Hybrid UML-COSMIC Approach to Functional Sizing and Cost Estimation for a Marketing Goods Monitoring

Nurul Alifah Puspita Sari [1*], Dwi Nur Kholidah [2], Hotma Uli Arta Sitorus [3], Rizal Akbar Kurniawan [4], Muhammad Syauqi Kahfi Sin [5], Riska Dhenabayu [6], Renny Sari Dewi [7]

[1, 2, 3, 4, 5, 6, 7] **Digital Business, Universitas Negeri Surabaya, Surabaya, Indonesia.**

***Email:** nurul.23460@mhs.unesa.ac.id

Abstract. This study proposes a hybrid approach that integrates UML-based modeling and the COSMIC functional size measurement method to design and evaluate a web-based Marketing Goods Monitoring System (MGMS) tailored for small and medium-sized enterprises (SMEs). The system is designed using use case and sequence diagrams to model 24 functional requirements supporting CRUD operations across five distinct user roles: Admin, Warehouse Staff, Sales Staff, Marketing Manager, and Finance Admin. Additionally, five non-functional requirements—security, performance, accessibility, scalability, and usability—are incorporated to ensure deployment readiness. The system's complexity is assessed using the COSMIC method, resulting in a functional size of 82 COSMIC Function Points (CFP), including 14 Entry, 19 Write, 14 Read, 10 Exit operations, and 25 validation processes. This size places the MGMS in the high-complexity category, indicating a system with multiple interacting components and rich functionality. Furthermore, a cost estimation is conducted using a productivity rate of 7 CFP per day and a labor rate of IDR 135,800 per hour, yielding a projected development effort of 82 working hours and a total cost of IDR 11,135,600 (≈ USD 750). These results demonstrate the system's technical feasibility, functional completeness, and economic viability, offering a scalable and affordable solution for SMEs transitioning from manual to automated operations.

Keywords: Cost Estimation, COSMIC Function Point, Marketing Goods Monitoring System, UML Modeling

Introduction

In recent years, the monitoring of marketing goods has become a critical function for businesses seeking to improve operational efficiency and customer satisfaction (Kotler & Keller, 2016). The increasing complexity of supply chains and the dynamic nature of consumer demand have driven organizations to adopt information systems that provide real-time monitoring and management capabilities. Several studies, such as those by Azevedo et al. (2019) and Zhao et al. (2020), have demonstrated the benefits of implementing technological solutions in inventory and goods management. However, many organizations, especially small and medium enterprises (SMEs), still face difficulties in developing customized monitoring systems that align precisely with their operational processes (OECD, 2021).

A review of the existing literature reveals that prior research has largely concentrated on macro-level solutions involving sophisticated technologies like RFID (Ngai et al., 2011), blockchain (Kamble et al., 2020), and big data analytics (Gunasekaran et al., 2017). While these innovations undoubtedly enhance supply chain visibility, they often require significant investments and technological infrastructure that may not be feasible for many businesses. Moreover, most existing studies focus on the implementation phase or the technological impacts, with limited attention paid to the early system design phase where requirements are gathered and modeled using structured methodologies such as Unified Modeling Language (UML) (Booch et al., 2005).

This indicates a research gap in developing a practical and systematic design approach for marketing goods monitoring systems. In addition to the research gap, a phenomenon gap can be observed in the industry. Many businesses, particularly in developing economies, still rely on semi-manual processes or fragmented digital tools for goods monitoring (World Bank, 2021). This results in inconsistencies, data inaccuracy, and delayed responses to stock changes or market demands. Although digital transformation is a prevailing trend, the lack of clear system design frameworks prevents organizations from efficiently transitioning from manual to fully automated monitoring systems (Ross et al., 2019). This phenomenon underscores the pressing need for model-driven system development approaches that are accessible, cost-effective, and easily comprehensible to both technical and non-technical stakeholders. To address these gaps, this study incorporates the COSMIC (Common Software Measurement International Consortium) method to measure the functional size of the proposed information system. By quantifying software requirements in terms of functional user requirements, COSMIC provides a standardized basis for comparing functionality and complexity across systems. The functional size measurement further supports early-stage project planning, including the estimation of development efforts and costs. A preliminary cost estimation is also conducted using parametric modeling based on the COSMIC functional size, allowing stakeholders to make informed decisions about resource allocation and budgeting.

