PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

PES UNIVERSITY

Department of Computer Science and Engineering

UE21CS341A: Software Engineering

ASSIGNMENT / PROJECT GUIDELINES

Acknowledgment: Authors of the document

Purpose: As part of the software engineering course, students are expected to do a project incorporating all stages of software development life cycle. As the units will be taught in the classes, phase wise inclusion needs to be done.

Outcomes: Team Details, Project Title, Synopsis, all intermediate SE(SDLC) Documents and a complete Project Report (Formats attached for reference).

Evaluation and Complete Process: The details of how this project must be done and evaluated are as described below:

Guidelines:

1: Project team must identify an application case study ex: e-ticket/Banking/Recruitment System/Trading System.

2: Team must prepare a synopsis to enlist all the features of the chosen project / software application.

3: Prepare a document consisting of requirements, planning, design, implementation, and testing details.

4: Project Report consisting of all the relevant documents must be submitted with all results.

Project Team:

The project will be done by a group of 4 students. One or two teams in a class may be an exception with prior approval of the subject teacher. Teams must be among students belonging to the same section. Each team member needs to complete one full functionality. Weekly evaluations will be done based on individual performance and contribution made towards the project. A mid-point review of the project and corresponding document submission is mandatory. This will carry a portion of the total marks awarded for the project.

PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

UE20CS341A SE - Project-Guidelines

Software Requirements Specification

for

FLIGHT BOOKING SYSTEM

Version 1.0 approved

Prepared by

Sai Lithish Degapudi(PES2UG21CS456)

Sachin Ramesh Kulkarni(PES2UG21CS449)

Sai Rahul Reddy Kona(PES2UG21CS458)

PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

S Ramanujan (PES2UG21CS441) -----> TEAM - 7

PES UNIVERSITY ECC

05-09-2023

PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Table of Contents

PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Revision History

Name	Date	Reason For Changes	Version

Introduction

Purpose

The purpose of this document is to outline the software requirements for a Flight Booking System website, defining the features, functionalities, and constraints of the system.

The Flight Booking System encompasses the following key components and functionalities:
1)Flight Search and Booking
2)Flight Management
3)Customer Support and Notifications
4)Reporting and Analytics
5)Integration with External Systems: The Flight Booking System may integrate with external systems such as airline reservation systems, payment gateways, and third-party services for additional features and functionalities.

Intended Audience

This document is intended for software developers, testers, project managers, and stakeholders involved in the development of the Flight Booking System.
The document is organized in the following way:

Introduction:Purpose,Scope and Definition

Overall Description:Product Perspectives and Functions,User Classes and Characteristics,Operating Environment, Assumptions and Dependencies

Specific Requirements:Functional and Non-Functional Requirements,External Interface Requirements,System Features and Use Cases.

System Models:Data Models,ER Diagram , State transition Diagram.

Appendices:Additional Information, such as reference materials, legal considerations, or technical documentation.

Index and References:An index to help readers quickly locate specific information in the document.

Product Scope

The Flight Booking System will allow users to search for flights, make reservations, and manage their bookings. It will provide a seamless and user-friendly experience for booking domestic and international flights.

References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>
TO BE COMPLETED

Overall Description

Product Perspective

Origin: The idea for this flight booking system originated from the increasing demand for online travel services and the need for a user-friendly, efficient, and secure platform for booking flights.
Self-Contained Product: This flight booking system is designed as a standalone product, and it does not rely on or replace any existing systems. It is intended to cater to a broad range of users, including individual travelers, travel agencies, and airlines.

External Interfaces: The flight booking system interacts with various external entities, including:
Airline Systems: The system may connect to airline databases or APIs to retrieve real-time flight availability, schedules, and pricing information.
Payment Gateways: To facilitate secure online payments, the system interfaces with payment gateway services.

##External Data Sources: Weather information, airport details, and geographic data may be sourced from external databases or APIs.##

User Communication: The system communicates with users via email or SMS for booking confirmations, notifications, and support.

User Interface: The system provides a user-friendly web-based interface for users to interact with. Users can access the system via web browsers on desktop or mobile devices.

Product Functions

User Registration and Authentication: Allow users to create accounts and log in securely.

