[Nagarro]-[Amcart]

DAR Document

Nagarro Software Pvt. Ltd.

<Shubham>

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| Version | Date | Author/Contributor | Comments |
| 0.1 | 9 -12-2023 | Shubham Agrawal | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1 Introduction 4](#_Toc129721414)

[1.1 Objective and scope of document 4](#_Toc129721415)

[2 Requirements at a Glance 4](#_Toc129721416)

[3 Available tools 5](#_Toc129721417)

[3.1 Layered or N-tier Architecture 5](#_Toc129721418)

[3.2 Monolithic Architecture 5](#_Toc129721419)

[3.3 Microservices Architecture 5](#_Toc129721420)

[4 Comparison Analysis 6](#_Toc129721421)

[4.1 Weightage Matrix 6](#_Toc129721422)

[4.2 Comparison Table 6](#_Toc129721423)

[5 Recommendation 6](#_Toc129721424)

[6 Assumptions 7](#_Toc129721425)

[7 Risks 8](#_Toc129721426)

[8 Appendix 9](#_Toc129721427)

[8.1 References 9](#_Toc129721428)

# Introduction

This is an online shopping website, which can be used to purchase the clothes for both men and women.

## Objective and scope of document

The objective is to design online clothes shopping website for women and men.

Design should be technology agnostic.

# Requirements at a Glance

Various functional modules that will be implemented by the system will be -

• **Login**: Customer log -in into the system by entering valid user id and password for the

shopping.

• **Registration**: Customer needs to be registered to log-in into the application and buy

products

• **Administrator Page**: The administrator should be able to add, update and remove items.

• **Search or choose product**: Customers should be able to search for products.

• **Product Catalog**: This should display a list of products available on site for shopping.

• **Sale**: The products available in this section will have special discount offers.

• **Stock**: Site should be able to check the availability of products.

• **Price**: System should keep check on displayed price.

**• Add to cart**: Direct functionality of adding product to cart should be possible.

• **Payment**: For customer there are many types of secure billing will be direct bank transfer,

check or bank draft. The security will be provided by the third party like Pay-Pal etc.

• **Order Confirmation**: After the payment or surf the product the customer will logged out and

an email will be sent to him.

# Available tools

* **Layered or N-tier Architecture**
* **Monolithic Architecture**
* **Microservices Architecture**

## Layered or N-tier Architecture

The N-layer application involves the segmentation of application logic into distinct layers. The N-layer architecture pattern, a well-established approach, entails the segregation of different logical layers into individual physical tiers within applications.

## Monolithic Architecture

The conventional model for software program design is a monolithic architecture, characterized by its unified structure. In this context, "monolithic" signifies that the software is composed as a single, cohesive unit.

## Microservices Architecture

Microservices architecture, commonly referred to as microservices, represents an architectural approach or development style for applications. It revolves around breaking down extensive applications into smaller, self-sufficient functional units that can operate and communicate autonomously.

# Comparison Analysis

Below is the DAR comparison for the architecture type for various factors.

## Weightage Matrix

<point assignment as per features>

|  |  |
| --- | --- |
| Feature | Points |
| Scalability | 25 |
| Fault Tolerant | 20 |
| Complexity | 15 |
| Flexibility | 15 |
| Performance | 25 |

## Comparison Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | N-tier Architecture | Monolithic Architecture | Microservices Architecture | Justification |
| Scalability | 15 | 15 | 20 | Microservices are readily scalable horizontally |
| Fault Tolerant | 15 | 15 | 18 | Microservices are potentially less fault-tolerant, as a single point of failure can lead to system downtime.. |
| Complexity | 13 | 13 | 10 | Microsevice’s are not easy to manage when are on bulk. |
| Flexibility | 10 | 10 | 15 | Microservice’s provide flexibility for multiple teams to work in multiple stack simultanously. |
| Performance | 15 | 15 | 20 | Microservice’s are made granuarly and are losely coupled. They provide better performance. |

# Recommendation

From the comparison we can see that the **microservice** **architecture** will be best suited to our objective.

# Assumptions

The application is designed to exhibit high scalability.

The system is intended to maintain constant availability, ensuring users can access it through a web browser, with downtime limited to server-related issues.

Security holds paramount importance in all internet transactions, necessitating the secure and encrypted handling of all data.

# Risks

Scaling and optimizing independently hosted and deployed services entail intricate coordination among diverse components spread across separate servers. While microservices architecture is often lauded for its scalability, achieving it can pose challenges.

Microservices are deployed across varied cloud environments, interacting through different infrastructure layers. This setup reduces control and heightens the obscurity of components, leading to increased security vulnerabilities. Testing for vulnerabilities and ensuring network security becomes more challenging in this microservices structure.

These standalone microservices, each functioning independently, must communicate with one another. Poorly configured infrastructure layers that facilitate resource sharing among services can result in a suboptimal application with a sluggish response time.

Appendix

## References

1. https://www.c-sharpcorner.com/
2. https://microservices.io/