United International University
Department of Computer Science and Engineering

CSE 4495: Software Testing and Quality Assurance
Final Examination : Fall 2023
Total Marks: 40 Time: 2 hours

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

Answer all the questions. Numbers to the right of the questions denote their marks.

1. (a) UIU is planning to implement an attendance management system for students. This system will aid in recording and processing attendance data which will increase productivity in class and during publishing of results. So far, the developers have implemented the following features in their system prototype- [3]

   i. Students can use the scanners placed at each classroom to scan their ID cards. This will automatically mark the student present for that class in UCAM.

   ii. If students forget to bring their ID Card they can login to a web portal and ask for attendance which later needs to be verified by the course teacher. If the teacher approves the attendance will be updated.

   iii. Each student will receive their attendance marks automatically based on their attendance, which relieves the faculty from calculating and publishing this mark.

   As a QA engineer you are tasked with testing this system. Mention one performance, one availability, and one security requirement that you think would be necessary for this software. Also, suggest any modification of the existing features or any additional feature that you might find useful.

   (b) Find the MTBF, MTTR and probabilistic availability of the system from following testing data obtained over 15 hours of stress testing: [3]
   Duration = 900 minutes, failures = 4, crashes = 2, start time : 12 AM

   i. Reason of failure #1: Server timeout, timestamp: 3:27 AM,

   ii. Reason of failure #2: Invalid output, timestamp: 7:35 AM.

   iii. Reason of failure #3: Server crash, timestamp: 10:30 AM. Restart required, repair time = 30 minutes

   iv. Reason of failure #4: Server crash, timestamp: 1:50 PM, Restart required, repair time = 20 minutes

   (c) Briefly describe security as a software quality attribute using *CIA* characterization. [2]

2. This is a snapshot from UIU e-LMS. When any faculty member wants to post an assignment they complete this form to control the availability and allowed file type for the assignment. Imagine you are tasked to test out whether this system level feature works properly. [6+2=8]

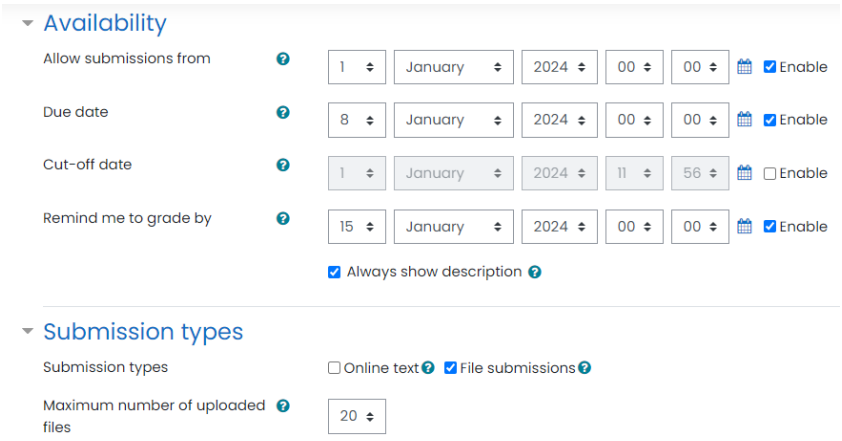

Figure 1: Snapshot for Ques-2

   (a) Identify your choices while testing this feature. Choices are aspects that you have control over that can alter the output of the program. Also enumerate representative values for your choices.

   (b) Calculate the number of test specifications for your choices.

3. You want to test out the a new game that has been launched recently. It is called "World of Witchcraft", which is an online role playing game where users design their avatar/character and roam around a virtual world. Some

Gender	Class	Skill	Hair
Male	Warrior	Sword	Brown
Female	Wizard	Bow and Arrow	Blue
	Assassin	Spear	

possible characterizations you can be apply to your character are tabulated above. Due to time limitation you cannot test out all the combinations of these attributes. Therefore, some test specification reduction technique is required.

(a) Which test selection technique is most applicable in this scenario and why? [2]

(b) Design a covering array to perform pairwise combinatorial interaction testing (CIT). [6]

4. Consider the following class diagrams for users in an online shopping platform. Here each user has a personal cart where he/she can add items to for checkout. Each user also has a membership status. As the system is in beta phase there are only two possible values for the integer User. membership_status : 0 = 'Not membership, 1 = 'VIP membership'. The difference being users with VIP membership can apply a coupon which provides a 20% discount on the cart subtotal, whereas a non-member isn't eligible to apply. Now, devise two executable test cases for testing the applyCoupon() method of User class in the JUnit notation. The test case specifications are described for you -

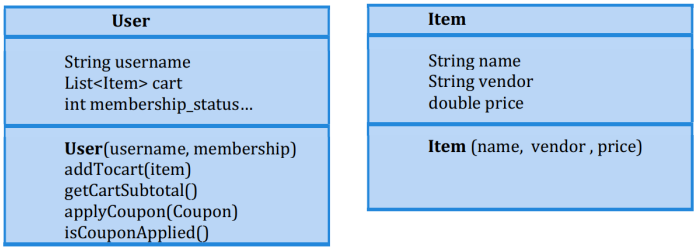

Figure 2: Class diagrams for Ques-4

(a) write a test case for a user with VIP membership. Try to add atleast three items to the cart and apply coupon. It is expected to be applied successfully. Check the cart subtotal and coupon status for consistency. You can check whether the coupon was added successfully or not with *isCouponApplied()* method. This is a boolean method reflecting the status of the cart, i.e. whether or not any coupon has been applied. [4]

(b) write a test case for a user with no membership Try to add atleast two items to the cart and apply coupon. The system should throw "*NotaMemberException*" with message *"This coupon is not applicable for you. Please upgrade to VIP."* Also check the cart subtotal and coupon status for consistency. [4]

5. Consider the following code:

```
1. public int inflections(int[] a, int n) {
2.     int v = 0; // number of inflections
3.     int d = 0; // current run direction (+/-)
4.     while (n > 1) {
5.         n = n - 1;
6.         if ((d * (a[n]-a[n-1])) < 0) // direction change
7.             v = v + 1; // => inflection point
8.         if (a[n] != a[n-1])
9.             d = a[n] - a[n-1]; // record direction
10.     }
11.    return v;
12. }
```

(a) Draw the control flow graph for the function ***inflections***. [4]

(b) Trace the control flow path in your CFG for the following test inputs: [2]
$inflections([10, -5, 6, 4], 4)$, $inflections([-10, -5, 6, -4], 3)$

(c) Does the tests mentioned above achieve branch coverage. If not design more test cases and add them to the suit so that branch coverage is achieved. [2]