

Build and Design of Voyage Account Applications Using C#, WPF, and SQL Server 2012 (Case Study Company X)

Desi Dyah Sulistyarini, Reza Kamaluddin Isman, Hata Maulana

Politeknik Negeri Jakarta

Jl. Prof. G.A. Siwabessy, Kampus Baru UI, Kota Depok Jawa Barat, Indonesia 16424

Email: desidyahs@gmail.com

Abstract

Voyage Account is an application that record consumption data on each voyage, then the data is used to calculate the profit or loss of each voyage. The application interface was created using Windows Presentation Foundation (WPF). Prototyping development methods was used to create these applications, C# as the programming language, and SQL Server 2012 as the database. This study discusses the making of the interface, backend, and database of application. WPF has been selected, because WPF is the latest technology developed by Microsoft after WinForms. WPF presents a display that can be customized to user needs. The test results showed that the Voyage Account application is already functioning in accordance with wishes of the user, the test is done using sql query by entering the appropriate username and password and match the query returns results with what is displayed by the application. As for testing the interface shows if WPF can be flexibly adapted to the screen resolution of 1366 x 768, 1920 x 1080 and 1280 x 720.

Keyword: *Voyage Account, WPF, prototyping, C#, SQL Server 2012.*

1. Preliminary

The main business of Company X is focused on the coal transportation to supply PLN power plants, a subsidiary of PLN, and Independent Power Producers (IPP). Consumption from every voyage captured and recorded to quantify the costs incurred by the company. Calculation from recorded data is made to be used as an evaluation for the Operation Department and Fleet Department.

Each voyage which is run by fleet has the income and cost. Costs to be charged in one voyage, among others MFO (Marine Fuel Oil), MDO (Marine Diesel Oil), CO (Crude Oil), FW (Fresh Water), and HSD (High Speed Diesel). When fleet do not travel in accordance with a predetermined time, the operational costs of fleet will increase. There are many factors that cause the fleet did not arrive at their destination on time, one of which is the weather. Because of that factor, Voyage Account application is made so that can be used to record data obtained during the voyage and this application can calculate profit or loss that is earned by the company in one voyage. Voyage Account application can shorten the time of calculation of profits and losses which originally was done manually. In addition, the information generated by the application can be used as an evaluation of each voyage.

2. Result

(Soni, 2012) [1] argued that WPF is a reliable technology in the manufacture of a GUI. When combined with the method MVVM, processing applications can be more focussed, because the method of MVVM that divides an application into three parts, namely Model (back end Voyage Account), ViewModel (back end Window Voyage Account), and View (front end or Window) Technology WPF and MVVM pattern be applied in applications Voyage account.

According Akiki, Pierre A. Aroscha K. Bandara, and Yijun Yu [2] applications design that created already adapted to the whim of the client. However, sometimes users who use these applications do not necessarily have the same opinion on the application design that has been made, because of the experience and knowledge of each user is different. Thus, if the user desires varied met, factors such as performance, price, and time will increase, impacting applications and users themselves. This problem can be solved with adaptive design, in which the method of this design, the varied desires can be met. Because, applied adaptive design is the simplest design application and each user can add their own functionality according to the needs and desires of each. For example, the default view Adobe Illustrator CS6, functions basic functions such as to draw a square or circle that has been provided on the toolbox left of the application, while using

- Sometimes the user become focused on data being inputted and less scrupulous.
- Common human error in the calculation with Excel formulas.
- Users often to perform copy and paste existing report formats to make the voyage a new report.
- Data that has been completed voyage report difficult to find because it is mixed with other data.

Once the problem is finished presented, solutions have been successfully obtained then expressed to the parties concerned. The solution is obtained by changing the input of each data obtained from only shaped excel file into an application called Voyage Account. In Voyage Account applications, it takes some window, the window to get into the app, window main menu, window to enter, view, and update the ship, window to enter, view, and update the barge, the window to enter and update the port, the window for inputting and updating fixed cost, the window to enter and view the voyage, window to personalize the employee, and the window for setting. During the display interface design for applications Voyage Account, the thing to do is sketch design display interface is used and then translate these sketches into WPF display. Here is shown a sketch design has been created and has been converted into WPF display.

