Introduction

🏆 A2.2 Learning Activity

Software architectural patterns

📘 Instructions

- Based on the software architecture styles and patterns documentation, realize the system architecture for the case study.
- Every activity or challenge must be realized using **Markdown style with .md extension** and VSCode development environment, it has to be as a **single page** document, which means if the document has images, links o another external document it must be accesed from tags and links, and must be named with the nomenclature **A2.3_ActivityName_StudentName_Team.pdf**
- As a requirement the .md file must contain a tag of the link to the repository of your Github document, for example a **Link to my GitHub**. At the end of the assignment, the .md file must be uploaded to github.
- From the **.md** file export a **.pdf**, that must be uploaded to classroom in the corresponding section, serving as evidence of turned in, since being the **official** platform here you will receive the activity's result.
- Considering that the .pdf file was obtained from the .md file, both must be identical.
- Your repository, besides containing a **readme**.md inside the root directory containing student's information, team, subject, career, teacher's information, and also a logo or pictures, must contain a section of contents or index, which are **links to your md documents**, try not to use text to indicate internal or external links.
- It's suggested to use a structure similar to the one indicated below, however, you can use any other structure that can help you organize your repository.

```
| readme.md
| | blog
| | | Cx.1_NombredelaActividad.md
| | | Ax.1_NombredelaActividad.md
| | diagrams
| | docs
| | html
| | img
| | pdf
```

✏️ Development

1. Consider apply to the case study the architecture patterns below:

    - Client-server architecture pattern
    - Layered architecture pattern
    - Model-view-controller architecture pattern
    - Micro-services architecture pattern
    - Pipe-filter architecture pattern
    - Event-driven architecture pattern
    - Blackboard architecture pattern
    - Publish/subscribe architecture pattern
    - Master-slave architecture pattern
    - Peer-to-peer architecture pattern

2. Based on the architectural views like so realized UML diagrams, apply the pattern that consider the most appropriate to each of the next scenarios:

    - 2.1. Considering that it is desired to maintain and scale the system, it seeks to develop the application through the decomposition of small independent and isolated services, which consume an external interface to communicate with a database server.

      • Micro-services architecture pattern

      • Layered architecture pattern

    - 2.2. Being confidentiality and security attributes or requirements for the case study, it seeks to structure each of the components that are going to be programmed into groups of sub-tasks, where each of these sub-tasks must communicate an intermediate layer and this to another layer above.

      • Layered architecture pattern

      • Model-view-controller architecture pattern

    - 2.3. Looking for the availability of the system, it is proposed to install two service servers, where they make their requests to a third server in which the database would be stored.

      • Client-server architecture pattern

    - 2.4. Identifying that the integrity of the data is a requirement, it is proposed to replicate and synchronize the database stored within the main server to another, considering the main one as the master and the secondary one as the slave.

