

# The Design of UML-Based Sales Forecasting Application

Rendra Gustriansyah, Nazori Suhandi, Fery Antony

**Abstract:** It has now become an important research direction by applying UML as a concept of software development model. The Unified Modeling Language (UML) is a standard modeling language for visualizing the design of a system to make it easier for software developers to build a system. In this research, we designed software for sales forecasting application using best worst method (a new multi criteria decision making method) and adopted RFM concept (recency-frequency-monetary, one of the most popular data mining techniques) based on UML (use case diagram, class diagram, and sequence diagram). The proposed design is expected to accelerate the software development process, reduce complexity and minimize errors while building the software. Finally, through usability testing, sales forecasting application that has been designed based on UML had a user acceptance score of 85.0 (excellent level), so that the design of UML-based sales forecasting application can be used by all users.

**Index Terms:** data mining, design, sales forecasting application, UML

## I. INTRODUCTION

Forecasting is a method to estimate a future value using historical data. Meanwhile, sales volume forecasting is an activity to estimate how much product sales by producer or distributor in certain time period and marketing area, so that demand fluctuation can be overcome. Sales volume forecasting is also part of the management function as one of the contributors to the success of a company [1].

In this research, we focus on designing software for sales volume forecasting based on UML, because the design of good software will make software development time more efficient, effective, and can minimize common encoding errors.

Software can be defined as instructions (computer programs) that when executed provide the desired features, computing functions, delivering content to the end users and performance [2].

Software is divided into the following categories [2]: (1) system software, which is a collection of programs written to service other programs, such as compilers, editors, and file management; (2) application software, which is a stand-alone program that solves certain business needs, such as the

development of a toolkit for calculations [3]; (3) engineering/scientific software, which uses “number crunching” algorithm, such as software to detect attacks over a network [4]; (4) embedded software, which is in a product or system and used by the end user to control the features and functions of the system itself, such as control of electrical devices using microcontroller [5]; (5) product-line software, designed to provide a specific capability for use in a limited and esoteric marketplace, such as decision support system capable of providing alternative solutions to a problem [6]; (6) web software (WebApps), is a network-centric software that includes a wide range of applications; such as software to be designed in this research; (7) artificial intelligence software, which uses non-numerical algorithms to solve complex problems, not for easy computing or analysis, such as expert system-based cellular software [7].

For information, the methods that will be involved in this sales forecast are BMW and adopt the RFM concept. The best-worst method (BWM) is a new multi-criteria decision-making method (MCDM) introduced by Rezaei in 2015, which can derive criteria and alternatives by comparing the best criteria with the worst criteria based on simpler pairwise ratios [8]. Meanwhile, RFM (recency, frequency and monetary) is a powerful and well-known concept in database marketing, and is widely used to measure customer value based on their previous purchase history. The RFM concept has also been integrated into the mining process in recent years [9].

The software design in this study is based on UML. The Unified Modeling Language (UML) is a standard modeling language for visualizing the design of a software to make it easier for software developers to build a software [2]. UML is commonly used to model software architectures as per requirements and it includes a set of graphical notation techniques to create an intensive visual model of software systems. The use of UML diagrams that are standard, complete, correct, and unambiguous in building software can generate efficient and fast program code.

UML diagrams consist of two main categories, namely behavioral diagrams and structural diagrams. This research will use a behavioral diagram (use case diagram) and structural diagram (class diagram). The next section will discuss the diagram.

## II. RESEARCH METHOD

The system in this research was developed with PHP and Android using a monthly dataset in the one-year period (from January to December 2015) obtained from [6]. The dataset of the research were imported from the table of product

Revised Manuscript Received on 30 March 2019.

\* Correspondence Author

**Rendra Gustriansyah\***, Faculty of Computer Science, Universitas Indo Global Mandiri, Palembang, Indonesia.

**Nazori Suhandi**, Faculty of Computer Science, Universitas Indo Global Mandiri, Palembang, Indonesia.

**Fery Antony**, Faculty of Computer Science, Universitas Indo Global Mandiri, Palembang, Indonesia.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

# The Design of UML-Based Sales Forecasting Application

(6,877 items), the table of sales order (399,738 records) and the table of sales order details (3,956,683 records).

