system complies with industry norms and rules, safeguarding traveler information and transactions while upholding security and privacy standards.

Acknowledgments

As we reflect on the journey of developing the Airline Reservation System, we are filled with gratitude towards all those who have contributed their time, expertise, and resources to make this project a success. This section is dedicated to acknowledging the invaluable support we have received.

Project Team

Our heartfelt thanks go to each member of the project team. Your commitment, dedication, and tireless efforts have been the driving force behind this project. Your collaborative spirit and innovative ideas have transformed challenges into opportunities and made this system what it is today.

Project Supervisors

We extend our sincere gratitude to our project supervisors, whose guidance and insights have been crucial. Your constant support and constructive feedback have shaped our approach and propelled us towards excellence. Your mentorship has not only guided this project but has also prepared us for future professional challenges.

Stakeholders and Clients

We are immensely thankful to our stakeholders and clients for their trust and collaboration. Your engagement and feedback have been essential in aligning our project with real-world needs and expectations. Your contributions have ensured the system's relevance and usability.

Faculty and Mentors

Special thanks to the faculty members and mentors from the Faculty of Computing at Plymouth University. Your academic support and the resources provided were instrumental in our ability to undertake this ambitious project. The knowledge and expertise you shared have been invaluable to our success.

Technical Support Staff

We acknowledge the technical support staff whose behind-the-scenes work made our project possible. From troubleshooting software issues to providing necessary hardware, your prompt and reliable support kept our project on track.

Peer Reviewers

Thank you to our peers who participated in reviewing and testing phases, providing critical feedback that was integral to refining the Airline Reservation System. Your unbiased critiques have helped us enhance the system’s functionality and user experience.

We are indebted to everyone mentioned and unmentioned who contributed to the project's success. Your support has not only made this project possible but has also enriched our learning experience and personal growth.

Background

Literature Study

Overview of Existing Research :

The research surrounding airline reservation management systems encompasses a wide array of topics, including system architecture, user experience optimization, pricing strategies, and technological advancements. Numerous academic papers, industry reports, and articles delve into these areas, offering insights into the challenges and opportunities present in the domain. Additionally, a variety of products and solutions already exist in the market, ranging from comprehensive systems utilized by major airlines to more specialized offerings tailored to specific needs.

Evaluation of Existing Products/Solutions :

Existing airline reservation management systems exhibit both strengths and weaknesses. Major solutions often boast robust features, scalability, and extensive integration capabilities. However, they may also suffer from complexities in usability, high implementation costs, and rigidity in customization. On the other hand, niche solutions may offer simplicity and cost-effectiveness but might lack the breadth of functionality required by larger airlines. Evaluating these solutions provides valuable insights into areas for improvement and innovation.

Theoretical Framework for the Solution

System Development Theory :

Our project adopts an Agile methodology for system development, emphasizing iterative development, collaboration, and adaptability. Agile methodologies are well-suited for dynamic environments like airline reservation management, allowing for frequent updates, responsiveness to changing requirements, and early delivery of valuable functionality. By

embracing Agile principles, we aim to ensure the timely delivery of a high-quality solution that meets the evolving needs of our stakeholders.

Design Level Theory :

The design of our airline reservation management system is guided by principles of usability, accessibility, and user-centered design. Drawing upon concepts from human-computer interaction (HCI) and interaction design, we prioritize intuitive interfaces, clear navigation, and seamless user experiences. Accessibility considerations ensure that our system is usable by individuals with diverse needs and abilities, enhancing inclusivity and user satisfaction.

Justification :

The selection of Agile methodology aligns with our project's objectives of delivering a flexible, responsive, and customer-centric solution. By embracing iterative development cycles and continuous feedback loops, we can rapidly adapt to changing market dynamics and stakeholder requirements. Likewise, our focus on user-centered design principles ensures that our system prioritizes the needs and preferences of end-users, ultimately enhancing usability and satisfaction.

User requirement

Identification of Users (Stakeholders)

1. **Airline Staff:** Including reservation agents, ticketing staff, and customer service representatives.
2. **Passengers:** Individuals booking flights through the system.
3. **Administrators:** System administrators responsible for managing user accounts, system settings, and overall system functionality.
4. **Management:** Executives and decision-makers within the airline company who require access to performance metrics, financial data, and strategic insights.

