Equivalent Class Testing

#### 

#### 

Team 16:

Raghav Gupta

Chelsea Jaculina

Bala Nyan Kwaw

Mike Lee

Yihua Li

**March 18th, 2018**

**Professor Badari**

**CMPE 187**

# Table of Contents

[**Table of Contents**](#_fvgztqx5fz61) **1**

[**Overview**](#_8pvx7xuczgkg) **2**

[Introduction](#_ynfxzoxduftu) 2

[Purpose](#_ld166kvhvjc4) 3

[**Question I**](#_1wh8gx8z0kic) **4**

[**Question II**](#_ysdx22aa9f8n) **5**

[**Question III**](#_bvsa2uhob1tm) **6**

[Equivalent Class Partition Diagram](#_xz6umbyuwj3r) 6

[Name Field](#_oywztewwyarb) 6

[Age Field](#_fv35j61vgu5h) 6

[Boundaries](#_6jfvn7torjwy) 7

[**Question IV**](#_12gndt7ndpng) **7**

[Test Oracle](#_jlu7nyf28ez) 7

[Environment](#_hxp4hhr1f9w7) 7

[Strategy](#_pde375v0bgqr) 8

[Equivalent Classes](#_n9qu3t7vzey7) 8

[Age-Discount](#_ikjhsug9g7xj) 8

[Destination-Fare](#_cda8kbqfnd5c) 8

[Boundary Values](#_okv1n8wh6ej3) 9

[Age Discount](#_5ohwh18vyz7z) 9

[Test Suite](#_gxzblx6galfh) 10

[Inputs](#_x313y9bl1koy) 10

[Test Cases](#_c89lwdn9g2qa) 10

[Outputs](#_c5og6zfs1zup) 14

[**Lessons Learned**](#_zain1bfg61tw) **15**

# Overview

## Introduction

In this particular project, we are given an example form that include fields such as Name, Age, Description, and a few buttons like OK and Cancel (**Image 1**). In addition we are also given a discount table which dictates the fare-price for each age tab (**Table 1**) and a fare table which dictates fares to a specific region (**Table 2**). Using these tables and based off the information, we are asked to answer five questions using equivalent class testing and boundary value testing concepts.



**Image 1:** User Example Form

|  |  |
| --- | --- |
|  |  |
| **Table 1:** Discount Table | **Table 2:** Fare Table |

## **Purpose**

The purpose of this project is to use the concept of “Equivalent Class Testing” and “Boundary Value Testing.” For us to understand the concept of what equivalent class and boundary value testing, we need to correctly define equivalent class and gather enough test vectors to cover all partitions and scenarios. By gathering enough test vectors, we are able to perform equivalence testing and draw a clear Equivalent Partition Diagram. In addition, the project also allows us to practice how to write a Test Oracle and generate an Output - Input relationship based on the test vectors we present and scenarios we cover.

# Question I

The test values that our team will input for the age field and the expected output is written in tabular format and can be seen in **Table 3**.

|  |  |  |
| --- | --- | --- |
| **Test Input** | **Expected Result** | **Equivalence Class** |
| -1 | Invalid age | Invalid |
| 0 | Travel for free | Age group: 0-4 |
| 4 | Travel for free | Age group: 0-4 |
| 5 | 50% discount | Age group: 5-15 |
| 15 | 50% discount | Age group: 5-15 |
| 16 | Full price | Age group: 16-64 |
| 64 | Full price | Age group: 16-64 |
| 65 | 25% discount | Age group: 65 or older |
| 99 | 25% discount | Age group: 65 or older |

**Table 3:** Test Input Values

# 

# 

# Question II

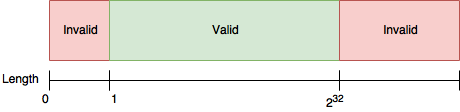
Based on several input values for our Name and Destinations, the expected outputs are of the following shown in Table 4.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Destination** | **Test Name** | **Expected Result Before Discount** | **Equivalence Class** |
| Manchester | John Doe | £42.30 | Manchester |
| Manchester | 123 | Invalid name | Invalid |
| Liverpool | John Doe | £42.30 | Liverpool |
| Liverpool | John\_Doe | Invalid name | Invalid |
| Warrington | “”(empty) | Invalid name | Invalid |
| Warrington | John Doe | £42.30 | Warrington |
| Glasgow | !@##$ | Invalid name | Invalid |
| Glasgow | John Doe | £61.80 | Glasgow |
| Edinburgh | John D0E | Invalid name | Invalid |
| Edinburgh | John Doe | £61.80 | Edinburgh |
| Cardiff | John Doe | £31.10 | Cardiff |
| Cardiff | Jo\_hn Doe 23! | Invalid name | Invalid |
| Swansea | John Doe | £31.10 | Swansea |
| Swansea | JohnDoe | £31.10 | Swansea |
| Newport | John Doe | £31.10 | Newport |
| Newport |  | Invalid name | Invalid |

