United International University
Department of Computer Science and Engineering

CSE 4495: Software Testing and Quality Assurance
Final Examination : Spring 2024
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) *DangerCore softwares Inc.* has developed a new consumer application that you can use to track your daily fitness, exercise duration, calories burned etc. You are working as a crowd tester for this app and make the following observations. [3]

   i. Each user has a separate profile, but everyone can view their physiological statistics. Their is no way to hide a user's daily activity from another user.

   ii. The API communication with the server is not encrypted and user Id is exposed.

   iii. If you can find out the user id of a user you can send API requests to server pretending to be that person and the server actually executes the request.

   Which quality attribute do you think is mostly lacking in this system? Explain with proper reasoning.

   (b) A given system has the following requirements for release- availability of at least 99% and *ROCOF* less than 6 errors per 16 hour period [3]
   After the testing is done you receive the following report from your testing team - "During 10 days of testing the system processed 7560 requests.

   i. After each crash servers were restarted. Each restart took 30 minutes on average."

   ii. 97.5% of all requests were successful.

   iii. 65.1% of failed requests ended up causing a system crash.

   Now depending on this report measure the availability and *ROCOF,POFOD* of your system. Also decide whether your product meets the requirements given for release. Also Can you calculate the MTBF?

   (c) What is the difference between latency and turnaround time? Testers can design several classes of performance tests such as load testing, stress testing, spike testing, endurance testing and concurrency testing etc. Identify in which situations each of these tests are applicable for a software system? [4]

2. You have become fed up with the existing course registration system. So with your friends you decide to create a portal yourselves that will be much better. While implementing this idea you have modeled the whole process with following two classes initially. Class Student represents an individual student and has the ***register(Course c)*** and ***complete(Course c)*** methods to register for any course and to complete it. There is also a Course class where you can set any prerequisite course. As you have written all the code now you need to write unit test cases in JUnit notation to test the ***register()*** method in following scenarios-

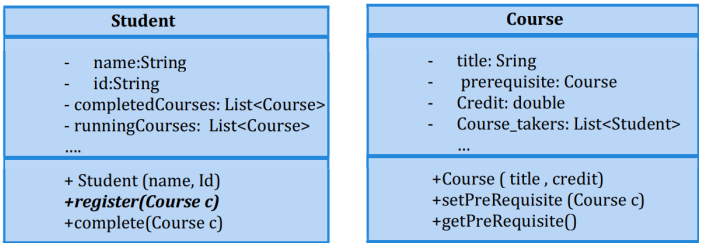

Figure 1: Class diagrams for Ques-2

   (a) First write a test case to test whether a student can register for a course after completing its prerequisite course. In this case the register method should return true. [4]

   (b) Write another test to check if it is possible to register for a course without completing it's prerequisite. In this case the register method should throw a RequirementUnsatisfiedException with the message - "Course requirement not fulfilled". [4]

3. Imagine there is a web platform called *LuggageLoaders* that enable you to buy and ship high quality products from abroad. The system enables you to create a **delivery chain**. A delivery chain can consist of up to three travelers. A

traveler is a person with the following information – Name (6-8 letters, Alphanumeric), Origin location, Destination Location, Journey start, Journey End.

Now if there exists two travelers such as –

- Name: John, Origin: New York, Destination: London, Start: 15/05/2024 8 pm, End: 16/05/2024 10 pm
- Name: Rokib, Origin: London, Destination: Dhaka, Start: 16/05/2024 12 pm, End: 16/05/2024 10 pm

These two travelers are eligible to form a delivery chain that can deliver some product from New York to Dhaka via this route – $NewYork(John) \longrightarrow London(John) \longrightarrow London(Rokib) \longrightarrow Dhaka(Rokib)$. Here you need to consider the following scenarios -

- If the destination of traveler and source of the next traveler is different they cannot form a delivery chain.
- If the time difference between first traveler's end time and the next travelers start time is less than 1 hour they might not be able to meet up to handover the delivery. In that case they are not eligible to form a delivery chain either.

So you need to write test cases to test this feature that verifies if a delivery chain is valid (Figure 2). It will show one of the following outputs –

- ***Connection Valid*** : If the travelers are eligible to create the chain.
- ***Connection short***: If the travelers are eligible but the time difference is less than one hour.
- ***Connection Invalid*** : If travelers location mismatch.
- ***Wrong Information*** : For any other error such as if traveler doesn't exist in database etc.

