

¹ Iqbal Aurelio
Tawakal *

Prototype Data Processing System Wide Information Management Using Architectural Design Model View Controller



Abstract: - System Wide Information Management is a standard, infrastructure, and governance that enables the management of information related to ATMs and their exchange between qualified parties through interoperable services. SWIM aims to improve interoperability, efficiency, environment, and security and reduce spending. However, in Indonesia itself, the implementation of the Wide Information Management System has yet to be announced, so the exchange of data from ATS Message, Aeronautical, and Meteorological has yet to be fully integrated. Therefore, it is necessary for data processing to integrate some of the information data but still present the main functions listed in the ICAO Doc. 10039 by adapting the Model View Control (MVC) architectural design pattern, which has advantages over conventional ones. To implement the MVC architectural design, we will use the PHP Laravel framework and MySQL database, which is implemented by the functions of SWIM, namely converting raw data, displaying data, converting compressed XML in GZIP format, and being able to exchange data between users. A sending and receiving data simulation will be carried out to exchange data between countries using SWIM Indonesia with Testbench Singapore via the TCP/IP protocol.

Keywords: SWIM, Laravel, PHP, XAMPP, Database, MVC, MySQL, Raw data

1. INTRODUCTION

International Civil Aviation Organization (ICAO) is an organization with countries as its members aiming to develop the principles and techniques of international air navigation, encourage the planning and development of international transport, and promote the development of all aspects of international civil aeronautics.

ICAO has a work plan called the Global Air Navigation Plan (GANP) to achieve these goals and development. GANP is a comprehensive framework that includes aviation policy principles to help the region ICAO, sub regional, and state with the preparation of regional and state air navigation planning.

The purpose of GANP is to increase the capacity and efficiency of global civil aviation and improve or at least maintain security. GANP also includes strategies for addressing the strategic objectives of ICAO. GANP has a plan framework called Aviation System Block Upgrades [1]. ASBU provides a roadmap to assist air navigation service providers in developing their respective strategic plans and investment decisions to achieve global interoperability of aviation systems [2].

System-wide information Management (SWIM) is a standard, infrastructure, and governance that enables the management of related information ATM and its exchange between eligible parties through operable services. On ASBU, there are also blocks and modules related to SWIM, such as B1-SWIM and B2-SWIM. B1-SWIM explains improving performance using the application of SWIM and implementation of services SWIM in terms of applications and infrastructure, thereby creating an aviation intranet network based on model standard data and internet protocols to maximize interoperability. Meanwhile, B2-SWIM contains the application of aircraft participation in ATM collaboration via SWIM and the application of connections from aircraft as information on the application of SWIM participation in the ATM collaboration process with access to increasing volumes of dynamic data including meteorology. SWIM manages some information data through exchange models that apply to information, including ATS Message, Aeronautical, and Meteorological[3].

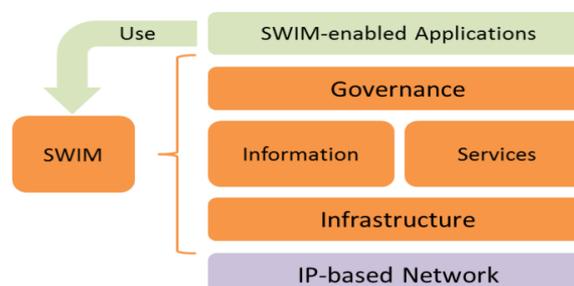


Figure 1. Scope of SWIM
Source: ICAO Doc. 10039

¹ Air Navigation Engineering Study Program, Ministry Transportation of Republic of Indonesia, Indonesia

*Corresponding author: iqbalarelio@gmail.com

Copyright © JES 2024 on-line : journal.esrgroups.org

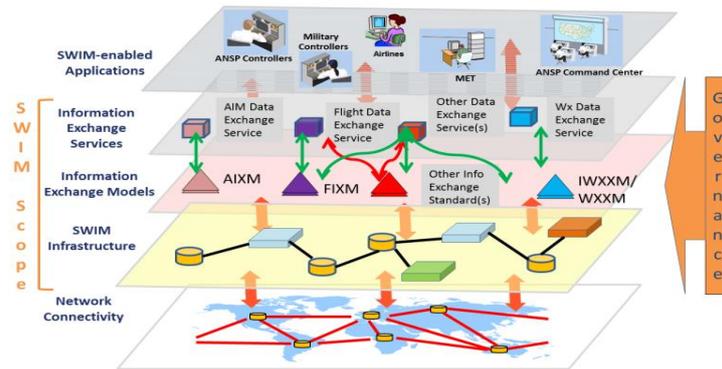


Figure 2. Scopes of SWIM

Source: ICAO Doc. 10039

In addition to improving interoperability, SWIM has several advantages in terms of efficiency, environment, security, and small expenditure [1]. In Indonesia, the SWIM application has yet to be launched, so data exchange from Message, Aeronautical, and Meteorological still needs to be fully integrated. SWIM is also one part of supporting the implementation of the national aviation navigation order following the concept ASBU ICAO in building an efficient, targeted, and integrated national aviation communications network [4].

To enable SWIM to manage certain information data in the form of exchange models, data processing is necessary. Data processing is simplifying data into a form that is easier to read and interpret to obtain conclusions about evaluation results [5]. Data processing must be user-friendly so that all groups can use it, and it must also be easy to carry out the process of maintenance or maintenance. All these advantages are found in MVC. Where MVC is an application design architecture that breaks the system into classes Model, View, and Controller so the system is more structured [6].

Therefore, it is necessary for data processing to integrate several of these information data but still provide the main functions listed in ICAO Doc. 10039 by adapting architectural design patterns of Model View Controller, which has advantages compared to native or conventional.

Previous research regarding SWIM Laily has done it [7], which makes bank data plans for SWIM, and then Faisal [8] implements the bank data design with the application design SWIM uses PHP Native. Luo Yi et al. [9] did other research regarding SWIM, which analyzes and compares three alternative technical architectures that SWIM needs to improve because SWIM will become a developing and perfect system if it is developed continuously.

Mohammed carried out research in creating web-based applications with MVC architectural design using Laravel, which makes code development easier and increases application security. Problem identification includes the disintegration of the ATS Message, Aeronautical, and Meteorological data exchange systems, and the need for a user-friendly SWIM system. Problem constraints include restrictions on prototypes, data processing in XML format, and testing with data samples from a specific time range. The problem formulation includes designing a SWIM application with an MVC design, applying the design pattern, and evaluating it with BlackBox Testing. The goal is to become a reference prototype for creating SWIM applications, implementing architectural design patterns in SWIM, and evaluating SWIM with MVC architectural design patterns. The benefit is to provide an overview of the implementation of SWIM in Indonesia following ASBU and ICAO Doc 10039 in the future.

2. METHODS

2.1 Design Method

The design method used in this design is the prototyping method. The prototyping method is a method that consists of 3 phases, namely analysis, design, and implementation. These three phases will be repeated until a system is successfully realized. After a prototype is installed, improvements continue until the system presents something new or becomes a new system.

2.1.1 Planning

The purpose of making a prototype SWIM to fulfill the system that will be implemented in the future because it has the aim of integrating exchange Model ATS Message, Aeronautical, and Meteorological based on ICAO Doc.9750 in the form of XML to make it easier to read the system. Data from ATS Message, Aeronautical, and Meteorological were taken from Perum LPPNPI JATSC Branch over time.

2.1.2 Analysis

This phase analyzes SWIM's functionality requirements and system limitations, which will be developed. The data obtained previously is raw data, which is still formatted as WORLD, FRM, and SELL. So that the user can see the data in table form, the integration process is carried out with raw data in the database MySQL. According to ICAO Doc. 10039, Data already in tabular form can be converted into a form XML. Data that has been formed XML is automatically compressed into a zip with format .gz which format .gz this is a compression technique for

all data types ICAO [41]. Depending on the conversion stage, different programming languages will be used for the conversion process. For more details, see the following table.

Table 1. Data Conversion Stage

Data Conversion Stage	Programming Language Used
Conversion Raw data into table form	SQL
Convert tables into XML form	PHP
Data compression XML into .gz form	PHP

2.1.3 Design

In this phase, determine design from the prototype SWIM it works. The SWIM design will use the PHP Laravel framework because prototype SWIM will use architectural design Model View Controller which has advantages compared to native. With the separation between Model, View and Controller then the process of creating and improving applications will be easier and faster. Design MVC this will combine the functions that have been mentioned in the stage analysis thus producing a system prototype SWIM which is functional. For more details on the integration of functions SWIM towards design MVC.

2.1.4 Implementation

The final phase is a phased implementation, which implements the previous stages. As explained in stage design, to simulate data exchange will use Testbench Singapore making the testing process easier. The testing process will use a protocol TCP/IP because it doesn't allow him to use the system AMHS.

2.2 Tools and materials

Planning Prototype data processing System Wide Information Management requires several tools and materials, including the following.

