Colorado Technical University

CS492

Team Project II

Unit 1 – Review and Setup

Group 5:

Claudia Rivera, Brandon Meadows, Josh Mann, Ryan Shippy, Victor McMurray-Kendall

Table of Contents

[Project Outline 3](#_Toc197881717)

[Project Description and Scope 3](#_Toc197881718)

[System Features:. 4](#_Toc197881719)

[Team Roles 5](#_Toc197881720)

[Collaboration Methodology 5](#_Toc197881721)

[Definition of Done 6](#_Toc197881722)

[Product Design 6](#_Toc197881723)

[Home Page Design 6](#_Toc197881724)

[Menu Pages 7](#_Toc197881725)

[About Us 8](#_Toc197881726)

[Book a Flight 9](#_Toc197881727)

[Checkout 11](#_Toc197881728)

Sprint 1 Retrospective Summary Report……………………………………………………………………………………………16

Sprint 2 Retrospective Summary Report…………………………………………………………………………………………….18

References…………………………………………………………………………………………………………………………………………19

# Project Outline

## Project Description and Scope

The project aims to develop an online reservation system for an airline company looking to expand its market to online shoppers. The system will enable customers to search for flights, purchase tickets, check flight statuses, and manage reservations. The project scope includes:

* **Flight Search & Purchase**:

Customers can search for available flights based on various criteria such as destination, departure date, and return date. They can then purchase flight tickets directly through the system. Customers can also check the status of their flights in real-time. The ability to filter search results are based on criteria like price, number of stops, airline, departure/arrival times. Option to view flights availability and prices across a range of dates.

* **Reservations & Ticket Sales Tracking**:

The system keeps track of all reservations made by customers. The system also records all ticket sales for flights to various destinations. Search by flight number or booking reference number. Customers can access their bookings to view details, change seats, request upgrades, or cancel flights. Clear display of cancellation fees and rules.

* **Payment Processing**:

The system supports payments by credit card only, ensuring secure and convenient transactions.

* **Flight Cancellations & Notifications**:

The system supports flight cancellations. Customers are notified of flight cancellations through email or SMS notifications.

* **Agent Sales Tracking**:

The system keeps track of agents and their individual sales. Detailed reports on agent sales can be generated for management purposes.

## System Features:

* **User Accounts:**
  + Registration: Easy registration process for new customers.
  + Login/Logout: Secure login/logout functionality.
  + Profile management: Ability for customers to manage their profile information (name, contact details, frequent flyer number - optional).
  + Password management: Password reset and recovery options.
* **Notifications:**
  + Email notifications: Automated emails for booking confirmations, flight status updates, cancellations, and other important information.
  + SMS notifications: Option for customers to receive SMS notifications for flight status updates and other time-sensitive information.
* **Payment:**
  + Secure payment gateway integration: Integration with a PCI DSS compliant payment gateway.
  + Credit card support: Support for major credit card types.
  + Fraud prevention: Implementation of fraud prevention measures.
* **Security:**
  + Data encryption: Encryption of sensitive data (e.g., credit card information, passenger details).
  + Access control: Role-based access control to restrict access to sensitive data and functionalities.
  + Security audits: Regular security audits to identify and address vulnerabilities.
* **Scalability:**
  + Scalable architecture: Design the system to handle a large volume of transactions and users.
  + Performance optimization: Optimize the system for speed and performance.
* **Reporting & Analytics:**
  + Sales reports: Generation of reports on ticket sales, revenue, and other key metrics.
  + Customer analytics: Tracking of customer behavior and preferences.
* **Administration Panel:**
  + System configuration: Ability for administrators to configure system settings.
  + User management: Management of user accounts and permissions.
  + Reporting and analytics: Access to system reports and analytics.

## Team Roles

In a Scrum framework, the team roles typically include.

* **Product Owner: Claudia Rivera** - Responsible for defining the product vision, managing the product backlog, and prioritizing tasks.
* **Scrum Master: Josh Mann**- Ensures the team follows Scrum practices, facilitates meetings, and removes any impediments.
* **Development Team: Ryan Shippy, Brandon Meadows, Victor McMurray-Kendall** - A cross-functional group that designs, develops, tests, and delivers the product increments.