In detail, the dataset consists of eight tables as follows:

- 1) Product: a table to store all product data.
- 2) Transaction: a table which is the result of pre-processing from sales order table and sales order details table.
- 3) Scoring: a table to store scores of each criterion of each product.
- 4) Forecast: a table to store data forecasting sales volume of each product.
- 5) Realization: a table to store actual data of sales volume of each product every month.
- 6) Error: a table to store the difference between the forecasting sales volume data and actual sales volume data.
- 7) Classification: a table to store values from the lower limit and upper limit of each criterion.
- 8) Weights: a table to store the weight of each criterion which is the result of evaluation using the best-worst method (BWM).

After all components are set up, the next step is to design software using UML. UML 2.0 provides 13 different diagrams for use in software modeling, and in this paper is simply represented by diagrams of 'use case', 'class', and 'sequence'.

Meanwhile, user interface design will be evaluated the usability. Evaluation of usability is an evaluation conducted to figure out the easiness of the system usage. This evaluation is used to measure the user acceptability of the application being developed. John Brooke adopted the ISO 9241-11 in measuring usability with three main criteria: Effectiveness (the ability of users to complete tasks using the system, and the quality of output of the tasks), Efficiency (level of consumption of resources in performing tasks), Satisfaction (user's subjective reactions to using the system) [10].

The usability level is measured using an evaluation tool 'System Usability Scale' (SUS) [10]. The SUS can measure rapidly how people's subjective perceptions of the usefulness of their systems/applications. SUS is a simple usability scale, consisting of 10 subjective statements. Each statement is evaluated using the likert scale of points 1 through 5. Statements 1, 3, 5, 7, and 9 (odd items) are positive statements, while statements 2, 4, 6, 8, and 10 (even items) are negative statements [11]. Table I shows the answer score of the statement [12], and Table II shows the list of statements in the questionnaire [13].

**Table I. The score of the answers [12]**

Score	Description
1	Strongly Disagree
2	Disagree
3	Neutral
4	Agree
5	Strongly Agree

**Table II. The list of statements in the questionnaire [13]**

No	Statements
1	I think I will often use this application
2	I found the application unnecessarily complex
3	I thought the application was not difficult to use.
4	I thought I would need someone's help to be able to use this application.
5	I saw various functions in the application well integrated

- 6 I thought there were too many inconsistency in the application
- 7 I would imagine that most people would learn to use this application very quickly
- 8 I found this application is very complicated to use
- 9 I felt very confident using the application
- 10 I needed to learn a lot of things before I could get going with this application

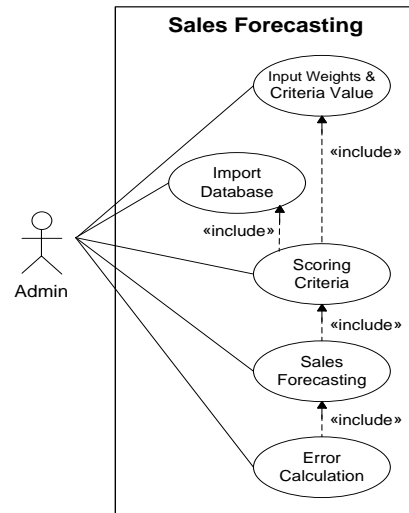
Scores of every odd (positive) statement, calculated in a manner, scale each statement (Si) minus 1 or (Si-1). Meanwhile, scores of every even (negative) statement are calculated in a way, 5 minus the scale of each statement (Si) or (5-Si), so the total score is obtained by summing all the scores of each statement (even and odd). The final score of SUS is the total score multiplied by 2.5 and the final score is in the range 0-100. Based on SUS final score will be known how much the level of usability and acceptability of application design developed. The acceptance category in SUS consists of three categories namely "Not Acceptable" with a score range 0-50.9, "Marginal" 51-70.9, and "Acceptable" 71-100 [13].

