



<Flight Booking> SYSTEM REQUIREMENTS SPECIFICATION

#	Name	ID	Group
1	Rodyna Amr Fathy	20227012	G2
2	Mohamed Yasser	20226095	G2
3	Mohamed Essam	20226093	G2
4	Mohamed Khaled Abdelhakim	20225056	G2
5	Mazen Wael	20226084	G2

Project Team leader email: rodynaamr@icloud.com
Project Team leader mobile: 01060840111

TA. <Ahmed Ashraf>

TABLE OF CONTENTS

1.Introduction:	3
1.1 purpose.....	3
1.2 Scope.....	3
1.3 Stakeholders	3
1.4 Abbreviation & Glossary	4
2.System Requirements:	5
2.1 functional requirements	7
2.2 Non-functional requirements	11
3. Use Cases Tables:	16
3.1 Use case (User Registration and Authentication):	16
3.2 Use case (Search and Booking):.....	17
3.3 Use case (Notification and Confirmation):	18
3.4 Use case (Cancellation and Refund):	19
3.5 Use case (User Feedback):	20
3.6 Use case (Seat Selection):	22
4. Use Case Diagram	24

1.INTRODUCTION:

1.1 purpose

This document is the general overview of the software requirements for the flight booking app. The flight Booking app's goal is to give users an easy-to-use and effective online platform to book flights, reserve seats, pay online, and peruse airline schedules. The application seeks to improve user experience, accelerate secure and convenient transactions, and streamline the booking process for flights.

1.2 Scope

This document describes the standards by which the Flight Booking app's initial release will be judged. The flight booking app's developers, testers, and end users are among the target audience. The features and functionalities expected for the project's initial phase are included in the scope.

1.3 Stakeholders

The primary users of the Flight Booking app are individual [customers](#) seeking to book flights. The secondary users are [system administrators](#) responsible for managing flights, schedules, and user data.

- Customer: Users who intend to book flights.
 - ◆ Role: Primary users of the system who intend to book flights.
 - ◆ Interactions:
 - Create an account for flight bookings.
 - Search for flights based on travel preferences.
 - Select and reserve seats.
 - Make online payments for flight reservations.
 - Receive booking confirmations and notifications.
 - Provide feedback on their booking experience.

- Admins: System administrators responsible for managing flights, schedules, and user data.
 - ◆ Role: System administrators responsible for managing flights, schedules, and user data.
 - ◆ Interactions:
 - Monitor and manage user accounts.
 - Receive and analyze feedback from users.
 - Perform system maintenance and updates.
- Airlines: The companies that the app integrates to provide airline services.
 - ◆ Role: Companies integrated into the system to provide airline services.
 - ◆ Interactions:
 - Provide flight information and schedules to the system.
 - Receive and process flight bookings from users.
 - Manage seat availability and configurations.
 - Handle cancellations and refunds as per policies.
- Payment Gateway: Services that enable electronic payments.
 - ◆ Role: Services that enable electronic payments for flight bookings.
 - ◆ Interactions:
 - Process secure online payments for flight reservations.
 - Ensure the integration of secure payment systems.
 - Handle payment-related issues and inquiries.

1.4 Abbreviation & Glossary

- E-ticketing: Electronic ticketing, a digital form of flight ticket.
- E-payments: Electronic payments, including credit cards, digital wallets, etc.
- Seat Reservation: The process of reserving specific seats on a flight.
- Caching: Caching refers to the practice of storing a copy of specific data on a user's device that has been previously fetched from the server. This locally stored copy allows for quicker retrieval when the same data is requested again, reducing the need for repeated server queries.
- Servers: A server is like a super-smart computer that stores and shares information. When you use apps, websites, or play online games, a server is the computer helping

make everything work. It listens to what you want, gets the needed information, and sends it back to your device so you can see or do what you asked for.

2.SYSTEM REQUIREMENTS:

- **Authentication and User Registration:**

The system shall implement a robust authentication mechanism requiring users to register with their personal data, including email and password.

User registration data shall be securely stored in the database to maintain confidentiality.

To ensure user legitimacy, various verification techniques such as security questions, password verification, or email-based verification codes shall be employed.