Table 3. Tools and Materials

No.	Tools and materials	Function
1.	Laptop HP Pavilion 15-cw	As a medium for making designs
2.	ATOM 1.60.0	As a text editor
3.	Laravel 9	A Framework PHP making architectural designs MVC
4.	XAMPP v3.3.0	As a provider application localhost and MySQL
5.	MySQL 5.7	As an operation database
6.	phpMyAdmin 5.20	As a management application database from MySQL
7.	PHP 5.4	As web server index programming

2.3 Design Flowchart

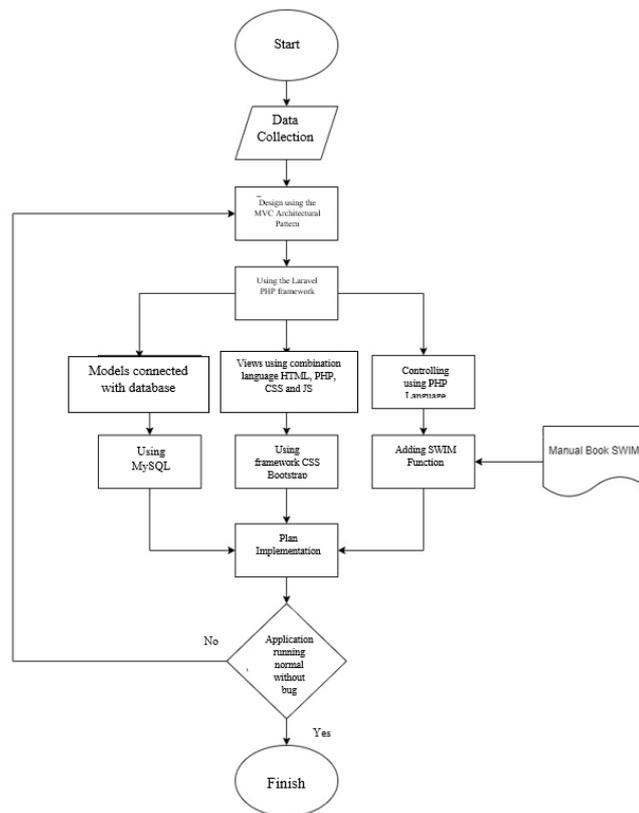


Figure 3. SWIM Planning Flowchart

2.4 Planning Criteria

Criteria are really needed because they will determine the needs that will be used to design a tool. The initial stage in designing is determining needs according to ICAO Doc. 10039 and is expected to achieve the general design objectives.

- 1) Can connect to database ATS Message, Aeronautical and Meteorological.
- 2) Can differentiate privilege between users.
- 3) Can convert raw data into forms XML which is compressed GZIP.
- 4) Can display raw data in tabular form.
- 5) Can send and receive messages XML between user.
- 6) Can change status user.

3. RESULTS AND DISCUSSION

3.1 Application Planning

3.1.1 Planning

SWIM is a system that will be implemented in the future because it aims to integrate the exchange model ATS Message, Aeronautical, and Meteorological. Planning Prototype SWIM provides data exchange ATS Message, Aeronautical, and Meteorological between users. This data exchange can also convert data ATS Message, Aeronautical, and Meteorological into an extended form XML.

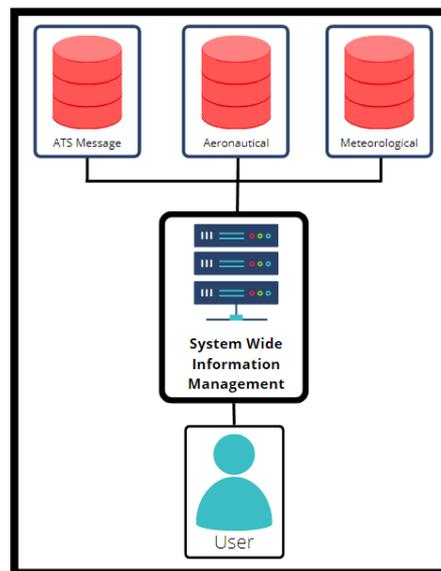


Figure 4. Data integration by SWIM

3.1.2 Analysis

Analysis of system functionality requirements SWIM according to the level planning previously. The data that will be processed are ATS Message, Aeronautical, and Meteorological. From raw data derived from AMHS, the format FRM, WORLD, and SELL will be processed into a Graphical User Interface (GUI) so that users can easily interact with datasets Message, Aeronautical, and Weather forecast astrology. The user can change the data to XML Format Data and compress it into .gz form, then forward it to another user if that user wishes. System SWIM has different privileges. Up to a few users have limitations in accessing other data. The following are privileges from each user. To support the implementation of Testbench trials, a Testbench was created in Singapore with a dashboard different from the dashboard default. Testbench Singapore intended to simulate data exchange between plans SWIM and Testbench Singapore. Therefore, a database was made that contains users and the country to differentiate between users with Indonesia and Singapore. Every user's SWIM dashboard is different according to each county.

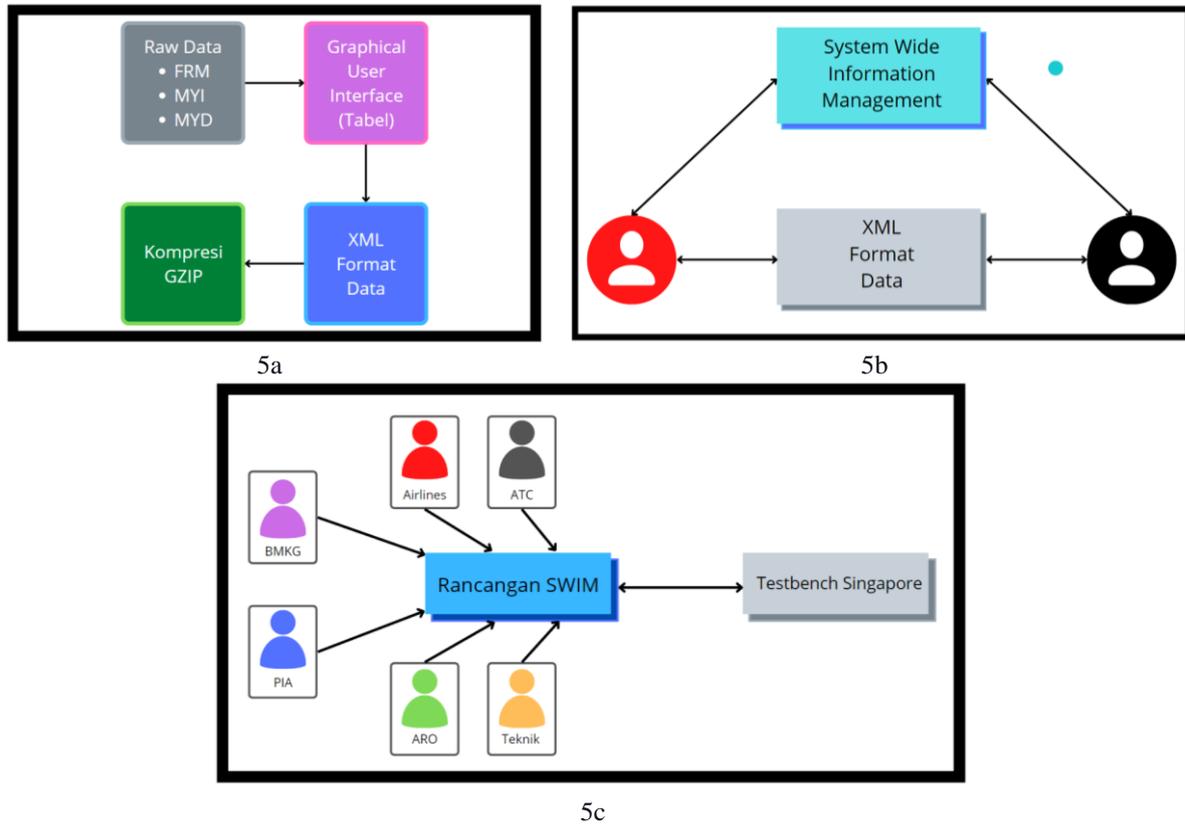


Figure 5a. SWIM data conversion process; 5b. Data exchange interactions between SWIM users; 5c. Visualization of users connected to the SWIM design

3.1.3 Design

This stage explains the process of designing a system SWIM, from design to architectural design, containing designs MVC and UML.

1) Design

System SWIM will use an architectural design model view controller, which has advantages over native ones. With the separation between Model, View, and Controller then, the process of creating and improving applications will be easier and faster. For data, ATS messages and aeronautical and meteorological messages can be accessed and processed by users. The user interface of the SWIM system must be made functional and easy to use. To make this happen, the system menu SWIM was made in such a way. Where is the system SWIM has the ability to process data from ATS Message, Aeronautical, and Meteorological. Another ability is to change raw data into XML format, which is compressed in form GZIP, and file XML compressed GZIP. This can be sent or received by Inter user SWIM.

