Task 1)

There is the organizational level, the system level, and the building block level. The organizational level is characterized by the fact that the various organizations and how they work together within the framework of their various roles are presented at this level (page 73 above). And on this level, one deals with business processes, for example. (page 75)The system level is characterized by the fact that the IT systems within the organization and how they are used in the context of the organization's business processes are presented here (page 73 above). And at this level, one deals with system use cases, for example. (page 75) The building block level is characterized by showing the interfaces, users, building blocks that make up the system, responsibilities, and interactions (page 73 above). And one is concerned on this level e.g. with the production of system modules the module application cases convert. (Page 75) The distinction is used to create high quality architectures because the architectures do not mix problem/aspects of the levels, appropriate levels are assigned, and the influences on an architecture can be better understood. (page 75)

The authors understand by a level change (page 82,79) a decomposition or that one looks at the individual parts of an element and then looks at the individual parts of the individual parts until the abstaction has decreased to a certain degree that one is in another level.

The macro architecture differs from the micro architecture in that the macro architecture has a low level of detail or a high level of abstraction and the micro architecture has a high level of detail or a low level of abstraction. The transition from macro to micro architecture is recognizable when the abstraction decreases more and more and the details increase (page 78), or when one looks so far into system building blocks in macro architecture that one encounters non-supporting system building blocks, then one is in micro architecture. (page 79)

There are the basic architecture views:

-Conceptual view (business view) which is characterized by the fact that it is suitable to convey an architecture to non-technical interests. The distinction in a conceptual view serves the documentation of the architecture requirements.

-Logical view() This is characterized by the fact that it describes the system components and their relationships to each other in detail. The distinction in a logical view serves the documentation of the architecture design.

- Execution view (distribution view) is characterized by the fact that it shows the physical distribution

of the system building blocks to the runtime describes and to itself also technical stakeholders addresses. The distinction into an execution view is used to document the physical distribution of software building blocks.

and there can be any others according to architecture-view-model

The distinction in the architecture views in general serves that different interest groups can also understand the architecture of the system (page 90, 89).

There are any number of architectural styles, but some commonly used styles can be divided into categories that are characterized by the following features: Independent Component, Call and Return, Data Centered, Data Flow, Virtual Machines.

Task 2)

a) Client/provider architecture: Server, It can be recognized that there are clients, which are the users, and providers, which are the server.

Multilayer architecture: Browser, It is to be recognized there since a browser uses clearly separated technologies like Javascript, Html and Css, Cookies, SSL connections here would be e.g. the separation between which is visible and which not and which connects with networks and which not. Html and Css would be in the graphical part that communicates with the logical layer the javascript or the program that executes the javascript, this on the other hand then communicates with the network layer with the SSL etc.. And the upper layer, the graphical layer communicates
e.g. not with the netwerk layer.

Event-driven system: user interface/gui, it is recognizable that everything is controlled by the user and thus by events that the user generates.

Data flow networks: rendering pipelines, it is recognizable because a lot of data, the values for each pixel are calculated simultaneously by a similar process that has several steps, these steps are for example the evaluation of the different shaders, vertex shader that calculates the triangles in the 3D scenes and the fragment shader that calculates the colors of the triangles or pixels.

Web architecture: web page, it is recognizable because it is used on the Internet or web as the name suggests.

b)

1.) Event-driven system style, or event-driven software architectures patterns, because they handle multiple events at once, so users don't have to wait for one process to finish before starting the next. Thus the response time depends

does not depend on whether a process has already been completed.

