Name : Sajjad Younas

Id : 13850

Instructor: Mam Aasma Khan

Course Name: Software Design And Architecture

Date : 23-6-2020

"software design and architecture"

Q 01:-

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

Ans (a):-**Software Architecture:-** Software architecture is concerned with the selection of architectural elements, their interaction, and the constraints on those elements and their interactions. Architecture is specifically not about details of implementations (e.g., algorithms and data structures.).

Why software architecture design is so 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

Due to this reason software architecture is so important.

b)Explain any four tasks of architect?

Ans(b):-

(1) Perform static partition and decomposition of a system into subsystems and communications among subsystems.

>A software element can be configured, delivered, developed, and deployed, and is replaceable in the future.

(2) Establish dynamic control relationships among different subsystems in terms of data flow, control flow orchestration, or message dispatching.

(3) Perform tradeoff analysis on quality attributes and other nonfunctional requirements during the selection of architecture styles.

> For example, in order to increase a distributed system's extensibility, portability, or maintainability, software components and Web services may be the best choice of element types, and a loose connection among these elements may be most appropriate.

(4) Consider and evaluate alternative architecture styles that suit the problem domain at hand.

****

Q 02:- Explain Architecture Business Cycle (ABC) in detail with figure?

Ans:-

>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)*.**

i. Case studies of successful architectures crafted to satisfy demanding requirements, so as to help set the technical playing field of the day.
ii. Methods to assess an architecture before any system is built from it, so as to mitigate the risks associated with launching unprecedented designs.
iii. Techniques for incremental architecture-based development, so as to uncover design flaws before it is too late to correct them.

****

Q 03:- Explain ABC Activities?

Ans 04:-

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 architectural 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

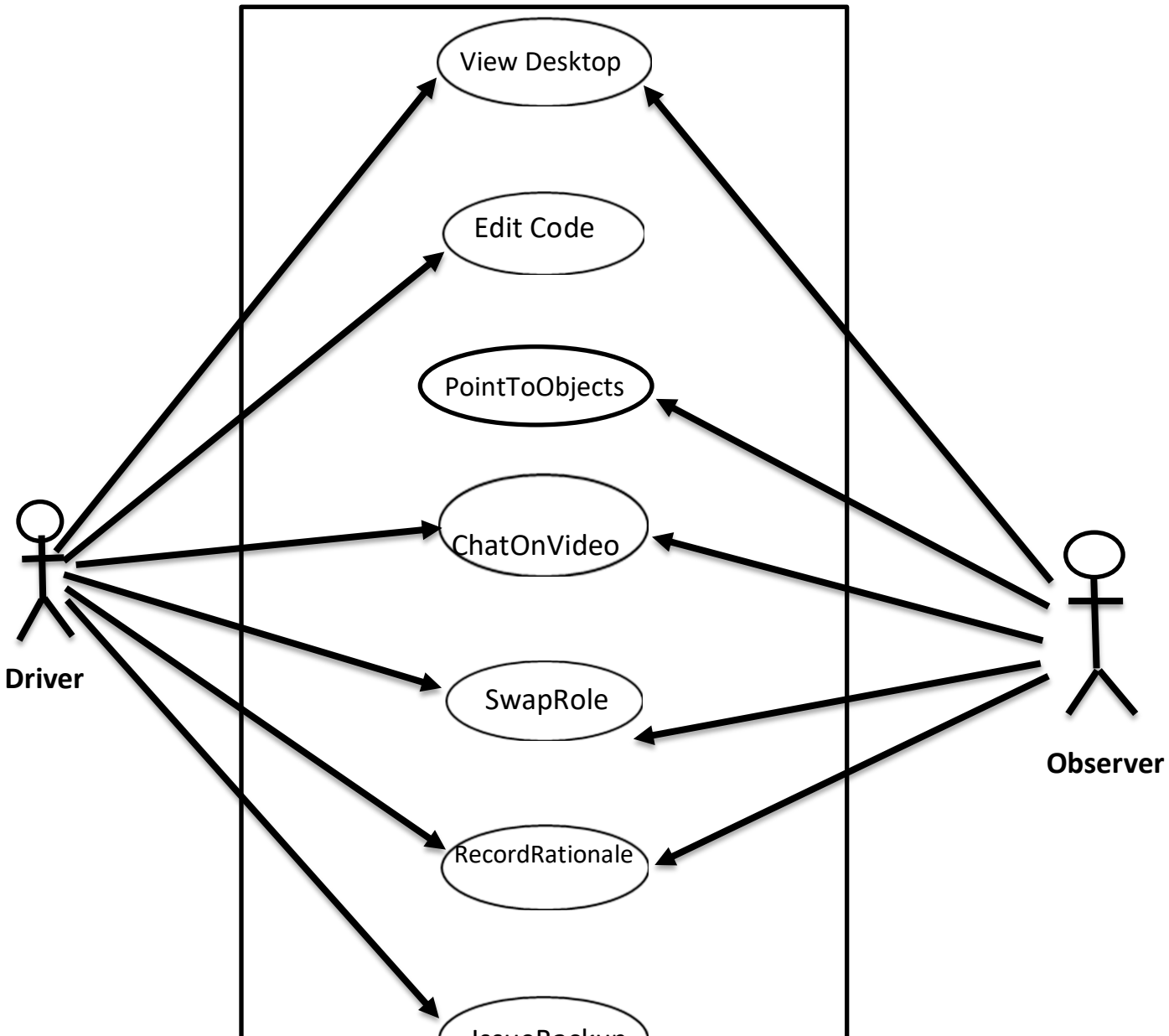
>Architecture maps the stakeholders' requirements/needs

6 Implementation based on architecture

7 Ensuring conformance to an architecture

Answer:

Use-Case Diagram



Assumptions: when the Driver edits code, we assume that the Observer can see the changes in realtime through the ViewDesktop use case, thus there is no arrow pointing back to the Observer for the EditCode use case. A similar assumption is made for the PointToObjects use case, so no arrow points back to the Driver.

we assume that both the Driver and Observer can initiate the ViewDesktop, ChatVideo, SwapRole, and RecordRationale use cases.

Nonfunctional:

- **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.
- **Portability** - the programmers should be able to use the system regardless of what computer and operating system used by the programmers.

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

Answer:

Example 1: "**Portability-** the system should be portable" is a NFR. This NFR may lead to a constraint on the programming language used for the implementation of the system (e.g., the programming language Java (rather than C and C++) might be preferred in order to meet this NFR).

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

Answer is on next page….

Class Diagram