Technical Requirements Document

< Web Application>

<20 February 2024>

<Version 1>

<By: Dondolo Kazembe>

TABLE OF CONTENTS

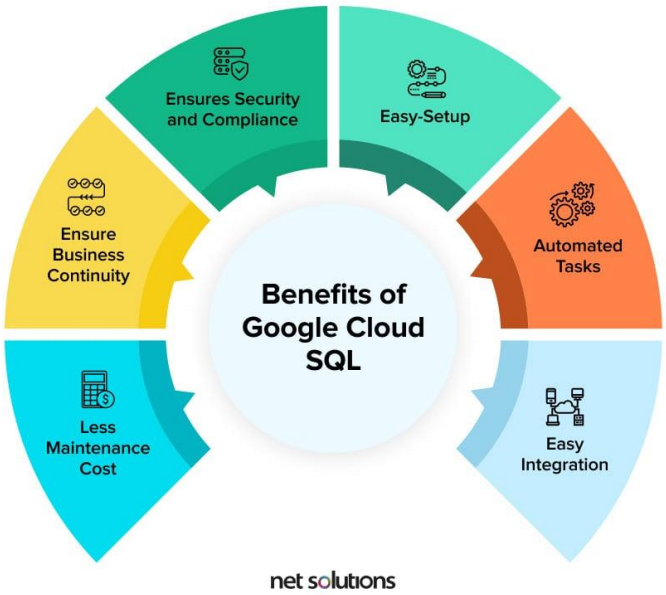
1. INTRODUCTION

The purpose of this document is to evaluate each architecture used to develop and deploy the EPI-USE employee system. This document will delve into the overarching solution architecture as well as the various layers and components that constitute the employee system. Furthermore, this document will examine the design patterns employed to address recurring design challenges, flexibility, and maintainability within the system. Furthermore, this document will shed light on the technologies used to implement key system functionalities, encompassing databases, programming languages, frameworks and enhance the overall performance of the web application.

2. DATABASE ARCHITECURES

SQL database on SQL Server Management Studio

The web application developed made use of a SQL database and was developed through the platform SQL Server Management Studio (SSMS). SSMS is a graphical user interface (GUI) tool that allows a user to manage SQL Server databases and provides many database activities such as, creating and managing databases as well as writing, executing queries, designing, and deploying database objects. A SQL database was selected as the backbone for the web application due to its inherent advantages in maintaining data integrity and structure, enabling efficient and effective data manipulation. This structured database environment facilitated swift creation, storage, and retrieval of both managerial and employee data, thereby promoting code efficiency and overall system performance. The diagram below illustrates the overall benefits of using a SQL database and reaffirms it effectiveness for the EPI-USE employee system:



Utilisation of Azure for Database Deployment

Azure is a cloud computing platform that offers a wide range of cloud-based services, including computing, storage, networking, databases, machine learning, and analytics. This platform was used to host the online database and perform all data queries. Azure provides adaptable pricing structures for SQL database hosting, which made it an ideal cloud platform choice. Leveraging its free trial, which grants users 200 credits to access any cloud computing service, I capitalized on this opportunity to use this cost-effective cloud computing platform. Thereafter, developed resource groups, servers, and databases were created ensuring efficient allocation of resources while optimizing costs. Furthermore, Azure provides tools and features for optimizing SQL database performance, such as intelligent performance tuning, query performance insights, and automatic index management. Additionally, Azure's global network of data centres ensures low latency and high throughput for SQL database workloads worldwide therefore, stored procedures created to manipulate that data will operate and an optimal level.

3. APPLICATION ARCHITECTURES

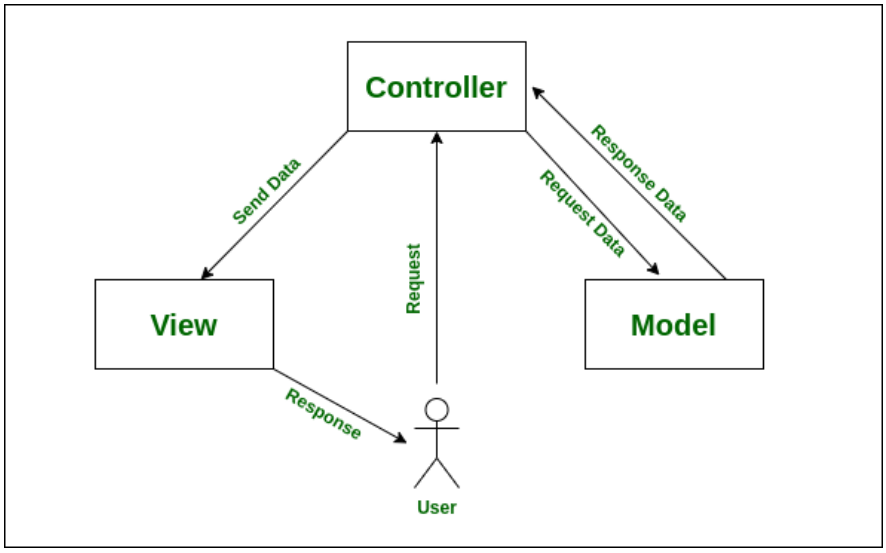
C# as Programming Language (Application Backend Code)

C# is an object-oriented programming language that is typically used to develop various applications including desktop, web, mobile, and gaming applications. This programming language was used to develop the EPI-USE employee system because it is based on easy-to-understand syntax which makes development easy and efficient. Additionally, C# has become a cross-platform language, allowing for the development of web applications to run on Windows, macOS, and Linux environments. This cross-platform capability enhances flexibility and facilitates deployment in diverse hosting environments which is essential in a fast-paced industry that EPI-USE occupies. Furthermore, C# development for web applications is well-supported in Microsoft's Visual Studio IDE, offering a rich set of tools, debugging capabilities, and integrations with other services such as Azure for cloud deployment. This made C# the ideal programming language as Visual Studio was the IDE used to create the web application.

Model-View-Controller (MVC) as application framework

The web application made use of the Model-View-Controller (MVC) framework which is an architectural design pattern that organizes an application's logic into distinct layers, each of which carries out a specific set of tasks. This framework was used as it allows easy modification of the entire application. Adding or updating the new type of views is simplified in the MVC pattern (as a single section is independent of the other sections). Therefore, any changes in a certain section of the application will never affect the entire architecture. In turn, this leads to an increase in flexibility and scalability of the application. Additionally, in the MVC framework, each component operates independently, promoting modularity and reusability of code. This makes it to reuse models, views, and controllers across different

parts of the application or even in separate projects, improving development efficiency. The diagram below illustrates how the MVC framework operates:



4. DESIGN PATTERNS

HTML and CSS as frontend application languages

Since MVC was the framework of the application, HTML and CSS were the frontend programming languages used to develop the views of the application. HTML and CSS are standardized technologies supported by all modern web browsers, ensuring consistent rendering and behaviour across different platforms and devices which made them an ideal frontend language for the web application. Furthermore, user experience is enhanced as HTML offers offline browsing which means that the offline cache produces significant performance enhancements as more of the site or system code and content is accessed quickly and locally.

5. CONCLUSION

In summary, the development of the EPI-USE web application prioritized efficiency and affordability, utilizing optimal resources to fulfil the specified functional requirements outlined in the technical assessment. A robust SQL database was developed and hosted on Azure, leveraging the scalability and reliability of cloud infrastructure to support the application's data management needs. Additionally, the MVC (Model-View-Controller) architecture was the framework for the application, ensuring a structured and organized approach to development that promotes maintainability and reliability. This strategic combination of technologies and platforms laid a strong foundation for the successful deployment of the web application.