User Interviews/Observations and Surveys (Fact Gathering)

Conducted interviews with airline staff, passengers, administrators, and management to gather insights into their needs, pain points, and preferences regarding the reservation

management system. Additionally, observational studies were conducted to observe user interactions with existing systems and identify areas for improvement. Surveys were distributed to a representative sample of users to collect quantitative data and validate findings from interviews and observations.

Use Case Analysis

Developed use cases to capture the functional requirements of the system from the perspective of different user roles. Use cases were documented to describe various scenarios, including flight booking, reservation modification, ticket cancellation, user authentication, system administration, and reporting.

Persona Development

Created personas representing typical users of the system, including reservation agents, frequent flyers, system administrators, and airline executives. Each persona was based on insights gathered from user interviews, observations, and surveys, allowing for a deeper understanding of user goals, behaviors, and pain points.

Requirements Prioritization

Prioritized requirements based on their impact on user satisfaction, system functionality, and business objectives. High-priority requirements were identified through stakeholder consultations, user feedback, and strategic alignment with the airline company's goals. This prioritization process ensured that essential features and functionalities were addressed first, while less critical requirements were scheduled for subsequent iterations.

Functional/Non-functional Requirements

Documented functional requirements specifying the system's behavior and capabilities, such as flight search, booking management, payment processing, and reporting functionalities. Non-functional requirements were also identified, including performance, security, scalability, usability, and accessibility criteria. These requirements guided the design, development, and testing of the system to ensure it met user expectations and industry standards.

Validation and Verification of the Findings

Validated requirements through user testing, prototyping, and feedback sessions to ensure alignment with user needs and expectations. Verification activities were conducted to verify that the implemented system functionalities met the specified requirements and quality standards. Continuous validation and verification processes were employed throughout the development lifecycle to address any discrepancies and refine the system iteratively.

Functional Specification

Requirement ID	Requirement Description	Dependencies	Acceptance Criteria	Priority
FR001	User Authentication	None	Users must be able to securely log in to the system using unique credentials.	High
FR002	Flight Search	None	Users should be able to search for available flights based on specified criteria (e.g., date, destination).	High
FR003	Booking Management	FR001, FR002	Users should be able to book flights, select seats, and make payments securely.	High
FR004	Reservation Modification	FR003	Users should be able to modify existing reservations, including changing flight dates or passenger information.	High
FR005	Ticket Cancellation	FR003	Users should be able to cancel booked tickets and receive refunds according to the airline's cancellation policy.	High
FR006	User Profile Management	FR001	Users should be able to update their personal information, including contact details and frequent flyer numbers.	Medium
FR007	Reporting and Analytics	None	Administrators should have access to comprehensive reports and analytics on booking trends, revenue, and more.	Medium

FR008	System Administration	FR001	System administrators should have the ability to manage user accounts, system settings, and access permissions.	Medium
FR009	Payment Processing	FR003	The system should securely process payments using various payment methods (e.g., credit cards, PayPal).	High
FR010	Seat Selection	FR003	Users should be able to select preferred seats during the booking process, subject to availability.	Medium
FR001	User Authentication	None	Users must be able to securely log in to the system using unique credentials.	High

Technical Specifications

User Interface Design (UI/UX)

UI Design:

- The user interface will be designed with a focus on simplicity, intuitiveness, and consistency across different devices and platforms. It will feature modern design elements, clear navigation paths, and interactive components to enhance user engagement.

UX Design:

- User experience will be prioritized to ensure seamless interactions, efficient task completion, and minimal cognitive load for users. Iterative user testing and feedback loops will be employed to refine the UI/UX design throughout the development process.

Data Model (DFD/DB Design)

Data Flow Diagram:

- A DFD will be created to illustrate the flow of data within the system, including processes such as flight booking, payment processing, and reporting.

Database Design:

- The database schema will be designed to efficiently store and retrieve data related to flights, reservations, users, payments, and system configurations. Normalization techniques will be applied to ensure data integrity and minimize redundancy.

System Architecture

Microservices Architecture: The system will be designed using a microservices architecture to promote modularity, scalability, and maintainability. Each functional module, such as flight booking, payment processing, and reporting, will be developed as a separate microservice with its own database and APIs for communication.