Flight Search: Enable users to search for flights based on criteria like departure/arrival locations, dates, class, and passengers.

Booking: Facilitate flight selection, seat reservation (if available), and booking with real-time price updates.

Payment Processing: Securely handle payments through various methods (credit card, PayPal, etc.).

Booking Confirmation: Generate booking confirmations and send notifications to users.

User Account Management: Let users manage their profiles and booking history.

Customer Support: Provide access to customer support via various communication channels.

Admin Functions: Allow administrators to manage flights, user accounts, and resolve issues.

User Classes and Characteristics

These user classes are defined based on their roles, privileges, and interaction with the system. Here are some of the key user classes, along with their pertinent characteristics:
Passengers:

Occasional users.

Typically, limited privileges such as booking flights, managing personal profiles, and making payments.

May vary in age, language preferences, and accessibility needs.

Airline Staff (Agents and Customer Service):

Frequent users with moderate to high technical expertise.
Access to advanced booking and reservation management functions.
Need to handle customer inquiries, resolve issues, and provide assistance.
Must adhere to strict security and privacy protocols.

Administrators:

Expert users with in-depth knowledge of the system.
Full control over system configuration and user management.
Responsible for system maintenance, updates, and security.

System Administrators (IT):

Responsible for system deployment, maintenance, and performance optimization.
Need access to backend systems and databases.
Focused on ensuring system reliability and security.

Third-Party Integrators (API Users):

Developers or technical experts.
Utilize APIs to integrate the Flight Booking System with external applications or services.

Frequent Flyers and Loyalty Program Members:

Passengers who frequently use the system.
May have access to special features related to loyalty programs, rewards, and miles.
Require a streamlined experience for quick booking and managing their accounts.

Call Center Operators:

Moderate technical expertise.
Assist passengers with phone reservations and inquiries.
Need access to the system's booking and customer support functions.

Web Server Software:

Operating Environment

Our software will have the following environemnets:
Web Technologies and APIs:HTML ,CSS
Programming Languages and Frameworks:We are using Python Programming Language
Database Management System (DBMS):MySQL for databases

Design and Implementation Constraints

1)Regulatory Compliance:

Constraint: The Flight Booking System must adhere to aviation industry regulations, data protection laws (e.g., GDPR), and payment card industry standards (e.g., PCI DSS).
Impact: Compliance requirements will influence data handling, security measures, and user consent procedures. Regular audits and reporting may be necessary.

2)Hardware and Performance Limitations:

Constraint: The system must perform efficiently on the chosen hardware platform, which may have limitations in terms of processing power, memory, and storage.
Impact: Developers must optimize code and database queries to ensure acceptable performance. Load testing and performance tuning are crucial.

3)Integration with External Systems:

Constraint: The Flight Booking System must interface with external systems, such as airline reservation systems and payment gateways.
Impact: Developers must implement and maintain reliable integration points, ensure data consistency, and handle potential downtime or communication issues.

4)Technology Stack:

Constraint: The organization may have predetermined technology choices for web servers, programming languages, databases, and frameworks.
Impact: Developers must work within the specified technology stack, which can impact development speed and scalability options.

5)Security Considerations:

Constraint: Stringent security measures are essential due to the sensitive nature of passenger data and payment information.
Impact: Developers must implement encryption, access controls, authentication, and authorization mechanisms. Security audits and regular updates are necessary.

6)Design Conventions and Coding Standards:

Constraint: The organization may have established design conventions, coding standards, and development practices that must be followed.
Impact: Developers must adhere to these standards, which can affect code readability, maintainability, and collaboration among development teams.

7)Parallel Operations and Concurrency:

Constraint: The system should support concurrent booking and reservation activities without data conflicts.
Impact: Developers must implement locking mechanisms, database transactions, and concurrency control to prevent data inconsistencies.

8)Maintenance Responsibility:

Constraint: The customer's organization may be responsible for maintaining the delivered software.
Impact: Developers should provide clear documentation, support, and training to facilitate ongoing maintenance and updates by the customer's team.

2.6 Assumptions and Dependencies

1)Third-Party APIs and Services:

Assumption: The availability, reliability, and performance of third-party APIs (e.g., airline reservation systems, payment gateways, geolocation services) will meet system requirements.

