# FC723 Application Part A

1. Software Development process:

For the project, I will be using the Waterfall development model because:

1. Clear Requirements: The project has defined and clear requirements: seat booking, availability checking, database integration)
2. Predictable phases: Given that the project involves initial development then database integration (at a later stage). This model allowed a structured progression of tasks.
3. Documentation-Heavy: Since the project includes diagrams, requirements specifications, and written essays. Waterfall allows the opportunity to produce documentation and necessary things to mention at each phase.
4. Functional Requirements Specification:
   1. Introduction:

-Purpose: This document specifies the functional requirements for the seat-booking system developed for Apache Airlines. The system will handle seat availability, booking, freeing of seats, and display of booking status for passengers

* 1. Functional Requirements:
  2. Check Seat Availability:

-Description: The system shall allow users to check the availability of any seat on the airplane. Available seats are denoted by the letter 'F', and reserved seats are denoted by 'R'. The system shall display a message showing the current status of the seat (either free or reserved) based on the seat input provided by the user.

* 1. Seat Booking:

-Description: The system shall allow users to book an available seat. A user must enter the seat identifier (e.g., 1A, 2B). The system shall verify if the seat is available ('F') and, if so, mark it as reserved ('R'). If the seat is already reserved, the system shall display an error message indicating that the seat is unavailable.

* 1. Seat freeing:

-Description: The system shall allow users to free a previously booked seat. The user shall enter the seat identifier, and the system will verify if the seat is reserved ('R'). If so, the system will mark the seat as free ('F'). If the seat is not reserved, an error message will be displayed.

* 1. Show Booking Status:

-Description: The system shall display the current booking status of a seat. This will show whether a seat is free ('F'), reserved ('R'), an aisle ('X'), or a storage area ('S'). Users will enter the seat identifier to retrieve the status of that specific seat.

* 1. Exit program:

-Description: The system shall provide an option to exit the program. This function terminates the application gracefully.

* 1. Non-functional Requirements:

3.1 Performance: The system should be able to process seat booking, freeing, and status checking in less than 2 seconds for a single seat.

3.2 Usability: The user interface should be easy to navigate, with clearly labeled buttons and input fields for the users to interact with. The system will provide error messages when invalid inputs are given.

3.3 Security: The system does not handle sensitive data in the provided scope, but it must ensure that the seat booking and freeing operations are protected from invalid inputs and unauthorized changes.

1. Activity diagram:

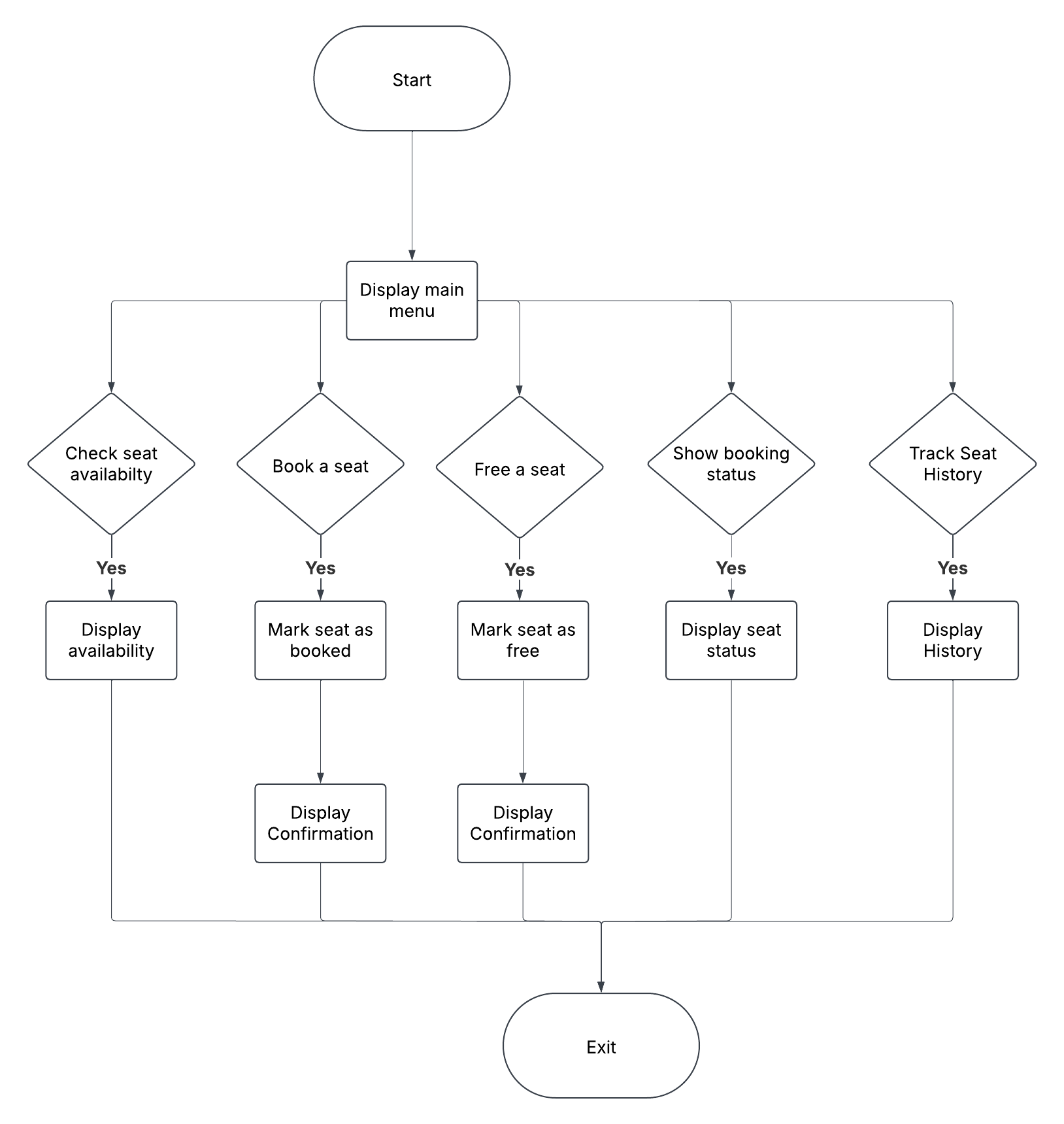
A diagram of a display

AI-generated content may be incorrect.

1. The code for the application is provided in the attached GitHub repository. It implements all the specified functionalities with error handling, and includes comments for documentation purposes.
2. A common feature available in airline booking systems, but not described in the original Apache Airlines' system, is the Seat Reservation History. This feature allows the system to track and display the history of reservations for each seat. For example, when a seat is reserved or freed, the system records the date and time of the transaction, along with the passenger's information.

The implementation of this functionality was achieved by adding a reservation\_history dictionary to the SeatBookingSystem class. The add\_to\_history() method records each action, and the show\_history() method allows users to retrieve and display the reservation history for a given seat.

The updated activity diagram:



1. I first created the public github repository (<https://github.com/lamawahab/F023_Project_3>)

Then, I cloned it into my local machine and run “git add .” to add all my files to Git, then committed using “git commit -m “Create book-seating.py”

I pushed my code into the repository using “git push origin main”

The repository is now publicly accessible, and the link to the repository is provided.