Final -Term Examination

| | | |
|---|---|---|
| Name | : | Shahid Ali |
| Id | : | 13942 |
| Subject | : | Software Design And Architecture |
| Submitted To | : | Mrs. Aasma Khan |
| Degree | : | BS(SE) 6th semester |
| Date | : | 23/06/2020 |

**Q.1:**

**a) What is software Architecture? Why is software architecture design so important?**

**Ans: Software Architecture:.** The fundamental organization of a system embodied in its components, their relationships to each other, and to the environments, and the principles guiding its design and evolution.

**Software architecture design important:**

- A poor design may result in a deficient product that
  - does not meet system requirements,
  - is not adaptive to future requirement changes,
  - is not reusable,
  - exhibits unpredictable behavior, or
  - performs badly.

**b) Explain any four tasks of architect.**

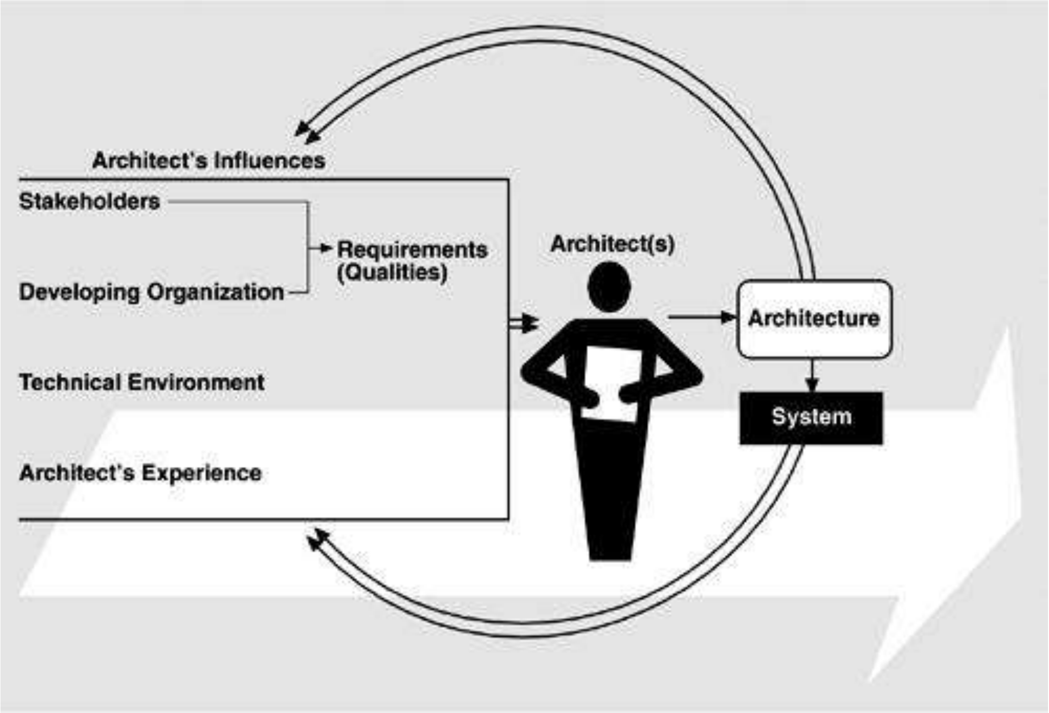
**Ans: Tasks:**

1) Perform static partition and decomposition of a system into subsystems and communications among subsystems.
2) Establish dynamic control relationships among different subsystems in terms of dataflow, control flow orchestration, or message dispatching.
3) Consider and evaluate alternative architecture styles that suit the problem domain at hand.
4) Perform tradeoff analysis on quality attributes and other nonfunctional requirements during the selection of architecture styles.

**Q.2: Explain Architecture Business Cycle (ABC) in detail with figure.**

**Ans: Architecture Business Cycle (ABC):**

Software architecture is a result of technical, business and social influences. These are in turn affected by the software architecture itself. This cycle of influences from the environment to the architecture and back to the environment is called the Architecture Business Cycle ( ABC ). 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).



1. The organization goals of Architecture Business Cycle are 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.
2. Three things required for ABC are as follows:
   a. **Case studies:** Case studies of successful architectures crafted to satisfy demanding requirements, so as to help set the technical playing field of the day.
   b. **Methods:** Methods to assess an architecture before any system is built from it, so as to mitigate the risks associated with launching unprecedented designs.
   c. **Techniques:** Techniques for incremental architecture-based development, so as to uncover design flaws before it is too late to correct them.