Impact: If third-party services experience downtime or performance issues, it can disrupt system functionality and user experience. Contingency plans may be needed.

2)Data Availability:

Assumption: Assumption that data from airlines, including flight schedules, seat availability, and pricing, is available through reliable data feeds.

Impact: If data sources change or become unreliable, it can affect real-time flight information and booking accuracy. Data quality assurance measures should be in place.

3)Internet Connectivity:

Assumption: Users will have consistent and reliable internet connectivity when accessing the system.

Impact: Poor internet connectivity may lead to a subpar user experience. Offline access options or graceful degradation of features may be needed.

4)Regulatory Changes:

Assumption: Assumption that aviation industry regulations and data protection laws will remain stable during system development.

Impact: Changes in regulations can require significant system modifications and compliance efforts, potentially impacting timelines and costs.

5)User Behavior and Load Assumptions:

Assumption: Assumptions about user traffic patterns, peak booking times, and system load may influence scalability and performance planning.

Impact: If user behavior deviates significantly from assumptions, it can lead to underperforming systems during peak times or over-provisioning of resources during low traffic periods.

6)Data Migration Assumptions:

Assumption: Assumptions about the ease and accuracy of migrating existing data (if any) into the new system.

Impact: Data migration challenges can affect project timelines and data integrity if not properly addressed.

External Interface Requirements

User Interfaces

Passenger-Facing User Interface:

Characteristics: A user-friendly web interface for passengers to search, book, and manage flights. It follows modern web design principles, offers easy navigation, and provides clear error feedback.

Airline Staff and Administrator Interface:

Characteristics: Role-specific dashboards for airline staff and administrators to manage flights, reservations, and configurations. Designed for efficiency, it may include keyboard shortcuts and consistent error handling.

Mobile Interfaces:

Characteristics: Mobile-optimized interfaces for smartphones and tablets, featuring responsive design, touch-friendly navigation, and user-friendly buttons. Error messages are adapted for mobile touch screens.

Software Interfaces

1)Database Interfaces:

Database System: MySQL
Purpose: The Flight Booking System interacts with the MySQL database to store and retrieve data, including flight schedules, passenger details, reservations, and payment records.

Booking data: Passengers' flight choices, seat selections, and payment information.

Flight details: Information about available flights, seat availability, and pricing.

Passenger information: Data related to passenger profiles and reservations.

2)Payment Gateway Interfaces:

Payment Gateways: Various payment gateways (e.g., Stripe, PayPal, and others).
Purpose: Facilitating secure payment processing for flight reservations.

Services Needed: Integration with payment gateway APIs for processing transactions securely.

Communication: The system communicates with payment gateways using their respective APIs, adhering to the protocols and security standards provided by each gateway.

3)External APIs:

APIs: Airline reservation systems, geolocation services (e.g., Google Maps), and other external services.

Purpose: Retrieving real-time flight data, including schedules and seat availability, and accessing geolocation data for mapping and route planning.

Communication: The system communicates with external APIs via RESTful requests or other standard protocols as specified by each service provider. Detailed API documentation is referenced for protocols.

4)Third-Party Libraries and Frameworks:

Libraries and Frameworks: For example, a specific web framework (e.g., Django, Ruby on Rails).

Purpose: Utilizing third-party libraries and frameworks for web application development.

Communication: The system integrates with third-party libraries and frameworks according to their respective documentation and usage guidelines.

Communications Interfaces

Web Communication: Uses HTTPS for user interactions, ensuring data security and low-latency responses.

Email Communication: Sends HTML-formatted emails for booking confirmations and notifications.

Network Server Communications: Communicates with external services via RESTful APIs over HTTPS for real-time synchronization.

Electronic Forms: Captures user data through structured submissions over secure HTTPS connections.

Hardware Interfaces: Connects to physical devices like barcode scanners, using custom or standard protocols.

Communication Security and Encryption: Ensures data protection through encryption (e.g., HTTPS) and secure hashing.

Data Transfer Rates and Synchronization: Optimizes data transfer for high volumes and real-time updates with external services.

