0032

National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Software Testing	Course Code:	CS4036
Degree Program:	Computer Science	Semester:	Spring 2023
Exam Duration:	180 Minutes	Total Marks:	75
Paper Date:	26th May, 2023	Weight	40
Section:	8A & 8B	Page(s):	3
Exam Type:	Final Exam		

Student : Name: [redacted] Roll No. [redacted] Section: [redacted]

Instruction/Notes:

Attempt all questions on the answer sheet
1 A4 size page (both sides) is allowed as cheat-sheet during the exam
Take Assumptions where required and note them down along with your answers.

Question #1:[10+10+3+2]

You have been asked to test the web api of a project management tool. The tool uses the micro-service architecture pattern. Following three are among the many services. A few API endpoints are also mentioned for each service:

- *Authentication Service*
  - POST /api/login
    - Request Parameters
      - '{"username":"ABCD", "password":"abcd"}'
    - Responses
      - In case of valid credentials: '{"status": "200", "token":"xxxx-xxxxxxxx-xxxx", }'
      - In case of invalid credentials: '{"status": "401", "message":"Either username or password is incorrect"}'
- *Project Service*
  - GET /api/projects
    - Request Parameters
      - '{"token":"xxxx-xxxxxxxx-xxxx", "filter":"open"}'
        - Filter field can be set to open, closed or archived.
    - Responses
      - In case of valid credentials:
        - '{"status": "200",
        - "projects": [
        - {"id": "zzzzzzzzzz","name":"board1","status":"open"}
        - {"id": "zzzzzzzzzz","name":"board2","status":"open"}
        -]}'
- *Task Service*
  - POST /api/tasks
    - Request Parameters
      - '{"token":"xxxx-xxxxxxxx-xxxx", "projectId":"zzzzzzzzzz", "summary":"Some Text","description":"Some text}'
    - Responses
      - '{"status": "200", "id":"yyyyyyy","projectId":"zzzzzzzzzz", "summary":"Some Text","description":"Some text }'

You are required to automate the integration tests for above API endpoints by covering following scenarios:

- **Scenario-1:** Make an api call to /api/login endpoint with invalid credentials and expect that the application does not return a token. Then make calls to /api/projects and /api/tasks without a token and expect that both endpoints return 401 responses.
- **Scenario-2:** Make an api call to /api/login endpoint with valid credentials and expect that the application will return a token. Then make calls to /api/projects and expect that no project is created and therefore the project array is empty in the response
- **Scenario-3:** Make an api call to /api/login endpoint with valid credentials and expect that the application will return a token. Then make calls to /api/projects and expect that list of projects is returned. Then use id of one of the projects and call the /api/tasks endpoint to create a task and expect that the task is successfully created

a) Suggest a folder structure and modules/class structure to create a modular integration testing framework that uses programing principles like KISS, DRY, single responsibility principle.

b) Automate above three scenarios using your preferred language according to the above suggested modular approach

c) Assuming that modules have been developed in the following order due to multiple practical reasons, suggest your strategy of using stubs/drivers to perform integration testing at each stage

   i) Project

   ii) Task

   iii) Authentication

d) Logically speaking, which of the above three scenarios does not need to be a part of the integration testing suite. Give reasons.

<u>**Question #2:[5+5+5]**</u>

Write load, stress and spike tests for following requirements while testing https://mywebsite.com/

i) For concurrent users upto 5000/sec with ramp up time of 10 mins and duration 30 mins
   1) P80 of response time < 1s
   2) Error rate < 0.1%

ii) For concurrent users upto 10000/sec with ramp up time of 10 mins and duration 30 mins
   1) P80 of response time < 3s
   2) Error rate < 0.5%

iii) For concurrent users upto 20000/sec with ramp up time of 30 seconds and duration 5 mins
   1) P80 of response time < 30s
   2) Error rate < 10%

Question #3:[2+10+8]

Your company "Jiggle" has a product "JDocs", where people can collaborate to manage their shared documents word/sheets. For collaboration the user can use the sharing feature of the JDocs. Currently JDocs offer following options when sharing

- You can share Files or Folders
- You can give following access rights
  - read-only permissions
  - read and write permissions
  - Owner permissions
- You can share with
  - Individuals
  - All members of an organisation
  - Anyone that has the link

a) Which technique among the following two is most appropriate for this situation to optimise your number of test cases. Explain why?
   i) Combinatorial testing
   ii) Reduced decision tables

b) Apply the above recommended technique to identify the appropriate test cases

c) Write a BDD test using gherkin scenario outline to cover the above identified test cases

Question #4 [10+5]:

```
-   void AddressBook::Open()
-   {
-      if(!fs.ExecuteOpen()) return;
-      filename = fs;
-      FileIn in(filename);
-      if(!in) {
-         Exclamation("Unable to open [* " + DeQtf(filename));
-         return;
-      }
-      array.Clear();
-      while(!in.IsEof()) {
-         Vector<Value> q;
-         for(int i = 0; i < 4; i++)
-            q.Add(in.GetLine());
-         array.Add(q);
-      }
-   }
```

a) Compute the cyclomatic complexity of the above ***open*** function in a class ***AddressBook*** and identify the paths that you will choose to achieve 100% basis path coverage. (Focus on the code structure without worrying about the logic.)

b) If you are asked to use Test Driven Development (TDD) approach then will you use base path coverage as a test-basis? Briefly explain