## Collaboration Methodology

The team will use the Scrum methodology for collaboration, which includes:

* **Sprint Planning**- A meeting where the team decides what work will be performed during the sprint.
* **Daily Scrum**- A daily meeting (usually 15 minutes) for the team to synchronize activities and create a plan for the next 24 hours.
* **Sprint Review**- A meeting at the end of the sprint to review the work completed and determine whether it meets the definition of done.
* **Sprint Retrospective**- A meeting for the team to reflect on the past sprint and identify improvements for the next sprint.

The benefit of scrum allows the team to adapt quickly to changes and new requirements. With regular meetings and visible progress, it helps maintain transparency in the project's development. The repetitive nature of sprints allows for continuous improvement and early detection of issues. Scrum fosters a collaborative environment where team members work closely together.

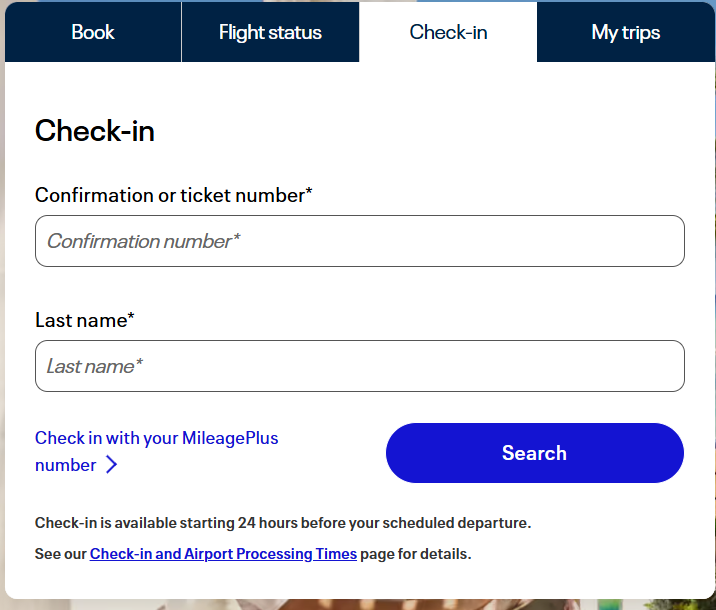
## Definition of Done

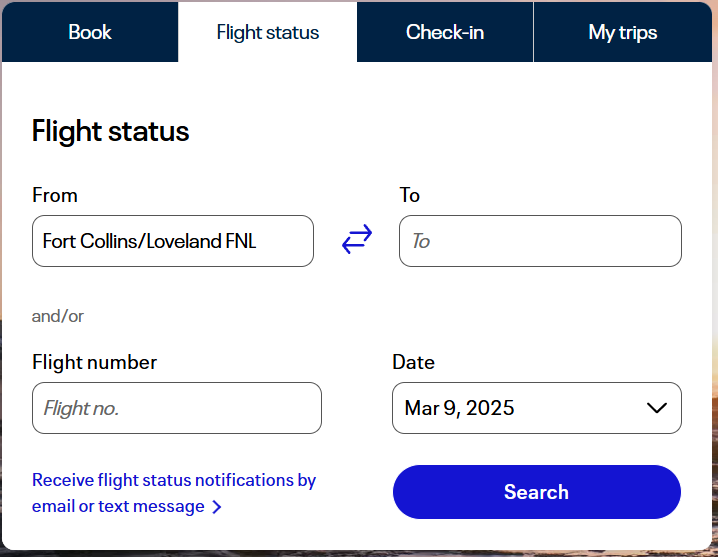
The Definition of Done (DoD) is a clear and concise list of criteria that a product increment must meet to be considered complete. It includes:

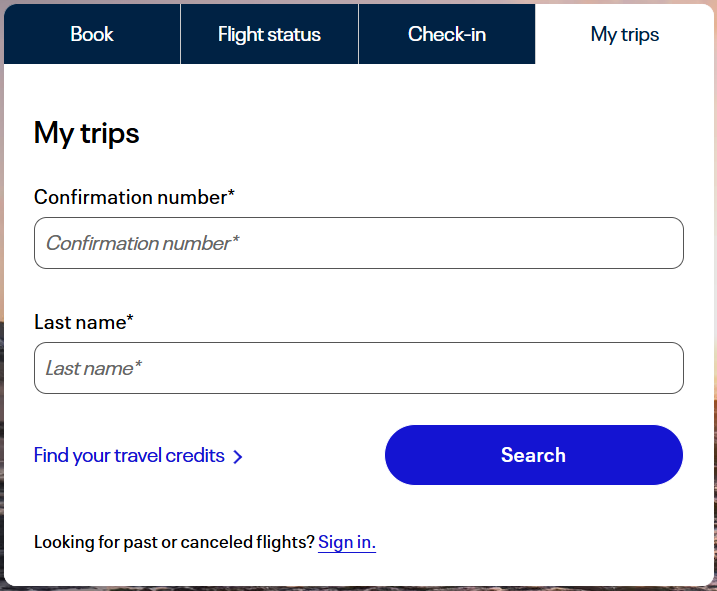
1. Code is written, reviewed, and tested.
2. Documentation is updated.
3. All acceptance criteria for user stories are met.
4. The increment is potentially shippable and meets quality standards.