- **Looking and Making a Reservation:**

The system should facilitate easy flight searches by allowing users to input travel details such as dates, hours, departure, arrival, and the number of passengers.

Filter options shall be provided to refine search results based on user preferences, including stops, airline reputation, and other factors.

Comparison of flights based on factors like cost, duration, stops, and airline reputation should be enabled.

Additional details such as plane information and seat availability shall be presented to users for an informed decision-making process.

- **Booking Procedure:**

The application should provide a streamlined booking process, allowing users to easily choose their preferred seats.

Integration with secure payment methods shall be implemented to facilitate a seamless payment experience.

Upon successful payment, users should receive a confirmation email containing a unique reference number.

Users shall be promptly notified of any changes to the gate, delays, cancellations, or updates in flight status.

- **Returns and Cancellations:**

The system should support a straightforward cancellation procedure initiated by users through the app.

Clear policies and refund eligibility criteria based on ticket type, fare restrictions, and cancellation time should be provided.

The refund process shall be transparent, disclosing refund amounts and estimated timelines for processing.

- **User Input:**

Various feedback mechanisms, including comments, suggestions, and complaints, shall be provided to gather user insights.

Real-time feedback submission shall be encouraged, allowing users to share their experiences immediately after completing their trip.

A structured feedback form shall collect relevant information for analytical purposes.

Analytics tools should be employed to analyze feedback data, identify patterns, and generate reports for management.

- **Seat Selection:**

The system shall enhance the user experience by visually representing the aircraft's seating configuration.

Real-time seat availability information shall be displayed, allowing users to pick and reserve specific seats based on their preferences.

Users should have the option to choose between aisle or window seats, and interactive information about each seat shall be provided through pop-ups or tooltips.

Assumptions and Constraints

Assumption: Users have access to the internet for booking flights.

Constraint: The system will be developed for web browsers and may not support all browsers.

Operating System Compatibility: The application is compatible with the following operating systems:

- Windows
- macOS
- iOS
- Android

Web Browser Compatibility: The application is designed to work seamlessly with the following web browsers:

- Microsoft Edge
- Google Chrome
- Safari.

2.1 functional requirements

2.1.1 User Registration and Authentication:

The user shall create an account if they have an account already. The system will display a login screen so he can login with his information ([Mail and Password](#)). If the information of the user is incorrect the system should ask the user to re-submit it. If the does not have an account, he/she should create an account then a sign-up screen will be displayed asking for personal information ([Name, Phone number, Mail, Password \(should be strong and not less than 8 characters\), Birth date, profile picture, Personal id](#)) if the information is missed or incorrect, he/she will be asked to submit them again. And to make sure it's the real user the system will authenticate the user by asking some questions or submitting the password twice or sending a verification code to the user's mail to make sure he is not a bot or malware. After the data is submitted its securely saved in the database in the system and the user can login in the application successfully.

- Users create an account with personal information.
- Returning users log in with email and password.
- Authentication methods ensure user validity.

2.1.2 Search and Booking:

This functional requirement involves several key components and functionalities that are essential for users to search for available flights, compare options, and complete the booking process.

2.1.2.1 Flight Search

- Users should be able to enter their travel information into the app, including the (dates, preferred times, locations of departure and arrival and number of passengers).
- The app should provide filter options to refine search results based on these preferences so the users may have preferences like non-stop flights, specific airlines, layover durations, or price ranges.
- Users should be able to compare different flights based on various parameters, such as price, duration, layovers, departure/arrival times, and airline reputations. This helps users make informed decisions.
- Providing additional details about the aircraft, seat availability, onboard services, and amenities can assist users in choosing the best-suited flights.
- The application should display a list of available flights that match the entered criteria. This includes details such as (flight numbers, departure/arrival times, durations, layovers, airline options, and price).

2.1.2.2 Booking Process

After the user finish his search and the list of the available flights is displayed the user can:

- Select their preferred seats on the available flights, considering factors like window or aisle seats, extra legroom, etc.
- Pay for their reservations using a variety of payment methods (credit/debit cards, e-wallets, etc.) on integration with secure payment systems.

