April 18th, 2024

**Comfort Airlines**

**Requirements Document Outline**

Client: Professor Michael Oudshoorn

Group Name: Byte Me

Contractors: Joellen A., Kyle B., Hanna Z., Alex H., Joshua T., JR V.

**About Requirements Document**

The "Requirements Document" for Comfort Airlines, crafted by the Byte Me group, provides a detailed outline of both functional and non-functional requirements for a software system designed to enhance airline operational efficiencies. This document meticulously describes how each component of the system operates, including databases, GUIs, and timetables, ensuring that all stakeholders have a clear understanding of the expected system behaviors and outputs. The document serves as a comprehensive guide, detailing the interactions between user inputs and system outputs, and is essential for ensuring that the final product aligns perfectly with the client's needs and expectations.

Introduction

This document details the functional and non-functional requirements and describes the observable behavior of our software system. Our aim is to ensure that upon reading this document, our client will have a comprehensive understanding of the product deliverables, encapsulating not just the functionalities but also the expected inputs and outputs, thereby enabling an intuitive grasp of the system's operations. Each component is described below with an emphasis on observable behavior and the cause-and-effect relationship between inputs and outputs, as well as an example of what output looks like based on input.

**Functional Requirements**

**Paragraph describing requirements**

Database

**Description:** Our database is a foundational system used to store and manage all flight-related data. This includes the tables: flight with entities: FlightNumber, DepartureAirport, DestinationAirport, NumberOfPassengers, ScheduledDepartureTime, ActualDepartureTime, ScheduledArrivalTime, ActualArrivalTime, TailNumber, airport with entities: AirportCode, ArrivalTime, DepartureTime, ArrivingFlightNumber, DepartingFlightNumber, NumberOfArrivingPassengers, NumberOfDepartingPassengers, GateUsed, AircraftTailNumber, aircraft with entities: TailNumber, FlightDate, FlightNumber, DepartureAirport, DestinationAirport, DepartureTime, ArrivalTime, NumberOfPassengers and passengers with entities: PassengerID, SourceAirport, DestinationAirport, FlightNumber, ScheduledDepartureTime, ActualDepartureTime, ScheduledArrivalTime, ActualArrivalTime We use mariadb to create the tables and entities that will be later queried. The tables each have been edited to include a speed entity on the aircraft table and on the flight table. The database will store the information for the GUI, and it will pull the data and run it through code to output the information needed.

**Observable Behavior:** When a user queries for flight information (input), the system promptly retrieves and displays the relevant details (output), ensuring secure and efficient access. This query will work with the flight GUI, as the user selects the flight from a dropdown, querying the specific flight information connected to that flight number to be displayed.

Language Used: mySQL

When a user interacts with a MySQL database (input), the system swiftly processes the request and retrieves the requested data (output), ensuring reliable and secure access to information. For instance, in a web application featuring flight details, users may input their desired flight number through a dropdown menu on the graphical user interface (GUI). Upon selection, the system queries the database for the specific flight information corresponding to the chosen flight number, promptly displaying the relevant details to the user. This seamless interaction underscores MySQL's efficiency and versatility in handling user requests and delivering accurate data in real-time.

Input Sample: (Not actual final input)

1. Flight Table:

- FlightNumber: FL123

- DepartureAirport: JFK (John F. Kennedy International Airport)

- DestinationAirport: LAX (Los Angeles International Airport)