Literature Review

COSMIC Functional Size Measurement (COSMIC-FSM) is a new generation method for measuring the functional size of software developed by the Common Software Measurement International Consortium (Abran et al., 2010). This method was born to overcome the limitations of lines of code (SLOC) and Function Point Analysis (FPA)-based measurements developed by Albrecht, which although effective for Management Information Systems (MIS) applications, have proven less suitable for non-MIS software such as embedded systems and real-time systems. COSMIC-FSM measures software size based on the amount of data movement (Entry, Exit, Read, Write) that occurs in response to user requests, focusing on the logical function of the software from the user's perspective, regardless of programming language or specific technology. Use case diagrams and sequence diagrams are two essential tools in system design. Use case diagrams are employed to model the system's functionalities, whereas sequence diagrams are used to model the sequence of operations performed by the system (Booch et al.2005). In information system development, these diagrams are often used in tandem to ensure that both functional and non-functional requirements are comprehensively addressed.

$$\text{CFP} = E + W + R + X \quad (1)$$

Literature studies in Table 1 show that MGMS must have several characteristics, namely:

Table 1: Summary of Previous Research Results.

Author, Year	Summary of Results	Method
Ardhiawan Anshori, Eman Setiawan (2023)	This study designs a web-based Sales Force Automation (SFA) information system at PT. Garam. This system aims to overcome the problem of manual recording in sales and customer visit schedules, as well as monitoring sales more efficiently.	Waterfall, UML
Mustapha Maidawa, Hussaini Mamman (2015)	Design a wholesale inventory management system using UML to improve the efficiency of the wholesale business. This study emphasizes the benefits of UML class diagrams in documenting the system.	UML, Class Diagram.
Rendra Gustriansyah, Nazori Suhandi, Fery Antony (2019)	This study designs UML-based sales forecasting software using the Best Worst Method (BWM) and the RFM (Recency, Frequency, Monetary) concept.	UML, BWM, RFM

Author, Year	Summary of Results	Method
Hari Krishna Mahat (2013)	This study aims to design a logical data model for a sales and inventory management system. This model is created to help manage customer orders and product distribution of the company.	Logical Data Model, ER Diagram, Requirement Analysis
Egwali A.O.,Ikhalo E.I.(2024)	This study develops a web-based stock management system to improve the efficiency of inventory tracking, demand forecasting, and storage optimization. This system utilizes the latest technologies such as blockchain and mobile applications to improve supply chain visibility.	UML, PHP, Client-Server Model
Loh,K.Y., Haslina Mohd, Hazlyna Harun, Barraood,S.O., Muhammad Zulfazli Bin Bakri (2024)	This study designs and develops a product sales management system for small and medium-sized manufacturing companies using the Rapid Application Development (RAD) approach. This system improves the management of product, customer, and order information.	UML, RAD
Ade Hendini (2016)	This study discusses UML modeling in the sales and stock monitoring information system at Distro Zhezha Pontianak. This system helps in monitoring stock and sales transactions to improve business efficiency.	UML, Use Case Diagram, Activity Diagram
Hanafi Kambivi, Eko Junirianto, Nisa Rizqiyah Fadhliyah (2023)	This study develops an inventory management application using the Laravel framework with a Point of Sales (POS) approach to help SMEs in managing purchasing and stock data.	UML, Flowchart, Laravel
Narayon Chandra Sarker (2018)	An online inventory and ordering management system for pharmacies designed to automate the entire process of managing drug stock and ordering goods to improve efficiency and reduce product loss.	UML, SaaS, PHP, Bootstrap
Y.Wibawanti,Z. Niswati, D. R. Wardhani, F.A. Mustika,A. Primawati,M. Firdaus,I.B. Rangka (2018)	A sales and purchasing information system for UML-based cooperatives that allows automatic recording of member transactions to improve operational efficiency.	UML, JAVA

Methods

Here is the illustration of the research workflow that was conducted (Fig. 1).

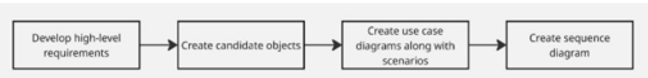

Figure 1. Explanation of research workflow

Fig. 1 illustrates the process flow in designing an information system using use case diagrams and sequence diagrams. The first stage involves formulating high-level requirements, which means defining the core needs of the system in a general manner. At this stage, the main features required for the smart village information management system are identified, including both functional and non-functional requirements. Once the system requirements are established, the next step is to identify candidate objects, which entails recognizing the entities or objects to be utilized within the system. These objects may include system users (actors), primary data to be managed, as well as elements that interact within the system. The selection of candidate objects is based on the system needs formulated in the previous stage.