2.1.3 Notification and Confirmation:

- After successful payment, users should receive booking confirmation mail along with a unique reference number that they can use for future references or modifications.
- The application also sends notifications or emails to users regarding flight status updates, gate changes, delays, or cancellations, ensuring they stay informed about their travel plans.

2.1.4 Cancellation and Refund:

It's a feature of an airline booking application that lets consumers cancel their scheduled flights and get refunds. This feature guarantees an easy-to-use and seamless approach for addressing refunds and travel plan modifications.

2.1.4.1 Cancellation Process

- Users should have the ability to initiate the cancellation process for their booked flights through the app.
- Clear and thorough cancellation regulations, including costs and deadlines for cancellations, should be displayed on the application. Refund eligibility should also be indicated based on ticket type, fare restrictions, and cancellation time.
- Users can submit cancellation requests by selecting the booked flight they wish to cancel and confirming the cancellation. After initiating the cancellation process, users should receive immediate confirmation that their cancellation request has been received and is being processed.

2.1.4.2 Refunding Process

- The app should outline the conditions under which users are eligible for refunds, such as within a specific cancellation window or fare type. Different fare classes may have varying refund policies.
- Provide transparent information about the refund amount that users will receive after cancellation, considering deductions for cancellation fees or penalties as per the airline's policies. In cases where partial refunds are applicable, provide details on how these are calculated and processed.
- Provide an estimated timeline for when users can expect to receive their refunds. This timeline may vary based on payment methods, banking institutions, and airline policies.
based

2.1.5 User Feedback:

2.1.5.1 Feedback types:

- It should be possible for users to provide various kinds of feedback, including suggestions, praises, and grievances.
- Several feedback categories pertaining to the flight booking procedure must be supported by the system.

2.1.5.2 Real-time Feedback Submission:

- Users should be able to share their experiences as soon as they complete their trip by providing feedback in real-time.

2.1.5.3 Feedback Form:

- There should be a structured feedback form accessible that gathers pertinent data such as flight information, booking reference, and departure date.
- Give users the option to enter comments and thoughts in open-ended fields.

2.1.5.4 Feedback Analytics:

- To improve the user experience overall, employ analytics tools to examine feedback data that has been gathered, spot trends, and extract insights.
- Provide management with reports so they may evaluate the application's performance and implement data-driven enhancements.

2.1.6 Seat Selection:

Users should have the ability to view the available seats on selected flights and choose their preferred seats during the booking process.

2.1.6.1 Visual Representation:

- Display a visual representation of the aircraft's seating arrangement during the booking process.
- Illustrate seat configurations, divisions (e.g., by class), and other relevant details to help users understand the layout.

2.1.6.2 Seat Availability Information:

- Provide real-time information on seat availability, distinguishing between occupied and available seats.
- Implement color-coded indicators or labels to clearly signify the status of each seat.
- Seat Availability

2.1.6.3 Seat Selection and Reservation:

- Allow users to interact with the visual representation to select and reserve specific seats.
- Provide details such as seat numbers, proximity to windows or aisles, and any additional features (e.g., extra legroom).

2.1.6.4 Personal Preferences:

- Incorporate a preference-setting feature, allowing users to input their preferences, such as window or aisle seats.
- Utilize a filtering mechanism to highlight seats that align with the user's specified preferences.

2.1.6.5 Interactive Seat Information:

- Enable users to click or tap on individual seats to access detailed information, including pricing, amenities, and any restrictions.
- Implement tooltips or pop-ups to display relevant details when a user hovers over a seat.

2.2 Non-functional requirements

2.2.1 Reliability and Availability:

The Flight Booking application aims to provide a [dependable and continuously available service](#) for users, reducing the impact of potential failures and ensuring a seamless and reliable experience.

Robust Server Failure Handling: Implement robust mechanisms to handle server failures effectively, minimizing the risk of data loss. This includes regular backups, fault-tolerant architectures, and failover systems to ensure continuous data integrity.

High Availability (24/7): [Ensure the availability of servers 24/7](#), minimizing downtime for users. This involves designing the system with redundancy and load balancing to distribute traffic efficiently, mitigating the impact of potential failures.

Scheduled Maintenance to Minimize Disruptions: Carefully schedule maintenance and updates to avoid disrupting user services. Plan maintenance during low-traffic periods to minimize the impact on users and maximize system availability.