**Table 4:** Expected Output Values

# Question III

## Equivalent Class Partition Diagram

[](https://www.draw.io/#G1UzSCkqomiRPjpIFTKRqCO-6J8Y6JZs09)

**Diagram I:** Name Field Partition Diagram



**Diagram II:** Age Field Partition Diagram

### Boundaries

|  |  |
| --- | --- |
| Field | Boundaries |
| Name | No name can be empty or have more than 232 characters |
| Age | * From 0 - 4 years old : 100% Discount * From 5 - 15 years old: 50% Discount * From 16 - 64 years old: No Discount * From 64 and older: 25% Discount |

# Question IV

## Test Oracle

### Environment

The hardware used for programming and testing is a Macbook Pro 13.3 from 2015, with a 2.7GHz Intel Core I5 processor, 8GB 1867MHz DDR4 of memory. The software used for programming and testing is OS X High Sierra version 10.13.3. The program is written and tested in Java code using the Eclipse version Oxygen, with Java version 8 build 1.8.0\_144. The tests are run in Eclipse IDE.The program used for creating the Equivalent Class Partition diagram was draw.io (a plugin by Google) and Code2flow ( <https://code2flow.com/app> ) for verification of the diagram through a third party software.

### Strategy

The strategies that we perform to test the example is that the tester will open the Eclipse IDE and execute the code in the console. Secondly, the tester will put in several values and execute the test cases. Lastly, the tester will verify that all test cases will have a pass rate of 100%.

## 

## Equivalent Classes

### Age-Discount

|  |  |
| --- | --- |
| **Age** | **Discount** |
| 0 - 4 years old | 100% |
| 5 - 15 years old | 50% |
| 16 - 64 years old | 0% |
| 64+ years old | 25% |

### Destination-Fare

|  |  |
| --- | --- |
| **Destination** | **Fare** |
| Manchester | £42.30 |
| LiverPool | £42.30 |
| Warrington | £42.30 |
| Glasgow | £61.80 |
| Edinburgh | £61.80 |
| Cardiff | £31.10 |
| Swansea | £31.10 |
| Newport | £31.0 |

## Boundary Values

### Age Discount

|  |  |
| --- | --- |
| **Age** | **Boundary** |
| -1, 0 | Invalid and 0-4 |
| 4, 5 | 0-4 and 5-15 |
| 15, 16 | 5-15 and 16-64 |
| 64, 65 | 16-64 and 65+ |

## Test Suite

A test suite is a set of tests that tend be in done in units. Each test suite helps validate if the program is working the way that it is expected to. In **Table 1**, shows a table consisting of our test cases that our team has written for our Student average grade program.

### Inputs

We will use 3 inputs in order for us to apply Equivalence Class Testing and Boundary Value Testing. Our inputs can be found in Table 5.

|  |  |
| --- | --- |
| **Input** | **Description** |
| Age | A positive integer that represents the age of the customer |
| Name | A String without any numbers or special characters that represents the name of the customer |
| Destination | A String that matches 1 out of the 8 destinations that is explicitly stated in the destination-fare equivalent class |

