

JAHANGIRNAGAR UNIVERSITY

Institute of Information Technology



ASSIGNMENT 2 (SPECIAL)

PMIT 6111 SOFTWARE TESTING & QUALITY ASSURANCE

Summer Semester 2023 Intake

Date of Submission: September 16, 2023

Submitted by

HASAN TAHSIN RAFSAN

Professional Masters in Information Technology (PMIT) Program

1st Trimester Regular Batch - Section A

ID 232137

Submitted to

DR. FAHIMA TABASSUM

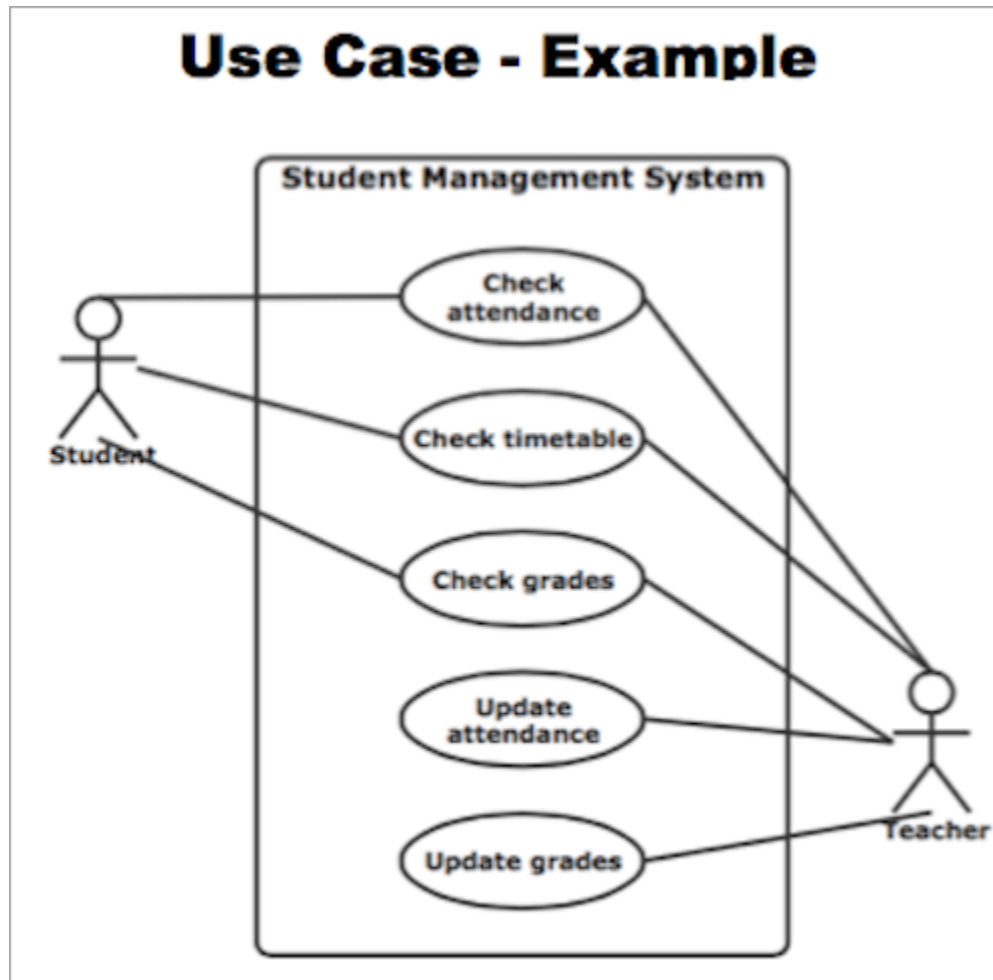
Professor

Institute of Information Technology

Assigned Tasks

Consider the following use case diagram given below. Complete the following task based on the use case diagram.

1. Identify at least two more use cases for one or both the actors (student and teacher)
2. Write test cases for "check grades" use case.
3. Draw a flowchart for anyone of the following use cases. Then find coverages for three inputs.
4. If possible, apply any of the following test techniques on the any use case you write in 1.
 - a. EP with BVA
 - b. Decision Table



Answer 1

The diagram represents a Student management system. Here the two actors in the system are Teacher and Student. It represents the particular functionality of the entire system with a total of five use cases.

Here, each actor interacts with a specific use case. First actor “student” can check his/her attendance. He/she can also check his/her timetable and grades on the system or that application. This actor can perform only these three interactions with the system even though other use cases remain in the system. However, various use cases are also remaining in the system. Although each actor does not need to interact with all the use cases, it can happen.

In the diagram, the name of the second actor is “Teacher”. It is an actor that can interact with all the functionalities of the entire system. The teacher actor is also able to update the student's grades as well as attendance. These interactions of the student, as well as teacher actors together, summarize the whole student management application.