Analysis Models

Actors:

Customer (to search, book, and manage flights)

Administrator (to manage flights, user accounts, and bookings)

Use Cases:

Search for Flights

Book Flight

Cancel Booking

Manage User Account

Manage Flight (for administrators)

Generate Booking Confirmation

ER diagram

Entities:

User

Flight

Booking

Payment

Airport

Relationships:Users can make bookings, and each booking is associated with a flight and payment.

Flights depart from and arrive at airports.

System Features

1. User Registration and Authentication:
Allow users to create accounts.
Authenticate users securely.
Provide options for password recovery and account management.

2. Flight Search and Booking:
Enable users to search for available flights based on criteria such as destination, date, and class.
Display a list of available flights with details like price, departure time, and airline.
Allow users to select flights and book tickets.Manage seat availability and reservations.

3. Payment Processing:

Integrate payment gateways to facilitate secure online payments.
Support multiple payment methods (credit/debit cards, PayPal, etc.).

4. User Profiles and Preferences:

Allow users to create and manage profiles with personal information.
Provide options to set preferences like seat preferences, and frequent flyer numbers.
Save booking history and upcoming trips.

5. Notifications and Alerts:

Send confirmation emails and SMS notifications upon successful booking.
Provide real-time flight status updates and alerts for delays or cancellations.
Notify users about upcoming flights and check-in reminders.

6. Booking Management:

Enable users to modify or cancel bookings.
Implement refund and cancellation policies.
Allow for seat upgrades or add-ons like extra baggage.

7. Loyalty Programs and Rewards:

Integrate frequent flyer programs for earning and redeeming rewards.
Display loyalty points, tier status, and benefits to users.

8. Admin and Staff Management:

Provide administrative interfaces for managing flights, airlines, and pricing.
Allow staff to process reservations, handle customer support, and manage bookings.

9. Reporting and Analytics:

Generate reports on booking trends, revenue, and user demographics.
Utilize analytics to optimize pricing, routes, and user experience.

Integration with External Services:

Integrate with external systems for weather updates, visa information, and airport details.
Connect with third-party travel insurance providers.

Customer Support and Feedback:

Provide a customer support portal with live chat, email, and phone support.
Collect user feedback and reviews for continuous improvement.

Marketing and Promotions:

Support promotional codes and discounts.
Implement marketing campaigns and newsletters for user engagement.

Compliance and Regulatory Requirements:

Ensure compliance with aviation regulations and industry standards.
Maintain records for auditing and compliance purposes.

Flight Booking and Reservation Management:

5.1.1 Description and Priority

1. User Registration and Authentication:
Description: This feature allows users to create accounts and log in securely.
Priority: High
Priority Components: Benefit - 8, Penalty - 7, Cost - 5, Risk - 6

2. Flight Search and Booking:
Description: Users can search for and book flights.
Priority: High
Priority Components: Benefit - 9, Penalty - 7, Cost - 6, Risk - 6

3. Payment Processing:
Description: Facilitates secure online payments.
Priority: High
Priority Components: Benefit - 9, Penalty - 8, Cost - 7, Risk - 7

4. User Profiles and Preferences:
Description: Users can manage personal information and preferences.
Priority: Medium
Priority Components: Benefit - 7, Penalty - 6, Cost - 5, Risk - 5

5. Notifications and Alerts:
Description: Sends real-time updates and flight alerts.
Priority: Medium
Priority Components: Benefit - 7, Penalty - 6, Cost - 5, Risk - 5

6. Booking Management:
Description: Allows users to modify or cancel bookings.
Priority: Medium
Priority Components: Benefit - 6, Penalty - 6, Cost - 6, Risk - 5

7. Loyalty Programs and Rewards:
Description: Integrates frequent flyer programs.
Priority: Medium
Priority Components: Benefit - 6, Penalty - 5, Cost - 5, Risk - 4

8. Admin and Staff Management:
Description: Provides administrative interfaces for staff.
Priority: High
Priority Components: Benefit - 8, Penalty - 7, Cost - 6, Risk - 7

9. Reporting and Analytics:
Description: Generates reports on system performance and user data.
Priority: Medium
Priority Components: Benefit - 7, Penalty - 6, Cost - 6, Risk - 5

Accessibility and Multilingual Support:
Description: Ensures the system is accessible and supports multiple languages.
Priority: Medium
Priority Components: Benefit - 6, Penalty - 5, Cost - 5, Risk - 4