## III. RESULTS AND ANALYSIS

### A. UML Use Case Diagram

The use case diagram is a technique used in the development of a software or system to capture the functional requirements of a system. The use case diagram is used to construct behavioral things in a model, since the use case diagrams can explain the interactions that occur between the users and the system itself. A use case diagram can define functionality and software features from user's perspective.

The design of sales forecasting application that adopt RFM concept is done by modeling the use case diagram as shown in Fig. 1.



**Fig. 1: The use case diagram for the sales forecasting application**

The use case diagram in Fig. 1 can be used as the initial stage of development of this system. In this use case diagram, you can see five use cases and one actor. Each use case is self

explanatory and can represent the interaction between functions of the sales forecasting application. Meanwhile, the actor describes someone who interacts with the system. There are five use cases used in the system, the first is the function to input the value of each weight and criteria that most influence the sales volume based on the evaluation of the best-worst method. Second is the function to import data from the database of a case company. The third use case is the function for projecting the customer value into numbers. The fourth use case is sales forecasting that serves to forecast the sales volume of each product for the coming month, and the last use case is the calculation of sales forecasting errors (accuracy).

**B. UML Class Diagram**

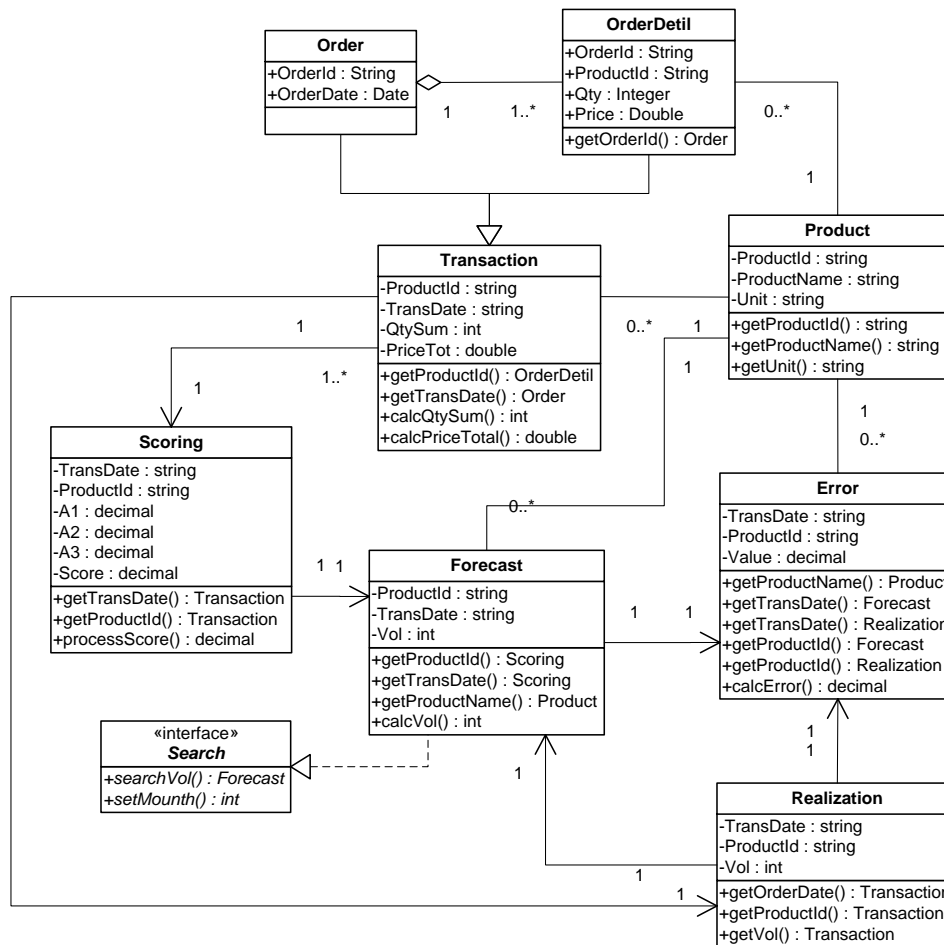
The class diagram is one of the main diagrams of UML that describes a class or blueprint object on a system. The analysis of class diagram formation is a core activity that greatly

influences the software architecture designed up to the coding stage.

