Name: Danyal Ahmad
Class: BSSE
Course: Software Design & Architecture
ID # 13838
Module: Semester-VI
Submitted To: Aasma Khan
Spring-2020 Final-Term Examination
Date: 23/06/20

- **Attempt all questions.**
- **Marks will be given as per the DEPTH of the answer, not LENGTH.**

Q1(a):What is Software Architecture? Why is software architecture design so important?

Answer 1(a):

Software Architecture:

The software architecture of a system depicts the system's organization or structure, and provides an explanation of how it behaves. A system represents the collection of components that accomplish a specific function or set of functions. In other words, the software architecture provides a sturdy foundation on which software can be built.

A series of architecture decisions and trade-offs impact quality, performance, maintainability, and overall success of the system. Failing to consider common problems and long-term consequences can put your system at risk.

There are multiple high-level architecture patterns and principles commonly used in modern systems. These are often referred to as architectural styles. The architecture of a software system is rarely limited to a single architectural style. Instead, a combination of styles often make up the complete system.

The Importance Of Software Architecture Design :

A software architecture is the foundation of a software system. The design of the architecture is significant to the quality and long-term success of the software. A proper design determines whether the requirements and quality attributes can be satisfied.

There are a number of reasons why a good software architecture design is critical to building useful software.

1. Software architecture design is when key decisions are made regarding the architecture.

2. A software architecture design communicates the architecture to others.

3. The design provides guidance to the developers.

4. The impact of the software architecture design is not limited to technical concerns. It also influences the non-technical parts of the project.

Q1(b): Explain any four tasks of architect.

Answer 1(b):

Tasks performed by the Architect:

In the software industry, the role of a software architect is interpreted in many different ways. In some cases, an architect may work in an established enterprise company and hand down instructions on technology stacks to the developers. At the other extreme Agile development, a team may work without the involvement of an architect.

1. A software architect needs to interact with clients, product managers, and developers in order to envision, model and provide initial models and designs that can be built. This role also may cover the meeting potential or current customers.

2. A software architect has to constantly review the code to ensure the quality of the design by avoiding complexity, advocating clarity and to do this with the team. This usually requires hands-on work in terms of developing prototypes, contributing code or evaluating technologies.

3. The role of a software architect includes collaborative working with a degree of humility and providing mentoring as required. Such collaboration also allows the architect to become familiar with the skills and interests in the team and to share their knowledge with the rest of the team. Humility is required to ensure that all the team is listened to, as they

may have more specific experience or knowledge for the problem at hand.

4. Architects can use various software architectural models that specialize in communicating architecture.

Q2: Explain Architecture Business Cycle (ABC) in detail with figure.

Answer(2):

Architecture Business Cycle (ABC):

Software architecture is a result of technical, business, and social influences. Its existence in turn affects the technical, business, and social environments that subsequently influence future architectures. We call this cycle of influences, from the environment to the architecture and back to the environment, the Architecture Business Cycle(ABC).

The organization goals of architecture Business Cycle re beget requirements, which beget an architecture, which begets a system. The architecture flows from the architect's experience and the technical environment of the day.