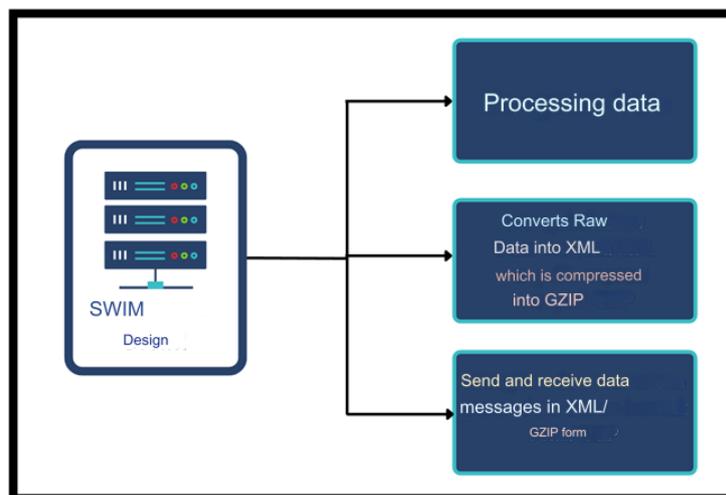


Figure 6. Menus that will display the SWIM Plan

2) Architectural Design

This time, the design will use the PHP Laravel framework because the SWIM design will adopt the MVC architectural design.

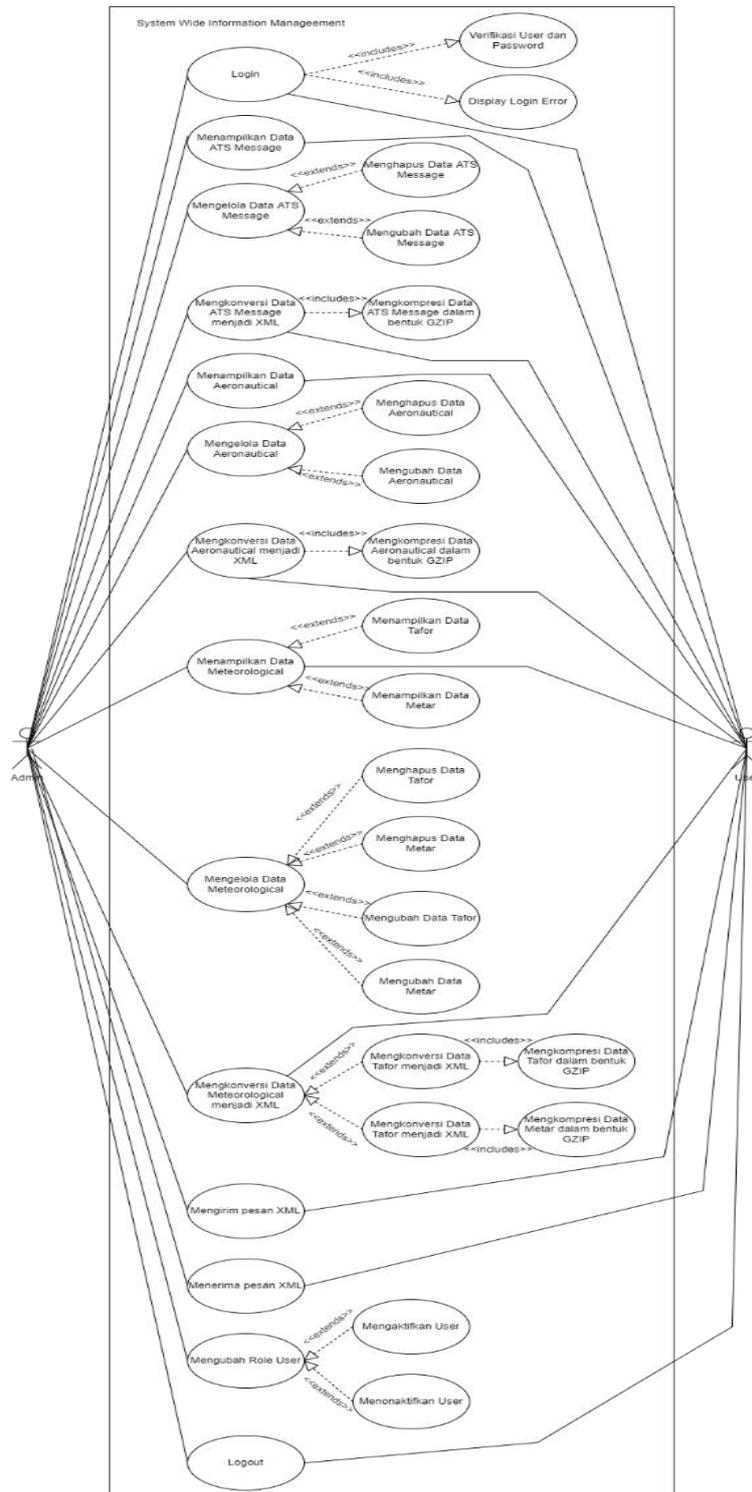


Figure 7. Use Cases from SWIM Design

3) Implementation

Implementation is the stage after the design of a system has been carried out. After the design of the system SWIM has been implemented, the overall implementation must be carried out in terms of software installation, implementing the design, and testing the system design SWIM.

4) Design Implementation

After the software is installed, the process of implementing the SWIM design continues. Implementation takes the form of integrating Laravel projects, databases, and MVC.

a. Accessing Laravel Project

Access the previously created Laravel project using the Composer command, which will generate a file structure.

b. Running MySQL

To run MySQL, you must use the XAMPP application to activate the MySQL module. If MySQL is already running, the Start button will change to the Stop button.

c. Creating Databases

After installing the software, a MySQL database can be created using phpMyAdmin, which is integrated with XAMPP. With phpMyAdmin, the swim MVC database is created to operate with the SWIM system. To activate phpMyAdmin, run the Apache module as localhost and select the admin button on the MySQL menu to enter the phpMyAdmin display page. After the phpMyAdmin display appears, create a database.

d. Integration of Raw Data into the database

Integrating raw data into the swim mvc database that has been created. The following is the raw data file from ATS Message, Aeronautical, and Meteorological. Copy all the files into the /XAMPP/data/MySQL/SWIMVC folder so that all the files are integrated into one single database.

e. Connect Database

Connecting Database swims MVC with projects SWIM has been done previously using Laravel by changing file .env using ATOM in the folder project SWIM. Connect the project SWIM with the database swim MVC by changing the file .env in the project Laravel SWIM_MVC and the code.

f. Column Creation Users and Messages

In order for the function of SWIM It work well from the start, a column was created for users. Users this was created to accommodate Admin and user information from the SWIM This system. Both are in terms of status, role, country, privilege, password, and other functions regarding the user's identity.

Apart from column users, a message column is also created, which aims to accommodate message data XML sent from each user to the user. The message is left blank at first because it is intended to contain messages.

g. Making Models, Views and Controllers

Making Models, Views, and Controllers refers to the architectural design subchapter. These three classes are implemented as one unit. After all the classes from Models, Views, and Controllers are integrated into the Laravel project, the structure of the Laravel project is created.

h. Make Dashboard SWIM Indonesia and Testbench Singapore

Two dashboards will be made to simulate data exchange as stated in stage analysis: Dashboard Indonesia and Dashboard Singapore. To differentiate between Dashboard Indonesia and Dashboard Singapore, create a code for the layout onViews.

i. Running a Project Laravel

After all processes have been carried out, the system has become one unit, resulting in a system SWIM. To run the project you need to use the command. After executing this command, the IP Address will appear. The IP address is the address used to access the project Laravel SWIM. Open the IP Address via Chrome.

5) Design Trial

After implementation, the SWIM design will be tested according to the functions that have been implemented based on the analysis stage.

a. Login to SWIM

Before entering the main system, the login page will be displayed for the first time. Access is required to enter the system by entering a username and password. If the username and password are not registered in database users, then the system will deny access. Conversely, if username and password input is registered in the user's database, the system will provide access and display the dashboard page according to user, role, and privilege here.

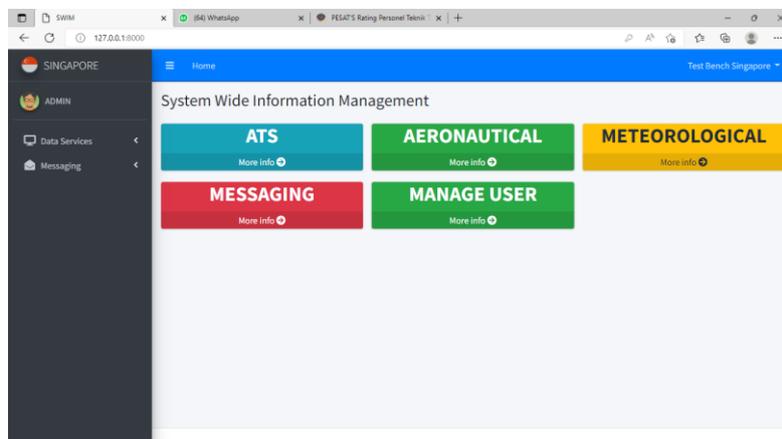


Figure 8. Views Testbench Singapore Dashboard Login

b. Viewing Data

The SWIM system is designed so that all users can see raw data that has been converted into tabular form. So that users can search for the data they need.

1) ATS

Users can view the data ATS if logged in as ATTENTION. To view and search data, you can use the feature Data Services ATS. Steps were taken to view and search for data.

