***High Level Architecture Research***

***What is software architecture?***

Software architecture defines how different parts of a software will be organised, how they’ll work together and what rules they will follow. It mainly contains 3 parts:

* Components
* Relationships
* Quality attributes (the software needs to meet certain quality standards).

***What is good software architecture?***

A good software architecture is defined as a set of factors that contribute to the effectiveness, efficiency and sustainability of a software. Some of these factors/attributes are:

* Understandability (the architecture must be clear and easy to understand)
* Modularity (specific components have specific functionality, this allows code reusability and ease of testing).
* Scalability: the architecture should allow an increased number of users, data or features without the need for a complete redesign.

Other factors might be: *flexibility, performance, security, interoperability and documentation.*

***What are different software architectures?***

There are several software architectures we can use. Some of the most common are:

1. Monolithic Architecture: the application is built as a single unit.
2. Microservices Architecture: the application is broken down into small and independent services that communicate with each other through the use of APIs (This allows independent developments).
3. Client/server architecture: the application has an user interface and a back-end that communicates over a network.

***What are software architecture characteristics?***

As mentioned in *Question 2.* The characteristics that make a software architecture useful and complete are: *modularity, scalability, flexibility, maintainability, performance, reliability, security, interoperability and reusability.*

***What is high-level design?***

High-level design (HLD) in software development can be defined as an abstract representation of the software system without diving into the details of coding. It is the road map that allows the developers to get from the system's requirements to the actual implementation of the software. This design must be abstract and easy to understand.

***What are the best tools for lightweight high-level design?***

When looking for tools to use during the high-level design process of a software project, we immediately think about user-friendly diagrams that, once put together, can provide us with a roadmap/view containing the different steps to follow in order to create high quality software products. Some of the most common tools are: *draw.io, lucidchart, gliffy, microsoft visio, plantUML.*

*Note: If any of you wants to add extra information to this research, feel free to do so.*