Customer Support and Feedback:
Description: Provides customer support channels and collects feedback. - Priority: Medium
Priority Components: Benefit - 7, Penalty - 6, Cost - 6, Risk - 5

Marketing and Promotions:
Description: Supports promotional codes and marketing campaigns.
Priority: Medium
Priority Components: Benefit - 6, Penalty - 6, Cost - 5, Risk - 4

Compliance and Regulatory Requirements:
Description: Ensures compliance with aviation regulations.
Priority: High
Priority Components: Benefit - 8, Penalty - 7, Cost - 6, Risk - 7

5.1.2 Stimulus/Response Sequences

User Registration:
Stimulus: User selects the "Sign Up" or "Register" option on the system's home page.
Response: The system displays the registration form, prompting the user to input their details (e.g., name, email, password).

Stimulus: User enters their information and clicks the "Submit" button. Response: The system validates the input data, checking for valid email format and password complexity.

Stimulus: User's input data passes validation.
Response: The system creates a new user account and sends a confirmation email.

Stimulus: User receives the confirmation email and clicks the verification link. Response: The system verifies the email and activates the user account, displaying a confirmation message to the user.

User Login:

Stimulus: Registered user selects the "Log In" option on the system's home page. Response: The system presents a login form requesting the user's email and password.

Stimulus: User enters their login credentials and clicks the "Log In" button. Response: The system validates the credentials against the stored user data.

Stimulus: Provided credentials are correct.
Response: The system grants access to the user's account, redirecting them to their dashboard or a relevant landing page.

Stimulus: Provided credentials are incorrect.
Response: The system displays an error message and allows the user to retry.

Password Recovery:

Stimulus: User clicks on the "Forgot Password" or "Reset Password" option. Response: The system guides the user through the password recovery steps, typically starting with a request for their registered email address.

Stimulus: User provides their email address and submits the request. Response: The system sends an email with a password reset link to the provided address.

Stimulus: User receives the email and clicks the reset link.
Response: The system directs the user to a secure page where they can create a new password.

5.1.3 Functional Requirements

REQ-1: User Registration

1)The system shall provide a "Sign Up" or "Register" button on the home page.

2)The registration form shall include fields for the user to input their name,email address, and password.

3)The system shall validate the email address for correct format.

4)Password validation shall ensure a minimum length of 8 characters and contain at least one uppercase letter, one lowercase letter, one digit, and one special character.

5)Upon successful validation, the system shall create a new user account with the provided information.

6)The system shall send a verification email to the provided email address with a unique verification link.

7)In case of registration failure (e.g., invalid email, password requirements not met), the system shall display appropriate error messages and allow the user to correct the input.

8)The system shall provide a time-limited window (e.g., 24 hours) for the user to click the verification link in the email.

9)If the user clicks the verification link within the specified time, the system shall verify the email and activate the user's account.

REQ-2: User Login

10. The system shall provide a "Log In" option on the home page for registered users.

11)The login form shall include fields for the user to input their registered email address and password.

12)The system shall verify the provided email and password against the stored user data.

13)Upon successful login, the system shall grant access to the user's account and redirect them to their dashboard or a relevant landing page.

14)In case of login failure (e.g., incorrect email, password mismatch), the system shall display an error message and allow the user to retry.

REQ-3: Password Recovery

15. The system shall provide a "Forgot Password" or "Reset Password" option for users who have forgotten their password.

16)Users shall be prompted to enter their registered email address for password recovery.

17)Upon receiving the password recovery request, the system shall send an email to the provided address with a password reset link.

18)The system shall include a time-limited window (e.g., 24 hours) for the user to use the password reset link.

19)When the user clicks the password reset link within the specified time, the system shall direct them to a secure page to create a new password.

20)The system shall ensure that the new password meets the same complexity requirements as during registration.

PES
UNIVERSITY
BENGALU
RU

Department of Computer Science and Engineering

Flight Booking and Reservation System - Non-Functional Features.

Performance Requirements

1. Response Time for Flight Search:

Requirement: The system shall respond to flight search queries within 2 seconds for 90% of searches.

Rationale: Fast response times are crucial for providing a smooth user experience when searching for flights. Users expect quick results when planning their travel.