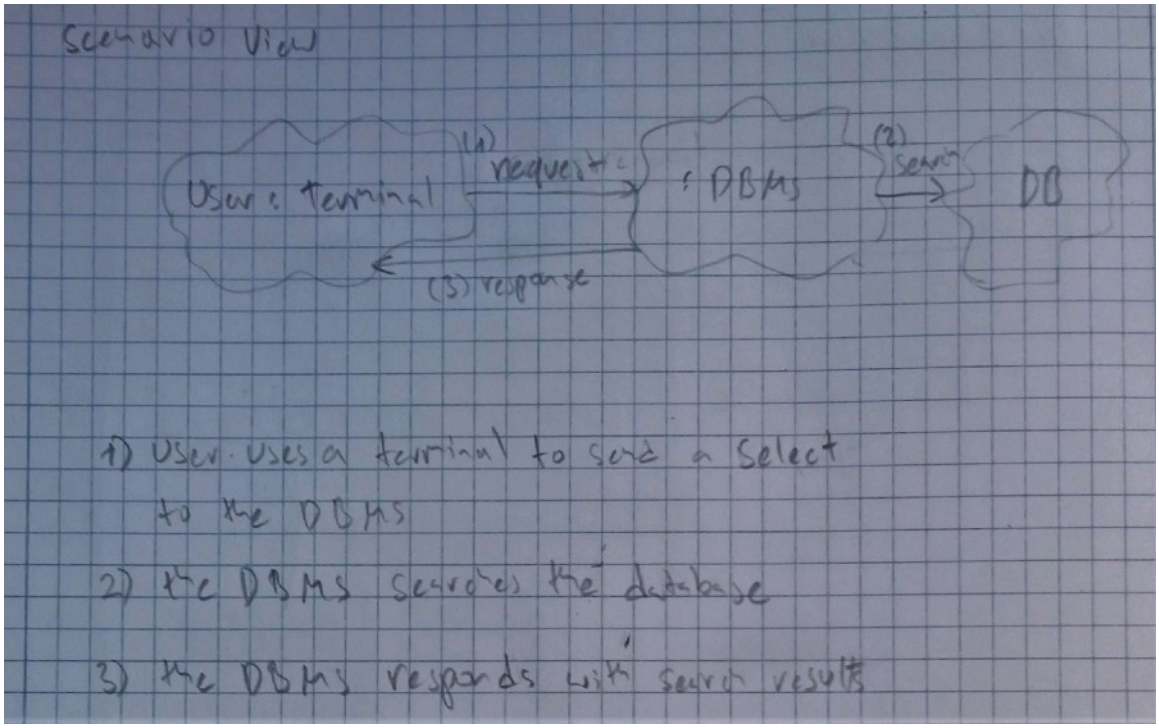
(source: https://www.infoworld.com/article/3669414/the-benefits-and-challenges-of-event- driven-architecture.html)

2.) Micro kernel styles, because in the style there is a main system that can run on the target operating system that contains everything the system needs to function and then it can run smaller plug in systems that are independent of each other. So once you have created one of these plugins you can use it directly on any system where the main system is already implemented.

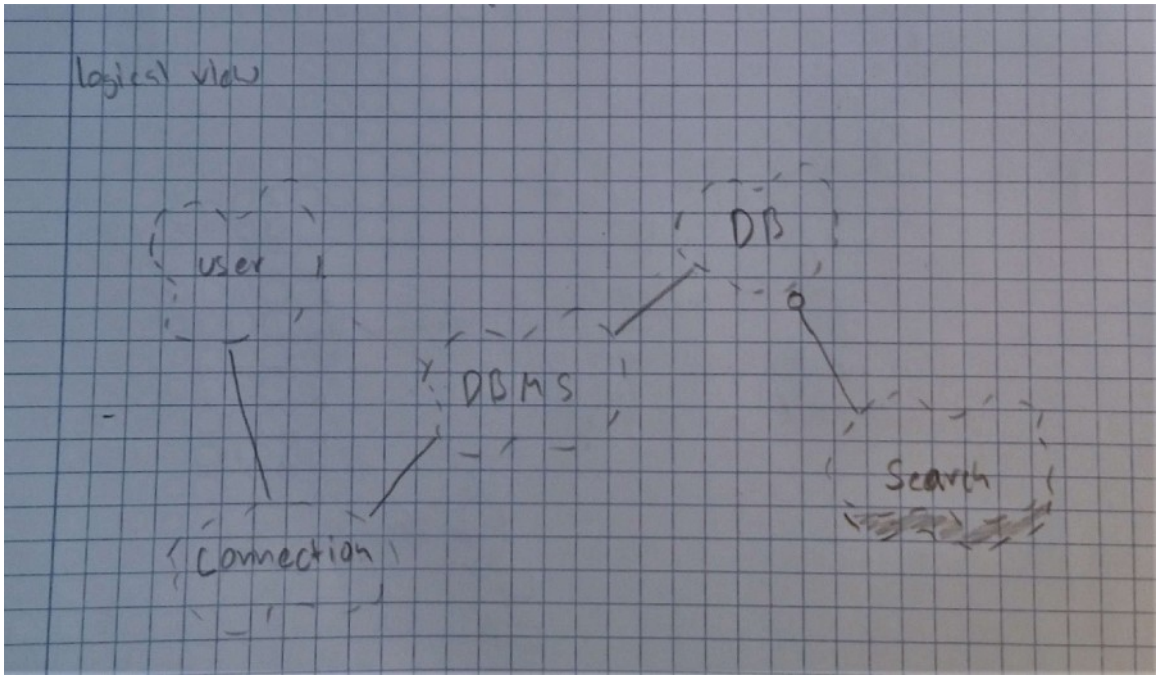
(Source: https://www.spiritofsoft.com/software-architecture-

patterns/) Task 3)