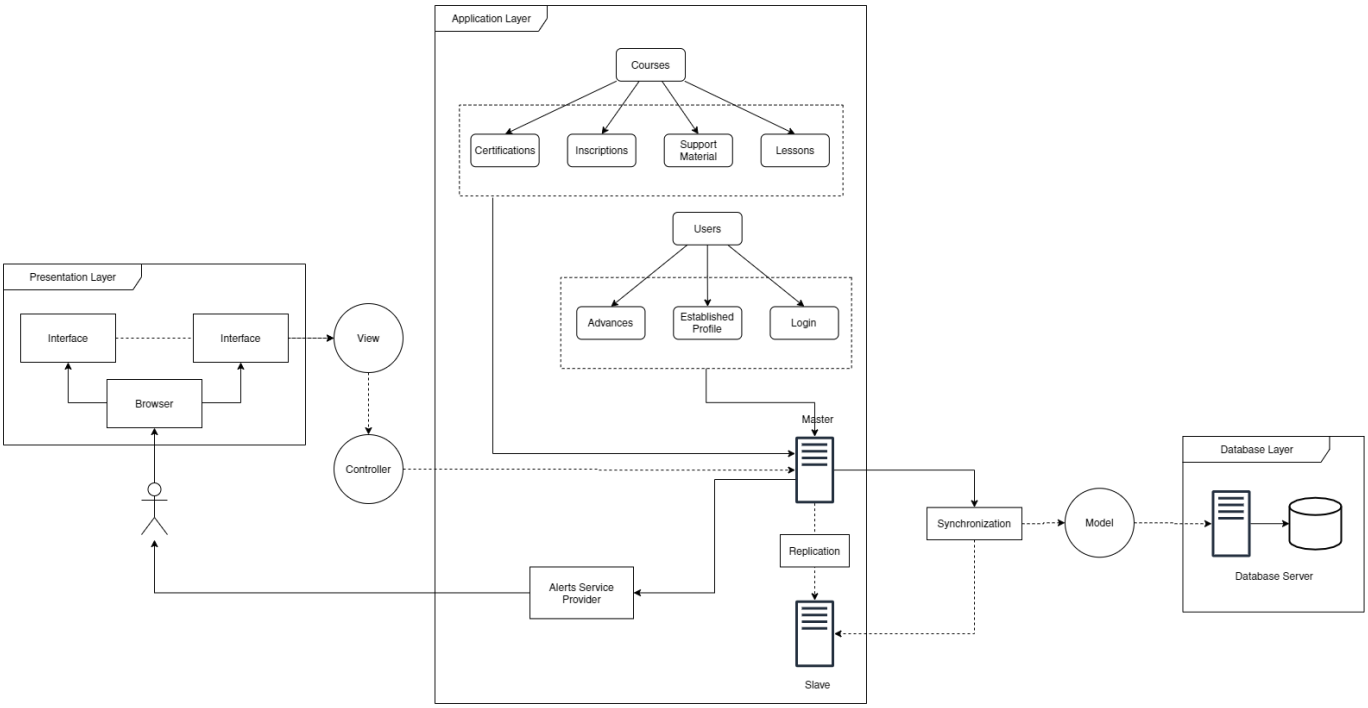
      • Master-slave architecture pattern

      • Pipe-filter architecture pattern

    - 2.5. Considering that the client requested that each time a failure occurs, alarms must be provided to the different users regardless of the place where they are, it is observed that a service provider must be hired to trigger the alarms and these reach the required users.

      • Event-driven architecture pattern

- Publish/subscribe architecture pattern



Conclusions

a) *Bañuelos Mendez Jordi's Conclusion*

This activity, in a similar way to previously developed activities, helped us provide a graphic representation of the interaction of the system, but this time as a whole; that is, representing each one of the system's components in one single diagram (including components, databases, replication processes, synchronization, etc).

b) *Castillo Medina Edgar Antonio's Conclusion*

In previous documents we've been designing and establishing some parameters of the project like requirements, standardized diagrams and selecting tools to develop what we are idealizing, but in this document we connect all that we have done in one (complete, extended and comprehensible) diagram that generalizes everything that will happen in the platform, if we'd decided to develop the software I believe the previous diagram would have been a perfect base for it.

c) *Villanueva Romero Carlos Daladier's Conclusion*

With the realized activity I learned that for specific scenarios there are patterns that will satisfy their needs, the ones that we choose for our study case seems to be the appropriates in case we would have develop the platform, I think that use this architectural patterns are a easier way to accomplish some phases of the developments strategies.

d) *Villegas Ramirez Luis Eduardo's Conclusion*

With the development of this activity, solutions to specific scenarios that the system will have are defined, the architecture patterns help us to understand and visualize the behavior that will take place at a certain time to

make everything workable and comply with what is requested, in addition, issues of interest for the client such as availability, integrity and security that, among other things, will form the construction of a complete system that allows its correct use.

Rubric

Criteria	Description	Score
Instructions	Is each one of the points indicated in the instructions section fulfilled?	10
Development	Was each one of the points requested within the development of the activity answered?	60
Demostration	Is the student present during the explanation of the functionality of the activity?	20
Conclusions	Is a personal opinion of the activity included by each of the team members?	10

Go to Home