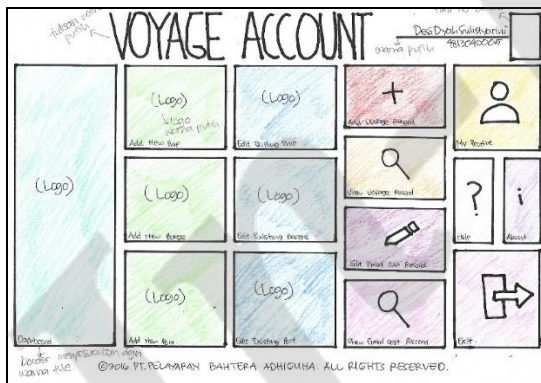


Figure 2. Window_MainMenu.xaml Sketch

Figure 2 is a sketch for Window_MainMenu.xaml. Window_MainMenu.xaml contains fifteen menu options that has been provided. Once converted into WPF, then Window_MainMenu.xaml will look like in Figure 3.



Figure 3. Window_MainMenu.xaml using WPFtechnology

In figure 3, a picture of an employee, NIP employee, employee name, and the application name in the upper part of the design of the menu display. The menu is yellow and has a user icon to personalize employee passwords and change your picture. Menu purple and have a question mark icon contains directions for use application Voyage Account. The menu is grayed out and have icon letter "i" contains a brief explanation about the application Voyage Account. The menu is pink and has a logout icon to exit the application. The menu is red and has a plus icon has a function to add record voyage. The menu is orange and has a search icon has a function to display a record of all the voyage. The menu is purple and has a pencil icon to edit record function fixed cost. The menu is pink and has a search icon has a function to display the record fixed cost. The menu is light blue and has a ship icon has a function to edit the data owned vessels. The menu is light blue and has the anchor icon is used to edit the data barge owned. The menu is light blue and has a function to edit the icon buildings owned port data. The menu is green and has a ship icon has a function to add the vessel. The menu is green and has the anchor icon is used to add data barge. The menu is green and has a building icon serves to add to the data port. Teal-colored menu icon box and has a function to see the last voyage, recent gains, total voyage, and graphs of all the voyage.

3.2. The Making of Database

Stan Gibilisco [7] found that SQL Server 2012 is a relational database management system (RDBMS) from Microsoft are designed for corporate environments. So the use of SQL Server 2012 is ideal for storing data obtained from the application Voyage Account. Once the requirements established, Entity Relationship Diagram (ERD) created by translating the data contained in an Excel file. ERD pieces can be seen in Figure 4. Ramez Elmasri (2016) [8] describes the type of

relationship (relationship) used in ERD generation. Types of relationships include One-to-one (1: 1), One-to-many (1: N), and Many-to-Many (M: N).