## Product Design

## Home Page Design

**Check flight status, check-in, and manage your trip from main page.**

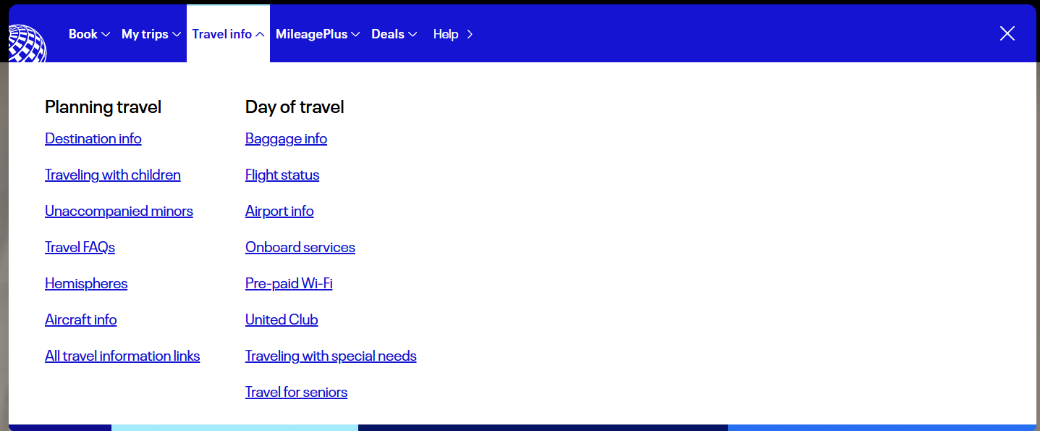


****

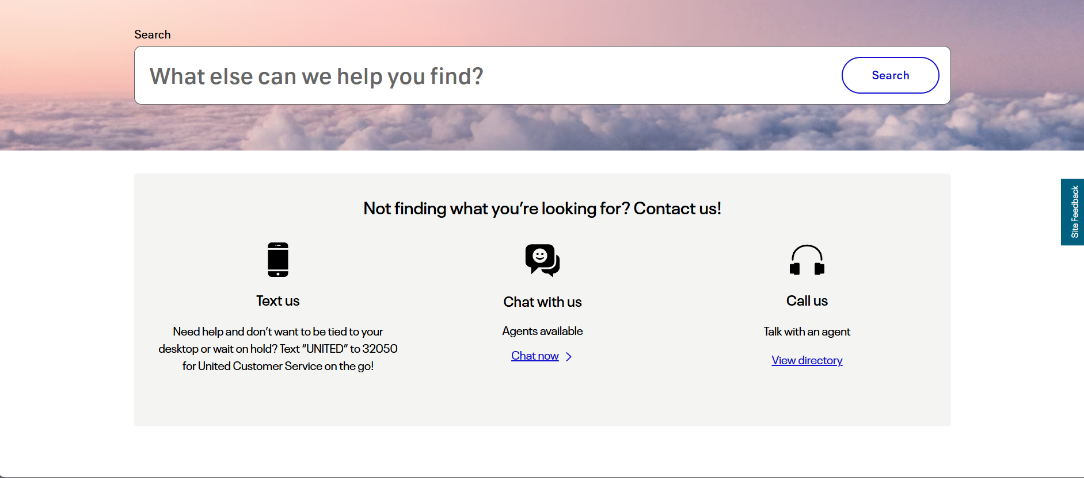