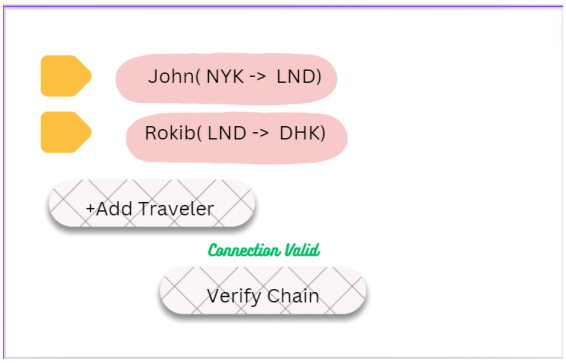

Figure 2: Snapshot for Ques-2

(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. [6]

(b) Calculate the number of test specifications for your choices. [2]

(c) Design a covering array to perform pairwise combinatorial interaction testing (CIT) for the choices tabulated below. These are choices for configuring a text editor. [4]

Character set	**Indent Style**	**Font Size**	**Bolded**
ASCII	Tabs	Large	Yes
UTF-8	Spaces	Medium	No
		Small	

4. (a) You want to test an API which updates the user data. This API is exposed by the following URL: $www.fakedata/update?id =< ID >$. A sample JSON request body contains two fields, "name" and "designation". Complete the request builder sections 1,2 and 3 in Figure 3 with appropriate data to consume this API and update the data for user_id=2 with name : "Jormund Stevensson", desgnation: "Software Tester". Write down the testing steps briefly. [5]

(b) Now you need conduct performance testing on the API mentioned in 4(a) with Jmeter. If you want to follow the load conditions shown in Figure 4 what would be your testing steps? Describe briefly with necessary details only. [5]

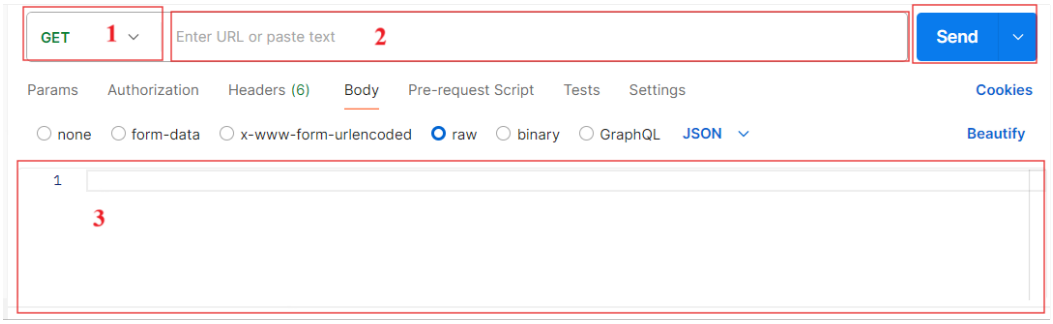

Figure 3: Postman request builder for Ques-4(a)

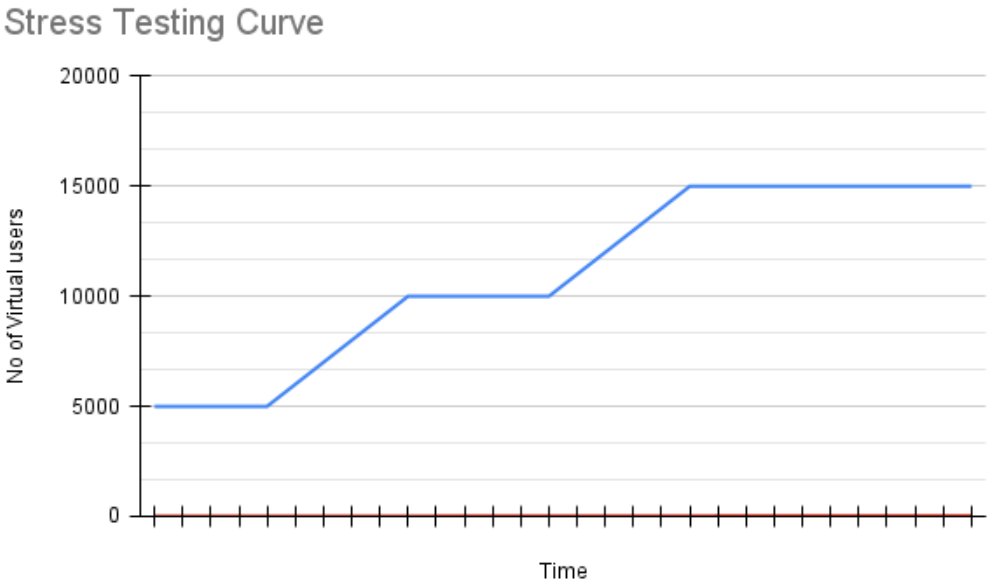

Figure 4: Load condition for Ques-4(b)