Ethan Booker

Dr. Mao Zheng

CS743 Software Verification and Validation

Assignment 1

**Parameters**

*Ticket numbers*: 1001 - 1150 [150]

*Row numbers*: 1 - 25 [25]

*Seat numbers*: 1 - 6 [6]

**Equivalence classes**:

* *Ticket number partitions*
  + S1 = { all numbers below 1000 inclusive } // invalid partition
  + S2 = { all numbers between 1001 - 1150 inclusive } // valid partition
  + S3 = { all numbers above 1150 non-inclusive } // invalid partition
* *Row numbers*
  + S1 = { all numbers below 1 non-inclusive } // invalid partition
  + S2 = { all numbers between 1 - 25 inclusive } // valid partition
  + S3 = { all numbers above 6 non-inclusive } // invalid partition
* *Seat Numbers* 
  + S1 = { all numbers below 1 non-inclusive } // invalid partition
  + S2 = { all numbers between 1 - 6 inclusive } // valid partition
  + S3 = { all numbers above 6 non-inclusive } // invalid partition

**Boundary values for each equivalence classes** (*Underline values are nominal values*)

* *Ticket number boundary values*
  + S1 & S3 are both invalid partitions
  + S2 = { 1000 | 1001 | 1002 | 1075 | 1149 | 1150 | 1151 } // valid
* *Row numbers*
  + S1 & S3 are both invalid partitions
  + S2 = { 0 | 1 | 2 | 13 | 24 | 25 | 26 } // valid
* *Seat Numbers*
  + S1 & S3 are both invalid partitions
  + S2 = { 0 | 1 | 2 | 4 | 5 | 6 | 7 } // valid

**Data table organization**

* The data table is organized in this manner:
* I moved the error message closer to the Current Seat Assignment Message for consistent placement [ Invalid … is an expected invalid error message]
* **Single Fault Assumption**
* Check the boundary values for ticket number then row number then seat numbers.
  + 3 tests will be the same, so I removed of two of them.
* I will use nominal value as the default value if I’m not currently testing for that input.