

**ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)  
ORGANISATION OF ISLAMIC COOPERATION (OIC)**

**Department of Computer Science and Engineering (CSE)**

**MID SEMESTER EXAMINATION**

**SUMMER SEMESTER, 2019-2020**

**DURATION: 1 HOUR 30 MINUTES**

**FULL MARKS: 75**

**SWE 4601: Software Design and Architecture**

**This is an open book exam.** You are allowed to use any printed material and online resources with a computer. You are not allowed to use your keyboard or other typing methods. Mobile phone or other touch devices are not allowed. Communicating with a person online/offline is not allowed.

There are **3 (three)** questions. Answer all of them. **Please start every answer on a new page.**  
**You can scan your answer script only after the exam time is over.**

1. a) Give an example of query optimization with column redundancy. Your example should 15 illustrate:
- A normalized database with 3 to 5 tables including relations between them.
  - A query that you are trying to optimize. It does not have to be a formal SQL query. For example, you may write: “get top 10 most expensive products”.
  - Which tables in the normalized database need to be joined for the query.
  - The same database but with column redundancy.
  - Which tables in the denormalized database need to be joined for the query.

**Notes:**

- The database should be presented in a formal or informal ER diagram.
- Use a different example than the kid’s shop database discussed in the class.
- It is sufficient to include only the columns to which are relevant to your example.

- b) Fowler attended a workshop hosted by Nilsson where the attendees voted on a few design principles related to layered architectures. The principle “Separate development teams by layer” got only 1 positive and 22 negative votes. If so, how should teams be built in a software development organization. 6

- c) When organizing source code, what are some drawbacks of using application layers (domain, data, view etc.) as top-level folders? 6

2. a) XSoft has hired you to review their new software project plan. They are planning to have two 5+5 software architects do a detail design in a form of blueprint UML. The blueprint will then be passed to four programmers to implement it.  
State your opinion about this plan in no more than 5 sentences. What alternative approach do you suggest?

- b) A year has passed and the first release of XSoft’s software is developed and deployed in a monolith style. Just like any software projects, this one also has a lot of features in the to-do list. The team is willing to migrate to microservice style. They are thinking of two options – (1) redesign the existing system to microservices and then add the new features. (2) implement the new features as microservices and gradually break the monolith into services.  
Which approach do you suggest? Explain according to applicable microservice characteristics.

- c) XSoft’s software now have a huge userbase. One of the services receives too many requests and is also very resource hungry (meaning it requires a lot of resources like CPU, memory etc), therefore, you are suggesting a horizontal scaling of this service. The service has a database associated with it.

Is this necessary to scale the database as well? Explain briefly.

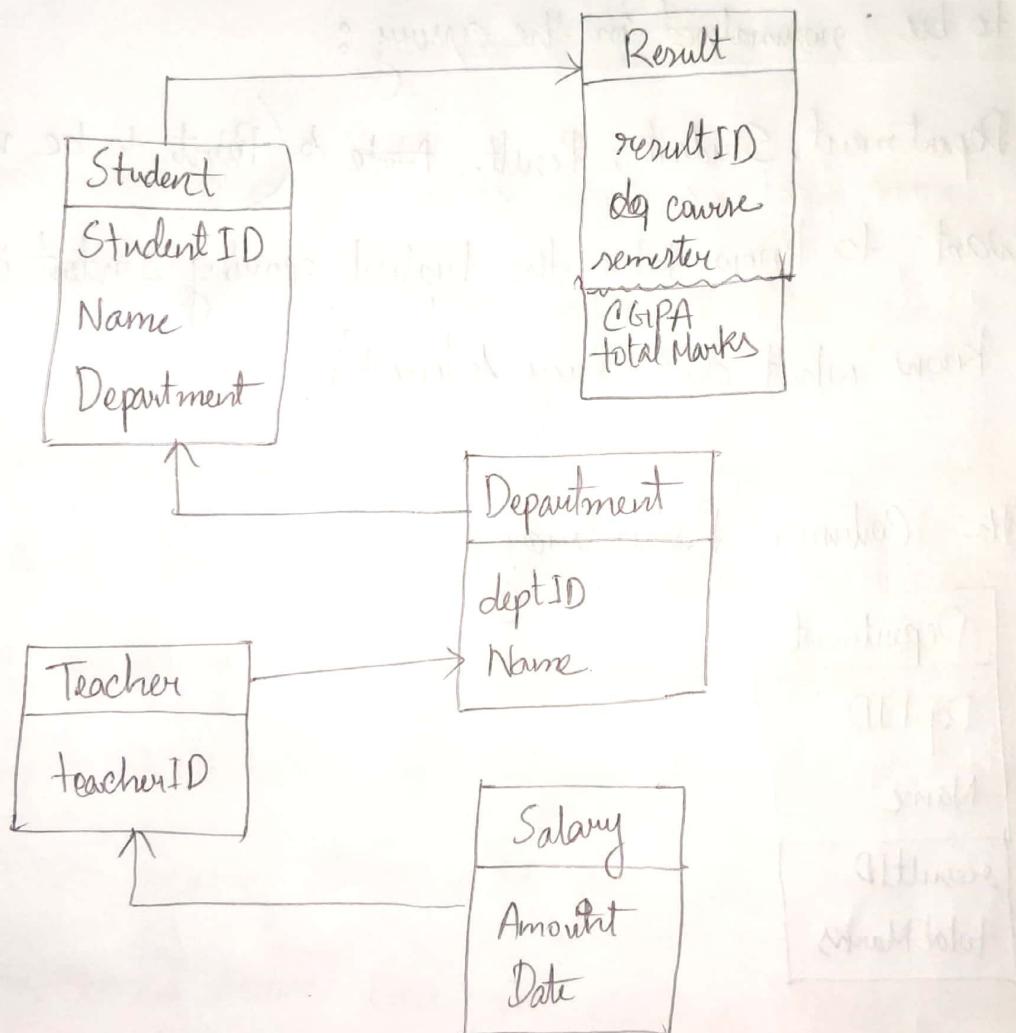
3. a) Differentiate between libraries and components in terms of microservices. What is a major benefit of components over libraries? What is a major drawback of components over libraries? 4+3
- b) According to Sadalage and Fowler's Evolutionary Database Design, is it possible to do a backward migration of production database? If yes, describe the process (only the backward migration part for production database). If no, explain why it is not a big problem. 8
- Note:** backward migration means that the application code is backdated than the database, and you need to revert the database to be compatible with the source code.
- c) In Sadalage and Fowler's Evolutionary Database Design, test data should be version controlled, however, it should not be inserted in the production database. If so, describe a process to keep the test data in version control but skip it from being inserted into production database? 7