- NumberOfPassengers: 150

- ScheduledDepartureTime: 2024-03-21 08:00:00

- ActualDepartureTime: 2024-03-21 08:30:00

- ScheduledArrivalTime: 2024-03-21 11:00:00

- ActualArrivalTime: 2024-03-21 11:30:00

- TailNumber: BA-456

- Speed: 550 mph

2. Airport Table:

- AirportCode: JFK

- ArrivalTime: 2024-03-21 11:30:00

- DepartureTime: 2024-03-21 08:30:00

- ArrivingFlightNumber: FL123

- DepartingFlightNumber: FL456

- NumberOfArrivingPassengers: 150

- NumberOfDepartingPassengers: 200

- GateUsed: 1

- AircraftTailNumber: BA-789

3. Aircraft Table:

- TailNumber: TN789

- FlightDate: 2024-03-21

- FlightNumber: FL456

- DepartureAirport: LAX

- DestinationAirport: JFK

- DepartureTime: 2024-03-21 12:00:00

- ArrivalTime: 2024-03-21 15:00:00

- NumberOfPassengers: 200

- Speed: 600 mph

4. Passengers Table:

- PassengerID: 001

- SourceAirport: JFK

- DestinationAirport: LAX

- FlightNumber: FL123

- ScheduledDepartureTime: 2024-03-21 08:00:00

- ActualDepartureTime: 2024-03-21 08:30:00

- ScheduledArrivalTime: 2024-03-21 11:00:00

- ActualArrivalTime: 2024-03-21 11:30:00

Timetable

**Description:** The timetable is a tool used to display all the flight information, divided by departures and arrivals, the time of each flight, flight numbers, number of passengers, the airports in which the flights will arrive and depart, and the tail number. The timetable is used in order to organize and output the data stored in the database upon user request.

**Observable Behavior:** Upon selecting a specific date and time (input), the timetable will display all corresponding flight operations (output), aiding in planning and operational efficiency.

Language Used to Design: PHP

### 

### Graphical User Interface (GUI) Requirements

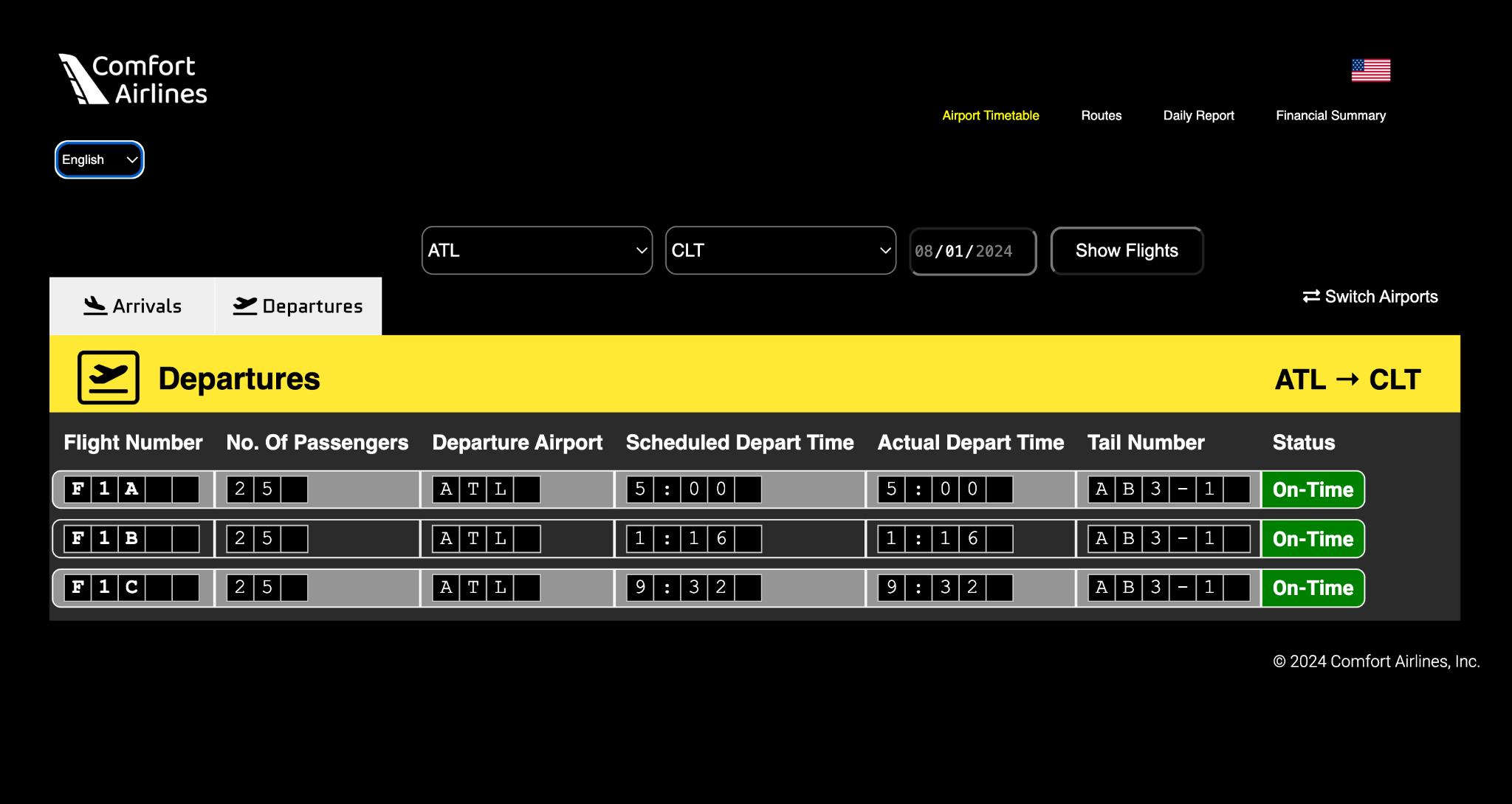
In the context of the Comfort Airlines software system, the graphical user interface (GUI) is not merely a point of interaction but the nexus of user experience, operational management, and data visualization. Our GUI is designed to embody the functional and non-functional requirements of our software system, providing a direct, intuitive connection between user inputs and system outputs. Below is a detailed description of the GUI components, emphasizing their observable behavior, and the cause-and-effect relationship between inputs and outputs.