Containerization: Docker containers will be used to package each microservice along with its dependencies, making it easy to deploy and manage across different environments.

Orchestration: Kubernetes will be utilized for container orchestration to automate deployment, scaling, and management of microservices.

Deployment and Infrastructure

Cloud Deployment: The system will be deployed on a cloud infrastructure provider such as AWS, Azure, or Google Cloud Platform to leverage scalability, reliability, and cost-effectiveness.

Continuous Integration/Continuous Deployment (CI/CD): CI/CD pipelines will be set up to automate the build, test, and deployment processes, ensuring rapid and reliable delivery of updates to the production environment.

Monitoring and Logging: Tools such as Prometheus and Grafana will be used for monitoring system performance, resource utilization, and error tracking, allowing for proactive identification and resolution of issues.

Testing Strategy

Unit Testing: Each microservice will be thoroughly unit tested using frameworks like JUnit or pytest to verify the functionality of individual components.

Integration Testing: Integration tests will be conducted to ensure that different microservices interact correctly with each other and external dependencies.

End-to-End Testing: End-to-end tests will be performed to validate the system as a whole, simulating real-world user scenarios and verifying system behavior across all layers.

Dependencies (API/libraries/OS/etc.)

APIs:

Integration with external APIs for services such as payment gateways, flight booking and aggregators.

Libraries/Frameworks:

Utilization of libraries and frameworks such as Spring Boot for backend development, React.js for front-end development, and use MySQL Work bench for creating database.

Operating System:

The system will be developed to run on Microsoft Windows-based operating systems, leveraging the following advantages:

- **Compatibility:** Microsoft Windows is widely used in enterprise environments, ensuring compatibility with existing infrastructure and software.
- **Developer Familiarity:** Many developers are familiar with Windows-based development environments, facilitating easier setup and configuration.
- **Tool Support:** Windows offers robust support for development tools and IDEs commonly used in the industry, such as Visual Studio and MySQL Workbench.

- **Integration:** Seamless integration with other Microsoft technologies and services, such as Azure cloud services and Active Directory, provides additional flexibility and capabilities.

Development Tools:

- **STS 4:** Spring Tool Suite 4 will serve as the primary integrated development environment (IDE) for backend development. Its features, including project scaffolding, code generation, and Spring Boot integration, streamline the development process within the Windows environment.
- **VS Code:** Visual Studio Code will be utilized for frontend development with React.js. Its lightweight yet powerful editor, extensive plugin ecosystem, and built-in Git integration make it well-suited for web development on Windows.
- **MySQL Workbench:** MySQL Workbench, available for both Windows and Linux, will be used as the graphical user interface (GUI) for database design, modeling, and administration. Its intuitive interface and rich feature set facilitate efficient database management and development on Windows.

Backend Framework:

- **Spring Boot:** The backend of the system will be developed using Spring Boot, a popular Java framework for building microservices-based applications. Spring Boot's convention-over-configuration approach, embedded application server, and extensive ecosystem of libraries simplify the development of robust and scalable backend services on Windows.

Frontend Framework:

- **React.js:** The frontend of the system will be built using React.js, a JavaScript library for building user interfaces. React's component-based architecture, virtual DOM, and declarative approach enable the creation of interactive and responsive web applications with minimal boilerplate code on Windows.

Database:

- **MySQL:** The system's data will be stored in a MySQL relational database management system (RDBMS). MySQL's performance, scalability, and reliability make it a suitable choice for handling the transactional data associated with airline reservations, user accounts, and system configurations on Windows.

Work breakdown project timeline

Project Phases and Deliverables

1. **Planning Phase:**
   Deliverable: Project plan document outlining objectives, scope, resources, and timeline.
2. **Requirement Gathering:**
   Deliverable: Requirements document detailing user needs, functionalities, and constraints.
3. **Design Phase:**
   Deliverable: UI/UX design mockups, database schema, system architecture diagrams.
4. **Development Phase:**
   Deliverable: Backend and frontend implementation, database setup, integration testing.
5. **Testing Phase:**
   Deliverable: Test plans, test cases, bug reports, and resolved issues.
6. **Deployment Phase:**
   Deliverable: Deployed system to production environment, user documentation, training materials.