## Menu Pages

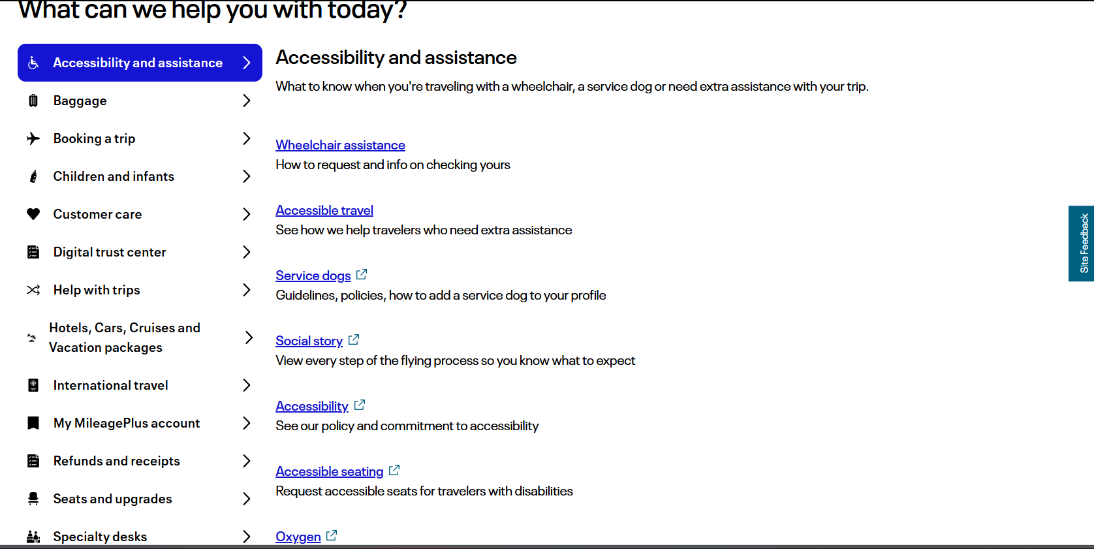
A screenshot of a computer

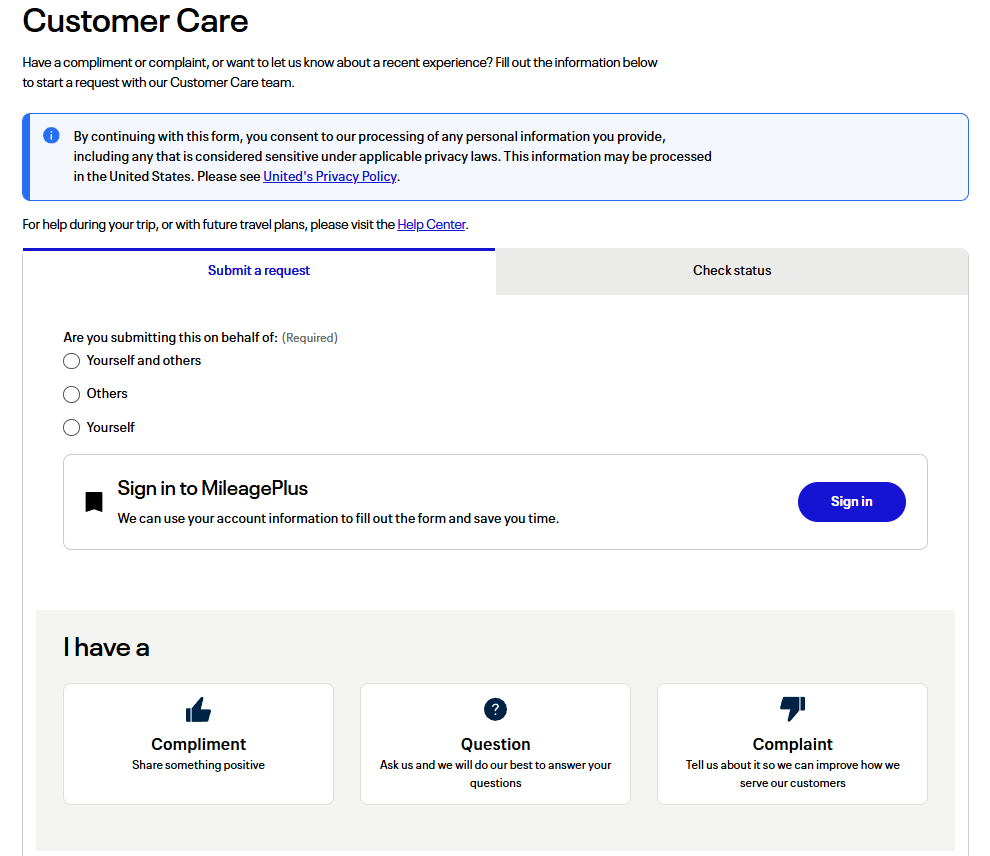
AI-generated content may be incorrect.