**Table 5:** Test Input Values

### Test Cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Purpose** | **Name Input** | **Age Input** | **Input**  **Destination** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | Test Age Domain | John Doe | -1 | Manchester | Error | Error | Pass |
| 2 | Test Age Domain | John Doe | 3 | Glasgow | 0.00 | 0.00 | Pass |
| 3 | Test Age Domain | John Doe | 15 | Glasgow | 30.9 | 30.9 | Pass |
| 4 | Test Age Domain | John Doe | 30 | Cardiff | 61.8 | 61.8 | Pass |
| 5 | Test Name Error | John Doe | 66 | Cardiff | 46.35 | 46.35 | Pass |
| 6 | Test Name Error | John\_\_Doe | 30 | Cardiff | Error | Error | Pass |
| 7 | Test Name Error | John Doe24 | 30 | Manchester | Error | Error | Pass |
| 8 | Test Name Error | [empty string] | 30 | Manchester | Error | Error | Pass |
| 9 | Test Name Error | John@gmail.com | 30 | Manchester | Error | Error | Pass |
| 10 | Testing discounts | John Doe | 0 | Glasgow | 0.00 | 0.00 | Pass |
| 11 | Testing discounts | John Doe | 0 | Manchester | 0.00 | 0.00 | Pass |
| 12 | Testing discounts | John Doe | 10 | Glasgow | 30.9 | 30.9 | Pass |
| 13 | Testing discounts | John Doe | 10 | Manchester | 21.15 | 21.15 | Pass |
| 14 | Testing discounts | John Doe | 22 | Glasgow | 61.8 | 61.8 | Pass |
| 15 | Testing discounts | John Doe | 22 | Manchester | 42.3 | 42.3 | Pass |
| 16 | Testing discounts | John Doe | 75 | Glasgow | 46.35 | 46.35 | Pass |
| 17 | Testing discounts | John Doe | 75 | Manchester | 31.725 | 31.725 | Pass |
| 18 | Testing large ages | John Doe | 8039 | Glasgow | 46.35 | 46.35 | Pass |
| 19 | Testing large ages | John Doe | 259 | Manchester | 31.725 | 31.725 | Pass |
| 20 | Testing large ages | John Doe | 102 | Manchester | 31.725 | 31.725 | Pass |
| 21 | Boundary values | John Doe | -1 | Manchester | Error | Error | Pass |
| 22 | Boundary values | John Doe | 0 | Manchester | 0.00 | 0.00 | Pass |
| 23 | Boundary values | John Doe | 4 | Manchester | 0.00 | 0.00 | Pass |
| 24 | Boundary values | John Doe | 5 | Manchester | 21.15 | 21.15 | Pass |
| 25 | Boundary values | John Doe | 15 | Manchester | 21.15 | 21.15 | Pass |
| 26 | Boundary values | John Doe | 16 | Manchester | 42.3 | 42.3 | Pass |
| 27 | Boundary values | John Doe | 64 | Manchester | 42.3 | 42.3 | Pass |
| 28 | Boundary values | John Doe | 65 | Manchester | 31.725 | 31.725 | Pass |

**Table 6:** Test Cases

Question V

Unfortunately there’s not enough information to be gathered to know exactly what the input vector will be especially when given when the final output price is an error. Our examples that cover all possible scenarios can be found in **Table 7**.

### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Output** | **Name** | **Age** | **Location** |
| 1 | 42.30 | John Doe | Between 16 - 64 | Manchester, Liverpool, or Warrington |
| 2 | 31.73 | John Doe | 64+ | Manchester, Liverpool, or Warrington |
| 3 | 31.73 | John Doe | Between 5 - 15 | Manchester, Liverpool, Warrington |
| 4 | 0 | John Doe | Between 0 - 4 | Manchester, Liverpool, Warrington, Glasgow, Edinburgh, Cardiff, Swansea, Newport |
| 5 | 0 | John Doe | Between 0 - 4 | Manchester, Liverpool, Warrington, Glasgow, Edinburgh, Cardiff, Swansea, Newport |
| 6 | 30.90 | John Doe | Between 5 - 15 | Glasgow or Edinburgh |
| 7 | 46.35 | John Doe | 64+ | Glasgow or Edinburgh |
| 8 | 61.80 | John Doe | Between 16 - 64 | Glasgow or Edinburgh |

**Table 7:** Outputs for All Possible Scenarios

As a result, we cannot precisely predict the input vector. It is hard to say which location the user will go to with fare of £42.30. They can end up in Manchester, Liverpool, or Warrington. Likewise, we cannot pinpoint the specific destination of Glasgow and Edinburgh with a fare of £61.80 or pinpoint the specific destination of Cardiff, Swansea, and Newport with a fare of £31.10. Lastly, it is proven that it is more difficult when the fare is £0, the user’s age is between 0-4 because the destination can be of any kind.

# **Lessons Learned**

There were many lessons that our team learned during this project. The first lesson that our team learned was how to correctly define equivalent classes. Another lesson that we learned was generating test vectors in order cover all partitions and cover all scenarios. Something else that our team was able to learn was how to draw a Equivalent Partition Diagram based on our test vectors. Lastly, we also practiced working on creating test cases in the form of an Oracle.

# 

## 

## 