Answer 2

TITLE	Check grades
DESCRIPTION	Actor can check the grades for the student ID
PRE-CONDITIONS	Actor should enter student ID
ASSUMPTION	Shows the grades for the respective student ID
PRIMARY ACTOR	Student & Teacher
TEST STEPS	Navigate to the System. Actor enters the Student ID of registered user Click Enter or Next
EXPECTED RESULT	The system output shows the grades of student ID, when the actor enters the correct student ID. If the actor enters the wrong student ID, then the system will output error message and prompts the actor to enter the student ID again.
FREQUENCY OF USE	Student will use it after exams to check the grades. The teacher may use it rarely to check the student's progress.

Answer 3

Input

0700: Invalid

1400: Accepted

2200: Invalid

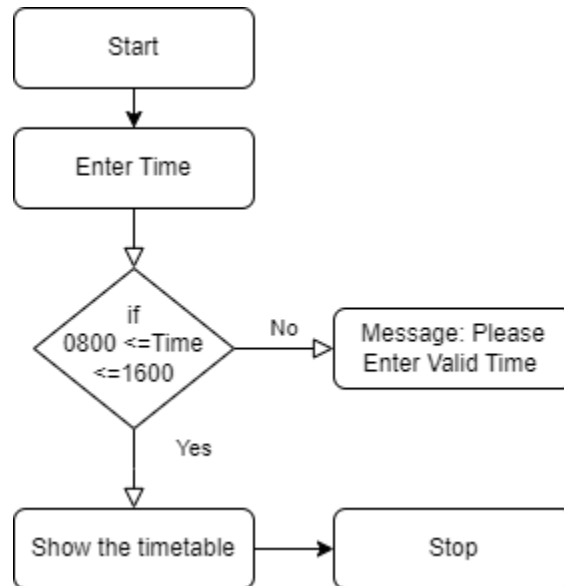


Fig: Flowchart of Use Case “Check Timetable” Here valid time is 08:00 to 16:00 for both actors

Coverage is defined as the number of items covered in testing divided by the total number of items. Coverage, therefore, defines the extent of code that has been tested out of the total code in the system. It can be a statement, branch, condition, multiple conditions, or a component.

Different inputs to the program, execute different decisions and branches. The system will be valid only when the input is greater 0800 to 1600 for second steps. As invalid input of 0700, fails the condition that time is less than or equal to 0800. Hence, an error message is displayed, and the user is prompted to enter a valid time. The second input of 1400, passes the condition that time is greater than or equal 0800 and less than or equal 1600. So it generates the timetable and reach the goal. Similarly, for third input. But it fails the second condition that time is greater than or equal to 1600. Hence, the system will stop. Different inputs execute different pieces of the code. Hence, Testers should build test cases using different input classes to ensure maximum statement, decision, and branch coverage.

Answer 4

In a. EP represents for equivalence partitioning. BVA represents for boundary value analysis. Equivalence partitioning is a method of deriving test cases when there is a large number of input data ranges. It helps to cut down exponentially on the number of cases required to test system. Where boundary value analysis is a black-box test designing technique. Here test cases are designed based on testing the boundaries between partitions and it will for both valid and invalid boundaries.

In b. Decision Table is a tabular representation of inputs versus rules/cases/test conditions. Decision table displays a combination of inputs with their associated outputs and/or actions. First one is known as “causes”, where the other one is known as “effects”.

For the use case of Student Management System lets consider **problem a**. That is EP with BVA.

In this method, for EP, classes of input conditions called equivalence classes are identified, with same kind of processing. Therefore, it leads to the generation of the same output. The implementation of equivalence partitioning includes:

Examining input and output

Dividing them into equivalence classes based on the behavior.

Inputs can be valid or positive, and invalid or negative.

And then for BVA Boundary value is an input or output value, which is on the edge of an equivalence partition or at the smallest incremental distance on either side of an edge. So we have to take that boundary level for this testing approach.

Let us consider the Check Timetable sections flow diagram. Here the valid time schedule is between 08:00 to 16:00. We can take the input as 0800, 1600. All of these four digits' data considers as clocks minute values 00 to 59 which is last two digits of every single data. Here these input will must be 4 digits. As we calculate time value so left most zero can be acceptable. System inputs have to be partitioned into “equivalence class” as follows.

For Equivalence Partitioning valid and Invalid equivalence partitions:

Invalid: < 0800

Valid: 0800 to 1600

Invalid: > 1600

For boundary value analysis test cases, at the boundary of each identified class using equivalence partitions technique, can be chosen as follows.

Category	Equivalence Class	Boundary Values
Invalid	<0800	0000,...,0759
Valid	0800 to 1600	0800, 0801,...,1559,1600
Invalid	>1600	1601,...,2359

References

1. Guru99
2. ISTQB Glossary
3. Ian Sommerville, “software Engineering”, 9th edition
4. Wikipedia
5. Edrawmax
6. Digarams.net
7. Ian Sommerville, “software Engineering” Slides
8. Slideshare
9. Class Lectures