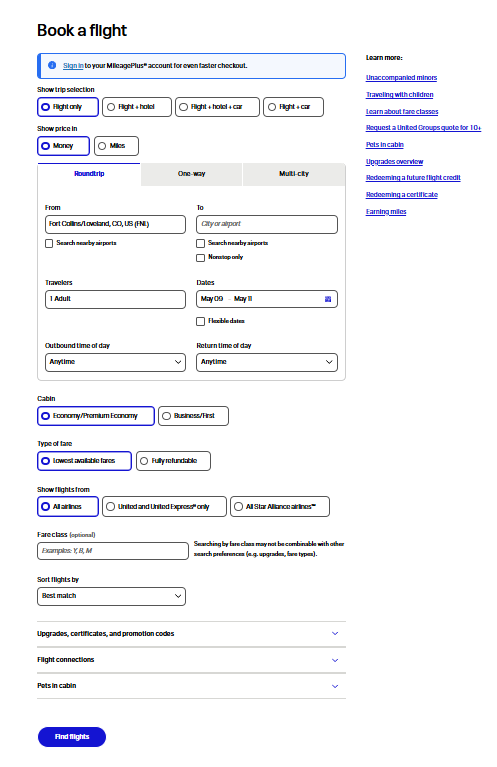
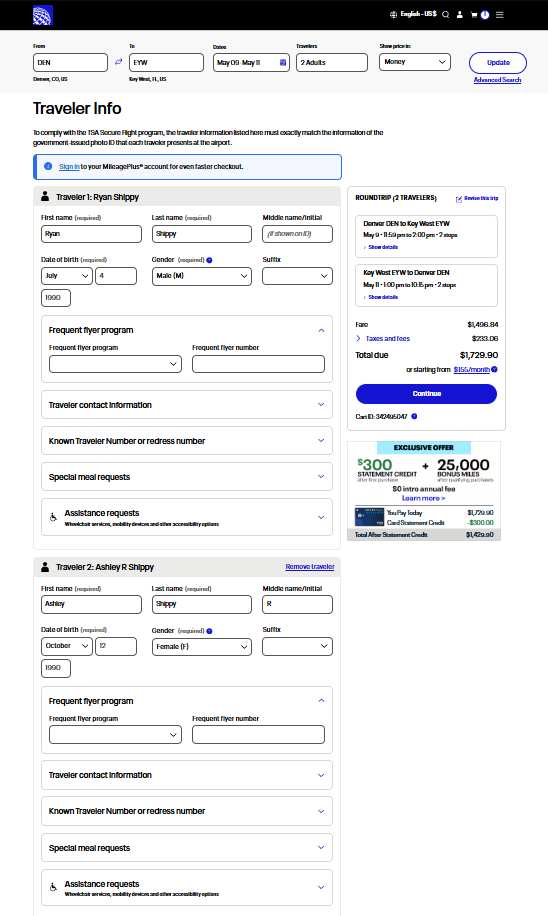
## About Us