Tasks Breakdown at Each Phase

Planning Phase:

- Define project scope and objectives.
- Identify project stakeholders and resources.
- Develop project plan document.

Requirement Gathering:

- Conduct user interviews and surveys.
- Analyze user requirements and document findings.

Design Phase:

- Create UI/UX design mockups.
- Design database schema.
- Define system architecture.

Development Phase:

- Set up development environment.
- Implement backend functionalities using Spring Boot.
- Develop frontend components using React.js.
- Integrate backend with frontend and database.
- Conduct integration testing.

Testing Phase:

- Develop test plans and test cases.
- Execute functional, integration, and regression tests.
- Report and resolve identified issues.

Deployment Phase:

- Deploy the system to production environment.
- Perform user acceptance testing (UAT).
- Prepare user documentation and training materials.

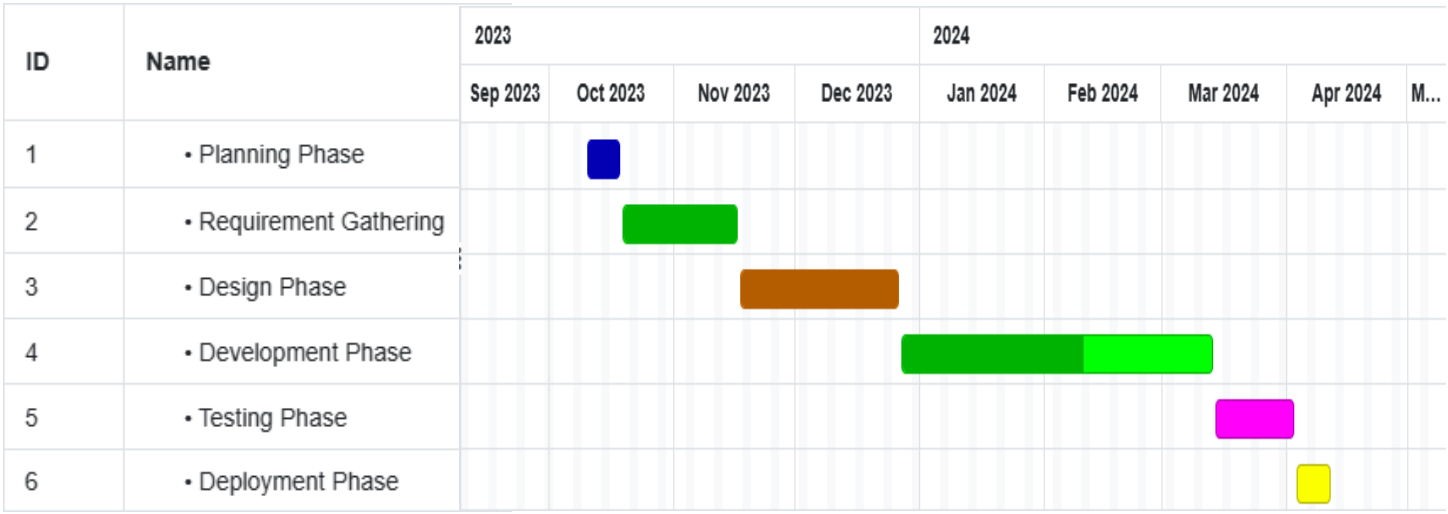
Project Timeline

- **Planning Phase:** 1 week
- **Requirement Gathering:** 3 weeks
- **Design Phase:** 4 weeks
- **Development Phase:** 8 weeks
- **Testing Phase:** 3 weeks
- **Deployment Phase:** 2 weeks

Resource Allocation (Work Distribution Among Members)

- **Backend Developers:** 2 members
- **Frontend Developers:** 2 members
- **Database Administrators**: 1 member
- **Tester:** 1 member

Gantt Chart



Current Status

1. **Project Timeline:**
   The project is in the initial phases, having completed the project initiation, requirements gathering, and system design. The development phase is ongoing, focusing on implementing core system features and conducting iterative development cycles. The timeline is on track based on the outlined milestones.

2. **Progress Update:**
   Progress has been steady, with the team successfully defining project objectives, assembling the project team, and conducting user interviews for requirements gathering. The system architecture has been created, and development is underway.

