Dhirubhai Ambani
Institute of Information and Communication Technology

Renting System

Group 6

Subject: Software Engineering

Prof. Saurabh Tiwari

Specification-based Test Case Generation

Name	**ID**
Prit Patel	201801261
Parth Santoki	201801257
Prayush Dave	201801021
Mihir Doshi	201801152
Utkarsh Asari	201801185
Nilay Mendpara	201801124
Ritika Lakdawala	201801226
Rishi Shivani	201801073
Chauhan Rahul	201801209
Dishant Goti	201801449

Contributions:

Mihir, Dishant, Ritika, Nilay, Rishi: worked on the first problem of building test cases for the previous date problem using Boundary value Analysis and Equivalence class partition Technique and also, made a working code for the problem.

Prit, Parth, Prayush, Utkarsh, Rahul: Worked on the second problem of building test cases for the eCommerce site's cart feature with Boundary value Analysis and Equivalence class partition Technique.

Q 1.

Consider a program for determining the previous date. Its input is triple of day, month and year with the following ranges 1 <= month <= 12, 1 <= day <= 31, 1900 <= year <= 2015.The possible output dates would be the previous date or invalid date. Design the equivalence class test cases?

Write a set of test cases (i.e., test suite) – specific set of data – to properly test the programs. Your test suite should include both correct and incorrect inputs.

1. Enlist which set of test cases have been identified using Equivalence Partitioning and Boundary Value Analysis separately.
2. Modify your programs such that it runs on eclipse IDE, and then execute your test suites on the program. While executing your input data in a program, check whether the identified expected outcome (mentioned by you) is correct or not.

Answer:

Equivalence Classes:

1. Days:
    a. D1: 1 <= Days <= 28
    b. D2: Days = 29
    c. D3: Days = 30
    d. D4: Days = 31
    e. D5: Days < 1
    f. D6: Days > 31
2. Month:
    a. M1: Months which have 31 days (1,3,5,7,8,10,12).
    b. M2: Months which have 30 days (4,6,9,11).
    c. M3: February Month.
    d. M4: Month < 1
    e. M5: Month > 12
3. Year:
    a. Y1: Leap Years Between 1900 to 2015
    b. Y2: Non-leap years Between 1900 to 2015
    c. Y3: Year > 2015
    d. Y4: Year < 1900

- **Test cases for Boundary Value Analysis**

Boundary Values	Day	Month	Year	Program Output
1	<1	ANY	ANY	INVALID
2	>31	ANY	ANY	INVALID
3	ANY	<1	ANY	INVALID
4	ANY	>12	ANY	INVALID
5	ANY	ANY	<1900	INVALID
6	ANY	ANY	>2015	INVALID

- **Test Cases for Equivalence Classes**

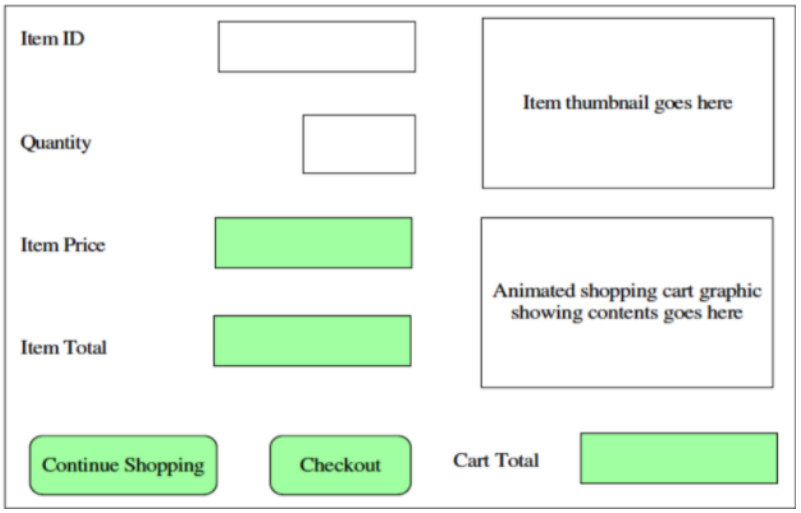
Test Case	Day	Month	Year	Program Output
1	D5	ANY	ANY	INVALID
2	D6	ANY	ANY	INVALID
3	ANY	M4	ANY	INVALID
4	ANY	M5	ANY	INVALID
5	ANY	ANY	Y3	INVALID
6	ANY	ANY	Y4	INVALID
7	D1	M1	Y1	PREVIOUS DATE
8	D2	M1	Y1	PREVIOUS DATE
9	D3	M1	Y1	PREVIOUS DATE
10	D4	M1	Y1	PREVIOUS DATE
11	D1	M1	Y2	PREVIOUS DATE
12	D2	M1	Y2	PREVIOUS DATE
13	D3	M1	Y2	PREVIOUS DATE
14	D4	M1	Y2	PREVIOUS DATE
15	D1	M2	Y1	PREVIOUS DATE

16	D2	M2	Y1	PREVIOUS DATE
17	D3	M2	Y1	PREVIOUS DATE
18	D4	M2	Y1	INVALID
19	D1	M2	Y2	PREVIOUS DATE
20	D2	M2	Y2	PREVIOUS DATE
21	D3	M2	Y2	PREVIOUS DATE
22	D4	M2	Y2	INVALID
23	D1	M3	Y1	PREVIOUS DATE
24	D2	M3	Y1	PREVIOUS DATE
25	D3	M3	ANY	INVALID
26	D4	M3	ANY	INVALID
27	D1	M3	Y2	PREVIOUS DATE
28	D2	M3	Y2	INVALID

Here in equivalence classes sets, sets for boundary value analysis are also included.
Link for the Code: https://ide.geeksforgeeks.org/JfuQyy3cZ3

Q 2.

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.

Answer:

On the basis of problem context, we can deduce the following constraints:

- ID: 00000 - 99999
- Quantity: 1 - 99
- Cart total : <= 999.99

Equivalence Classes:

1. Item ID
    a. A1: 0 <= ItemID <= 99999 - valid
    b. A2: ItemID < 0 - Invalid
    c. A3: ItemID > 99999 - invalid
2. Quantity
    a. B1: 1 <= Quantity <= 99 - valid
    b. B2: Quantity > 99 - Invalid
    c. B3: Quantity < 0 - Invalid
    d. B4: Quantity = 0 - valid

Test Case	Item ID	Quantity	Expected Output
1	A2	ANY	Invalid
2	A3	ANY	Invalid
3	ANY	B2	Invalid
4	ANY	B3	Invalid
5	A1	B1	Valid (Cart Total <= 999.99)
6	A1	B1	Invalid (Cart Total > 999.99)
7	A1	B4	Valid (Item will removed from a cart) [if Item with ID was purchased previously]
8	A1	B4	Invalid [if Item with ID was NOT purchased previously]