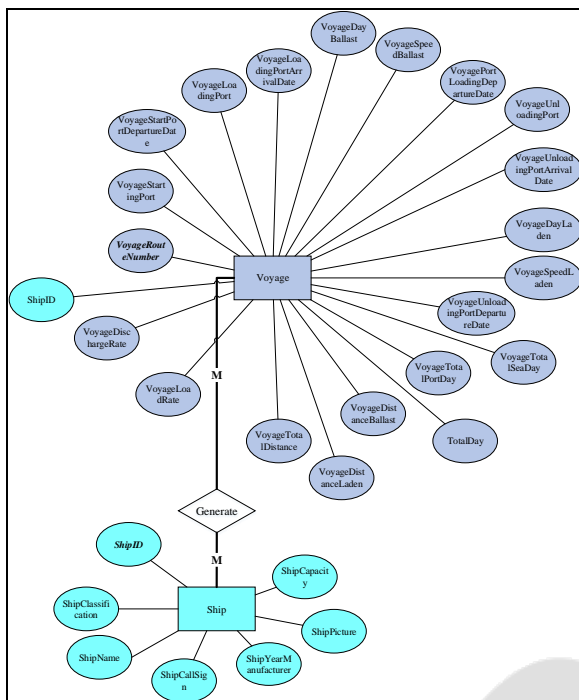


Figure 4. ERD translation of the Excel file Voyage 2015-2016 Period

SQL Server 2012 can be optimized, according to Costel Gabriel Corlatan and Marius Mihai Lazar (2014) [9] SELECT query optimizers in SQL Server, with the addition of index, view, and statistics. This is very important given the efficiency of the resources used (saving the use of Random Access Memory and Central Processing Unit). Searching data with this method requires a longer query than search in general as well as time savings of less than a second to search for some data from hundreds of existing data, but if users want to search data from millions of existing data rows, this method is perfect because can save time and save usage of RAM and CPU, this is possible because with the index method, SQL Server does not need to search the data throughout the millions of data, simply search the data on the index-index that is determined then the role of view on search, SQL Server does not need to work hard in menelurusi all the field names in the table, just browse the fields which are specified in the view.

In making the application of Voyage Account (after user requirements are met) the work is divided into two phases: creation of databases and back-end coding.

Then created a table in the database schemas based on user requirements. User requirements are

filled with a total of 14 tables. The following will be mentioned as much as 3 table as an example:

- Employees.Employee, table which contains details about the employee, such as Employee Identification Number (NIP), the name, place and date of birth, email, photos, and a password.
- Armada.Voyage, table which contains details of each trip fleets, such as trip number, company name, dates, duration of the trip, the name of the port of loading, unloading port name, the name of the power plant, the number of days over the sea, the number of days during dipelabuhan, the speed of the fleet with the charge, the speed of the fleet without charge, distance from the port of loading to the port of loading, the distance from the loading port to the power plant, the speed of loading and unloading speed.
- Armada.Ship (query can be seen in Figure 5), which contains a detailed table of ships, such as id ship, vessel name, call sign ship, ship classification, in shipbuilding, the number of cargo ships, and the image of the ship.

```

CREATE TABLE Armada.Ship
(
    ShipID INT IDENTITY(1,1) PRIMARY KEY NOT NULL,
    ShipName VARCHAR(100) NOT NULL,
    ShipCallSign VARCHAR(100) NOT NULL,
    ShipClassification VARCHAR(100) NOT NULL,
    ShipYearManufacturer VARCHAR(100) NOT NULL,
    ShipCapacity VARCHAR(100) NOT NULL,
    ShipPicture VARCHAR(100) NOT NULL
)
GO
CREATE TABLE Armada.Barge
(
    BargeID INT IDENTITY(1,1) PRIMARY KEY NOT NULL,
    BargeName VARCHAR(100) NOT NULL,
    BargeClassification VARCHAR(100) NOT NULL,
    BargeYearManufacturer VARCHAR(100) NOT NULL,
    BargeCapacity VARCHAR(100) NOT NULL,
    BargePicture VARCHAR(100) NOT NULL
)

```

Figure 5. SQL Query 2012 to make a table Armada.Ship and Armada.Barge in database schemas Fleet

Once the database, the database schema, and the table is made, the next thing to do is to make the relationship between the primary key table by using references. This is done to make the relationship one to one or one to many relationship and or otherwise.

3.3. The Making of Backend Application

Once the design of the database is complete, then proceed with the design of backend applications. Voyage Apps account has seventeen features used by the user. These features correspond to the usecase diagram in figure 7.