Rate of Failure Occurrence: Reduce the rate of failure occurrence by identifying and addressing vulnerabilities in the system. Regular audits, security assessments, and continuous monitoring contribute to a more reliable system with fewer incidents of failure.

Availability Metrics: Continuously monitor and measure system availability, utilizing key metrics to assess performance and identify areas for improvement. This includes tracking uptime, response times, and user experiences to ensure a reliable and responsive application.

2.2.2 Performance:

By focusing on these performance considerations, the Flight Booking application aims to deliver a fast, reliable, and user-friendly experience. The **commitment to rapid response times**, optimized page loads, and efficient server-side processing contributes to a **high-performance system** that meets user expectations and demands.

Rapid Response Times: Guarantee swift response times, aiming for all user interactions and transactions to complete within **3 seconds**. This commitment ensures a responsive and efficient user experience, contributing to overall satisfaction.

Optimized Initial Page Loads: Prioritize the optimization of initial page loads to deliver a seamless user experience. Implement techniques such as lazy loading and efficient caching to reduce load times, enabling users to access the application quickly and effortlessly.

2.2.3 Maintainability + Manageability:

The goal is to create a system that not only functions efficiently but also allow for seamless updates, quick issue resolution, and straightforward administrative control. These considerations contribute to the overall reliability and longevity of the Flight Booking application.

Thorough Code Documentation: Document the codebase comprehensively, providing detailed insights into its structure and functionality. This documentation serves as a valuable resource for developers, facilitating efficient maintenance and future updates.

Codebase Understandability: Prioritize code clarity to enable developers to understand and modify the code efficiently. This approach not only streamlines ongoing improvements but also ensures that new developers can quickly familiarize themselves with the codebase.

Ease of Component Fixing: Implement a system that minimizes the time required to fix components. Well-documented code and a clear architecture contribute to swift identification and resolution of issues, [reducing downtime](#) and [enhancing system reliability](#).

Administrator-Friendly Management: Develop an administrator interface that is user-friendly and intuitive. This ensures that administrators can efficiently manage the system, adjusting, and handling routine tasks without extensive training.

Recoverability Measures: Incorporate robust recoverability measures, allowing the system to recover quickly from faults or failures. This includes [efficient error handling](#), [automatic backups](#), and mechanisms to restore the system to a stable state after unexpected events.

Serviceability Considerations: Design the system with serviceability in mind, allowing for easy maintenance and updates. This involves implementing features and protocols that facilitate the installation of patches, updates, and new functionalities without causing disruptions to the overall system.

2.2.4 Compatibility:

The Flight Booking application aims to reach a broad user base, offering a consistent and reliable experience regardless of the chosen device or platform. This approach enhances accessibility and usability, contributing to the overall success of the application.

Web Browser Compatibility: Ensure broad compatibility by testing and optimizing the application for popular web browsers, including [Chrome](#), [Firefox](#), and [Safari](#). This ensures a consistent and reliable user experience across different browser environments.

Minimum Hardware Requirements: Define and communicate minimum hardware requirements to guarantee optimal performance. This may include specifications such as processor speed, RAM, and storage capacity. Ensuring compatibility with a range of hardware configurations enhances the accessibility of the Flight Booking app.

Supported Operating Systems and Versions: Clearly specify the supported operating systems and their versions. This ensures that users on different devices and platforms can reliably access the application. Regularly update compatibility to align with evolving operating system releases.

2.2.5 Scalability:

The Flight Booking application aims to provide a reliable and responsive experience, even during periods of high demand. The continuous assessment, infrastructure scaling, and performance optimization strategies contribute to the system's ability to handle increased loads and ensure consistent service delivery to users.

Infrastructure Scaling: Implement infrastructure scaling mechanisms to dynamically adjust resources based on demand. This includes provisions for additional servers, load balancing, and cloud-based solutions to optimize system performance during peak periods, such as Christmas, holidays, and other high-traffic events.

Code Performance Optimization: Continuously optimize code performance to enhance the system's ability to handle increased loads efficiently. Regularly review and refine algorithms, database queries, and other critical components to ensure they can scale seamlessly with growing user numbers.

