[Date]

Frank Schulz

NMit

SDV502

Assessment ONe

Unit Testing

Contents

[1. public decimal Adult\_Before\_5() 2](#_Toc81565916)

[2. public decimal Adult\_After\_5() 3](#_Toc81565917)

[3. public decimal Adult\_Tuesday() 4](#_Toc81565918)

[4. public decimal Child\_Under\_16() 5](#_Toc81565919)

[5. public decimal Senior() 6](#_Toc81565920)

[6. public decimal Student() 7](#_Toc81565921)

[7. public decimal Family\_Pass() 8](#_Toc81565922)

[8. public decimal Chick\_Flick\_Thursday() 9](#_Toc81565923)

[9. public decimal Kids\_Pass() 10](#_Toc81565924)

[Summary 11](#_Toc81565925)

## 1. public decimal Adult\_Before\_5()

Inputs – int quantity, string person, string day, decimal time

Output – decimal total\_price

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Equivalence Partitioning & Boundaries** | | | | |
| **Status** | **Person** | **Day** | **Quantity** | **Time** |
| **Acceptable** | Adult | Mon,  Wed,  Thu,  Fri,  Sat,  Sun | > 0 | <0500 |
| **Unacceptable** | Student,  Family,  Senior,  Child | Tue | < 1 | >=0500 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Schedule** | | | | | | |
| **Use Case** | **Quantity** | **Person** | **Day** | **Time** | **Expected** | **Result** |
| 1. One adult, normal day, before 5 | 1 | Adult | Monday | 4 | 14.50 | Pass |
| 2. Two adults, normal day, before 5 | 2 | Adult | Monday | 4 | 29.00 | Pass |
| 3. Zero adults, normal day, before 5 | 0 | Adult | Monday | 4 | -1 | Pass |
| 4. One student, normal day, before 5 | 1 | Student | Monday | 4 | -1 | Pass |
| 5. One adult, Tuesday, before 5 | 1 | Adult | Tuesday | 4 | -1 | Pass |
| 6. One adult, normal day, after 5 | 1 | Adult | Monday | 6 | -1 | Pass |

## 2. public decimal Adult\_After\_5()

Inputs – int quantity, string person, string day, decimal time

Output – decimal total\_price

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Equivalence Partitioning & Boundaries** | | | | |
| **Status** | **Person** | **Day** | **Quantity** | **Time** |
| **Acceptable** | Adult | Mon,  Wed,  Thu,  Fri,  Sat,  Sun | > 0 | >= 0500 |
| **Unacceptable** | Student,  Family,  Senior,  Child | Tue | < 1 | < 0500 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Schedule** | | | | | | |
| **Use Case** | **Quantity** | **Person** | **Day** | **Time** | **Expected** | **Result** |
| 1. One adult, normal day, after 5 | 1 | Adult | Monday | 6 | 14.50 | Pass |
| 2. Two adults, normal day, after 5 | 2 | Adult | Monday | 6 | 29.00 | Pass |
| 3. Zero adults, normal day, after 5 | 0 | Adult | Monday | 6 | -1 | Pass |
| 4. One student, normal day, after 5 | 1 | Student | Monday | 6 | -1 | Pass |
| 5. One adult, Tuesday, after 5 | 1 | Adult | Tuesday | 6 | -1 | Pass |
| 6. One adult, normal day, before 5 | 1 | Adult | Monday | 4 | -1 | Pass |

## 3. public decimal Adult\_Tuesday()

Inputs – int quantity, string person, string day

Output – decimal total\_price

|  |  |  |  |
| --- | --- | --- | --- |
| **Equivalence Partitioning & Boundaries** | | | |
| **Status** | **Person** | **Day** | **Quantity** |
| **Acceptable** | Adult | Tue | >= 1 |
| **Unacceptable** | Student,  Family,  Senior,  Child | Mon,  Wed,  Thu,  Fri,  Sat,  Sun | < 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Schedule** | | | | | |
| **Use Case** | **Quantity** | **Person** | **Day** | **Expected** | **Result** |
| 1. One adult, Tuesday | 1 | Adult | Tuesday | 13.00 | Pass |
| 2. Two adults, Tuesday | 2 | Adult | Tuesday | 26.00 | Pass |
| 3. Zero adults, Tuesday | 0 | Adult | Tuesday | -1 | Pass |
| 4. One student, normal day | 1 | Student | Tuesday | -1 | Pass |
| 5. One adult, normal day | 1 | Adult | Monday | -1 | Pass |
| 6. One adult, normal day | 1 | Adult | Friday | -1 | Pass |

## 4. public decimal Child\_Under\_16()

Inputs – int quantity, string person

Output – decimal total\_price

|  |  |  |
| --- | --- | --- |
| **Equivalence Partitioning & Boundaries** | | |
| **Status** | **Person** | **Quantity** |
| **Acceptable** | Child | >= 1 |
| **Unacceptable** | Student,  Family,  Senior,  Adult | < 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Schedule** | | | | |
| **Use Case** | **Quantity** | **Person** | **Expected** | **Result** |
| 1. One child | 1 | Child | 12.00 | Pass |
| 2. Two children | 2 | Child | 24.00 | Pass |
| 3. Zero children | 0 | Child | -1 | Pass |
| 4. One student | 1 | Student | -1 | Pass |
| 5. One adult | 1 | Adult | -1 | Pass |
| 6. Two adults | 2 | Adult | -1 | Pass |