#### **Index Page (index.php)**

The Index Page serves as the primary interface for users to interact with flight data. It is designed to offer an immediate overview of flight operations, including arrivals and departures, directly from the homepage. This page stands as the gateway for users to access detailed flight information, enabling selections based on airport codes to filter relevant flight data. The user-friendly layout ensures that all pertinent information, from flight status to passenger counts, is easily accessible, facilitating a better understanding of flight schedules and statuses at a glance.

**Inputs:**User selection of an airport code from a dropdown menu.

**Outputs:** A dynamically generated table displaying detailed flight information based on the selected airport code. Information includes flight number, passenger count, destination or departure airport, scheduled and actual times, and flight status (e.g., On-Time, Delayed, Canceled, Early).

GUI Output:* result of picking LAX and departures*

#### **Calculation Page (calc.php)**

The Calculation Page is pivotal for financial analysis and reporting within the Comfort Airlines system. It aggregates and processes flight data to generate detailed financial reports, covering aspects such as operational costs, revenue, and occupancy rates. This page is instrumental for internal stakeholders to assess financial performance, identify trends, and make informed decisions based on comprehensive cost and revenue breakdowns. The inclusion of maintenance and downtime costs further enriches the financial overview, providing a holistic view of the airline's operational efficiency and profitability over selected periods.

**Inputs:** Flight data fetched from the database, including tail numbers, flight dates, and distances.

**Outputs:** Comprehensive financial summaries presented through HTML tables. These summaries include operational costs per day, revenue generated per flight, actual occupancy rates, and the total operational costs over a two-week period. Additionally, it features calculations on maintenance costs, downtime costs, and a total Profit/Loss Summary for the specified period.