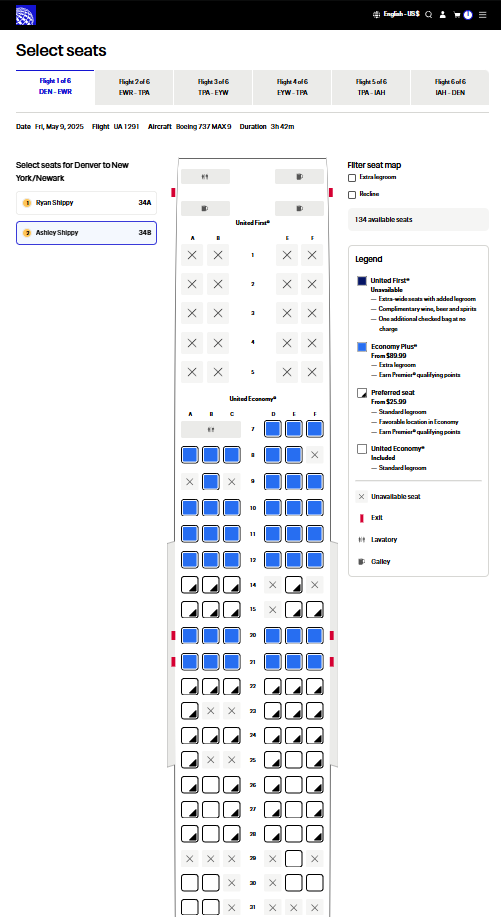




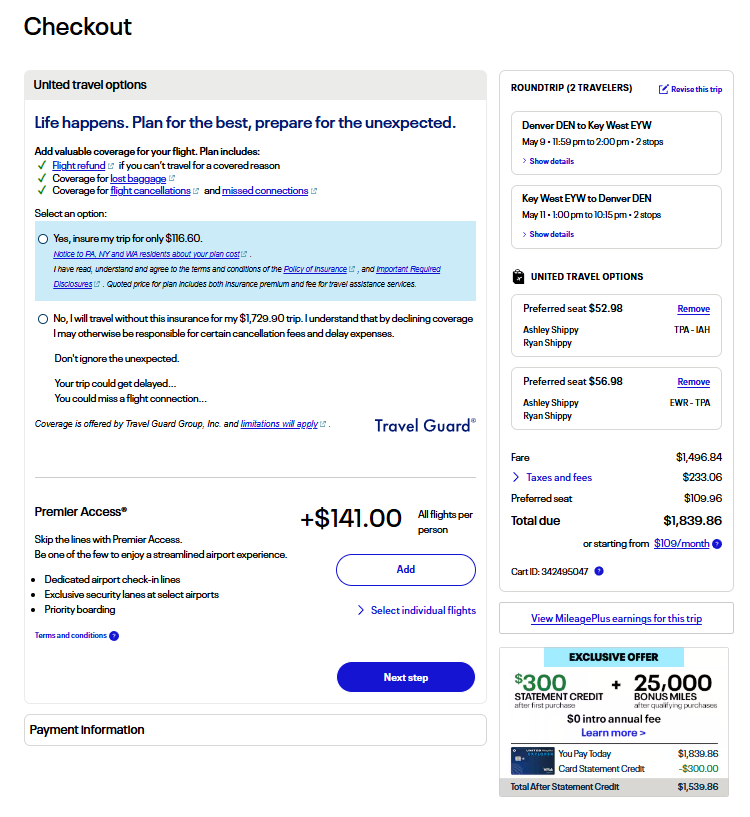


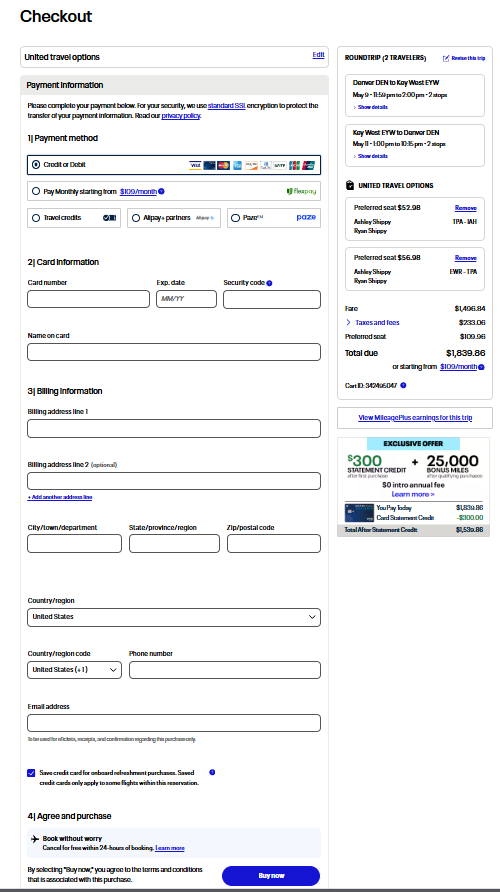
## Book a Flight





## Checkout





1. Product Design

The project scope for the Flight Reservation System includes with features like search, booking, payment, user management, booking confirmation notifications, agent sales and reporting, user profile customization, enhanced search filters, cancellation/ refund process, admin analytics, and admin dashboards.