2. Booking Transaction Processing Time:

Requirement: The system shall complete the flight booking transaction, including payment processing, within 5 seconds for 90% of transactions.

Rationale: Fast booking transaction times are essential to prevent user frustration and ensure a high conversion rate for flight bookings.

3. Scalability:

Requirement: The system shall be designed to scale horizontally to accommodate a 50% increase in concurrent user traffic during peak periods.

Rationale: Scalability ensures that the system can handle increased user loads during busy travel seasons without degrading performance.

4. Availability:

Requirement: The system shall have an uptime of at least 99.9% over any given month, including maintenance windows.

Rationale: High availability is crucial to prevent service disruptions and ensure users can access the system at any time.

5. Error Handling Response Time:

Requirement: The system shall display appropriate error messages, such as an invalid input during flight search or booking within a specific time limit.

Rationale: Swift error handling helps users quickly correct mistakes and continue using the system.

6. Load Testing:

Requirement: The system shall undergo regular load testing to ensure it can handle double the expected peak load without significant performance degradation.

Rationale: Load testing identifies potential bottlenecks and performance issues, allowing for proactive optimization.

Safety Requirements

1. Data Security:

Requirement: The system must implement robust data encryption protocols (e.g., SSL/TLS) to safeguard sensitive user information during transmission.

Rationale: Protecting user data during online transactions is essential to prevent data breaches and identity theft.

2. Compliance with Aviation Regulations:

Requirement: The system must adhere to all relevant aviation regulations and standards.

Rationale: Compliance with aviation regulations ensures the safety and security of flight operations and passenger information.

3. Privacy Protection:

Requirement: The system shall comply with data privacy regulations to protect user privacy and personal information.

Rationale: Safeguarding user privacy is essential to maintain trust and comply with legal requirements.

4. Error Handling and Redundancy:

Requirement: The system shall implement error handling mechanisms and redundancy measures to mitigate service interruptions and prevent data loss in case of system failures.

Rationale: Redundancy and error handling help maintain system availability and data integrity.

5. User Authentication:

Requirement: User authentication must be secure, with password policies that encourage strong passwords and mechanisms to prevent unauthorized access.

Rationale: Secure authentication prevents unauthorized access to user accounts and protects user data.

Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

Also include Domain requirements here.

Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Field Layouts

An Excel sheet containing field layouts and properties/attributes and report requirements.

Sample sheet with information required to register the customer

Field	Length	Data Type	Description	Is Mandatory
Account Number	16	Numeric		Y
ISFC code	11	Alphanumeric		Y
Card Amount	20	Numeric		Y
Mandate Start Date	8	Date	Date of Mandate Registration	N
Mandate End Date	8	Date	Date of Mandate Expiry	N
Status	25	Alphanumeric	Status of Registration	Y
Customer Name	60	String		Y
Reject Reason Code	4	String	Reject Reason code in case mandate is rejected	N

Sample Report Requirements: Include the fields to be included in the report

Registration Report	Transaction Report
Bank Account Number	Transaction Reference Number
ISFC Code	Bank Account Number
Bank Name	IFSC Code
Account Status	Bank Name
Account Type	Customer Name
Customer Name	Card Number
Card Number	Debit Transaction Amount
SI Start Date	Transaction Date
Status	Status
Remarks	Debit Attempt Number
	Remarks

Appendix C: Requirement Traceability Matrix

Sl. No	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID
1	REQ-1	User Registration					
2	REQ-2	User Login					

Format 3

Project Plan Document

1)Project Lifecycle Selection:

For this software engineering project(Flight Booking System), a widely accepted lifecycle model is the Agile methodology. Agile is chosen for its flexibility, adaptability to changing requirements, and frequent customer feedback. This is essential in a dynamic domain like flight booking where requirements can change frequently.

2)Tools Selection:

- Planning Tool:
  - Jira or Trello for agile project management.
- Design Tool:
  - Adobe XD or Sketch for UI/UX design, UML diagrams with tools like Lucidchart.
- Version Control:
  - Git with GitHub or GitLab for code versioning.
- Development Tool:
  - IDEs like Visual Studio Code, IntelliJ IDEA, or PyCharm.
- Bug Tracking:
  - Jira or Bugzilla for tracking and managing bugs.
- Testing Tool:
  - Selenium for automated testing, JUnit for unit testing, and tools like Postman for API testing.