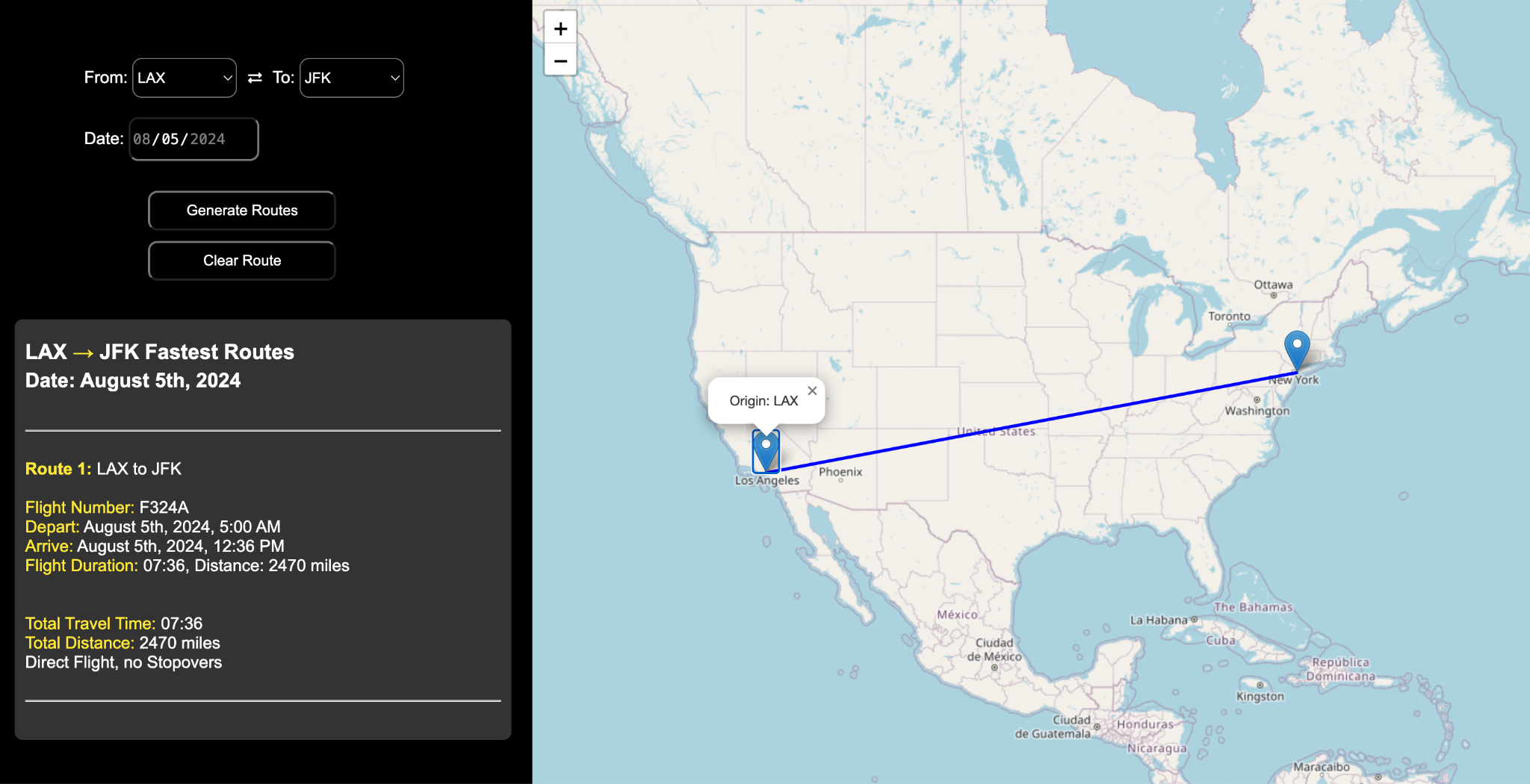
*Graphical representation of the financial summary for August 2024.*

#### **Route Navigator Page (Route.php):**

#### The Route Page is a user-centric portal designed for detailed insights into specific flight paths. It provides a comprehensive look at the intricacies of various routes, including frequency, capacity, and performance metrics. From this page, users can deep dive into the data, choosing specific route codes from a dropdown menu to filter and obtain the route-specific information.

#### **Inputs:** User selection of a route code from a dropdown menu.

#### **Outputs:** A dynamically generated report that includes information such as the number of flights per route, average passenger load, on-time performance statistics, and other operational data pertinent to the selected route. The layout ensures that the information is presented in an easily digestible format, allowing users to understand and analyze route efficiency and performance effectively.



*Screenshot showing the route details from JFK to LAX.*

#### **Simulation Report (Repo.php):**

#### The Report Page acts as a comprehensive dashboard for financial and operational reporting. It is designed to deliver a deep analysis of operational costs, revenue, and profitability across various timeframes. This interface allows users to select specific periods or dates to generate detailed financial reports.

#### **Inputs:** User selection of a date range or specific date from a date picker.

#### **Outputs:** A thorough financial report that outlines daily and total operational costs, including details on fuel, terminal, and leasing fees. It also aggregates revenue data from ticket sales and presents the total number of passengers transported. The final section synthesizes this data to reveal the profit or loss over the selected period, providing an essential tool for financial planning and analysis in the airline industry.

#### 

*Screenshot of the Daily Report for TailNumber AB3-10.*

#### **Database Connection (comfort\_db.php)**

While not a user-facing component, the Database Connection script is crucial for the seamless operation of the system. It handles the essential task of establishing a secure and reliable connection to the database, ensuring that the application has access to up-to-date and accurate data for all other components. In the event of a connection failure, it gracefully redirects the user to an error page, maintaining system integrity and user experience even under error conditions.

**Inputs:** Database credentials (DSN, username, password) for establishing a connection to the comfort\_airlines\_byteme database.

**Outputs:** Not directly observable through a GUI component, but successful database connection allows for the seamless operation of the index.php and calc.php pages. Failure in connection triggers the display of error.php.

Visual Representation Placeholder:

**Error Page (error.php)**

The Error Page is designed to handle unexpected situations gracefully, providing users with clear information about any issues encountered during their interaction with the system. This component enhances the system's robustness, ensuring that users are not left in the dark about errors, but instead are offered context and potential solutions to the problems at hand. This approach helps maintain trust and confidence in the system's reliability.