1. Enter the ATS Data Services menu.

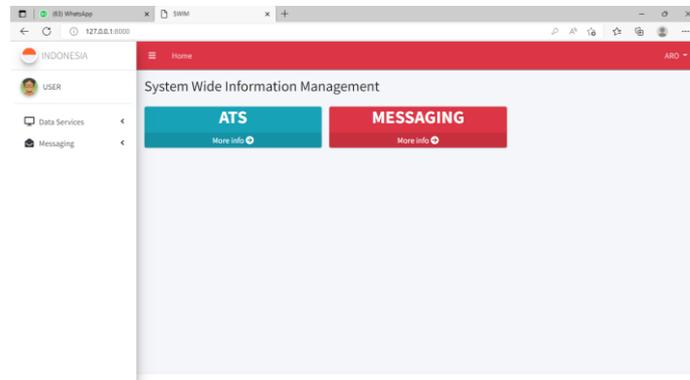


Figure 9. Dashboard display with ARO user with ATS privileges

2. Enter Aircraft ID, Year, From-To Date, From-To Time. In this case, Aircraft ID GIA553 is used with a time range from 01 March 2022 to 01 April 2022.

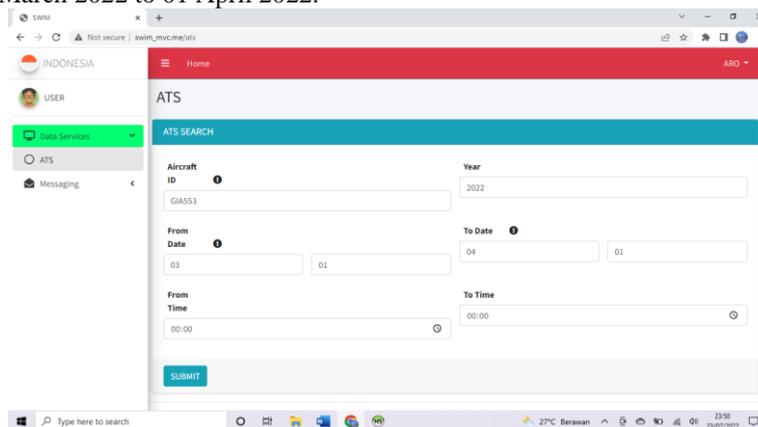


Figure 10. ATS Search page display

3. After submission, the data from GIA553 will appear in the form of a table as follows. The data that appears in the table is the data as previously input, namely GIA553 data from the time range 01 March 2022 to 01 April 2022.

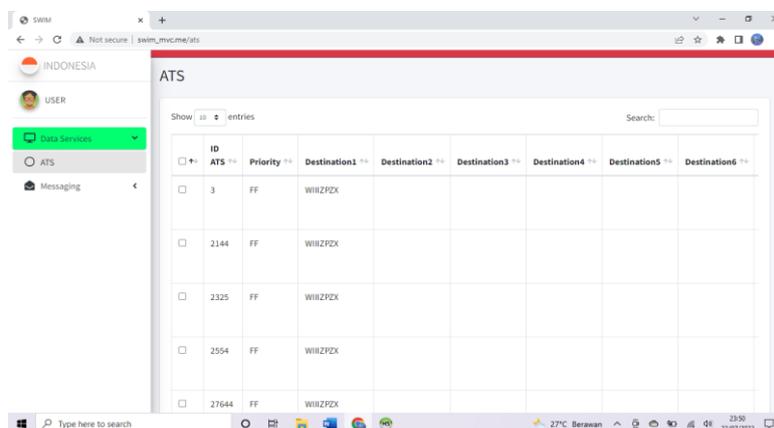


Figure 11. ATS search results in tabular form

2) Aeronautical

Users can view the ATS data if logged in as TOO. To view and search data, you can use the Data Services Aeronautical feature. Steps were taken to view and search for data. The steps taken to view and search for data are:

1. Enter the Data Services Aeronautical menu.

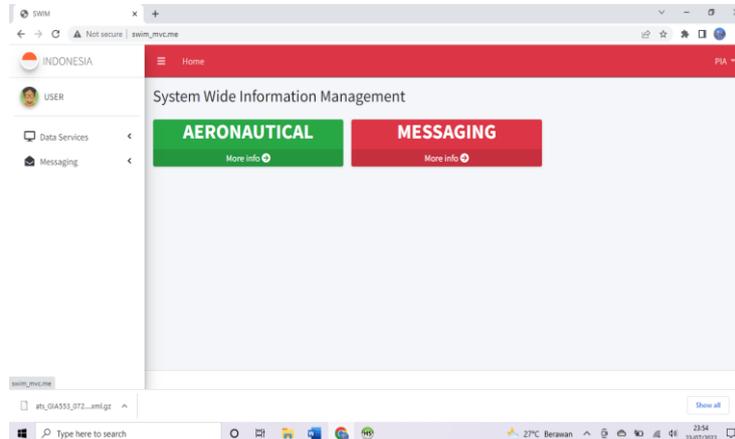


Figure 12. Dashboard display with PIA user with Aeronautical privileges

2. Enter Originator, Year, From-To Date, From-To Time. In this case, the WSSSYNYX Originator is used with a time range from 01 March 2022 to 01 April 2022.

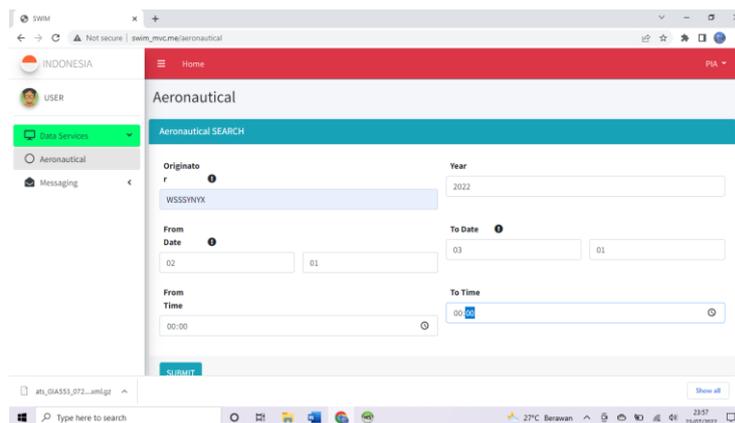


Figure 13. Aeronautical Search page views

3. After submission, data will appear from the WSSSYNYX Originator in the form of a table as follows. The data that appears in the table is the data as previously input, namely WSSSYNYX data from the time range 01 March 2022 to 01 April 2022.

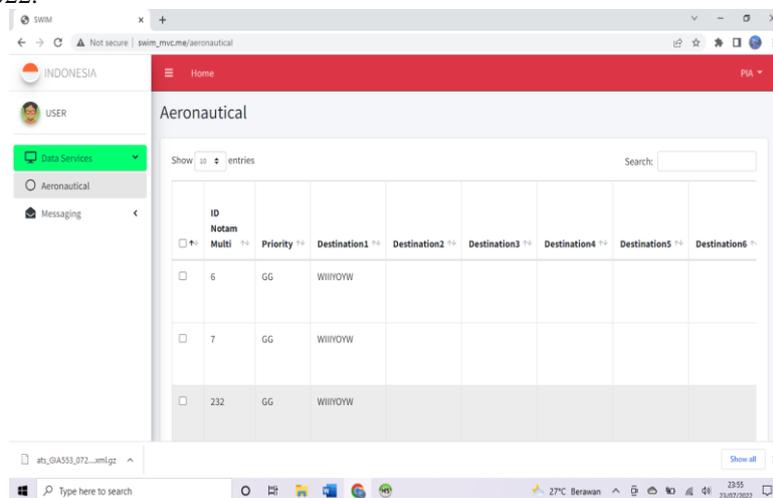


Figure 14. Aeronautical search results in tabular form

2) Meteorological

On Meteorological There are two search data, namely, *Tafor* and *Metar*. So when searching, you can select one of the data to search for. Users can view the data Meteorological if logged in as BMKG. To view and search data, you can use the feature Data Services Meteorological. Here are the step for view and search the metar data:

1. Enter Originator, Year, From-To Date, From-To Time with Data Type Metar. In this case, the WIDDYMYX Originator is used with a time range from 01 March 2022 to 01 April 2022.

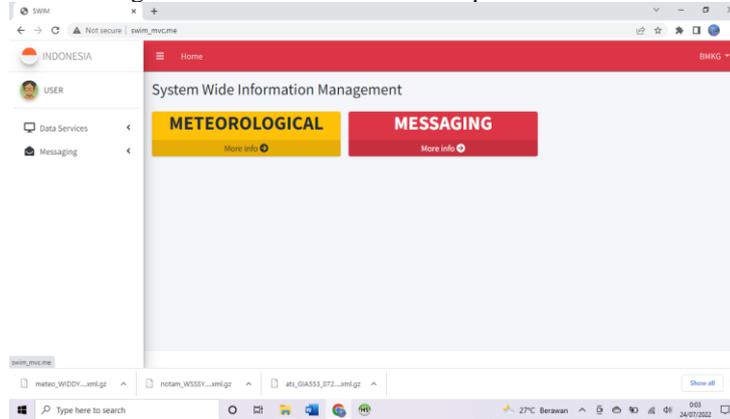


Figure 15. Dashboard display with BMKG user with Meteorological privileges

2. After submission, data from WIDDYMYX will appear in the form of a table as follows. The data that appears in the table is the data as previously input, namely WIDDYMYX data from the time range 01 March 2022 to 01 April 2022.