3. **Key Achievements:**
   Key achievements include defining objectives, assembling the project team, securing project funding, conducting user interviews, and completing the system design phase.

4. **Work Completed:**
   Project initiation, requirements gathering, and system design phases have been completed. The team has created a detailed system architecture, providing a foundation for the development phase.

5. **Work in Progress:**
   The development phase is currently in progress, focusing on implementing core system features. Testing and quality assurance activities are anticipated to begin shortly.

6. **Current Issues, Risks & Mitigation Strategy:**
   No major issues have been reported. Identified risks, including technical, operational, and project management aspects, are being actively monitored. The mitigation strategies outlined in the proposal are in place, such as detailed technology evaluation, robust security measures, change management strategies, and contingency plans for resource shortages.

7. **Next Steps and Resource Allocation:**
   The next steps involve completing the development phase, moving into testing and quality assurance, followed by deployment and ongoing support. Resources are allocated to ensure the timely completion of each phase, with a focus on adhering to the outlined project timeline.

8. **Introduction and Purpose:**
   The project acknowledges the need for a modern Flight Ticket Management System due to the evolving demands of the aviation industry. The current ticket booking system is outdated, lacking automation and user-friendly features.

9. **Objectives:**
   Clear objectives are established, focusing on simplifying the ticket booking process, automating ticket management, enhancing data security, improving customer experience, and increasing operational efficiency.

10. Scope:

The scope encompasses user-friendly interfaces, integration with payment gateways, real-time flight information, customer accounts, integration with existing systems, and robust security features.

11. Project Deliverables:

The project will deliver a fully functional FTMS, user documentation, security protocols, maintenance and support plan, comprehensive testing reports, and project closure documentation.

12. Budget:

Various cost components are identified, including development and integration costs, hardware and software costs, human resources, training and documentation costs, security measures, quality assurance and testing, and contingency expenses.

13. Risk Assessment:

The project identifies and assesses technical, operational, project management, and external risks. Mitigation strategies are outlined for each identified risk to ensure project success.

- The next steps would involve moving into the execution phase, starting with project initiation, requirements gathering, and subsequent development. Regular monitoring, risk management, and adherence to the proposed timeline will be crucial for successful implementation.
- Overall, the project is progressing according to plan, with a proactive approach to risk management and a clear path for the upcoming phases. The team remains committed to delivering a fully functional Flight Ticket Management System that meets the outlined objectives and scope requirements.

Conclusion/Summery

In conclusion, the development of the airline reservation management system presents a comprehensive solution to streamline the booking process, enhance user experience, and improve operational efficiency for airline companies. Throughout the project, several key points emerged, emphasizing the importance of thorough planning, stakeholder collaboration, and technical expertise.

The project began with meticulous requirement gathering, involving user interviews, observations, and surveys, which provided valuable insights into user needs and preferences. This phase was crucial in defining the scope and objectives of the system, ensuring alignment with stakeholders' expectations.

The design and development phases focused on creating an intuitive user interface, robust backend architecture, and scalable database schema. Leveraging tools such as Spring Boot for backend development and React.js for frontend development facilitated the implementation of core functionalities while maintaining flexibility and modularity.

Testing and deployment phases were integral to ensuring the quality and reliability of the system. Rigorous testing, including functional, integration, and user acceptance testing, helped identify and address issues early in the development lifecycle, resulting in a more stable and user-friendly product.

Looking ahead, several recommendations and suggestions can be made for potential deviations or improvements. Continuous stakeholder engagement and feedback loops are essential to accommodate changing requirements and evolving user needs. Additionally, implementing monitoring and analytics tools can provide valuable insights into system performance and user behavior, enabling data-driven decision-making and continuous optimization.

Furthermore, it's important to reflect on lessons learned throughout the project. Effective communication, collaboration, and project management are critical for success. Adapting to challenges and embracing iterative development methodologies promote agility and resilience in the face of uncertainty. By fostering a culture of learning and continuous

improvement, future projects can benefit from the experiences and insights gained during the development of the airline reservation management system.

In summary, the project has demonstrated the value of a systematic approach to software development, from requirement gathering to deployment. By leveraging technology, teamwork, and best practices, the airline reservation management system stands poised to deliver significant benefits to both airline companies and their customers.

-END-