

**ID:14269,**                      **DEP: BS(SE)**  
**NAME : Qazi bilal ,**      **SEM :5th**

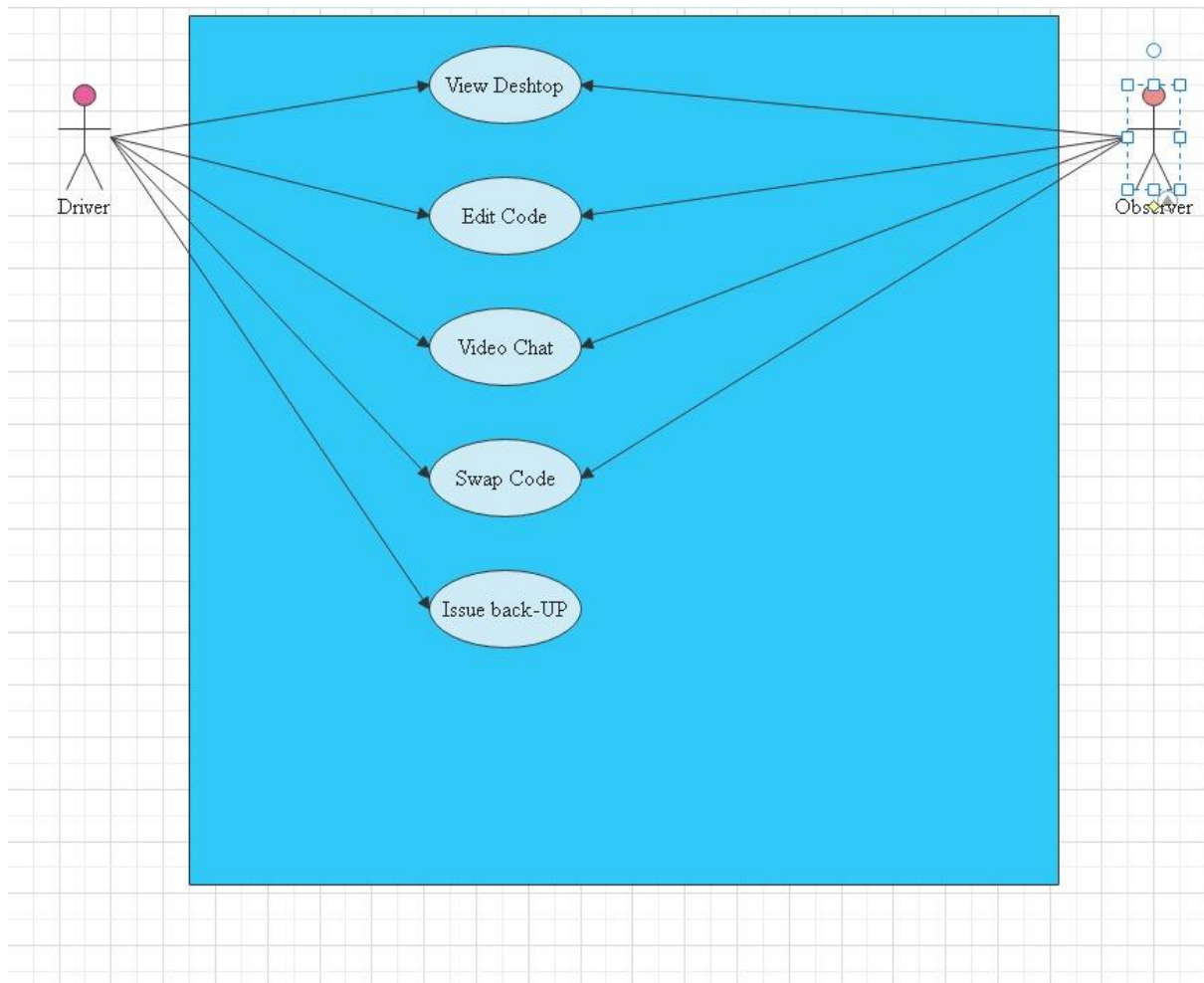
**Question No: 01**

**(20)**

Pair programming is an agile software development technique in which two programmers work together at one work station. One types in code while the other reviews each line of code as it is typed in. The person typing is called the driver. The person reviewing the code is called the observer. The two programmers switch roles frequently (possibly every 30 minutes or less).

Suppose that you are asked to build a system that allows Remote Pair Programming. That is, the system should allow the driver and the observer to be in remote locations, but both can view a single desktop in real-time. The driver should be able to edit code and the observer should be able to “point” to objects on the driver’s desktop. In addition, there should be a video chat facility to allow the programmers to communicate. The system should allow the programmers to easily swap roles and record rationale in the form of video chats. In addition, the 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.**
- It might be useful to break down huge user stories into the plumbing segment as well as a functional portion. For instance, it may be important to establish the blockchain infrastructure or write a few other automation to make the information available.
- Defects monitors issues with approved user stories. Just like with non-functional requirements, a few other deficiencies would be a greater priority than the recent feature task, but others may be fewer priority niggles.
- One or even more requirements for the security of a system and the data. Measurements can be deployed in several ways of breaking into the system. Do not talk about the solutions in the file on requirements.

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

The list of design constraints for the system is as follows:

- Principles of design of a company, team, or personal. For example, the designer who utilizes form follows the function of constraining designs.
  - A design that requires to work on other items like products, processes, services, systems, controls, information, and partners.
  - Functional requirements, like characteristics specification for the system.
  - Requirements defining intangible concept features. For example, there is a non-functional necessity that a structure is available.
- 
- **Propose a set of classes that could be used in your system and present them in a class diagram.**

