**Architecture**

TITLE PAGE

ABSTRACT

// brief summary of entire document

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

//Introduction to the entire document

// purpose and scope of the document

ARCHITECTURAL STYLE(S) USED

// As a whole, how does the architecture support

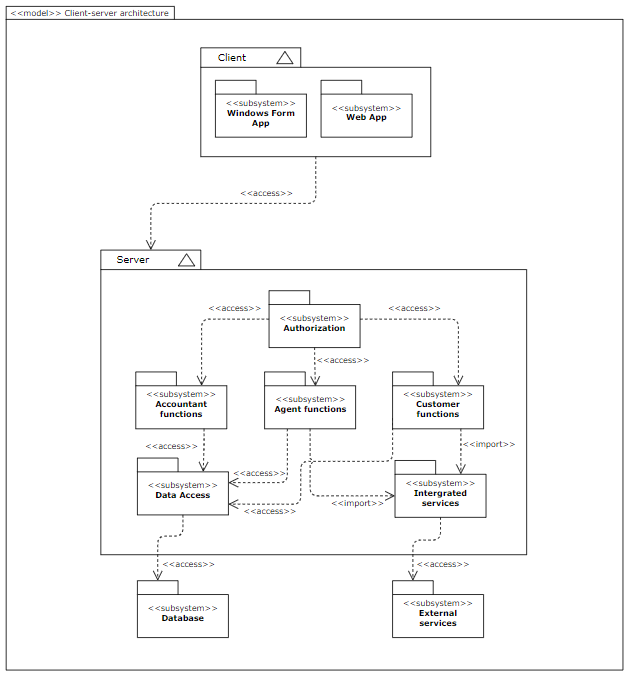
// various features of your application.

In brief, the architectural style used is the client-server architectural style. More specifically, the client applications are built based on the 3-layer for Windows applications and the MVC model for web applications; and the server is developed using the service-oriented architectural style and message-based communication.

ARCHITECTURAL MODEL

// Use packages stereotyped as subsystems <<subsystem>>

// No classes in the architectural model



TECHNOLOGY, SOFTWARE, AND HARDWARE USED

// Describe the technology used for implementing this project

// List all software and hardware required to support the technology

// Explain the communication between the application server and the database server

1. The technology, software, and hardware used

This project uses the programming language C#, .NET framework for Win form foundation; and ASP.NET MVC for the web application’s structure. In addition, no matter how the structural foundation of the applications is, they all connect with Microsoft SQL Server to get data from the database. Therefore, there are some technical requirements for the software and hardware used to implement these applications which are included in the following section.

* Software requirements:
  + Recommending to use Visual Studio 2019 or Visual Studio 2022.
  + Microsoft SQL Server 18 or higher.
  + Operating system: Windows 11 version 21H2 or higher, Windows 10 version 1909 or higher,...
* Hardware requirements:
  + · Windows computer or Macbook with special notes to run the software.
  + · 4GB RAM minimum [2].
  + · From 850 MB up to 210 GB of available space [2].

1. The communication between the application server and the database server

* Windows application: This application is developed on the 3-tier architecture (GUI, Business Layer, Data Access Layer). The Data Access Layer connects with Microsoft SQL Server by a connection string consisting of the information of data source, database name,..., therefore, it can access the database and perform actions on it. Then, This layer defines some proper functions which fulfill the requirements and gives permission for the Business Layer to perform these operations. In the top layer, GUI receives user requests and sends them to the Business Layer to process the requests. Thus, GUI does not access the database directly, and the development phase can be divided much more transparently.
* Web applications: This application uses the MVC (Model - View - Controller) model to operate the behind structure. In this model, the users interact with the interface supplied by the View, and the View sends these requests to the corresponding actions of the Controllers to process. The following step is that these actions connect with the model classes in the Model to execute proper processes. The reason for this connection is that the Model is the only place that has permission to access the database and operate actions on it. Similar to the Win Form, the Model also uses a connection string as a key to open the database in Microsoft SQL Server.

RATIONALE FOR YOUR ARCHITECTURAL STYLE AND MODEL

According to the distributor’s requirements, the system is built based on a mix of architectural styles. In the overall view, since the system is used by cross-platform applications including Windows forms and web applications, the client-server architectural style is applied to the system.

To begin with, the client side is developed based on the MVC model for Web applications and the 3-layer for Windows applications. The reasons for this choice are the characteristics of the software which demands fast deliverables, easy planning, and maintenance [1].

Furthermore, the service-oriented architecture is used to operate the communication between the client and server side. In detail, these interactions are performed using message-based communication. Due to the cross-platform applications, this architectural style enables the client apps to communicate with the server and enhances the flexibility of the system [2].

EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION

MANAGEMENT

REFERENCES

#### [1] Benefit of using MVC - GeeksforGeeks. (2021). Retrieved 16 April 2023, from <https://www.geeksforgeeks.org/benefit-of-using-mvc/>

[2] Liu, Y., Cruz, B. D., & Tilevich, E. (2022). Secure and flexible message-based communication for mobile apps within and across devices. *Journal of Systems and Software*, *193*, 111460.