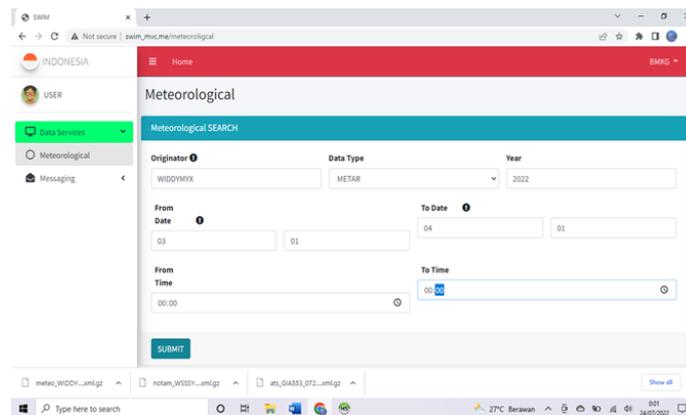


Figure 16. Metar Meteorological Search page views

3. After submission, the data from RJTDYPYX will appear in the form of a table as follows. The data that appears in the table is the data as previously input, namely RJTDYPYX data from the time range 01 March 2022 to 01 April 2022.

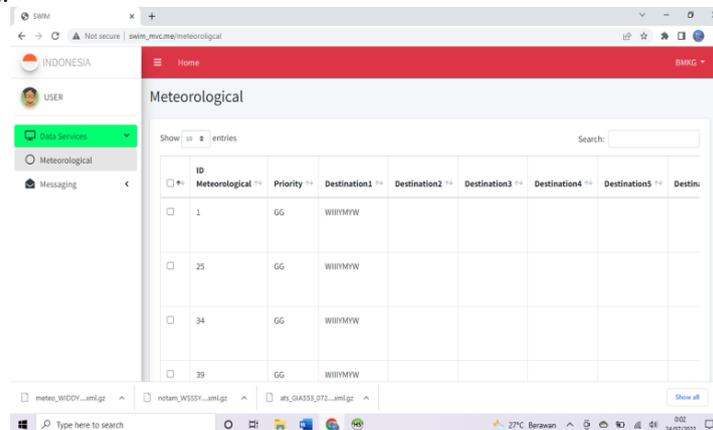


Figure 17. Metar search results in table form

The steps taken to view and search for Tafor data are:

1. Enter the Data Services Meteorological menu.
2. Enter Originator, Year, From-To Date, From-To Time with Data Type Tafor. In this case, the RJTDYPYX Originator is used with a time range from 01 March 2022 to 01 April 2022.

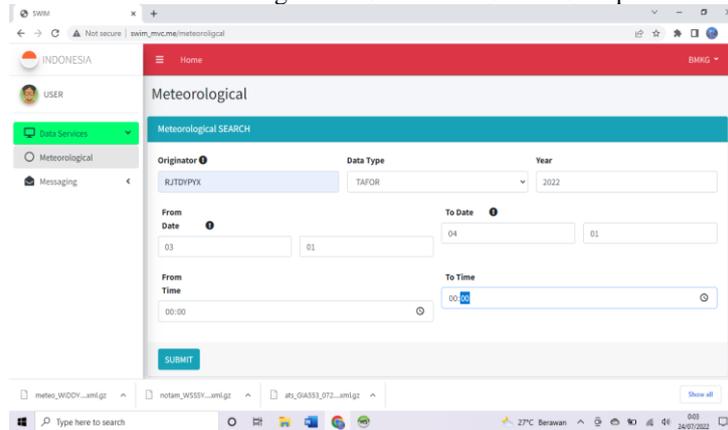


Figure 18. Tafor Meteorological Search page views

After submission, the data from RJTDYPYX will appear in the form of a table as follows. The data that appears in the table is the data as previously input, namely RJTDYPYX data from the time range 01 March 2022 to 01 April 2022.

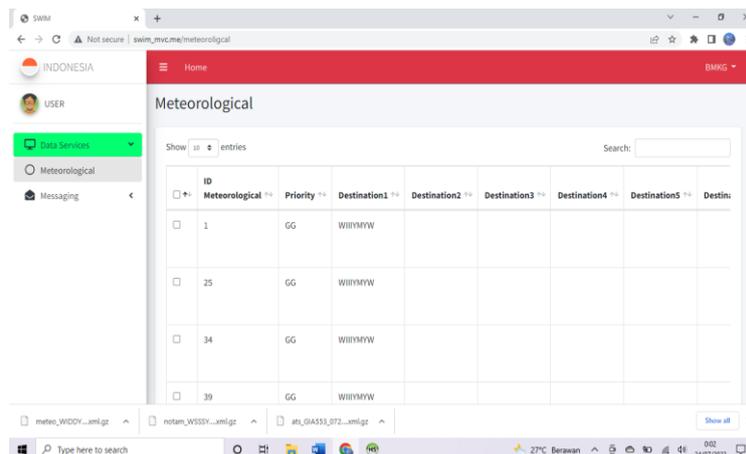


Figure 19. Tafor search results in tabular form

3) Changing Data

To change the data requires a user who has a role as an admin-like user. Step changing data is

1. By searching the data, for example using ATS data with Aircraft ID GIA553.
2. After the data is displayed in table form, select the update menu in the left corner of the table.

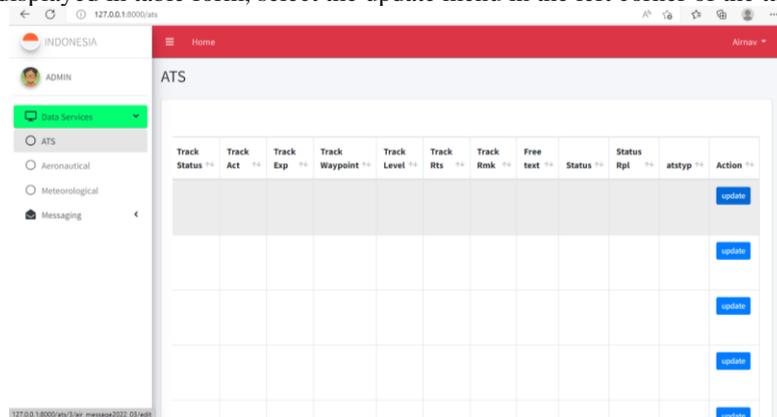


Figure 20. Display the data update page

1. After the update button is clicked, the Edit page will appear, which contains columns of GIA553 data. Each data column can be changed by the user.

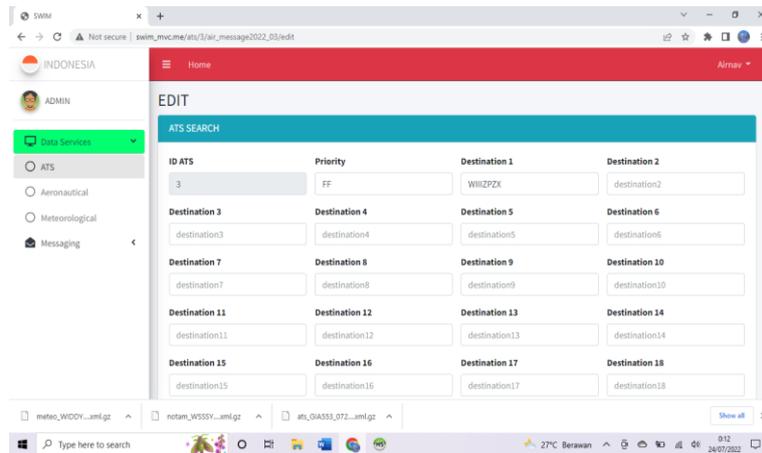


Figure 21. Display the results of the data edit page

4) XML Conversion

The SWIM system can convert table data that has been searched into an XML form. How to convert it is:

1. By searching the data, for example using ATS data with Aircraft ID GIA553.
2. After the data is displayed in table form, select which rows of data you want to convert.
3. After that, select the Convert to XML menu at the bottom of the table.

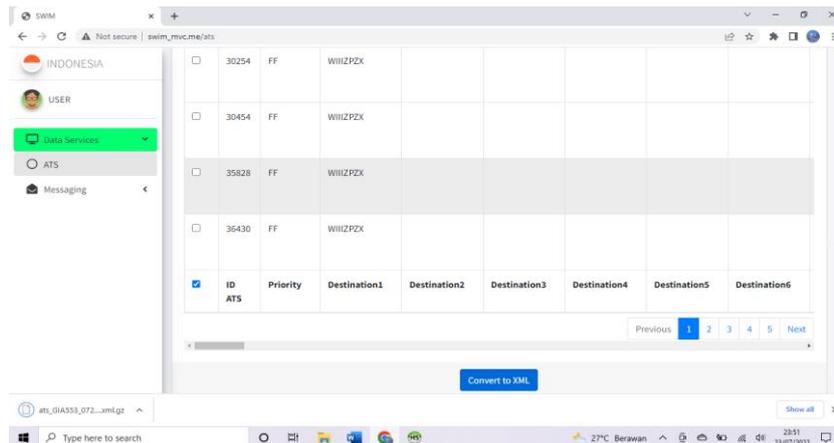


Figure 22. Convert table data into XML

8. The data will automatically be downloaded to the computer storage and the data will be in XML form which is compressed into .gz.