**Q.3: Explain ABC Activities?**

**Ans: ABC Activities:**

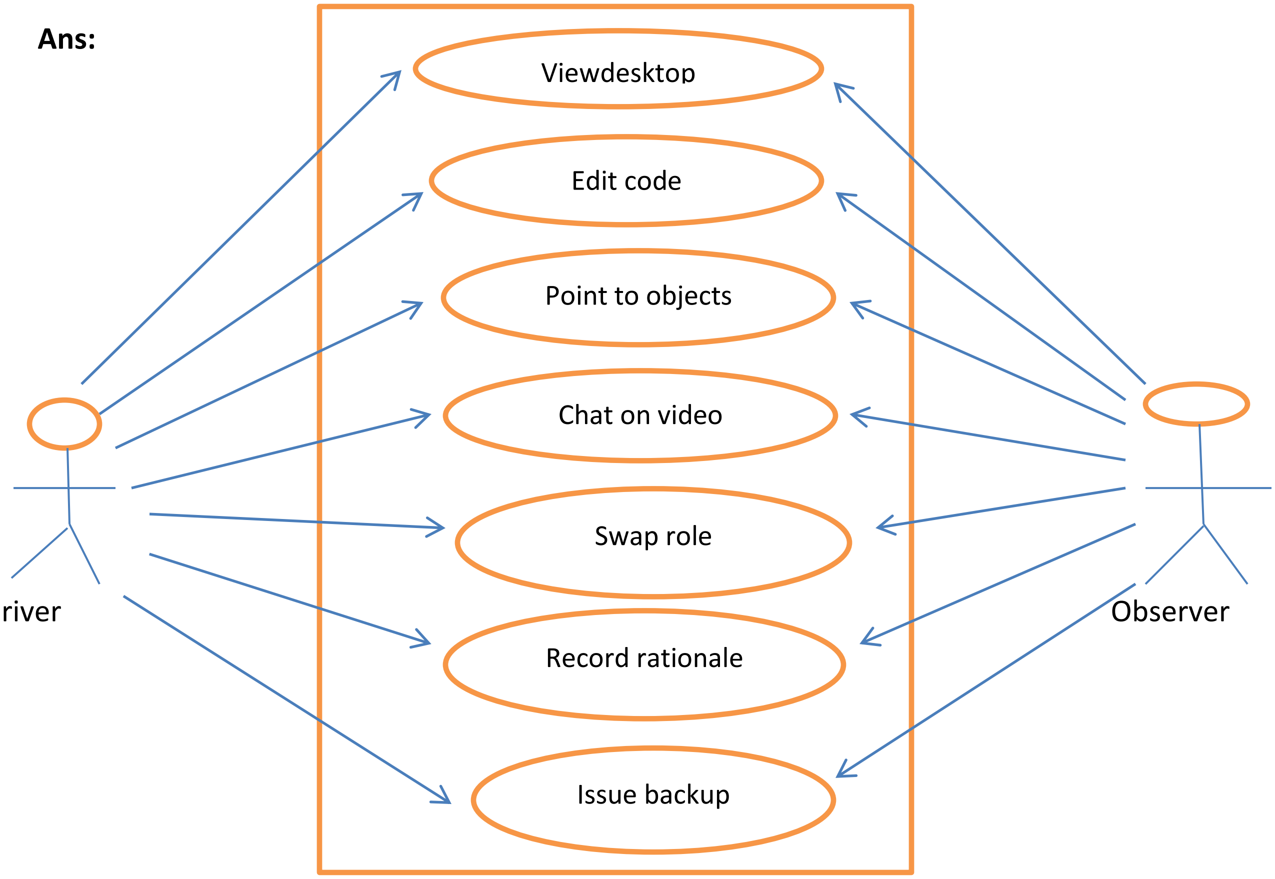
**1) Creating the business case for the system**

- Why we need a new system, what will be its cost?
- Time to market, integration with existing systems?

2) **Understanding the Requirements**
    - Various approaches for requirements elicitation i.e. , object-oriented approach, prototyping etc.
    - The desired qualities of a system shape the architecture decisions- architecture defines the tradeoffs among requirements
3) **Creating / selecting the architecture**
4) **Communicating the architecture**
    - Inform all stakeholders ( i.e.., developers, testers, managers, etc. )
    - Architecture's documentation should be unambiguous
5) **Analysing or evaluating the architecture**
    - Evaluate candidate designs
    - Architecture maps the stakeholders' requirements/ needs
6) **Implementation based on architecture**
7) **Ensuring conformance to an architecture**

**Q.4: a) Draw a use case diagram to show all the functionality of the system.**

**Ans:**



**Assumption:**

When the driver edits code, we assume that the observer can see the changes in real-time through the view desktop use case, thus there is no arrow pointing back to the observer for the edit code use case. A similar assumption is made for the point  to objects use case, so no arrow points back to the driver.

We assume that both the driver and observer can initiate the view desktop, chat video, swap role, and record rationale use cases.

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

**Ans:**

- **Ease of use-** the front-end interface must be simple and easy to use.
- **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.
- **Availability-** the system should be available to both programmers all the time.
- **Security-** the backup code should be kept securely and be protected from unauthorized access.