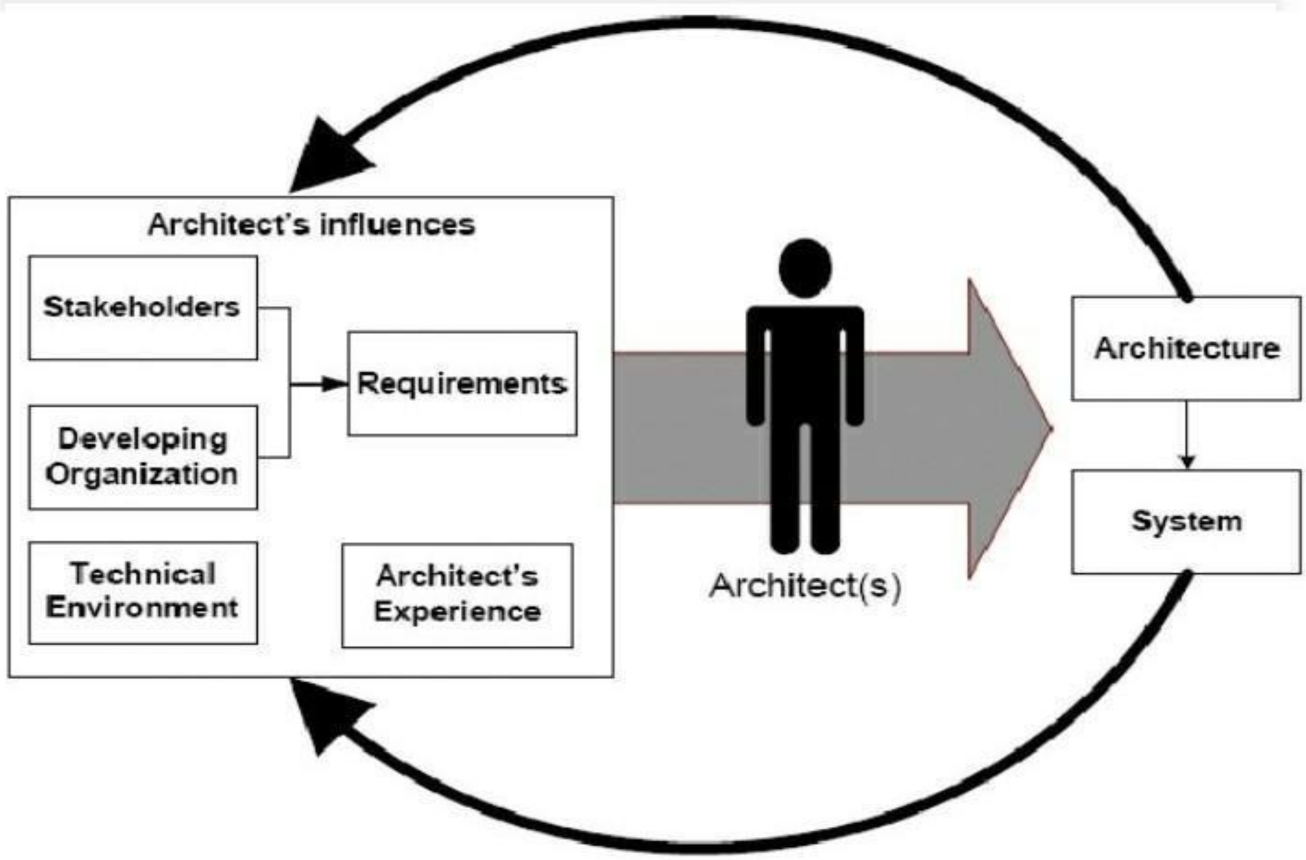
Three things required for ABC are as follows:

1. **Case studies** f successful architectures crafted to satisfy demanding requirements, so as to help set the technical playing field of the day.

2. **Methods to** assess an architecture before any system is built from it, so as to mitigate the risks associated with launching unprecedented designs.

3. **Techniques** for incremental architecture-based development, so as to uncover design flaws before it is too late to correct them.

DIAGRAM:



How The Cycle Works:

1. Architecture affects structure of developing organization

2. Architecture can affect goals of developing organization

3. Can affect customer requirements for next system

4. Process of system building will affect architect's experience with subsequent systems

5. Sometimes will influence and sometimes change software engineering culture.

Q3.Explain ABC Activities?

ANSWER(3):

Architecture Business Cycle Activities:

ABC includes the following activities:

a. Create the business case.
b. Understand the requirement.
c. Create the architecture.
d. Document & communicate the architecture.
e. Analyse the architecture.
f. Implement the system based on architecture
g. Confirms the implementation.

Creating the business case for the system :

It is simple to create a business case than understanding the needs of market How much should be the product cost? What is the Targeted market? What is the targeted time to market? Will it need to interface other system? Are there system limitations

Understanding the requirements

There are variety of techniques to understand requirements from stakeholders. Object oriented analysis: use cases & scenarios Safety Critical Systems: Finite state machine models Formal specification languages Quality attributes Prototypes Regardless of technique used, -- the desired qualities of the system to be constructed determine the shape of architecture. | Website for Students

Creating the architecture

Conceptual integrity A small no. of minds coming together to design the system's architecture.