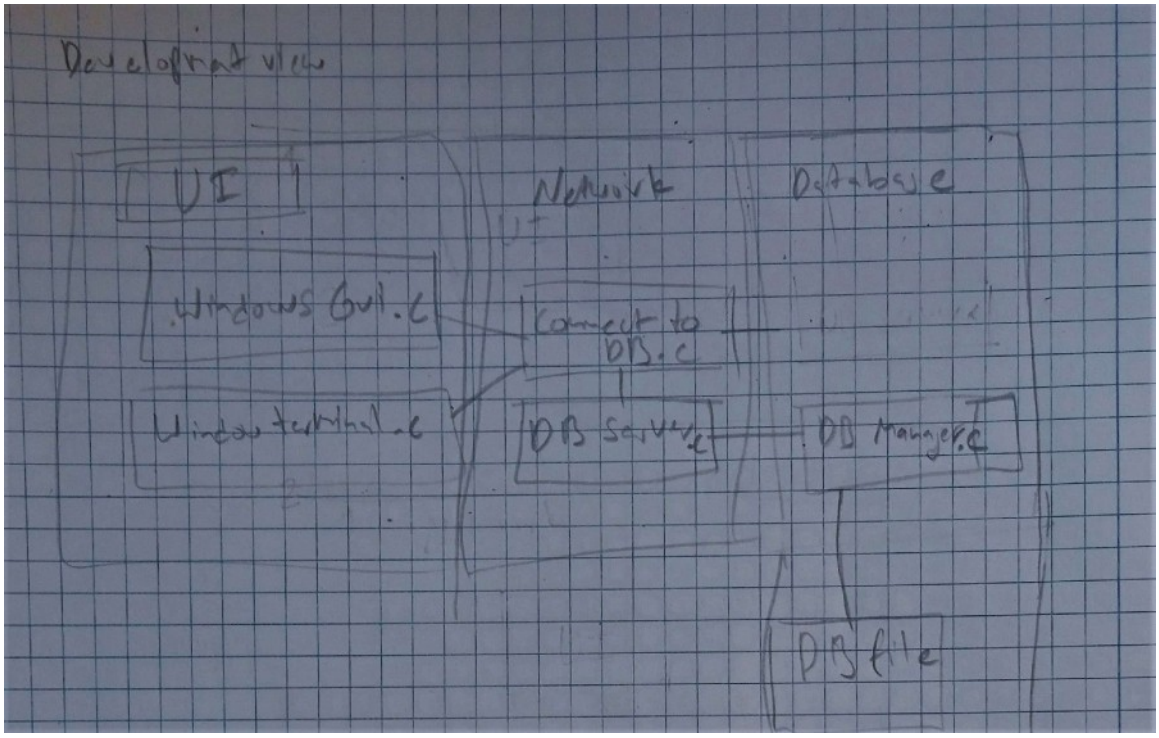
Use Case View:



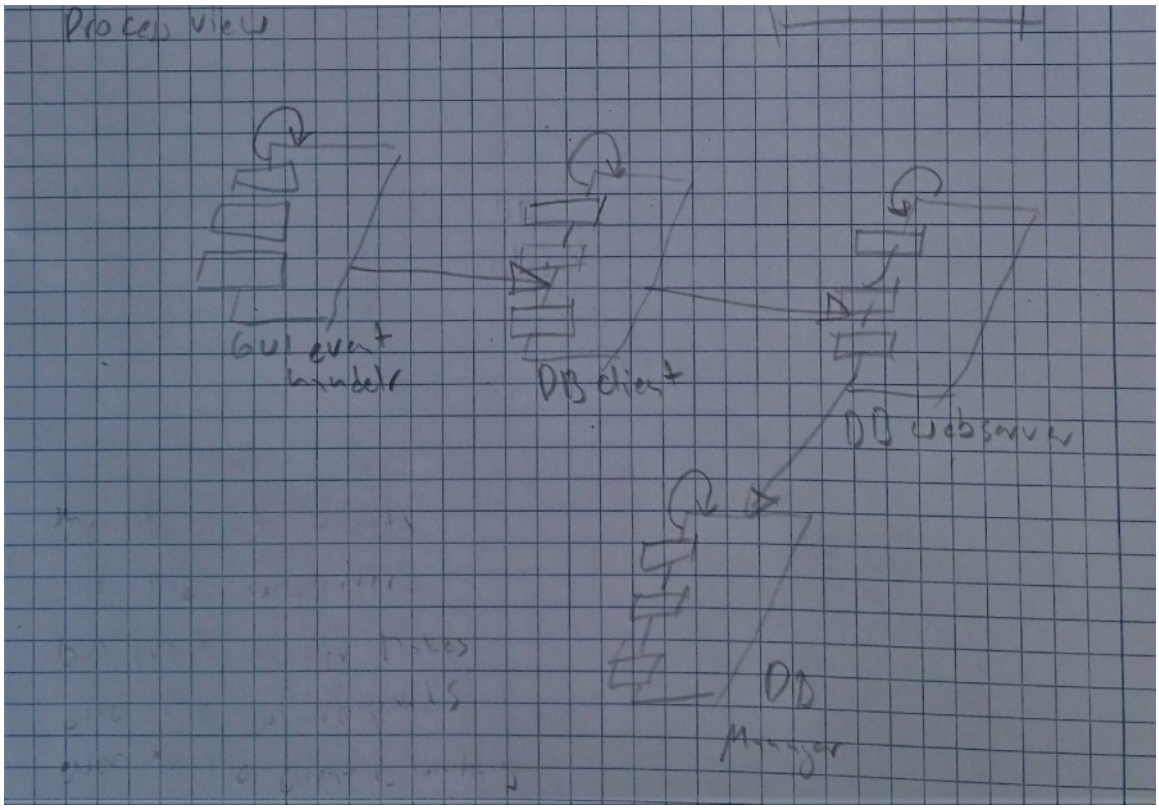
Logical view:



Implementation View:

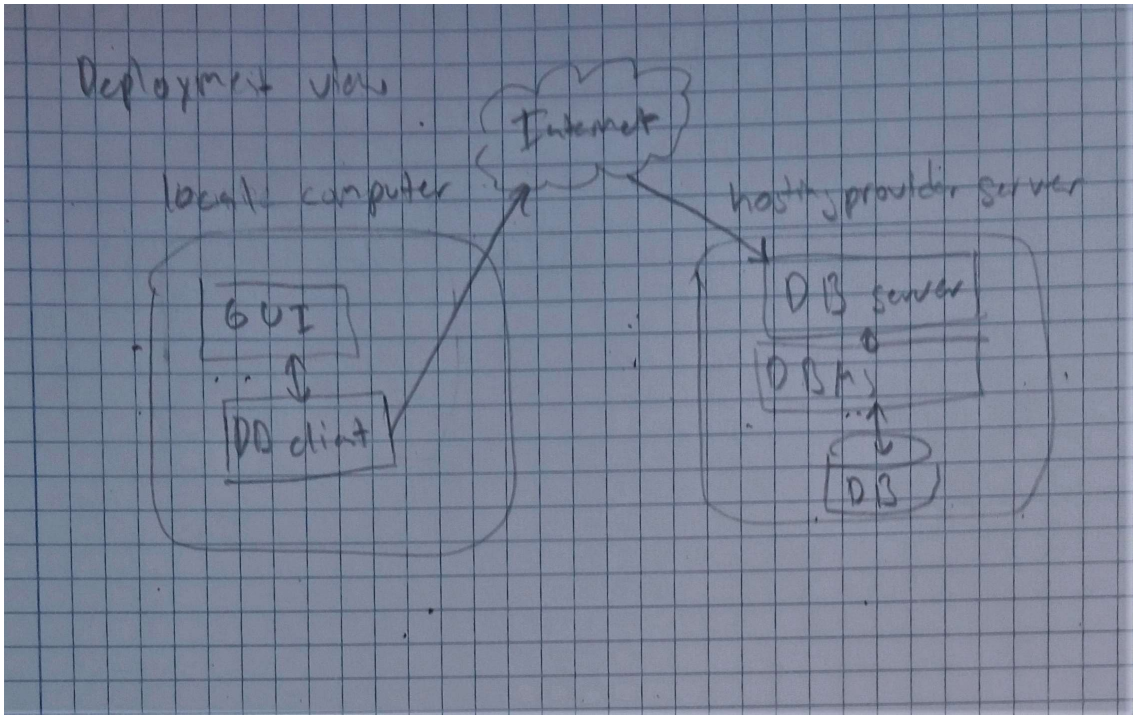


Process View:



Gui event handler ensures that the user can press the buttons without having to wait for the connection to be completed. DB client establishes the connection, waits for the server. DB webserver connects and sends information to DB client and talks to DB manager. DB manager processes data and searches data while server does other things.

Distribution View:


Deployment view
Internet
local computer
hosting provider server
GUI
DB client
DB server
DBMS
DB