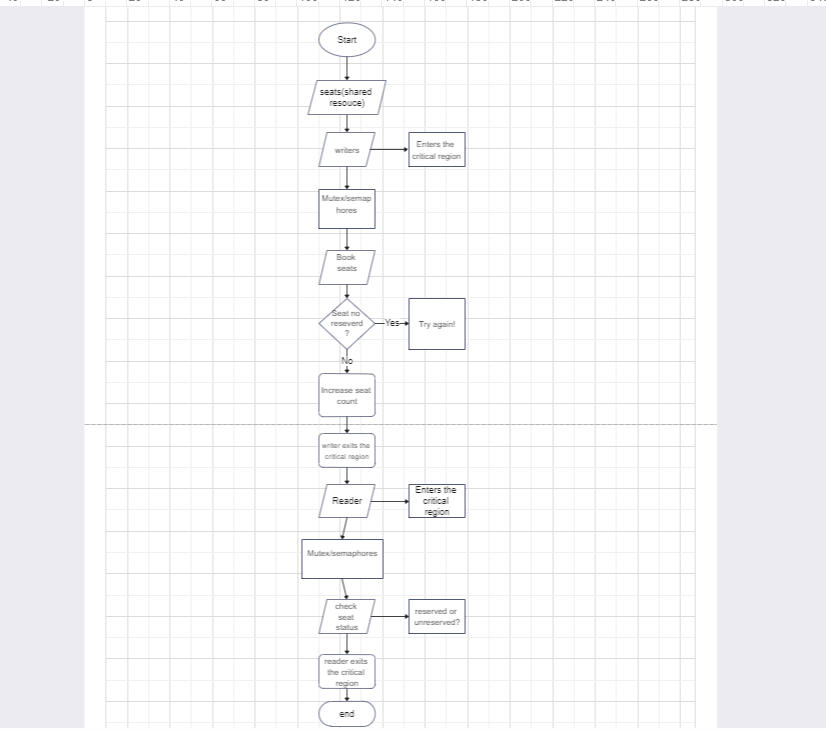
**Implementing Airline Reservation system using reader writers problem**:

**Student IDs: 21K-3288, 21K-3409, 20K-1889**

In this implementation, the seats array represents the shared resource that needs to be accessed concurrently. Readers read the status of a seat (reserved or unreserved), while writers reserve a seat if it is unreserved.

The architecture of the system is represented using a block diagram shown below, with the readers and writers as the input processes and the seats array as the output process. The mutex lock and the condition variables can be represented as intermediary processes that ensure synchronization and mutual exclusion between the readers and writers.

**PROJECT DIAGRAM:**



**WORK DIVISION PLAN:**

Talha: Implementing and testing writer’s function to ensure synchronization for the flight reservation system.

Zarah: Testing all the functions, creating main and display functions. Also creating a database that stores the reservation data needed to implement the reader-writers problem.

Sanjna: Implementing and testing readers function to ensure synchronization for the flight reservation system.