17 DE FEBRERO DE 2024

WIS ARCHITECTURE REPORT

GROUP C-1.047

Acme SF, Inc.

Repository link: https://github.com/Cargarmar18/Acme-SF-D01.git

Castillejo Vela, Manuel

Email: mancasvel@alum.us.es

García Martínez, Carlos

Email: cargarmar18@alum.us.es

Rodíguez Millán, María C.

Email: marrodmil@alum.us.es

Table of contents

1. Executive summary

In this report we defined the knowledges that precede the group from previous experiences, moreover, highlighting the knowledge accumulated from previous subjects of this grade. It is stablished from the more general concepts that previously were learned that can be useful in any type of software development process to finally emphasize some specific knowledge that we have applied in the development of past software systems applications.

2. Revision table

Revision number	Date	Description
1.0.0	17/2/2024	Base WIS architecture report

3. Introduction

In this document the students will manifest by writing down in a report document their previous knowledge about WIS architecture. This referral to WIS architecture is related to the structure and technical design that serves as sustain to the whole implementation and functionalities provided by the whole information system, in this case, planted and performing in the web. The previous knowledge acquired in other subject areas or by our part could be crucial in the development and performance of our selves in this subject reaching our full knowledge potential and acting according to what was previously learned by the student.

This analysis report has been manufactured in alignment with the annex. As have been defined it commences with a cover page featuring the author's report credentials. A version table follows, delineating modifications made in this document, classified by number and accompanied by their respective dates, and a brieve description of what have been added. After this, an executive summary is presented with the according information indicated in the annex. And this introduction that works as a precedent of the actual contents specified before.

4. Contents

4.1 Some key concepts

Our previous knowledge of WIS architecture is not extend or very detailed but stablish a base in which we can work and confront situations that could appear along all the process. Our main focus are the designs principles, architectural styles and design patterns.

4.2 Design principles

Our base is constituted with designs principles as DRY (Don't repeat yourself), in which our main focus is to do not have code which is a repetition or a really similar version of other code previous used, the main goal is to simplify the code and have a much better clarity reading and understanding the code.

Another important principle is YAGNI (you aren't gonna need it), which is strictly based in the implementation of the functions and code that we really need at that moment.

Furthermore, the knowledge of separations of concerns, information hiding, protected variations and high cohesion and low coupling are fundamentals to the development and design of the project.

4.3 Architectural styles and Design patterns

Another important factor is the knowledge of design patterns which are regular solutions to recurring designs problems. Some of them are SPA (single page application) and MVC (model view controller).

The MVC or model view controller patter it is based on the separation of responsibilities. In this pattern there are 3 components with 3 stablished functionalities. The model which is responsible for the data and application logic, the view which is responsible for the presentation to the user and the controller which acts as an intermediary between the model and the view and allows the connection between them. Some of the benefits of this pattern are the code reusability, the flexibility and adaptability to changes and the way in which helps the team to perform a better development as a group.

In the other hand, SPA or single page application it is a pattern used in web to load a page with all his data in the first load and the changes will be applied conform to the user interaction with the page. This pattern allows the user a more fluid interaction with the application as well as a navigation without time queues.

Furthermore, we can distinguish between architecture styles and design patterns knowing that styles establish restrictions on the architecture while patterns have a smaller scope focused on smaller scope problems and solutions. Some examples of Architectural styles include layers and microservices.

In layers we make a separation between "layers" as its name indicates. This structure allows us to divide the project into parts with specific responsibilities, this allows modularity and low coupling between layers.

In microservices the focus is on software development where application is decomposed in small decoupled services that communicate through well defined interfaces. This allows independency and decentralization.

4.4 Extra aspects about WIS architecture

Another aspect that could be part of our knowledge is a basic knowledge on databases management using languages like SQL or JPQL.

And as an end we can highlight the protection against some of the more typical hacking methods like SQL injection or GET hacking and how we need to know them to perform a robust environment which can prevail against those issues. And also a special mention to auditing and logging which is a crucial part of software development projects.

5. Conclusions

As a summary, we established the base in which we have been working in previous years acting as a preparation or conditioning to DP2 project objectives. Moreover, it is focused on WIS architecture and our experience with all the things that surround it. We have established this knowledge as a group consensus from what we have learned from subjects like DP1, IISSI 1 or IISSI 2.

6. Bibliography

Intentionally blank.