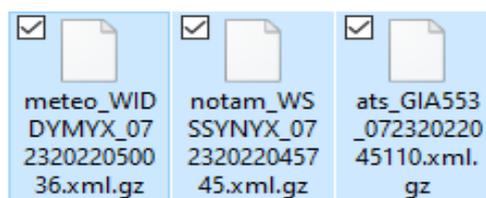


Figure 23. .gz compressed XML conversion data results

The following is the form of the XML file that has been previously converted.

```

ats_GIA553_07272022113109.xml
1 <?xml version="1.0"?>
2 <airmessage date="2022-03">
3   <data type7a="GIA553">
4     <id_ats id="3">
5       <priority>FF</priority>
6       <destination1>WIIIZP2K</destination1>
7       <destination2></destination2>
8       <destination3></destination3>
9       <destination4></destination4>
10      <destination5></destination5>
11      <destination6></destination6>
12      <destination7></destination7>
13      <destination8></destination8>
14      <destination9></destination9>
15      <destination10></destination10>
16      <destination11></destination11>
17      <destination12></destination12>
18      <destination14></destination14>
19      <destination15></destination15>
20      <destination13></destination13>
21      <destination17></destination17>
22      <destination18></destination18>
23      <destination19></destination19>
24      <destination20></destination20>
25      <destination21></destination21>
26      <filing_time>202357</filing_time>
27      <originator>W46GZP2K</originator>
28      <type3a>CH6</type3a>
29      <type3b></type3b>
30      <type3c></type3c>
31      <type5a></type5a>
32      <type5b></type5b>
33      <type5c></type5c>
34      <type7a>GIA553</type7a>
35      <type7b></type7b>
36      <type7c></type7c>
37      <type8a></type8a>
38      <type8b></type8b>
39      <type9a></type9a>
40      <type9b></type9b>
41      <type9c></type9c>
42      <time10a></time10a>

```

Figure 24. Fill in the data from the XML file

4) Delivery in XML format

Data that has been converted into XML can be sent between users using menus messaging. The way to send XML data between users is as follows.

1. Select the Compose menu in Messaging.
2. After the Compose Message page appears, select the user address that will be addressed in this example, namely the PIA user.
3. Type the title and body of the message, then upload the corresponding XML data file that has been converted previously.
4. Once everything is filled in, click the Send button.

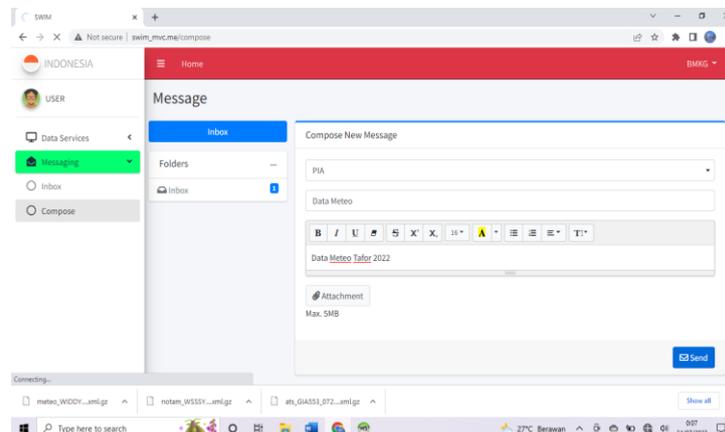


Figure 25. Compose Message page view

4) Acceptance in kind XML

The data that has been sent will go to the database until the message comes to the user, which is aimed. To view XML data messages, those who log in can use the Inbox menu. The way to view incoming XML data messages is as follows.

1. Select the Inbox menu in Messages.
2. After the Inbox Message page appears, you can see the incoming XML data message.
3. The example in the following image is an XML data message from a BMKG user with Meteorological data.

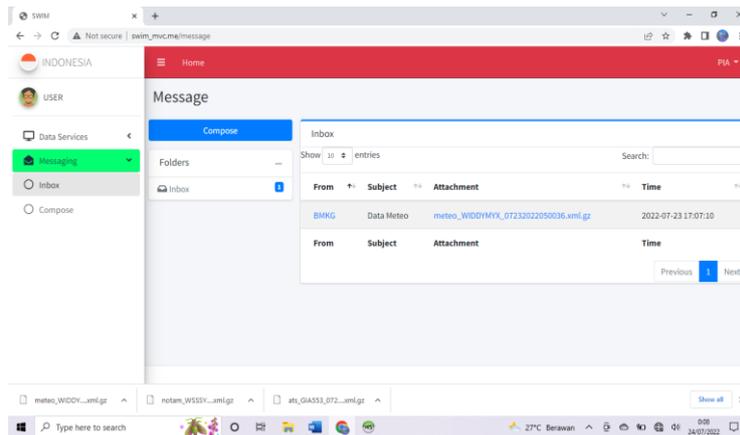


Figure 26. Inbox Message page display

To download the XML data file, click on the file name.

4) Changing User Status

To change the status of a user, a user who has the role of admin is needed. Steps to change the status of the user. Steps to change the status of a user are as follows.

1. If you have logged in using the admin user, in this example a technical user, then go to the Manage User menu.

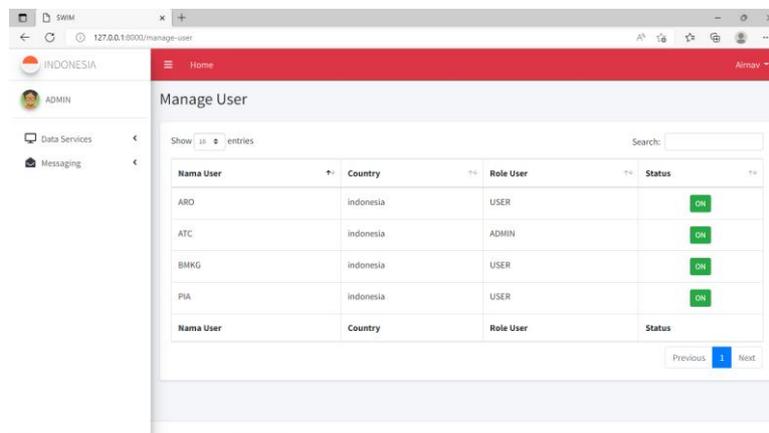


Figure 27. Display the manage user page

7. After the Manage User page appears, a table appears containing a collection of users along with Country, Role and Status.

8. To change the user status, click the ON/OFF button. Then the user's user status will change. In this example, user ARO is disabled.

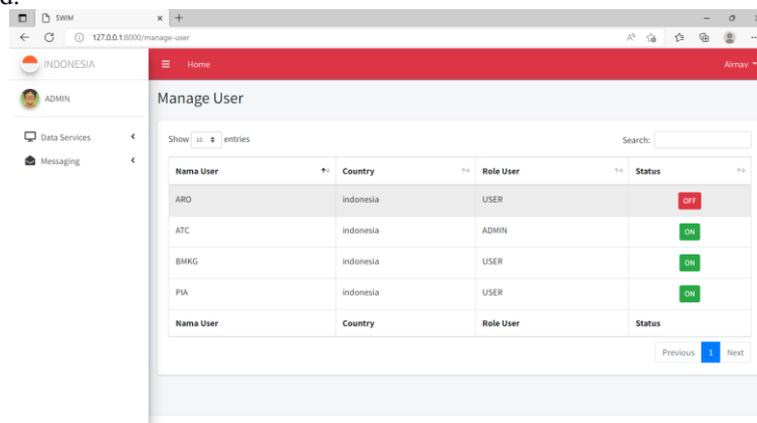


Figure 28. User management display after user ON/OFF

10. To ensure whether the ARO user is truly deactivated, log in using the username and password of the ARO user. Then the following display will appear.

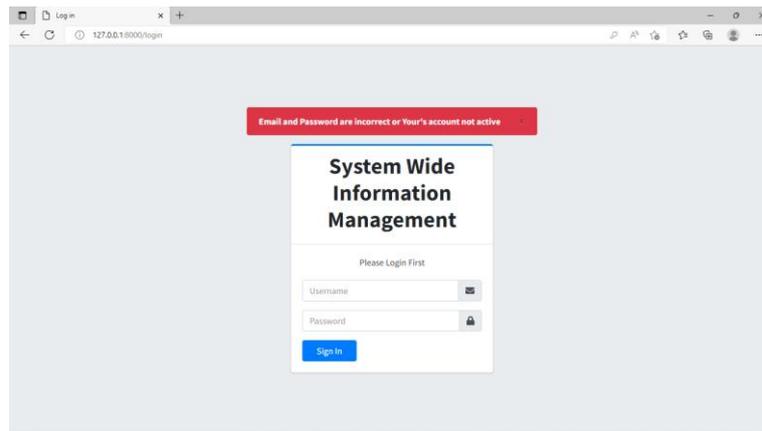


Figure 29. Display of user logins that have been disabled

4) Data Exchange with Testbench Singapore

To simulate data exchange between Plans SWIM and Testbench Singapore, testing is carried out in the form of sending and receiving data between designs SWIM and SWIM Singapore. In this test, we use user TO with country Indonesia sends data Aeronautical trouser Testbench Singapore. The data sent was received by the user Testbench Singapore.

The data sent is received by the Testbench Singapore user.