## 5. public decimal Senior()

Inputs – int quantity, string person

Output – decimal total\_price

|  |  |  |
| --- | --- | --- |
| **Equivalence Partitioning & Boundaries** | | |
| **Status** | **Person** | **Quantity** |
| **Acceptable** | Senior | >= 1 |
| **Unacceptable** | Student,  Family,  Child,  Adult | < 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Schedule** | | | | |
| **Use Case** | **Quantity** | **Person** | **Expected** | **Result** |
| 1. One Senior | 1 | Senior | 12.50 | Pass |
| 2. Two Seniors | 2 | Senior | 25.00 | Pass |
| 3. Zero Seniors | 0 | Senior | -1 | Pass |
| 4. One student | 1 | Student | -1 | Pass |
| 5. One Child | 1 | Child | -1 | Pass |
| 6. Two Children | 2 | Child | -1 | Pass |

## 6. public decimal Student()

Inputs – int quantity, string person

Output – decimal total\_price

|  |  |  |
| --- | --- | --- |
| **Equivalence Partitioning & Boundaries** | | |
| **Status** | **Person** | **Quantity** |
| **Acceptable** | Student | >= 1 |
| **Unacceptable** | Senior,  Family,  Child,  Adult | < 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Schedule** | | | | |
| **Use Case** | **Quantity** | **Person** | **Expected** | **Result** |
| 1. One Student | 1 | Student | 14.00 | Pass |
| 2. Two Students | 2 | Student | 28.00 | Pass |
| 3. Zero Students | 0 | Student | -1 | Pass |
| 4. One Child | 1 | Child | -1 | Pass |
| 5. One Adult | 1 | Adult | -1 | Pass |
| 6. Two Adults | 2 | Adult | -1 | Pass |

## 7. public decimal Family\_Pass()

Inputs – int quantity\_ticket, int quantity\_adult, int quantity\_child

Output – decimal total\_price

|  |  |  |  |
| --- | --- | --- | --- |
| **Equivalence Partitioning & Boundaries** | | | |
| **Status** | **Ticket Quantity** | **Adult Quantity** | **Child Quantity** |
| **Acceptable** | >= 1 | = 1  = 2 | >= 1 |
| **Unacceptable** | < 1 | < 1  > 2 | < 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Schedule** | | | | | |
| **Use Case** | **Ticket Quantity** | **Adult Quantity** | **Child Quantity** | **Expected** | **Result** |
| 1. Two Adult, Two Children | 1 | 2 | 2 | 46.00 | Pass |
| 2. One Adult, Three Children | 1 | 1 | 3 | 46.00 | Pass |
| 3. Two Adults, Two Children | 2 | 2 | 2 | 92.00 | Pass |
| 4. Zero Adults, Three Children | 1 | 0 | 3 | -1 | Pass |
| 5. Two Adults, Zero Children | 1 | 2 | 0 | -1 | Pass |
| 6. Two Adults, Three Child | 2 | 2 | 3 | -1 | Pass |

## 8. public decimal Chick\_Flick\_Thursday()

Inputs –int quantity, string person, string day

Output – decimal total\_price

|  |  |  |  |
| --- | --- | --- | --- |
| **Equivalence Partitioning & Boundaries** | | | |
| **Status** | **Quantity** | **Person** | **Day** |
| **Acceptable** | >= 1 | Adult | Thu |
| **Unacceptable** | < 1 | Senior,  Family,  Child,  Student | Mon,  Tue,  Wed,  Fri,  Sat,  Sun |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Schedule** | | | | | |
| **Use Case** | **Quantity** | **Person** | **Day** | **Expected** | **Result** |
| 1. One Adult | 1 | Adult | Thu | 21.50 | Pass |
| 2. Two Adults | 2 | Adult | Thu | 43.00 | Pass |
| 3. Zero Adults | 0 | Adult | Thu | -1 | Pass |
| 4. One Child | 1 | Child | Thu | -1 | Pass |
| 5. One Adult | 1 | Adult | Wed | -1 | Pass |
| 6. Two Students | 2 | Student | Thu | -1 | Pass |

## 9. public decimal Kids\_Pass()

Inputs – int quantity, string day, bool holiday

Output – decimal total\_price

|  |  |  |  |
| --- | --- | --- | --- |
| **Equivalence Partitioning & Boundaries** | | | |
| **Status** | **Quantity** | **Day** | **Holiday** |
| **Acceptable** | >= 1 | Wed | False |
| **Unacceptable** | < 1 | Mon,  Tue,  Thu,  Fri,  Sat,  Sun | True |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Schedule** | | | | | |
| **Use Case** | **Quantity** | **Day** | **Holiday** | **Expected** | **Result** |
| 1. One Child | 1 | Wed | False | 12.00 | Pass |
| 2. Two Child | 2 | Wed | False | 24.00 | Pass |
| 3. Zero Children | 0 | Wed | False | -1 | Pass |
| 4. One Child | 1 | Wed | True | -1 | Pass |
| 5. One Adult | 1 | Thu | True | -1 | Pass |
| 6. Two Students | 2 | Mon | True | -1 | Pass |

## Summary

The advantage of using Nunit Testing in the Cinema Functions App is that we can definitively show that the functions are properly executing and working as intended.  
I have implemented 6 test cases for each of the 9 functions for a total of 54 individual tests. This way I could cover the main potential problems and not spend an unnecessary amount of time trying to test for every potential edge case.

The tests that I have written mostly cover situations that could occur in real life, but I also wrote some that do not ( e.g. 0 people ) in order to ensure that those situations have also been covered.