3)

Reuse Components:

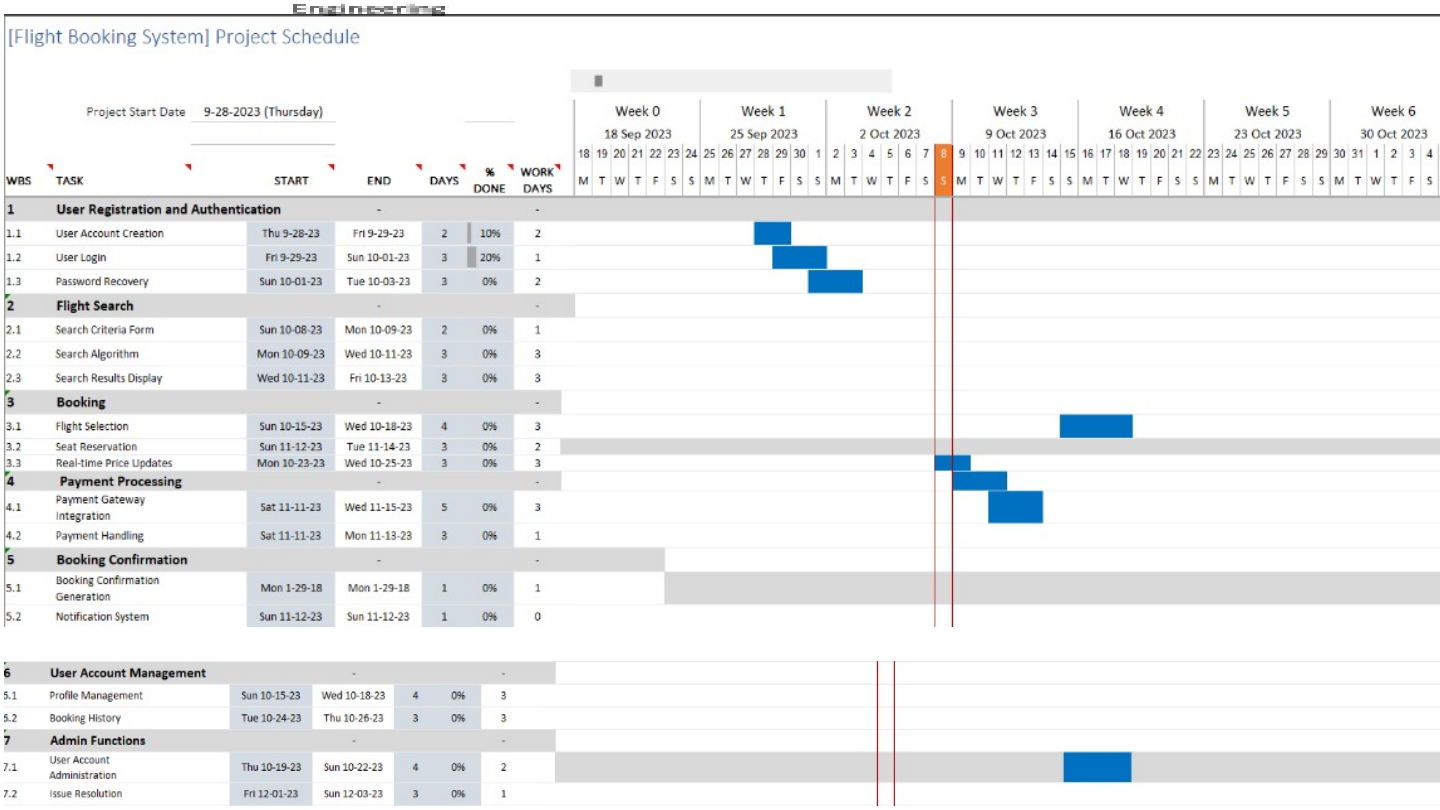
- User authentication module
- Payment processing module
- Flight search and booking engine

Build Components:

- User interface
- Database schema
- Reporting module

4: Create a WBS for the entire functionalities in detail.

here is the work breakdown structure represented as a gantt chart:



5)Here's a rough estimate in person months (PM):

- ➢ Project Initiation: 2 PM
- ➢ System Design: 4 PM
- ➢ Development: 20 PM
- ➢ Testing: 6 PM
- ➢ Deployment: 2 PM
- ➢ Maintenance and Support: 8 PM
- ➢ Total Effort Estimate: 42 PM

6)Project: Flight Booking System

| Task | Start Date | End Date | Duration |

| Project Initiation | 2023-10-15 | 2023-10-21 | 7 days |

| - Requirement Gathering | 2023-10-15 | 2023-10-17 | 3 days |

| - Stakeholder Analysis | 2023-10-18 | 2023-10-19 | 2 days |

| - Project Charter | 2023-10-20 | 2023-10-21 | 2 days |

| System Design | 2023-10-22 | 2023-10-31 | 10 days |

| - UI/UX Design | 2023-10-22 | 2023-10-26 | 5 days |

| - Database Design | 2023-10-27 | 2023-10-29 | 3 days |

| - Architecture Design | 2023-10-30 | 2023-10-31 | 2 days |

| Development | 2023-11-01 | 2023-11-20 | 20 days |

| - Front-end Development | 2023-11-01 | 2023-11-10 | 10 days|

| - Back-end Development | 2023-11-11 | 2023-11-20 | 10 days|

| Testing | 2023-11-21 | 2023-11-30 | 10 days|

| - Unit Testing | 2023-11-21 | 2023-11-24 | 4 days|

| - Integration Testing | 2023-11-25 | 2023-11-28 | 4 days|

| - User Acceptance Testing | 2023-11-29 | 2023-11-30 | 2 days|

| Deployment | 2023-12-01 | 2023-12-10 | 10 days|

| - Server Setup | 2023-12-01 | 2023-12-05 | 5 days|

| - Application Deployment | 2023-12-06 | 2023-12-10 | 5 days|

| Maintenance and Support | 2023-12-11 | 2024-01-31 | 52 days|

| - Bug Fixing | 2023-12-11 | 2023-12-31 | 21 days|

| - Feature Updates | 2024-01-01 | 2024-01-31 | 31 days|

Total Project Duration: 208 days

PES
UNIVERSITY

BENGALU
RU

Department of Computer Science and Engineering

Format 4

Test Plan Document

Instructions:

1: Prepare at least 8-10 test cases for each **implemented use case** (functional requirements) as per the below given template which should include Unit, Integration and System Test cases.

2: Carry out manual testing for all the test cases and populate the columns Actual Result and Test Result.

Template of a Test Case:

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Result	Test Result

Test Case ID : Each test case should be represented by a unique ID. To indicate test types, follow some convention like "UT_01" indicating "Unit Testing - Test Case#1."
Name of the Module : Specify the name of the **main module or sub module** being tested
Test Case Description : Specify the summary or test purpose in brief
Pre- Conditions : Any requirement that needs to be done before execution of this test case.
Test Steps : Mention all the steps in detail and specify the order in which it is to be executed.
Test Data : Input for the test case to be executed. Specify different data sets with precise values to be used as input. (create test case for both valid and invalid inputs)
Expected Results : Mention the expected results including error or precise messages that should be displayed on screen
Actual Results : After execution of test case fill this column with the result obtained
Test Result (Pass/Fail) : Mark this field as “fail” if the actual result is not same as expected result else mark as “pass”.

Sample Test Case:

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UT-01	User registration module	To test the login functionality	Access to Chrome Browser	1: Navigate to http://www.demo.com 2: Enter Username and Password 3: Click Submit	User name: PESU Student. Password : pes123	Login should be successful with “welcome message”	Login successful with “welcome message” displayed	Pass

PES
UNIVERSITY

BENGALU
RU

Department of Computer Science and Engineering

Format 5-Final Report

Department of Computer Science and Engineering

BENGALURU, KARNATAKA, INDIA.

B. TECH. (CSE) V

SEMESTER

Aug. – Dec. 2023

UE21CS341A – SOFTWARE ENGINEERING

PROJECT REPORT

ON

<PROJECT TITLE>

SUBMITTED BY

SRN -- NAME

Student Name
Student Name
Student Name

PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

TABLE OF CONTENTS