Lab 6 – Functional Testing

Create test cases for functional testing using the following techniques:

- **Boundary Value Analysis:**

Test Cases for Number of Passengers (Boundary Value Analysis):

Valid Input:

1. Test Case 1: Enter the minimum valid number of passengers (1).

   Test Data: 1

   Expected Result: The system should accept the input without errors.

2. Test Case 2: Enter a typical valid number of passengers (e.g., 3).

   Test Data: 3

   Expected Result: The system should accept the input without errors.

3. Test Case 3: Enter the maximum valid number of passengers (e.g., the system limit or a reasonable maximum, let's say 10).

   Test Data: 10

   Expected Result: The system should accept the input without errors.

Invalid Input (Lower Boundary)

4. Test Case 4: Enter a number of passengers less than the minimum valid value.

   Test Data: -4

   Expected Result: The system should display an error indicating that the input is invalid.

Invalid Input (Upper Boundary):

5. Test Case 5: Enter a number of passengers greater than the maximum valid value.

   Test Data: 20

   Expected Result: The system should display an error indicating that the input is invalid.

Edge Cases:

6. Test Case 6: Enter a number of passengers just above the maximum valid value.

   Test Data: 11

   Expected Result: The system should display an error indicating that the input is invalid.

7. Test Case 7: Enter a number of passengers just below the minimum valid value.

   Test Data: 0

   Expected Result: The system should display an error indicating that the input is invalid.

- **Equivalence Class Partitioning**

Test Cases for Age-Based Pricing (Equivalence Class Partitioning):

Valid Child Age – [0; 16)

Valid Adult Age – [16; 65)

Valid Senior Age – [65; 200)

Invalid Ages – (-inf; 0) || [200; inf)

Valid Adult Age:

1. Test Case 1: Enter the age of an adult within the valid range (e.g., 25).

   Test Data: 25

   Expected Result: The system should apply the standard pricing for adults.

Valid Child Age:

2. Test Case 2: Enter the age of a child within the valid range (e.g., 8).

   Test Data: 8

   Expected Result: The system should apply the discounted pricing for children.

Valid Senior Citizen Age:

3. Test Case 3: Enter the age of a senior citizen within the valid range (e.g., 65).

   Test Data: 65

   Expected Result: The system should apply the discounted pricing for senior citizens.

Invalid Age (Below Lower Bound for Adults):

4. Test Case 4: Enter an age below the lower bound for adults.

   Test Data: 15

   Expected Result: The system should display an error indicating that the age is below the valid range.

Invalid Age (Above Upper Bound for Senior Citizens):

5. Test Case 5: Enter an age above the upper bound for senior citizens.

   Test Data: 200

   Expected Result: The system should display an error indicating that the age is above the valid range.

Invalid Age (Negative Age):

6. Test Case 6: Enter a negative age.

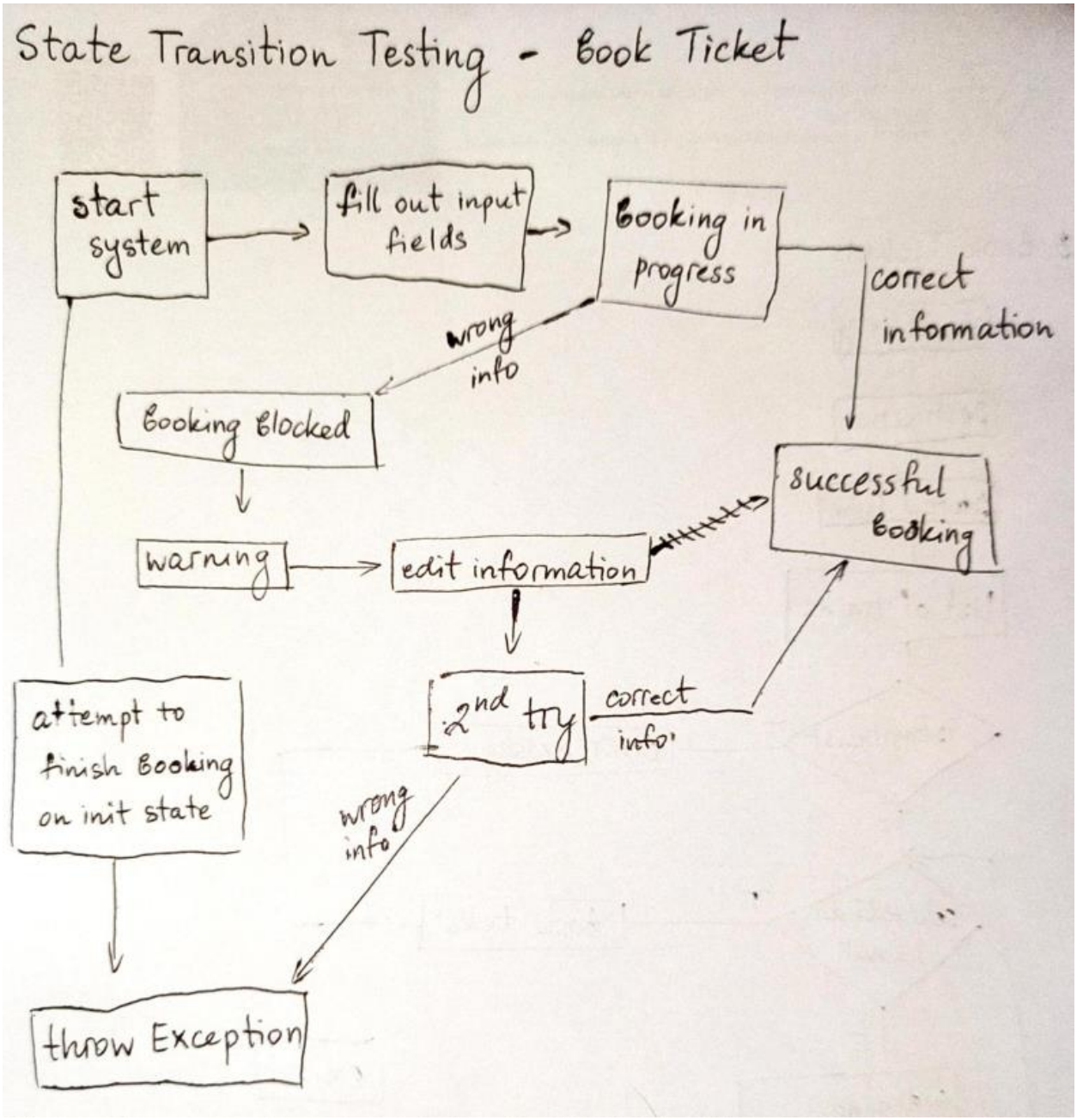
   Test Data: -5

   Expected Result: The system should display an error indicating that the age is invalid.

- Decision Table Testing – Book Ticket

Condition	Rule1	Rule2	Rule3	Rule4	Rule5
Entered start	F	T	F	T	T
Entered dest	F	T	F	T	T
Entered date	F	F	T	T	T
Avail. trains	F	F	T	T	T
Avail. seats	F	F	T	F	T
Output/ successful booking	F	F	F	F	T

- State Transition Testing – Book Ticket


State Transition Testing - Book Ticket
start system
fill out input fields
Booking in progress
correct information
wrong info
Booking blocked
successful Booking
warning
edit information
attempt to finish Booking on init state
2nd try
correct info
wrong info
throw Exception