2.2.6 Usability:

User-Friendly Interface: The application [features a simple and intuitive user interface](#).

Effortless Navigation: Users can easily move through the app from account creation to payment.

Clear Instructions: Concise and clear instructions guide users at every step of the booking process.

Visually Appealing Design: The interface is visually appealing, enhancing the overall user experience.

Simplicity in Account Creation: Creating an account is straightforward and user-friendly.

Smooth Flight Selection: Users can effortlessly browse and select flights that match their preferences.

Convenient Payment Process: The payment process is designed for ease, supporting various payment methods.

Accessible Management: Users can efficiently manage their flights, ensuring a hassle-free experience.

User-Centric Design: The application's design focuses on the user's needs for a convenient and enjoyable booking journey.

2.2.7 Portability:

When we talk about portability in the Flight Booking app, [we're looking at how easily it can adapt to different systems and devices.](#)

Percentage of Target-Dependent Statements: This is like checking how much of the app's language is specific to a certain type of system. The lower this percentage, the better. It means the app is more flexible and can smoothly work on different devices without needing a lot of changes.

Number of Target Systems: This is about how many different systems or devices the app can comfortably run on. The higher the number, the more portable or adaptable the app is. So, if it works well on various phones, tablets, and computers, it's considered highly portable.

3. USE CASES TABLES:

3.1 Use case (User Registration and Authentication):

Name	User Registration and Authentication
Actor	Users, Admin
Entry Condition	<ol style="list-style-type: none">1. Internet Access2. Device Compatibility3. Valid Email Address4. Personal Information5. Acceptance of Terms and Conditions6. Age Verification7. Email Confirmation
Exit Condition	<ol style="list-style-type: none">1. Successful Registration2. Email Verification3. Authentication Success
Flow Of Events	<p>For User Registration:</p> <ol style="list-style-type: none">1. User initiation2. Provide information.3. Create Credentials ==> User creates a username and password for their account4. Submit information.5. Confirmation <p>For User Authentication</p> <ol style="list-style-type: none">1. User Login:2. User Credentials => User entries their username and password

	3. System authentication => verification for the database 4. Redirect to Dashboard => User id redirected to their personalized dashboard or the main booking interface.
Alternative Events	Alternative Events for User Registrations: <ol style="list-style-type: none"> 1. Invalid Email 2. Weak Password 3. Duplicate Username 4. Email Verification Timeout Alternative Events for User Authentication: <ol style="list-style-type: none"> 1. Account Lockout 2. Forget Password 3. Inactive Account 4. Account Deactivation

3.2 Use case (Search and Booking):

Name	Search and Booking
Actor	User, Airline, Admin, Payment Gateway.
Entry Condition	-User is logged in and navigated to the search page. - User is on the search results page. - User has selected a flight and is ready to book.
Exit Condition	- Booking is successful, and payment is processed. -the user receives confirmation.
Flow Of Events	After the user login to the App by his account: 1) User enters travel details (dates, times, departure, arrival) in the search app. 2)Application check for details if validate. 3) User applies optional filters for preferences. 4) Application displays available flights based on criteria.

	<p>5) User compares and selects a preferred flight.</p> <p>6) App shows additional details of the selected flight.</p> <p>7) User chooses seats.</p> <p>8) Application checks seat availability and confirms the chosen seats.</p> <p>9) User enters payment details.</p> <p>8) Application processes payment through secure system.</p> <p>9) User receives a booking confirmation.</p> <p>10) Application sends the confirmation details to the user's app for display.</p>
Alternative Events	<p>1) "error in case1" If the entered details are invalid, display an error message.</p> <p>3) "error in case3" If no matching flights are found, inform the user, and suggest adjusting search criteria.</p> <p>6) "error in case6" If there is a technical issue, display an error message and help options.</p> <p>8) "error in case 8" If payment fails, notify the user, and provide alternative payment methods.</p> <p>9) "error in case9" If confirmation email delivery fails, prompt the user to check their spam folder or resend the confirmation.</p>

3.3 Use case (Notification and Confirmation):

Name	Notification and Confirmation
Actor	User
Entry Condition	User successfully completes the payment for flight booking.

