LAB1

C. Write a one page essay where you explain clearly why software architecture is important

Software Architecture deals with the overall important part of the program and stuffs that the engineers most familiar with the codebase and or the whole system have a shared understanding of the system. It is most important because over a long period of time, continuous delivery becomes faster compared to the software programs where architecture is not handled properly.

In a world where continuous delivery is very important, it is very important to make rapid changes to the codebase especially when adding new features. Usually, we have a agile methodology where the sprint goes up normally upto 2 weeks and each developer work on 3 to 5 story points. Most of the story points are based on adding new features, whereas some could be fixing the bug. So, having a clear architecture means that each of the component is modularized and there is a clear interface which can be extended on. It has a proper design pattern and follows basic rules of programming such as programming to interface, SOLID pattern and so on. In case of design, it could be the use of microservice architecture with asynchronous messaging using message brokers such as Kafka or RabbitMQ. So, having a proper software architecture like microservice nature could mean that each team could work independently and release new feature on a particular microservice.

Hence having a good software architecture is very crucial for the rapid release of new features which could mean a large economic gain to the company and the stakeholders involved.

D. Explain what the difference is between software architecture and software design

Software design means the tools and technologies that are involved in the program and deals with the how and is more detailed, whereas software architecture is the important high level structure of the system and deals with the what of the program and or requirements.

E. Explain what makes software architecture so difficult

Software architecture is difficult because the person who is architecting the software must be an expert developer with sufficient experience working on different kinds of projects who might be familiar with the bare necessity that the program could have for the start as well as the ability to handle scalability, consistency, modularity of the program once the user base rises. It should also handle availability and consider whether the system is available or not.

Furthermore, it should be dealt perfectly with the fact that whether the plans to accommodate some tools are a overkill or not and what could be important enough that it could cost the resources currently available.