* Flight Search and Booking:
  + Round-trip, one-way, and multi-city search options.
  + Flexible date search.
  + Filtering and sorting by price, airline, departure/arrival times etc.
  + Seat selection (including seat maps).
  + Baggage options and fees.
  + Fare types and restrictions.
  + Display of up-to-date flight availability and pricing.
* Payment Processing
  + Support for various payment methods (credit cards, debit cards, etc.)
  + Secure payment gateway integration.
  + Payment Card Industry Data Security Standard (PCI DDS) compliance to ensure secure handling of cardholder data.
  + Handling of payment failures and refunds.
* User Account Management
  + User registration and login (including social login options).
  + Profile management (name, contact details, travel preferences).
  + Password management (reset or change).
  + Storing travel history and booking details.
* Booking Confirmation Notifications
  + Email and/or SMS notifications for booking confirmations.
  + Ability for users to manage notification preferences.
  + Real-time updates on flight changes (delays or cancellations).
* Agent Sales and Reporting:
  + Dedicated interface for travel agents to manage bookings.
  + Commission tracking and reporting.
  + Access to customer support tools.
* User Profile Customization:
  + Allow users to personalize their profiles (ex., preferred airlines, seat preferences, meal choices).
  + Offer travel recommendations based on user preferences and past bookings.
* Enhanced Search Filters:
  + Provide advanced filtering options (ex., by aircraft type, number of stops, departure/arrival airports).
  + Allow users to save search criteria for future use.
* Admin Analytics Dashboard:
  + Provide administrators with insights into booking trends, revenue, and user behavior.
  + Tools for managing flights, users, and agents.
  + Reporting and data visualization capabilities.
* Security (Authentication, Authorization, Data Protection):
  + Implement robust security measures to protect user data and prevent fraud.
  + Adhere to industry best practices and comply with relevant regulations.

A screenshot of a computer screen

Description automatically generated

The use case diagram shows the interactions between users (actors) and the system. It helps identify the main functionalities of the flight reservation system.

1. Activity Diagram:

Flight Search and Booking

--> SearchFlights : User enters search criteria

SearchFlights --> ValidateCriteria : Validate input (origin, destination, date)

ValidateCriteria -- Invalid Criteria --> DisplayError : Show error message

ValidateCriteria -- Valid Criteria --> RetrieveFlights : Query flight database

RetrieveFlights --> DisplayResults : Show available flights

DisplayResults --> SelectFlight : User selects a flight

SelectFlight --> BookFlight : User initiates booking

BookFlight --> CheckAvailability : Verify seat availability

CheckAvailability -- Seats Available --> CreateBooking : Create booking record

CheckAvailability -- No Seats --> DisplayError : Show "no seats" message

CreateBooking --> InitiatePayment : Redirect to payment gateway

InitiatePayment --> ProcessPayment : Payment processing

ProcessPayment -- Successful --> SendConfirmation : Send booking confirmation

ProcessPayment -- Failed --> DisplayError : Show payment error

SendConfirmation --> [\*]

Activity Diagram: Payment Processing (Security Focus)

--> InitiatePayment : User initiates payment

InitiatePayment --> EncryptData : Encrypt payment data (using HTTPS, TLS)

EncryptData --> SendToGateway : Send encrypted data to payment gateway

SendToGateway --> AuthorizePayment : Payment gateway validates and authorizes

AuthorizePayment -- Authorized --> CapturePayment : Capture payment

AuthorizePayment -- Declined --> DisplayError : Show payment declined

CapturePayment --> UpdateBookingStatus : Update booking status

UpdateBookingStatus --> [\*]

The activity diagrams for Flight search and booking as well as Payment Processing shows the flow of activities for specific use cases. They will help in understanding the sequence of events and decision points chosen.

1. Non-Functional Requirements and Security Considerations

* Authentication: Implement secure authentication to protect user accounts and API endpoints.
* Authorization: Use role-based access control to restrict access to sensitive features (ex., admin dashboards, flight management) based on user roles.
* Data Encryption: Encrypt sensitive data (ex., payment information, user credentials) both in transit (HTTPS, TLS) and at rest.
* Input Validation: Implement robust input validation to prevent injection attacks (SQL injection, cross-site scripting).
* Rate Limiting: Implement rate limiting to protect against brute-force attacks and denial-of-service (DoS) attacks.
* Logging and Monitoring: Implement comprehensive logging and monitoring to detect and respond to security incidents.
* Compliance: Ensure compliance with relevant regulations (ex., Payment Card Industry Data Security Standard (PCI DSS) for payment processing, General Data Protection Regulation (GDPR) for data privacy).
* Performance: Design the system for high performance and scalability to handle a large number of users and transactions.
* Availability: Ensure high availability by using redundant systems and load balancing.
* Usability: Design a user-friendly interface that is easy to navigate and use.