Communicating the architecture

For effective architecture It must be communicated clearly and unambiguously to all stakeholders. Developers must understand work assignments. Testers must understand the task structures Management must understand the scheduling implications

Analyzing the architecture

Out of multiple designs, after analyzing, some design will be accepted or some are rejected. Evaluating an architecture for the qualities it supports is essential to ensure the stakeholders satisfaction (needs). Scenario- based techniques are for evaluation of architecture. | Website for Students

Implementing based on the architecture

Concerned with keeping the developers faithful to the structures. Should have an environment that assists developers in creating the architecture. Ensuring conformance to an architecture Finally, when an architecture is created and used, it goes into maintenance phase. Constant vigilance is required to ensure that actual architecture and its implementations remain faithful to each other.

Confirming the implementations

The final step in the cycle is to confirm the implementations and reviewed by a single architect or small group of architects. gather both the functional requirements and a well specified, prioritized list of quality attributes. be well documented, with at least one static view and one dynamic view. be reviewed by the system's stakeholders. be analyzed for applicable quantitative measures and formally evaluated for quality measures.

Q4. Pair programming is an agile software development technique in which……….driver should be able to issue the system to backup old work.

- **Draw a use case diagram to show all the functionality of the system.**
- **Describe in detail four non-functional requirements for the system.**
- **Give a prioritized list of design constraints for the system and justify your list and the ordering.**

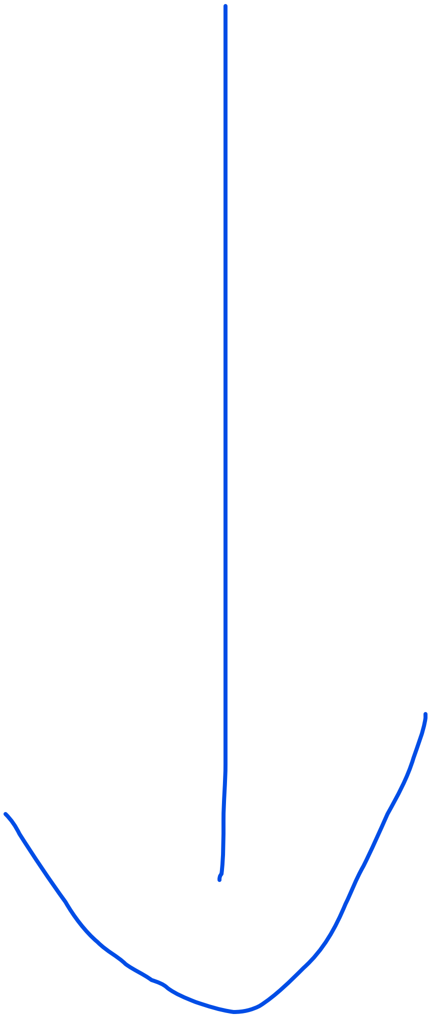
Propose a set of classes that could be used in your system and present them in a class diagram

