IT314 - Software Engineering

Lab Session IX – Specification based testing

Group 3:

Chirag Gupta-201801188
Arkaprabha Banerjee-201801408
Rahil Shah-201801252
Kartavi Shah-201801426
Archit Agrawal-201801043
Amruthsai Jilla -201801069
Meet Patel-201801415
Bhagyesh Ganatra-201801047
Ridham Suvagiya-201801006
Udit Meena -201801095

Q.1.

Equivalence classes:

For Day:

- D1: Day is less than 1
- D2: Day is between 1 to 28 (inclusive)
- D3: Day is 29
- D4: Day is 30
- D5: Day is 31
- D6: Day is greater than 31

For Month:

- M1: Month is less than 1
- M2: Month is Feb(2)
- M3: Month is from {April(4),Jun(6),Sep(9),Nov(11)}
- M4: Month is from {Jan(1),Mar(3),May(5),July(7),Aug(8),Oct(10),Dec(12)}
- M5: Month is greater than 12

For Year:

- Y1: Year less than 1900
- Y2: Year is non leap year
    - if(year%400 == 0) false;
      else if (year%100==0) true;
      else if(year%4==0) false;
      else true;
- Y3: Year is between 1900 and 2015
- Y4: Year is greater than 2015

The total number of permutations would be the product of the number of subclasses in each equivalence class : 6*5*4 = 120. However a lot of them would come under the valid condition owing to the common intersection region hence the table below has been designed accordingly.

No.	Day	Month	Year	Result
Day is <1	D1	any	any	Invalid
Day is >31	D6	any	any	Invalid

Month is <1	any	M1	any	Invalid
Month is >12	any	M5	any	Invalid
Year is <1990	any	any	Y1	Invalid
Year is >2015	any	any	Y4	Invalid
Day is 29 for feb and non leap year	D3	M2	Y2	Invalid
Day is 30 for feb	D4	M2	any	Invalid
Day is 31 for feb	D5	M2	any	Invalid
Day is 31 for M3 month	D5	M3	any	Invalid
Rest of 110 permutation	-	-	-	Valid

The total number of test cases
Boundary Value Analysis: For each equivalence class (sub classes as well), 4 values for the extremity are to be considered. Owing to the huge number of test cases only the most important ones have been highlighted.

No.	Day	Month	Year	Expected Op.
1	1	6	2000	31-5-2000
2	2	6	2000	1-6-2000
3	15	6	2000	14-6-2000
4	30	6	2000	29-6-2000
5	31	6	2000	Invalid
6	15	1	2000	14-1-2000
7	15	2	2000	14-2-2000

8	15	11	2000	14-11-2000
9	15	12	2000	14-12-2000
10	15	6	1900	14-6-2016
11	15	6	1901	14-6-1901
12	15	6	2014	14-6-2014
13	15	6	2015	14-6-2015
14	1	1	1900	Invalid
15	1	3	2000	29-2-2000
16	1	3	2001	28-2-2001

Q2
You are testing an e-commerce system that sells products like caps and jackets. The problem is to create functional tests using boundary-value analysis and equivalence class partitioning techniques for the web page that accepts the orders. A screen prototype for the order-entry web page is shown below.

The system accepts a five-digit numeric item ID number from 00000 to 99999. The system accepts a quantity to be ordered, from 1 to 99. If the user enters a previously ordered item ID and a 0 quantity to be ordered, that item is removed from the shopping cart. Based on these inputs, the system retrieves the item price, calculates the item total (quantity times item price), and adds the item total to the cart total. Due to limits on credit card orders that can be processed, the maximum cart total is $999.99

Equivalence Classes:

Valid Equivalence Classes:

1. 00000 <= id <= 99999,
   1 <= qty <=99,
   $0 <= cart total <= $999.99,

2. 00000 <= id <= 99999
   Qty =0,
   $0 <= cart total <= $999.99

Invalid Equivalence Classes:

1. id<0 (i.e. a negative number)
2. Id is not exactly 5 digit
3. qty<0 or Card Total < $0
4. Qty >= 100
5. Card total >= $1000

Functional Tests using Boundary-Value Analysis and Equivalence Class Partitioning

Equivalence Class	**Test Case**	**Expected Output**
00000 <= id <= 99999, 1 <= qty <=99, $0 <= cart total <= $999.99	Id = 12345 Qty = 53	**(Valid)** Card Total
00000 <= id <= 99999 1 <= qty <=99, $0 <= cart total <= $999.99,	**Id=00000** Qty = 50	**(Valid)** Card Total
00000 <= id <= 99999 1 <= qty <=99, $0 <= cart total <= $999.99,	**Id=99999** Qty = 50	**(Valid)** Card Total
Id is not exactly 5 digit	**Id=100000**	**Invalid ID**, has to be exactly 5 digit
Id is not exactly 5 digit	**Id=9999**	**Invalid ID**, has to be exactly 5 digit
id<0 (i.e. negative number)	**Id=-1**	**Invalid ID**,has to be

		positive
00000 <= id <= 99999, 1 <= qty <=99, $0 <= cart total <= $999.99	Id = 12345 **Qty = 1**	**(Valid)** Card Total
00000 <= id <= 99999, 1 <= qty <=99, $0 <= cart total <= $999.99	Id = 12345 **Qty = 99**	**(Valid)** Card Total
Qty >= 100	**Qty=100**	**Invalid quantity**, exceeded quantity limit
00000 <= id <= 99999, **Qty =0,** $0 <= cart total <= $999.99	Id = 12345 **Qty = 0**	**(Valid)** Remove from cart
qty<0 or Card Total < $0	Id = 12345 **Qty = -1**	**Invalid quantity**, quantity is negative
00000 <= id <= 99999 1 <= qty <=99, **Card total = $999.99**	Some ID and Some qty such that **Card total = $999.99**	**(Valid)** Card Total
Card total >= $1000	Id and qty such that **Cart total > $999.99**	**Invalid total,** exceeded credit card limit