UML Class Diagrams describe the static structure of classes within a system, which shows what interacts but not what happens when they interact. The class diagrams contain the system's classes, attributes, operations and relationships between classes.

Fig. 2 is a class diagram design that still requires customization when it will be used as a guide in building sales forecasting application at a later stage.

The class diagram in Fig. 2 shows that the class 'Forecast' is the main class of the system. It coordinates with the 'Scoring' class and the 'Realization' class. Meanwhile, the 'Scoring' class and the 'Realization' class coordinate with the 'Transaction' class. The 'Error' class coordinates with the 'Forecast' class and the 'Realization' class.



**Fig. 2: The class diagram for the sales forecasting application**

**C. UML Sequence Diagram**

UML sequence diagram is used to show how objects interact in a use case by describing the objects and the messages they pass. An important characteristic of a sequence diagram is that time passes from top to bottom: the interaction starts from left to right, descending and ends at the bottom. The sequence diagram in Fig. 3 shows the user (admin) interaction with 'Main' to input the weights and criteria values into 'Database'. After that, the admin imports the data into 'Database'. Admin may send a message to the scoring process to 'Main' which continues the message to 'Score' after receiving confirmation back that the data has been

successfully imported. Based on the confirmation, the admin can send a process message for forecasting sales volume through 'Forecast' and 'Database'. After the calcVol() process is obtained, 'Forecast' sends a confirmation back to 'Main' stating that the sales volume forecasting process is completed. Furthermore, Admin may send error check messages to 'Error' via 'Main'. The calcError() process is performed after obtaining the forecasting value and the actual value of the 'Database'. The error calculation results are sent back to



# The Design of UML-Based Sales Forecasting Application

'Main' to display error calculation results. Finally, the result of forecasting the sales volume of each product can be

displayed after 'Main' sends productID() message to 'Forecast' and receives the result to be sent back to Admin.