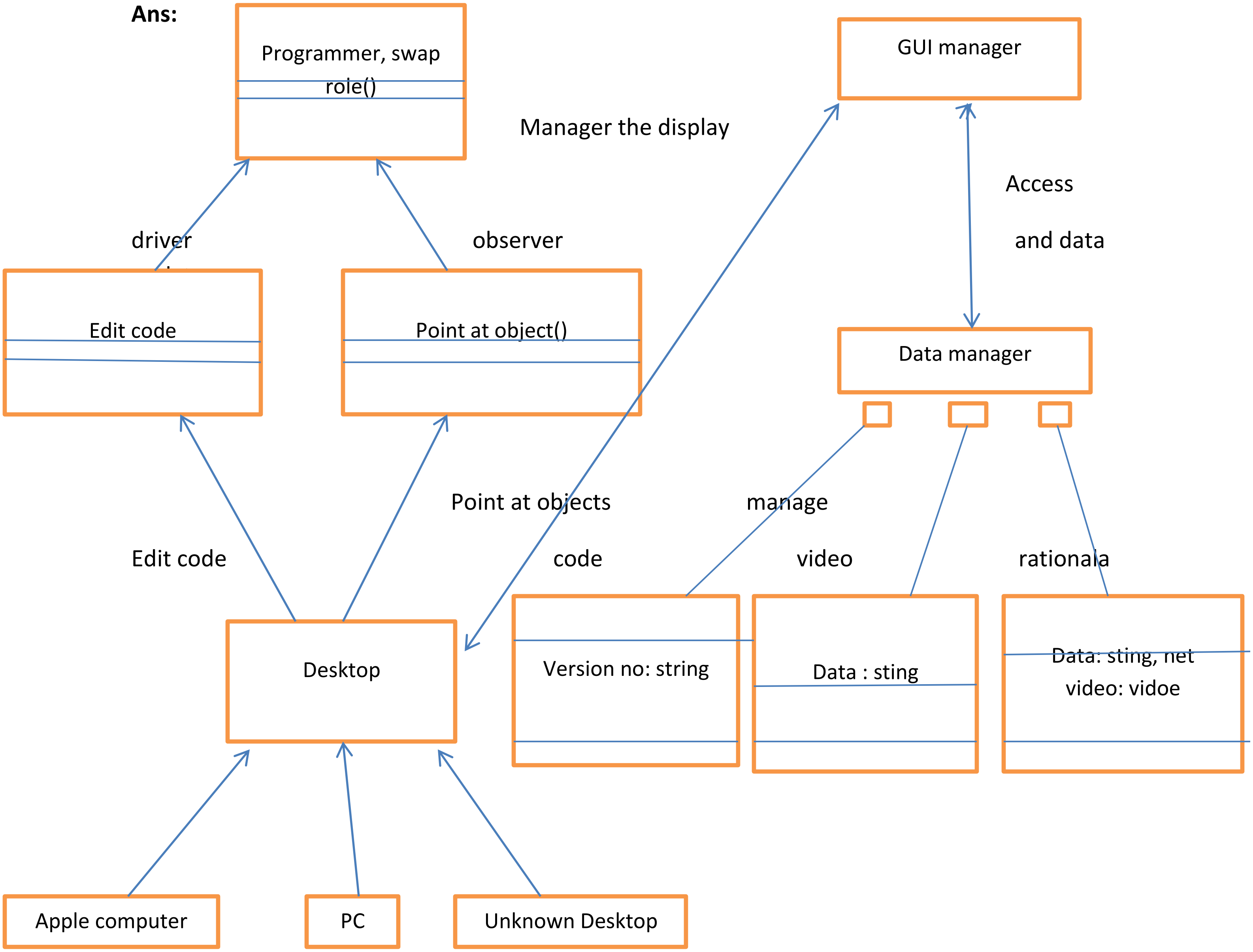
**c) Give a prioritized list of design constraints for the system and justify your list and the ordering.**

**Ans:** you should explain your ordering, E.g., if you put " ease of use" in position 1 and " real-time performance " in position 2, then justify why you think " ease of use" is more important than " real-time performance " . you should also discuss briefly whether there are conflicting design constraints as well and, based on your ordering, which design constraints should be emphasized on.

**Example:** "security- the system must be secured "  is a NFR. The design constraints could be a user authentication must be in place, the communication protocol must be encrypted, and/or the data must be stored on a server behind firewall.

**d) Purpose a set of classes that could be used in your system and present them in a class diagram.**

**Ans:**


Programmer, swap role()
GUI manager
Manager the display
Access
and data
driver
observer
Edit code
Point at object()
Data manager
Point at objects
code
manage
video
Edit code
rationala
Desktop
Version no: string
Data : sting
Data: sting, net
video: vidoe
Apple computer
PC
Unknown Desktop