Security Considerations in Detail:

* Authentication:
  + Users should log in with strong passwords and multi-factor authentication (MFA) if possible.
  + Use secure tokens for API authentication.
  + Implement password hashing for storing user passwords.
* Authorization:
  + Define clear roles (e.g., user, agent, admin) and permissions.
  + Use access control lists (ACLs) to manage access to resources.
* Data Protection:
  + Use HTTPS for all communication between the client and server.
  + Encrypt sensitive data in the database.
  + Regularly back up data and store backups securely.
* Input Validation:
  + Validate all user inputs on the server-side.
  + Use parameterized queries to prevent SQL injection.
  + Sanitize user inputs to prevent cross-site scripting (XSS) attacks.
* Rate Limiting:
  + Limit the number of requests per user or IP address.

Use a web application firewall to block malicious traffic.

# Sprint 1 Retrospective Summary Report

|  |
| --- |
| **Things That Went Well** |
| The team established clear communication from the start by actively using Discord and GitHub. Josh efficiently managed GitHub, set up repositories, provided tutorials, and actively assisted members with technical issues. Team roles were clearly defined and respected, with Josh effectively fulfilling the Scrum Master role by organizing meetings and facilitating communication. Victor consistently coordinated documentation and updated backlog tasks promptly. Claudia proactively communicated her schedule and issues to the team, allowing the group to adapt. Team members were supportive and responsive to each other’s challenges, fostering a positive collaborative environment. |
| **Things That Could Have Gone Better** |
| The main challenge was coordinating schedules and timely completion of coding tasks. Several members had scheduling conflicts, particularly with weekend meetings. Delays in accepting GitHub invites and uploading code created last-minute pressure to ensure visual and functional consistency across pages. Claudia accidentally deleting the "About Us" folder highlighted the need for clearer version control processes or more immediate availability for technical assistance. Additionally, some members faced challenges with correctly adding CSS files, leading to potential inconsistencies in website design. |
| **Things That Surprised Us** |
| The degree of dependency on each other's contributions surprised the team, particularly how small delays from individual members significantly impacted overall progress. The complexity involved in synchronizing code for visual consistency across different pages was also more challenging than initially anticipated. Team members were surprised by how quickly the sprint deadlines approached, highlighting the need for better planning and adherence to deadlines in subsequent sprints. |
| **Lessons Learned** |
| A key lesson learned is the critical importance of scheduling meetings at times convenient for all team members. Moving meetings to Saturdays mid-day, rather than Sundays, emerged as a significant improvement. Establishing clearer deadlines for code uploads and adhering to these timelines will reduce last-minute adjustments and stress. Early technical guidance, particularly with tools like GitHub, proved valuable and should continue in future sprints. Clearer communication regarding each member’s availability and progress throughout the week will enhance collaboration, prevent bottlenecks, and improve overall project outcomes. |

# Sprint 2 Retrospective Summary Report

|  |
| --- |
| **Things That Went Well** |
|  |
| **Things That Could Have Gone Better** |
|  |
| **Things That Surprised Us** |
|  |
| **Lessons Learned** |
|  |

# References

*Agile – Scrum Sprint Definition, Roles and Planning*. (2017, September 17). https://www.nutcache.com/blog/agile-scrum-sprint/. Retrieved February 16, 2025, from <https://www.nutcache.com/blog/agile-scrum-sprint/>

*What is a Definition of Done?* (n.d.). Scrum.org. <https://www.scrum.org/resources/what-definition-done>

*United Airlines*. (2025). https://www.united.com/en/us/. Retrieved March 9, 2025, from <https://www.united.com/en/us/>

Altexsoft Editorial Team. (2024, October 3). Flight booking process: Structure, Steps, and Key Systems. *AltexSoft*. https://www.altexsoft.com/blog/flight-booking-process-structure-steps-and-key-systems/

Atlassian. (n.d.). *Agile scrum roles and responsibilities*. https://www.atlassian.com/agile/scrum/roles

Ergun, O. (2024, September 15). *Understanding High-Level Design: An Introduction for Beginners | Orhan Ergun*. Orhan Ergun. https://orhanergun.net/understanding-high-level-design-an-introduction-for-beginners

Feldman, K. (2023, March 18). *The role of a scrum master in facilitating collaboration and productivity - Isixsigma.com*. isixsigma.com. https://www.isixsigma.com/dictionary/scrum-master/

Sutherland, J. (2022, June 21). *Definition of Done - Scrum Inc.* Scrum Inc. https://www.scruminc.com/definition-of-done/

*The Scrum Team | Scrum Alliance*. (n.d.). https://resources.scrumalliance.org/Article/scrum-team