**Inputs:** An error message variable provided by the PHP script encountering a connection issue or other errors.

**Outputs:** An HTML page that presents the error message in a user-friendly manner, offering suggestions for troubleshooting while maintaining visual consistency with the rest of the application.

#### **Database Connection Error Page (comfort\_db\_error.php)**

Similar to the general Error Page, the Database Connection Error Page specifically addresses issues related to database connectivity. It informs the user of the problem in a non-technical language, suggesting steps for resolution or further action. This specialized error handling reinforces the system's focus on user experience, ensuring that technical issues do not unduly impact the user's interaction with the application.

**Inputs:** An error message variable provided by the PHP script that failed to connect to the database.

**Outputs:** A styled HTML page that communicates the error message alongside troubleshooting tips, designed to inform the user about database connectivity issues in a clear, approachable format.

### 

### **Conclusion**

This requirements document has been carefully crafted to delineate the functional and non-functional aspects of the Comfort Airlines software system’s GUI. By specifying the inputs and expected outputs for each GUI component, alongside placeholders for visual representation, we aim to provide our client with a transparent, detailed overview of the system’s operations. This ensures that our client can visualize the end product and understand the direct relationship between user actions and system responses, reinforcing our commitment to delivering a user-centric, reliable software solution.

**Non-Functional Requirements**

Timeline and Commitment

**Team Meetings**

* Weekly team meeting in lab on Thursdays from 1:35pm - 3:15pm
* Additional team meetings when necessary on Fridays from 12:00pm - 2:00pm

**Individual Work Hours**

* Expected amount of time a team member should dedicate outside of the project hours is about 8 hours a week at minimum

**Availability and Communication**

* **Discord communication:** 
  + Includes channels for different portions of the project for files and documents to be shared
  + Channels:
    - Absent-dates
    - Due-dates
    - General
    - Questions
    - Userinterface
    - Githublinks
    - Database
    - Client-terminal-commands
    - meetings
* **iMessage communication:** 
  + Text messages for finalizing meeting times, last minute absences, and quick easy asking and discussion of the project
* **Shared Google Drive:**
  + Title - “Byte Me - Comfort Airlines”
    - Folders
      * Databases
      * Documentation
      * Formulas / Calculations
      * Pricing / Finances
      * Presentation
* **Github:**
  + Use github for file/code sharing

Overall

* We remain committed to delivering the outlined software components by the deadline of April 18, 2024, dedicating an average of 3 hours per group meeting and 8 hours of individual work per week to ensure timely and quality deliverables.

Airplane Decisions: All-First Class versus Mix-Class

In order to stand out from other competitors, we have decided to develop an all first class airline. The two types of airplanes that will be used are the Boeing 737-800 and the Airbus A220-300. After looking at dimensions of the types of airplanes, the Boeing 737-800 original airplane with premium and economy has a total of 12 premium seats and 162 economy seats, creating 174 total seats. However, when changing the seats to all premium options, with a pitch of 36 and a width of 17, then there will be an overall number of 156 seats, only an 18 seat loss. This is the most expensive airplane but will only be used when necessary and will maximize revenue. The rest of the airplanes used will be the Airbus A220-300, as not only it is the second lowest cost rent, but it also has the second most amount of seats from all four airplanes, even when the seats are all first-class. The Airbus A220-300 has 12 premium seats and 125 economy seats, creating a total of 113 seats. Based on measurements, when going all premium/first-class, the total number of seats on the airplane would be 132, increasing the amount of seats on the airplane by 7 seats. The Boeing 747-400 has 23 premium seats and 393 economy seats creating a total of 416 seats. Based on measurements the total number of premium seats would be 369 seats decreasing the number of seats by 47 seats, but giving quality seats to everyone for a long flight making the experience more enjoyable overall.

Conclusion

This revised conclusion provides a refined overview of the comprehensive functionalities and expected operational outcomes of the Comfort Airlines software system. With attention to both functional and non-functional requirements, this document has been structured to allow the client an easy comprehension of the software system and why decisions were made. By elaborating on the cause-and-effect dynamics in each system component, we offer an in-depth understanding of how user inputs translate to specific outputs. Through this approach, we ensure that our client possesses an understanding of the final product's capabilities and user experience, fostering confidence in the system's ability to meet and exceed operational expectations. This shows our dedication to delivering a software solution that not only addresses but anticipates the needs of Comfort Airlines.