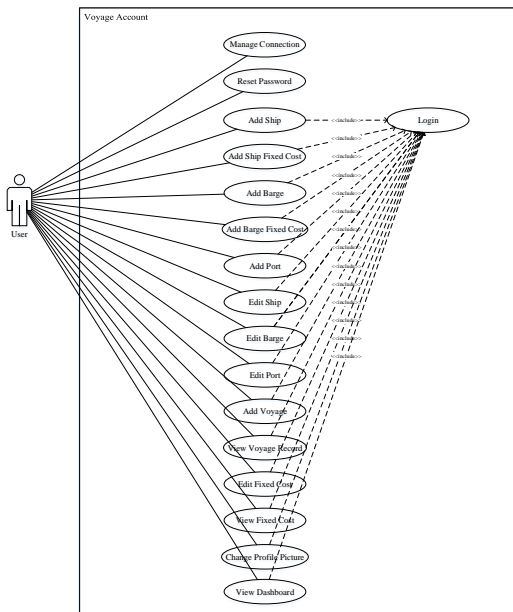


Figure 6. Usecase Diagram Voyage Account

The whole function is done in a separate class with CRUD.cs name (can be seen in a purple piece of code is the class name to the image in 8). Margaret Rouse [9] of the opinion that C # is a programming language that supports object-oriented programming (OOP). This is possible because C # supports method of object-oriented programming (OOP), so each of these functions being the object of which there are objects that are smaller. The framework used in the manufacture of Voyage Account application is .NET Framework 4.5.2. Microsoft [11] found himself .NET Framework is a framework used by developers of Windows-based applications. .NET provides a variety of options for use in manufacturing applications ranging from desktop applications (Windows Forms), a web-accessible applications (ASP.NET) to System.Xml (Defined Data Including XML using XSLT and XPath). With this method is applied, the workmanship and the use of each function can be maximized. For example scenario, if the latest voyage of data entered by the user wants, the user must log in first and then selected Add Voyage Record (the red button with the icon added in Figure 7) in Window_MainMenu and final process is data entered by the user voyage. In the process of inclusion of new data, three sets of functions through which the user, wherein the method of OOP three strands are taken of each object, so there are three functions that stand alone, ie login function, the function of the menu selection on the main menu, and the data storage function voyage into the database. With the sharing of the work into three functions, the work can be focused, because if there is a fault on one of the functions, enough to justify the wrong function does not

need the entire job. Another advantage possessed this method is able to use back functions the same as the example of the next scenario, a user wants to view the data voyage that has been stored, the first step is the user must be logged on, then Window_MainMenu traversed by choosing View Voyage Record (button is orange with magnifying glass icon in figure 7) and the last view data voyage. Login process is equally passed the first scenario (the latest voyage of data included user) and the second scenario (data stored voyage that users see), with OOP method login code (the code can be seen in figure 8) is made once and can be used for both scenarios.



Figure 7. Window_MainMenu Voyage Account Interface

```

using System;
using System.Collections;
using System.Data;
using System.Data.SqlClient;

namespace VA_BAG.Model
{
class CRUD
{
    Config con = new Config();
    private SqlConnection sqlcon;

    // ACCOUNT MANAGEMENT
    public string[] Login(string UN, string PW)
    {
        sqlcon = con.ConnectSQL();
        string[] details = new string[3];
        using (sqlcon)
        {
            sqlcon.Open();
            string sql = "SELECT EmployeeNIP,
EmployeeName, EmployeePhoto FROM Employees.Employee
WHERE EmployeeNIP = @NIP AND EmployeePassword =
@PW";
SqlCommand sqlcom = new SqlCommand(sql, sqlcon);
using (sqlcom)
{
            sqlcom.Parameters.Add(new SqlParameter("@NIP", UN));
            sqlcom.Parameters.Add(new
SqlParameter("@PW", PW));

            SqlDataReader dr =
sqlcom.ExecuteReader();
            while (dr.Read())
            {
                details[0] = dr.GetString(0);
                details[1] =
dr.GetString(1);
                details[2] =
dr.GetString(2);
            }
            sqlcon.Close();
        }
        return details;
    }
}
}

```

Figure 8. *public string[] function Login(string UN, string PW) in a group class Account Management Voyage Account CRUD.cs*

Bits of code in figure 8 are printed in red is a SQL query to retrieve data EmployeeNIP, EmployeeName, and EmployeePhoto in the database. Whether or not the inspection process and the presence or absence of employee data in the database with code snippets in blue in Figure 9 were checked through two parameters that the user entered (UN string, the string PW - UN means that the username and password PW mean). If the user data is in the database, then these data were taken with the piece of code in green and placed in a static array size of three index (string [] details = new string [3]; and each index used as a place for data EmployeeNIP the details [0], EmployeeName in details [1], and EmployeePhoto in details [2]) then the data already stored in the array is returned to Window_MainMenu (because after a successful login Window_MainMenu be displayed) with code snippets "return details;".