Exit Condition	1) Registration Confirmation Notification. 2) User receives a booking confirmation email with a unique reference number 3) User receives timely notifications about flight status, gate changes, delays, or cancellations.
Flow Of Events	1) Application sends a registration confirmation notification to the user. 2) User receives a booking confirmation email with a unique reference number 3) User receives timely notifications about flight status, gate changes, delays, or cancellations. 4) Application subscribes the user to relevant flight updates based on their booking.
Alternative Events	1) User does not receive the confirmation email. 2) - In case of critical updates, the system may attempt to contact the user through alternative means (e.g., SMS or phone call).

3.4 Use case (Cancellation and Refund):

Name	Cancellation and Refund
Actor	User, Airline, Admin
Entry Condition	The user has a booked flight and wishes to initiate the cancellation process.
Exit Condition	The cancellation request is successfully processed, and the user is informed of the refund status.

Flow Of Events	<ol style="list-style-type: none"> 1) User initiates the cancellation process 2) Application displays cancellation regulations 3) User reviews regulations and confirms cancellation 4) Application receives and confirms cancellation 5) Application outlines refund eligibility 6) Application calculates and displays refund amount 7) User confirms refund details 8) Application processes refund and notifies user 9) Application provides estimated refund timeline
Alternative Events	<ol style="list-style-type: none"> 1) User informs the system about the previous appointment. 2) System cancels the user tickets. 3) User requests a change in the appointment time. 4) System cancels the initial booking and starts the "current booking " use case. 5) if the user cancels within 24 hours it can be canceled by the Airline.

3.5 Use case (User Feedback):

Name	User Feedback
Actor	User, Airlines, Admins
Entry Condition	The user has completed booking a trip.
Exit Condition	The feedback is successfully submitted and recorded in the system.

Flow Of Events

- 1) System detects the completion of the booking process & checks for updates in seat availability.
- 2) User is prompted to provide feedback.
- 3) The system triggers a prompt for feedback to be provided by the user.
- 4) User selects feedback type: suggestions, praises.
- 5) The system triggers a prompt for feedback to be provided by the user.
- 6) Users have the option to share real-time feedback immediately after completing booking the trip.
- 7) The system directs the user to a feedback form.
- 8) Collects relevant data such as flight information, booking reference, and departure date.
- 9) Users shall receive a confirmation message acknowledging successful feedback submission.
- 10) Analytics tools periodically analyze gathered feedback.
- 11) Reports are generated and sent to management.

Alternative Events	<p>1) If a technical error occurs during the feedback submission</p> <p>2) The system notifies the user about the issue.</p> <p>4) The user is prompted to retry the feedback submission.</p> <p>5) If the issue persists, the system logs the error for investigation by technical support.</p>
--------------------	--

3.6 Use case (Seat Selection):

Name	Seat Selection
Actor	Customer, Airlines, Admins
Entry Condition	The user is in the process of booking a flight and has reached the seat selection stage.
Exit Condition	The user has successfully selected and reserved preferred seats, and the booking process advances.
Flow Of Events	<p>1. Customer reaches the seat selection stage during the flight booking process.</p> <p>2. Visual Representation: System displays a visual representation of the aircraft's seating arrangement, including divisions and relevant details.</p> <p>3. Seat Availability Information: System provides real-time information on seat availability, distinguishing between occupied and available seats.</p> <p>4. User Interaction: Customer interacts with the visual representation to select and reserve specific seats based on</p>

	<p>personal preferences.</p> <p>5. Preferences Input: Customer inputs personal seat preferences, such as window or aisle seats.</p> <p>6. Preference Filtering: System filters and highlights seats that align with the user's specified preferences.</p> <p>7. Interactive Seat Information: Customer clicks or taps on individual seats to access detailed information, including pricing, amenities, and restrictions.</p> <p>8. Reservation Confirmation: System confirms the selected seats, updates the booking information, and proceeds with the booking process.</p>
Alternative Events	<p>1)If the user encounters technical issues during seat selection, they are prompted to retry.</p> <p>2)If the system detects conflicting seat selections, it notifies the user and provides alternatives.</p>

4. USE CASE DIAGRAM