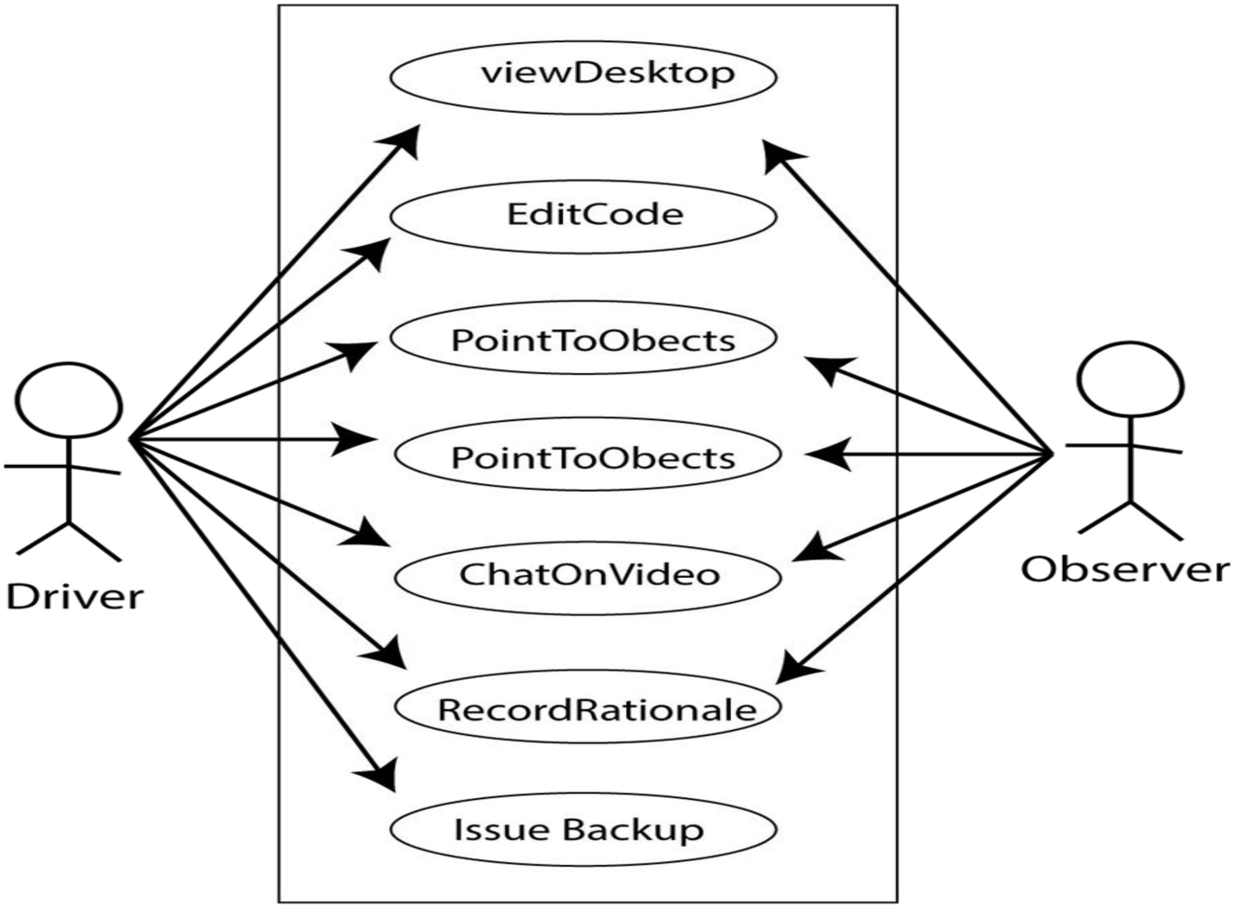
Answer(4):

Draw a use case diagram to show all the functionality of the system.

Answer:

Diagram:





🙰 **Describe in detail four non-functional requirements for the system.**

Answer:

1. Ease of use – the front-end interface must be simple and easy to use.

2. Real-time performance – the Observer should be able to see the changes made by the Driver immediately without delay; the video chat should be smooth without delay also.

3. Availability – the system should be available to both programmers all the time.

4. Reliability – the system should be reliable, i.e., it should not crash when the internet speed is slow and when the internet connection is suddenly down the user should be able to resume the session at a later time.

Give a prioritized list of design constraints for the system and justify your list and the ordering.

Answer:

We prioritized the list as follow

1. Easy to use

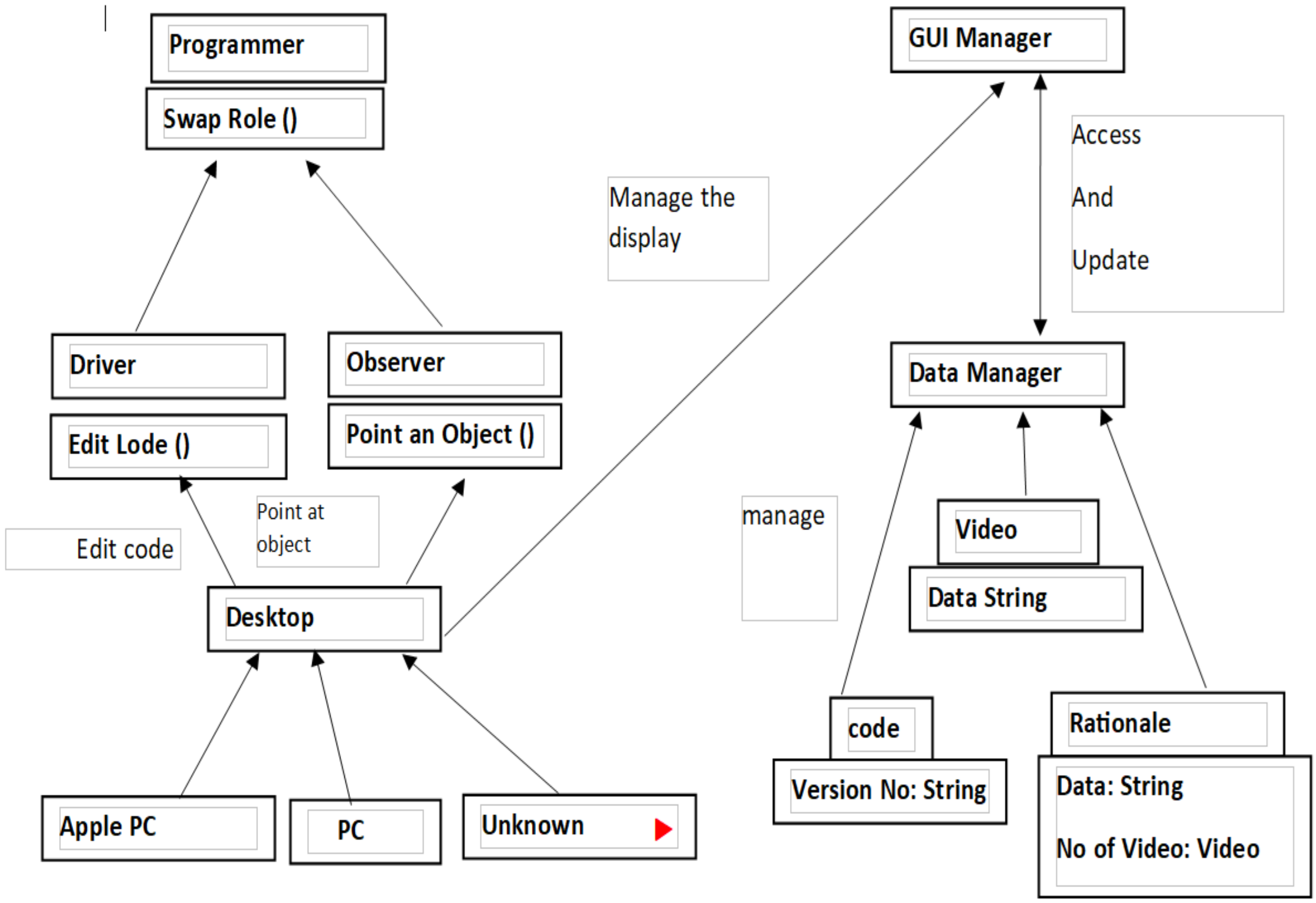
2. Availability
3. Probability
4. Cost

Our top priority is that the application is easy to use so every user can operate the application and will not have any kind of problem while working, the availability of the application for 24 hours is necessary but due to updates it might not provide 24 hour service , the probability of the device would be of great service regardless of the system specifications of the user and with less cost the user can easily acquire the product.

Example” Security”

The system must be secured is NFR. The design constraints could be user authentication and it must be in place., the communication protocol must be encrypted, and the data must be stored on a server behind firewall.

Propose a set of classes that could be used in your system and present them in a class diagram


Programmer
Swap Role ()
GUI Manager
Access
And
Update
Manage the
display
Driver
Edit Lode ()
Observer
Point an Object ()
Data Manager
manage
Point at
object
Edit code
Video
Data String
Desktop
code
Version No: String
Rationale
Data: String
No of Video: Video
Apple PC
PC
Unknown

THE END