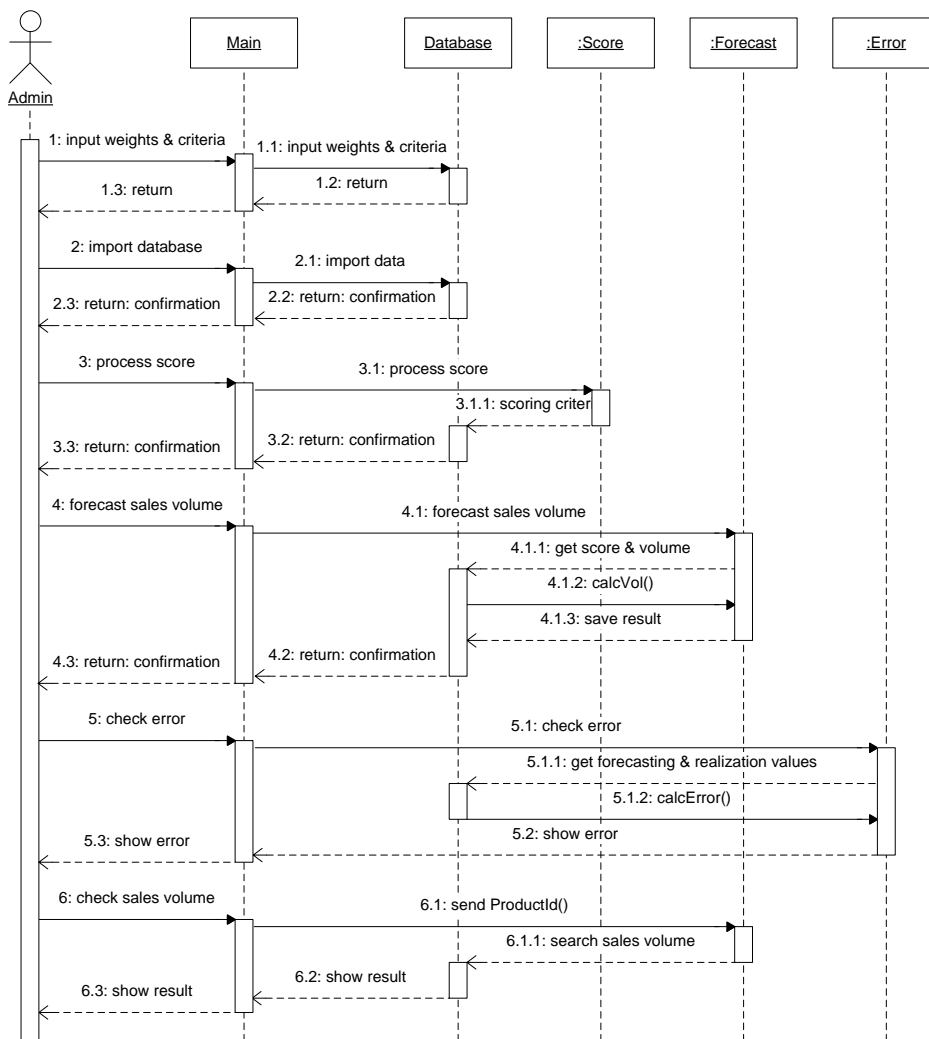


Fig. 3: The sequence diagram for the sales forecasting application

## D. User Interfaces

Based on the use case diagram in Fig. 1, user interfaces are built. The functions of each user interface are described in Table III, and one of the user interfaces to forecast sales via a mobile device is shown in Fig. 4.

No	Function Name	Usefulness
7	Check Sales Volume Forecasting	Forecast the sales volume of each product for next month via web or mobile device.
8	Logout	Exit the app

Table III. List of functions of each interface and their usefulness in the application

No	Function Name	Usefulness
1	Login	User authorization
2	Input Weights & Criteria Value	Input each criterion and its weight values
3	Import Database	Import the company database into the application dataset, and then calculate the sales volume of each product per month.
4	Scoring Criteria	Conduct the criteria scoring process for each product
5	Sales Forecasting	Forecast the sales volume of each product based on the score on the previous UI.
6	Error Calculation	Calculating forecasting errors (MAPE)



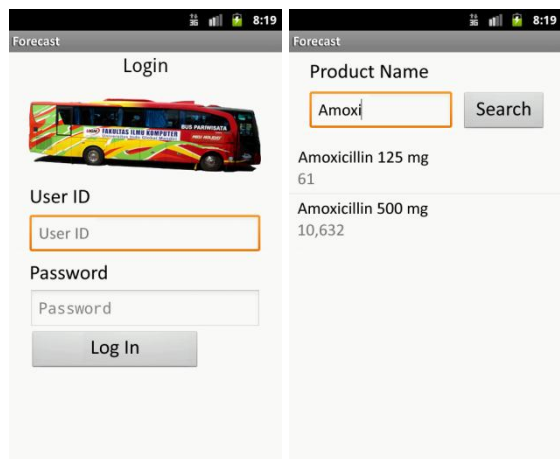


Fig. 4: User Interface on mobile device to login and check sales volume forecasting

Table IV. Recapitulation of the score of each SUS statement from each respondent

Respondent	S1	S2	S3	S4	S4	S5	S6	S7	S8	S9	S10	Total
1	5	3	4	2	5	1	4	2	5	3	5	80.0
2	4	2	4	2	4	1	5	1	4	2	4	82.5
3	5	2	4	3	4	2	4	1	5	2	5	80.0
4	5	2	5	3	5	2	5	1	4	3	5	82.5
5	5	2	5	3	5	2	5	1	4	2	5	85.0
6	5	1	5	1	5	1	5	1	5	1	5	100
7	4	1	5	3	5	2	5	1	5	3	4	85.0
8	5	2	4	2	4	2	5	1	4	2	5	82.5
9	5	1	5	1	5	2	5	1	4	1	5	95.0
10	5	2	4	2	4	2	5	1	4	2	5	82.5
Average SUS Score												85.0

IV. CONCLUSION

In this paper, we have proposed a good approach in designing the Sales Forecasting Application that is expected to be implemented in future application development. The impact is improved application performance by minimizing errors in programming. One example is performance improvement when forecasting the sales of each product needed by the sales manager or stock manager as a decision support system, especially when determining the volume of purchases per product per month.