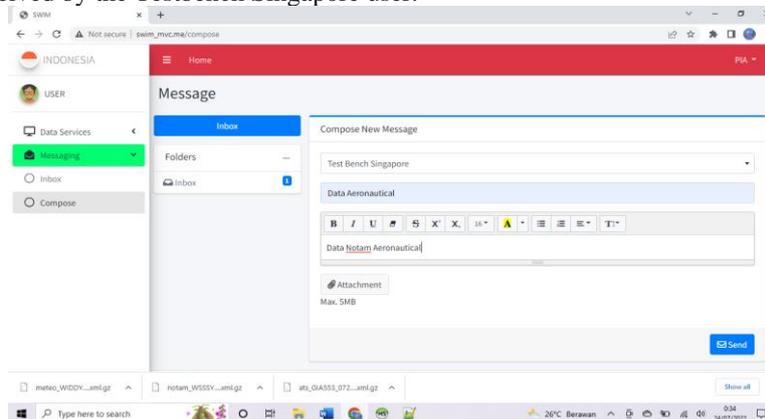


Figure 30. Sending data to Testbench Singapore

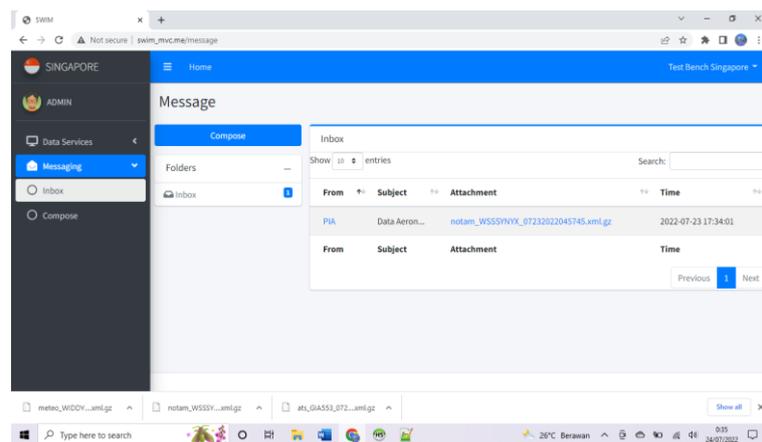


Figure 31. Receiving data in Testbench Singapore

For delivery testing, sending was carried out from the Singapore Testbench user containing Aeronautical data to PIA users with the country Indonesia.

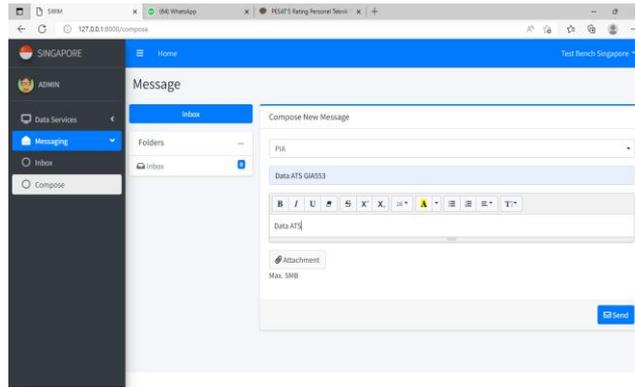


Figure 31. Sending data from Testbench Singapore
 After the Aeronautical data is sent by the Singapore Testbench user, the Aeronautical data is received in the Inbox from the PIA user with the country Indonesia.

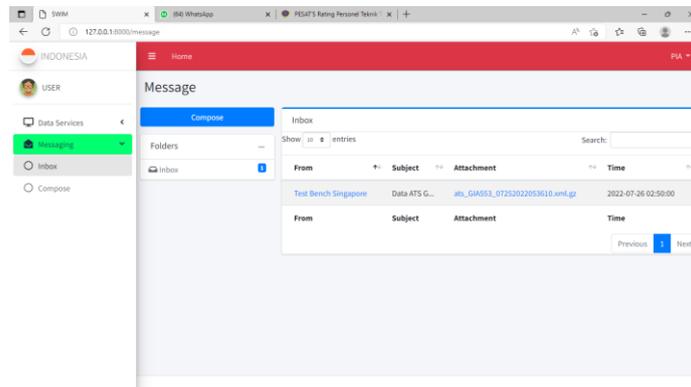


Figure 32. Receiving data from Testbench Singapore

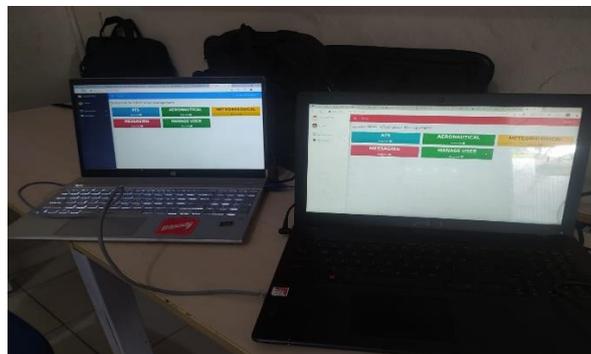


Figure 33. SWIM design connected to the Singapore Test Bench using TCP/IP Source: Personal Documentation

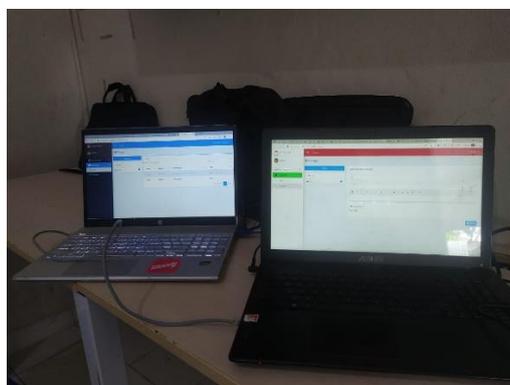


Figure 34. The SWIM design exchanges data with the Singapore Testbench using TCP/IP Source: Personal Documentation

6) Design Evaluation

Based on the design trials from the previous sub-chapter, the results of testing using the method are known as Black Box testing

7) Design Iteration

After processes such as analysis, design, implementation, testing, and evaluation have been carried out, the system will iterate. The process will return to the analysis and evaluation stages sequentially, thereby creating a system that is new and different from the previous one. Several factors influence the occurrence of iteration, such as user needs, the addition of new features, and coding errors in the program.

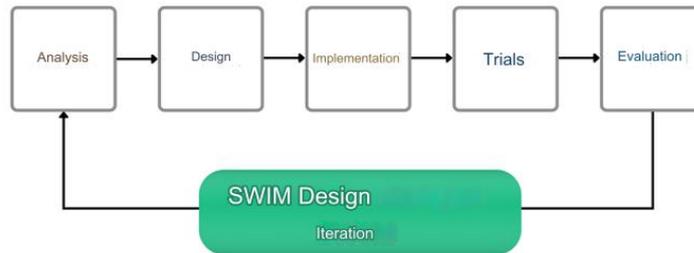


Figure 35. SWIM design iteration process

The following is an example of a coding error in the SWIM program that can cause iteration.

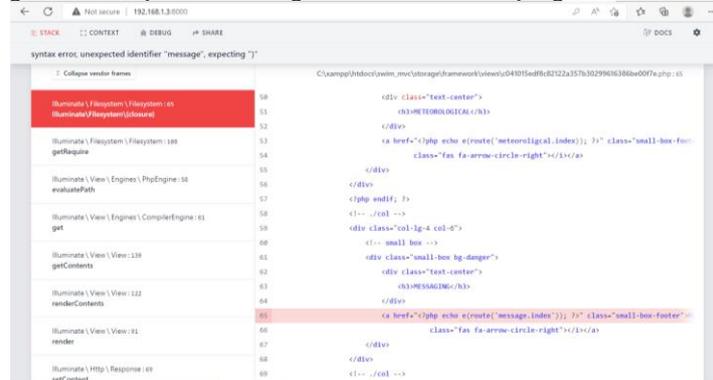


Figure 36. SWIM iteration process if a coding error occurs

By design, SWIM uses the PHP Laravel framework so that errors in the system can be found tracing quickly; when there are feature changes or feature additions in accordance with ICAO Doc. 10039, then plan SWIM. This, again, requires an iterative process. Developers will iterate so that the system SWIM will find a plan UI/UX that best fits the expectations and needs.

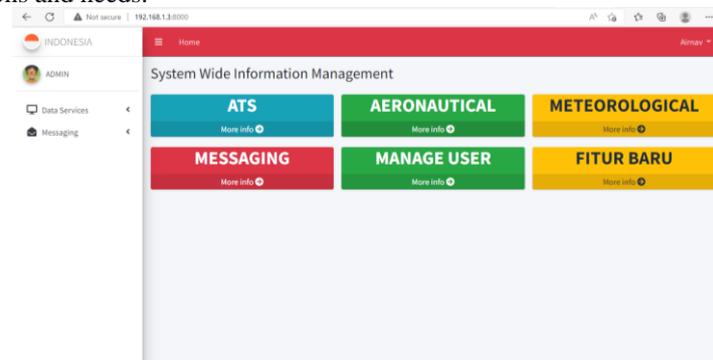


Figure 37. Iteration of SWIM design with additional features

4. CONCLUSION

Based on the test results, it can be concluded that the Wide Information Management System data processing prototype design was successfully created using the Models Views Controllers architectural design using the PHP Laravel framework. This prototype is capable of converting ATS Messages, Aeronautical, and Meteorological data into integrated data in tabular form that can be accessed and interacted with by users and can be converted into GZIP compressed XML for data exchange between SWIM Indonesia and Testbench Singapore users. Trials using the BlackBox Testing method produced results as expected, including login functions, data handling, XML

conversion, and data exchange with Testbench Singapore. The suggestion for developing this design is that System Wide Information Management can be connected to the X400 protocol on AMHS to increase data exchange capabilities.