Zahin - 170042032

51

Amt 2.1 @

Normalized DB with 3-5 tables & relations between:



The query to be optimized:

"Which dept has the student with highest score?"

Tables to be normalized in the query:

Department, Student, Result. ~~Note~~ (Point to be noted is we don't want to know who the highest scoring student is we just want to know what dept. they belong to).

DB with Column Redundancy:

Department
Dept ID
Name
resultID
Total Marks

resultID can also be replaced with student ID.

tables joined in denormalized DBs: only the Department tables give enough info.

~~Q) Tables to be joined in denormalized DB?~~

~~Result can be obtained from the Department Table only.~~

Ans to Q. 1(b)

♦ instead of dividing team by layers we can divide team based on module, with developers from all the layers in one team.

As Martin Fowler said, "developers don't have to be full stack, but ~~too~~ teams should be." In smaller systems, there might not be much difference between layer-based & domain-based approach, but as the application gets bigger, the layers start getting packed with too many things. In these ~~scenarios~~ scenarios, ~~domain~~ module-based teams face no problem because there is no change necessary in already existing team.

Ans to Q.1(c)

When organizing source code organizing them based on application layers can create the following issues:

- ① The more features are added into the software, the more complicated the hierarchy gets.
- ② If there is a bug in a module and we don't know which layer the bug is in, we have to go through all the folders, look for the specific ~~fold~~ file of that module among 10 possibly 100+ other files, which is a big hassle.

∴ it's better to use module as top-level folder can keeping the layered files (i.e. html, css, js) inside that folder in a preferable manner.

Ans to Q. 2(a)

My ~~opinion~~ opinion about the plan is given below:

- ① Since they are using blueprint UML their main focus is complete ~~work~~.
- ② the two ~~as~~ UML designers are probably senior developers & have more experience than the 4 programmers implementing it.
- ③ The process they are using forward engineering.
- ④ The UML design needs to suffice complete enough to implement
- ⑤ The blueprint should be detailed enough & straightforward so that programmers can implement them.

alternative suggestions Instead of working as 2 separate modules, the both parties can work on coding and design. That way errors can be fixed easily and one party won't continuously have to knock the other one if they get stuck. They can also use automated tools if the UML gets too tricky. Another interesting

approach can be reverse engineering, where the code is written first then as UML is used to explain it.

### Ans to Q. 2(b)

I think option 2 is a better approach. We can use the current monolith using an API and build the new feature as microservices. That way the new features can continue to use the monolith.

If we look at the microservice characteristic "Evolutionary design" in Matt Martin Fowler's article, we see that the migration from monolith to microservices need to be evolutionary, which means we implement new functionalities in new services referring to the original monolith & gradually split services from the actual monolith.

This can also be applied from the perspective of decentralized data management, where we decouple the databases.

Ans to Q. 2(c)

in case of horizontal scaling, they are increasing the service capacity by adding (CPU, memory etc). They are not adding new instances of the services. In that case, scaling of database is not absolutely necessary.

decentralized data management is one of the key characteristics of microservices. This means, every microservice should have a data storage mode of its own. So if there is a new service, it should have an instance of an old <sup>DB</sup> service or a completely new DB system. However in the current scenario we are not scaling the services vertically which means they are adding new instances of the services so they need a data storage system of their own.

### Ans to Q. 3(a)

difference between services & libraries:

Services are out-of-process components who communicate with a mechanism such as a web request or, remote call. Libraries are components that are linked to a program & called using in-memory fn calls. Services are independently deployable but components are deployed with the program. Library calls are cheaper than service calls. Service interfaces are more explicit than libraries so most language don't have a good mechanism to define them.

major benefit of services:

services can be invoked on demand. it doesn't have to be active all the time. Libraries don't have this advantage.

drawback of services:

calling services from out-of-process is an expensive operation. Library usage is cheaper.

Ans to Q. 3(b)

if we revert the database into a previous version then the current users who are using the application might not have access to all the data. This can be a problem. In that case, we need to notify the developers and collaborate necessary changes.

An alternative to this can be keeping a repository for the production database ~~as~~ that contains its previous versions, so the necessary version compatible with code can be pulled.

If we want to make large changes to the production data, then a better approach would be to ~~break~~ take the production DB down and then make changes.

Ans to Q. 3(e)

in order to keep the data in version control but skip it from being inserted in production database, we can keep separate files for test data & migration script. The script will be added in the production database, but the data can stay separated. They can → The data can also be stored in different folders, so the code doesn't get complicated.