In figure 9, explained about the data that has been retrieved and sent from class CRUD.cs towards

Window_MainMenu to be displayed in the user information.

```

namespace VA_BAG.View
{
    public partial class Window_MainMenu
    {
        CRUD MASTER = new CRUD();
        string setNIP, setName, setPhoto;

        public Window_MainMenu(string empNIP, string empName, string empPhoto)
        {
            InitializeComponent();
            setNIP = empNIP;
            setName = empName;
            setPhoto = empPhoto;

            this.lblNIP.Content = setNIP;
            this.lblName.Content = setName;

            BitmapImage Photo = new BitmapImage();
            Photo.BeginInit();

            string startuppath =
System.IO.Directory.GetCurrentDirectory();
            Photo.UriSource = new Uri(startuppath +
"\Picture\\" + setPhoto + "");
            Photo.EndInit();

            imageEmp.Source = Photo;
        }
    }
}

```

Figure 9. *Pieces of back end Window_MainMenuVoyage Account*

In figure 9, pieces of code are printed in red are required Window_MainMenu parameters, three parameters obtained from the contents of a static array in the process of a successful login (login code in figure 8). Then, code snippets in green in figure 9 is the way to put the Main Number Employees at the front end of applications, code snippets purple is a way to put the name of an employee at the front end applications, and code snippets shown in blue is the way to put photos of employees at the front end application. So it can be seen in the user information (figure 10 circled in red) is a front end view of applications the data retrieved from the database (figure 10).



Figure 10. *User information on Window_MainMenu Voyage Account*

3.4. The Interface Testing

When encoding has been completed, then the overall application testing is done. To test design, originally done installing applications with a screen resolution of 1366 x 768 (figure 11).

Design has been made with good produce output so as to make the application in accordance with the resolution of the screen.

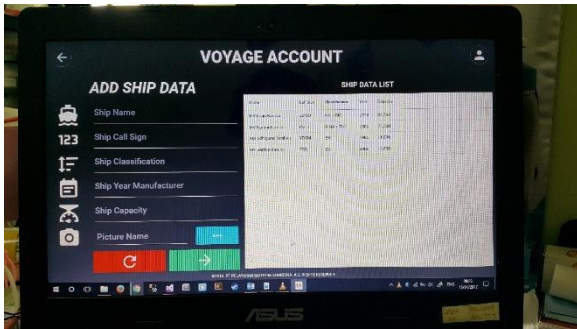
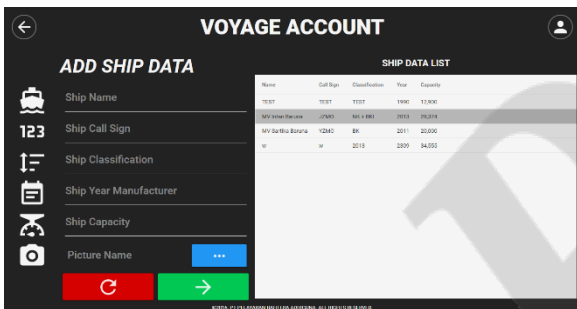


Figure 11. Interface display with 1366 x 768 resolution

Furthermore, the next design test with a screen resolution of 1920 x 1080 (figure 12). Design has been created to produce output properly so as to make the application in accordance with the screen resolution.



Gambar 12. Interface display with 1920 x 1080 resolution

The last design test is the tests carried out with a screen resolution of 1280 x 720 (Figure 13). Design has been created to produce output with either making the appropriate application of user expectations.



Gambar 13. Interface display with 1280 x 720 resolution

After fulfilling all user requirements and the tests on the design and features of, the Accounts Voyage applications have been made and are applied in the manufacture of voyage report. Here

is a perceived advantage by the Division of Business and Fleet Operations, which is:

- Users only need to enter the data that has been held in the space provided.
- Create a user focus, because the user must fill out each form with the sequence and in accordance with the groove. This makes the user must complete a voyage data input beforehand.
- Design a more simple front end to allow a user to perform the work.
- Safer, because all of a user's activities are recorded in the Application log.
- It takes a login to log into the application, so that people who have no interest can not use the application.