REFERENCES

- [1] ICAO, "2016–2030 Global Air Navigation Plan: Doc. 9750 fifth edition," *International Civil Aviation Organization*, 2016, [Online]. Available: <https://www.icao.int/airnavigation/Documents/GANP-2016-interactive.pdf>
- [2] MITRE, "Introduction to the Aviation System Block Upgrade (ASBU) Modules Strategic Planning for ASBU Modules Implementation," 2013.
- [3] Icao, "Manual on System Wide Information Management (Swim) Concept".
- [4] Direktorat Jenderal Perhubungan Udara, *Kp 008 Tahun 2017 Tentang Tata Negeri Jaringan Komunikasi Penerbangan Nasional*. 2017.
- [5] Badan Pengawasan Keuangan Dan Pembangunan, "Audit Kinerja Sektor Publik Pengumpulan & Pengolahan Data," 2007.
- [6] K. Wibowo, "Analisa Konsep Object Oriented Programming Pada Bahasa Pemrograman PHP," 2015.
- [7] L. Fitriana, "Rancangan Data Bank System Wide Information Management (SWIM) Dalam Mendukung Program Global Air Navigation Plan (GANP) di Indonesia," 2021.
- [8] F. Matin, "Rancangan Aplikasi System Wide Information Management (SWIM) Dalam Mendukung Program Global Air Navigation Plan (GANP) di Indonesia," 2021.
- [9] Y. Luo, Q. Xin, P. Wang, and X. Z. Chen, "System wide information management architecture designing," *Advanced Materials Research*, vol. 765–767, pp. 1585–1588, 2013, doi: 10.4028/www.scientific.net/AMR.765-767.1585.
- [10] M. T. Mahmood and O. I. A. Ashour, "Web application based on MVC laravel architecture for online shops," Sep. 2020. doi: 10.1145/3410352.3410834.
- [11] Alan Dennis, "System Analysis and Design," 2012.
- [12] S. Nidhra, "Black Box and White Box Testing Techniques," *International Journal of Embedded Systems and Applications*, vol. 2, no. 2, pp. 29–50, Jun. 2012, doi: 10.5121/ijesa.2012.2204.
- [13] P. Sulistyorin, "Pemodelan Visual dengan Menggunakan UML dan Rational Rose," 2009.
- [14] N. M. Rifqi Tsani, "Sistem Informasi Ujian Berbasis Web Smk Bina Islam Mandiri (Bisma) Kersana," vol. 1, no. 1, pp. 45–54.
- [15] H. Riyadli, A. Arliyana, and F. E. Saputra, "Rancangan Sistem Informasi Keuangan Gereja Berbasis Web Di Jemaat GMIM Bukit Moria Malalayang," *Jurnal Sains Komputer dan Teknologi Informasi*, vol. 3, no. 1, pp. 98–103, 2020, doi: 10.33084/jsakti.v3i1.1770.
- [16] ICAO, "Asia/Pacific Regional Aeronautical Telecommunication Network (ATN) Air Traffic Service (ATS) Message Handling System (AMHS) Description," 2005.
- [17] Dio Lavarino, "RANCANG BANGUN E – VOTING BERBASIS WEBSITE DI UNIVERSITAS NEGERI SURABAYA," *Science of Surveying and Mapping*, vol. 41, pp. 72–81, 2016.
- [18] A. Josi, "Penerapan Metode Prototyping Dalam Membangun Website Desa (Studi Kasus Desa Sugihan Kecamatan Rambang)," *Jti*, vol. 9, no. 1, pp. 50–57, 2017.
- [19] T. Yuliano, "Pengenalan PHP," *Ilmiu Komputer*, pp. 1–9, 2017.
- [20] D. Naista, "Bikin Framework PHP Sendiri dengan Teknik OOP dan MVC," *Bikin Framework PHP Sendiri dengan Teknik OOP dan MVC*, 2016.
- [21] F. Luthfi, "Penggunaan Framework Laravel dalam Rancang Bangun Modul Back-End," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 2, no. 1, pp. 34–41, 2017, doi: 10.14421/jiska.2017.21-05.
- [22] Eiril Obeit Choiri, "Pengertian Framework Laravel Beserta Fitur dan Keunggulannya," 2020. <https://qwords.com/blog/pengertian-framework-laravel> (accessed Mar. 01, 2022).
- [23] J. Armel, "Web application development with Laravel PHP Framework version 4," 2014.
- [24] R. Nelson, A. Shukla, and C. Smith, "Web Browser Forensics in Google Chrome, Mozilla Firefox, and the Tor Browser Bundle," 2020, pp. 219–241. doi: 10.1007/978-3-030-23547-5_12.
- [25] R. Abdulloh, "Easy & Simple Web Programming - Belajar Pemrograman Website Secara Efektif Dan Efisien," p. 5, 2016.
- [26] H. Alatas, "RESPONSIVE WEB DESIGN dengan PHP dan BOOTSTRAP," *Eduvantage*, vol. 1, no. 1, 2013, doi: 10.11635/2319-9954/1/1/18.
- [27] F. Effendy and B. Nuqoba, "Penerapan Framework Bootstrap Dalam Pembangunan Sistem Informasi Pengangkatan Dan Penjadwalan Pegawai (Studi Kasus:Rumah Sakit Bersalin Buah Delima Sidoarjo)," *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, vol. 11, no. 1, p. 9, 2016, doi: 10.30872/jim.v11i1.197.
- [28] M. R. Adhiasta, "Perancangan Informasi Batik Solo Berbasis Bootstrap," *Publikasi Ilmiah*, p. 2, 2016.
- [29] S. Mariko, "Aplikasi website berbasis HTML dan JavaScript untuk menyelesaikan fungsi integral pada mata kuliah kalkulus," *Jurnal Inovasi Teknologi Pendidikan*, vol. 6, no. 1, pp. 80–91, 2019, doi: 10.21831/jitp.v6i1.22280.
- [30] L. Beighley, *JQuery For Dummies*. 2010. [Online]. Available: <http://books.google.com/books?id=HH7epRPQMkEC&printsec=frontcover%5Cnpapers2://publication/uuid/66E8A968-973B-469F-AD67-F547A42D5C77>
- [31] A. ANDARU, "PENGERTIAN DATABASE SECARA UMUM," *Proceedings of the 1970 25th Annual Conference on Computers and Crisis: How Computers are Shaping our Future, ACM 1970*, pp. 1–7, doi: 10.1145/1147282.1147284.
- [32] A. Sofwan, "Belajar Mysql dengan Phpmysqladmin Pendahuluan," *Modul kuliah Graphical User Interface I (GUI) di Perguruan Tinggi Raharja*, pp. 1–29, 2011.

- [33] Rahmawati Erma Standsyah and Intannia Sari Restu N.S, "IMPLEMENTASI PHPMYADMIN PADA RANCANGAN SISTEM PENGADMINISTRASIAN".
- [34] H. Ramadhani and A. R. Nugraha, "Aplikasi web pengiriman dan penerimaan sms dengan gammu engine berbasis php," vol. 01, no. 01, pp. 191–200, 2018.
- [35] J. S. Pasaribu, "Penerapan Framework Yii Pada Pembangunan Sistem Ppdb Smp Bppi Baleendah Kabupaten Bandung," *Jurnal Ilmiah Teknologi Infomasi Terapan*, vol. 3, no. 2, pp. 154–163, 2017, doi: 10.33197/jitter.vol3.iss2.2017.132.
- [36] Pastima. Simanjuntak and A. Kasnady, "Analisis Model View Controller (Mvc) Pada Bahasa Php," *Jurnal ISD*, vol. 2, no. 2, pp. 56–66, 2016.
- [37] H. Deviana, J. T. Komputer, N. Sriwijaya, J. Srijaya, N. Bukit, and B. Palembang, "Penerapan XML Web service Pada Sistem Distribusi Barang," *Jurnal Generic*, vol. 6, no. 2, pp. 61–70, 2011, [Online]. Available: <http://www.nusphere.com>
- [38] A. Shah and M. Sethi, "The Improvised GZIP, A Technique for Real Time Lossless Data Compression," *EAI Endorsed Transactions on Context-aware Systems and Applications*, vol. 6, no. 17, p. 160599, Jun. 2019, doi: 10.4108/eai.1-10-2019.160599.
- [39] J.-L. Gailly, "GNU gzip," 2022.
- [40] A. M. Syafar, "Kendali Perangkat Listrik Dan Monitoring Daya Pada MCB Berbasis TCP/IP," *Jurnal Instek*, vol. 1, no. 1, 2016.
- [41] ICAO, "Guidelines For The Implementation Of OPMET Data Exchange Using IWXXM," 2020.