This research has resulted in the design of UML-based sales forecasting application with acceptability level of application interface is excellent, so that the design of UML-based sales forecasting application can be used by all users.

The results of this research are also expected to contribute to the development of sales forecasting system [1] and facilitate the development of toolkit [3] to forecast sales or inventory volume.

ACKNOWLEDGMENT

This work is supported by Directorate of Research and Community Service (DRPM), Directorate General of Research Reinforcement and Development of the Ministry of Research, Technology and Higher Education of the Republic of Indonesia, Number: SP DIPA- 042.06.1.401516/2018

REFERENCES

1. R. Gustriansyah, D. I. Sensuse, and A. Ramadhan, "A sales prediction model adopted the recency-frequency-monetary concept," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 6, no. 3, 2017, pp. 711–720.

E. Usability Test

Usability test is a test performed to determine the ease of use of the application. The usability level was measured using the System Usability Scale (SUS) evaluation tool.

The usability test of this application design involves 10 respondents consisting of programming practitioners and sales admin. The usability test results can be seen in Table IV.

Based on the results of data processing in Table IV, the average SUS score was 85.0. If the score is converted by standard [13], then the acceptability range of the application design can be classified to the 'Accepted' category or the application design has an 'Excellent' usability level.

2. R. S. Pressman, *Software engineering: a practitioner's approach*, 7th ed. McGraw-Hill Companies, Inc., 2010.

3. A. Sanmorino, R. Gustriansyah, Terttiavini, and Isabella, "The Toolkit of Success Rate Calculation of Broiler Harvest," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 15, no. 4, 2017.

4. A. Sanmorino and R. Gustriansyah, "An Alternative Solution to Handle DDoS Attacks," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 3, 2018, pp. 657–667.

5. A. A. Sasmanto, Zulkifli, and R. Gustriansyah, "Kendali Peralatan Listrik dengan SMS Menggunakan Arduino dan GPRS Shield," *Inform. Glob.*, vol. 6, no. 1, 2015, pp. 33–37.

6. R. Gustriansyah, D. I. Sensuse, and A. Ramadhan, "Decision support system for inventory management in pharmacy using fuzzy analytic hierarchy process and sequential pattern analysis approach," in *2015 3rd International Conference on New Media (CONMEDIA)*, 2015, pp. 1–6.

7. H. Sunardi, N. Suhandi, and R. Gustriansyah, "Pembangunan m-Bekam Berbasis Sistem Pakar," *Inform. Glob.*, vol. 5, no. 1, 2014, pp. 40–45.

8. J. Rezaei, "Best-Worst Multi-Criteria Decision-Making Method," *Omega - Int. J. Manag. Sci.*, vol. 53, 2015, pp. 49–57.

9. E. Gharavi and M. J. Tarokh, "Predicting customers' future demand using data mining analysis: A case study of wireless communication customer," in *The 5th Conference on Information and Knowledge Technology*, 2013, pp. 338–343.

10. J. Brooke, "SUS: A Retrospective," *J. Usability Stud.*, vol. 8, no. 2, 2013, pp. 29–40.

11. J. Brooke, *SUS - A quick and dirty usability scale*. London: Taylor and Francis, 1996.

12. J. Sauro, "Measuring Usability with the System Usability Scale (SUS)," *MeasuringU*, 2011. [Online]. Available: <https://measuringu.com/sus/>.

13. A. Bangor, T. Staff, P. Kortum, J. Miller, and T. Staff, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale," *J. Usability Stud.*, vol. 4, no. 3, 2009, pp. 114–123.