3.5. The Backend Testing

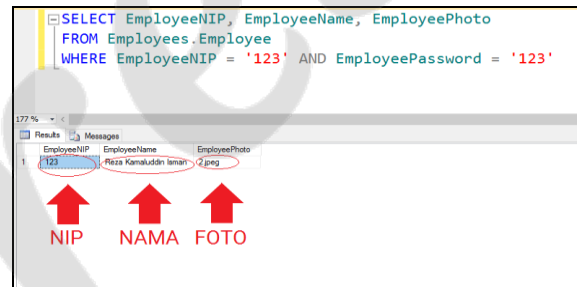


Figure 14. Testing the query login with username and password are correct so that to get the data employee

Figure 14 is a test backend applications via SQL Server 2012 through a query in the login function, query in figure 11 return a value of the stored data. With two parameters are correct (EmployeeNIP and EmployeePassword) then select the results in the database (figure 12 circled in red) in accordance with the figure 11 which is the information displayed on the front end application. With the incompatibility of values between figure 11 and 12 can be concluded that the backend (logic and code) applications are functioning properly.

4. Summary

Voyage Account is an application used to know all the details of every voyage. Back end applications created with the C # language provided by the .NET Framework. The database used is SQL Server 2012, the development method used is prototyping. Some of the advantages gained when using these applications is the gain or loss is calculated faster, more convenient because users can simply enter data and do not need to enter a formula (Excel), Voyage Report made automatically once the data is loaded, search Voyage Report easier because it

can be searched by date voyage or name of the fleet, more secure because it is needed to log when using the application and all activities performed by users will be logged.

5. Suggestion

The suggestions for application development Voyage Account include the addition of features already planned, which is:

- Tracking features, these features are used by the staff of the Division of Business and Fleet Operations to monitor the ship's position. This is done so that staff can quickly locate the vessel without the need to ask questions directly to the crew who served on board.
- Features a voyage to import data in Excel format, this feature is used to facilitate data entry voyage that has been held from the Excel format into the application Voyage Account.
- Realtime voyage features, this feature is intended to allow staff who work on board ships can enter the data CO, FW, HSD, MDO and MFO according to usage. This can allow the process of reporting done faster. Because when the application is made, the data inputted into the application data collected in advance from the staff on board who then reported to the Division of Business and Operations when the voyage is completed.

Bibliography

- [1] Soni, S., Prannali Dhete, Shirish Patil, and Dr. B.B. Meshram, "Industrial Automation using Windows Presentation Foundation & Model View View-Model Pattern", International Journal of Advanced Research in Computer Engineering & Technology, Vol. 1, 2012, pp. 149-152.
- [2] Akiki, Pierre A. Arosha K. Bandara, and Yijun Yu, "Using Interpreted Runtime Models for Devising Adaptive User Interfaces of Enterprise Applications", Paper presented at the meeting of the 14th International Conference on Enterprise Information Systems (ICEIS 2012), Worclaw, Poland, 2012.
- [3] Microsoft Developer Network, *Visual Studio IDE*, 2016.
<https://msdn.microsoft.com/en-us/library/dn762121.aspx>
- [4] Patil, S. Sameer Soni, Pranali Dhete, and Dr. B. B. Meshram, "Application Development Using WPF", International Journal of Advanced Research in Computer Engineering & Technology, vol. 1, 2012, pp. 480-483.
- [5] Shylesh, S. and Pai, V, "Use of Windows Presentation Foundation and Windows Forms in Windows Application Programming", International Journal of Advanced Trends in Engineering and Technology (IJATET), Vol. 1, 2016, pp. 138-141.
- [6] Microsoft, *Introduction to WPF*, 2015.
[https://msdn.microsoft.com/en-us/library/mt149842\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/mt149842(v=vs.110).aspx)
- [7] Gibilisco, Stan and Rouse, Margaret, *What is SQL Server 2012?*, 2012.
<http://whatis.techtarget.com/definition/SQL-Server-2012>
- [8] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems 7th Edition*, 2016.
- [9] Corlatan, C. G. and Lazar, M. M., *Query Optimization Techniques in Microsoft SQL Server*, Database Systems Journal, Vol. 5, 2014, pp. 33-48.
- [10] Rouse, Margaret, *Object-oriented programming defition*, 2014.
<http://searchsoa.techtarget.com/definition/object-oriented-programming>
- [11] Microsoft, *Welcome to the NET Framework*.
<https://www.microsoft.com/net/framework>