The third stage involves developing the use case diagram along with its scenarios. A use case diagram illustrates how actors interact with the system and the available features. Each use case is accompanied by a scenario that describes the interaction flow in greater detail, including the steps users take to perform specific functions.

The fourth step is to construct the sequence diagram, which depicts the order of interactions between objects in the system over time. A sequence diagram shows how messages are exchanged among objects to accomplish a particular process or scenario defined in the use case diagram. By using sequence diagrams, developers can gain a clear understanding of how data flows and communication occurs dynamically within the system. This entire process aims to ensure that the information system being developed has a clear structure, is easy to comprehend, and can be effectively implemented in alignment with user requirements.

This study, which involves the design of MGMS through the development of use case and sequence diagrams, alongside functional size measurement and software cost estimation using the COSMIC method, is classified as quantitative research. The rationale for this classification lies in the research's reliance on structured data analysis, focusing on objectively measurable numerical data to quantify software functionality and project cost parameters. Data were collected based on the detailed functional and non-functional requirements. Use case diagrams and sequence diagrams were developed using appropriate software tools.

Functional Requirement Information
Based on the identified 24 functional requirements, the system supports full CRUD (Create, Read, Update, Delete). The fifth stage is functional size measurement using the COSMIC method (Common Software Measurement International Consortium). COSMIC measures software size based on the functional user requirements defined in the previous steps. It quantifies the size by identifying data movements—Entry, Exit, Read, and Write—that occur in each functional process described in the use case scenarios. This step helps in obtaining an objective and standardized measurement of the system's functionality, independent of technology or development methodology.

The sixth stage is software cost estimation using the COSMIC-based approach. After determining the functional size, it is used to estimate the effort, time, and cost required to develop the system. This involves applying productivity rates (e.g., person-hours per COSMIC Function Point) to translate size into resource estimates. By leveraging COSMIC-based estimation, project managers can plan and allocate resources more accurately, mitigate risks, and ensure the project stays within budget and schedule, Delete) operations, divided across various user roles: Admin, Warehouse Staff, Sales Staff, Marketing Manager, and Finance Admin. The use case model was created to describe how users interact with the system. Each use case represents a specific system functionality, such as adding items, updating supplier data, or viewing transaction history. Below is a summary of the functional coverage:

The Create function comprises six sequences: adding items, categories, stock, suppliers, promotions, and marketing reports. These sequences facilitate the input of new data into the system. The Read function also includes six sequences, allowing users to view product lists, item details, marketing reports, supplier data, stock levels, and transactions, which support data monitoring and retrieval. The Update function involves six sequences for modifying existing data, such as product information, categories, supplier details, stock quantities, promotions, and transactions, ensuring data accuracy and relevancy. Lastly, the Delete function consists of five sequences focused on removing obsolete or irrelevant data, including items, categories, suppliers, marketing data, and expired promotions, thereby maintaining data integrity and system efficiency. The complete mapping between functional requirements and use cases is presented in Table 2 of this paper.

Table 2: Functional Requirements and Use Cases Mapping

FR Code	Functional Requirement Name	UC Code	Use Case Name
FR-001	Create a list of goods	UC-001	Adding items
FR-002	Create item categories	UC-002	Add item categories
FR-003	Create stock items	UC-003	Add stock items
FR-004	Create a marketing report	UC-004	Adding marketing reports
FR-005	Create supplier data	UC-005	Add supplier data
FR-006	Creating promos/discounts	UC-006	Add promo/discount
FR-007	View Item List	UC-007	Display Item List
FR-008	View Item Details	UC-008	Display Item Details
FR-009	View stock items	UC-009	Display Stock Items
FR-010	View Marketing Report	UC-010	Display Marketing Report
FR-011	View Supplier Data	UC-011	Display Supplier Data
FR-012	View Transaction History	UC-012	Display Transaction History
FR-013	Update Item List	UC-013	Update Item List
FR-014	Update Item Categories	UC-014	Update Item Categories
FR-015	Stock Item Update	UC-015	Updating Stock Items
FR-016	Update Supplier Data	UC-016	Update Supplier Data
FR-017	Promo/Discount Update	UC-017	Update Promo/Discount
FR-018	Transaction Update	UC-018	Update Transactions
FR-019	Delete Items	UC-019	Deleting Items
FR-020	Delete Item Category	UC-020	Delete Item Category
FR-021	Delete stock items	UC-021	Delete stock items
FR-022	Delete Supplier	UC-022	Delete Supplier
FR-023	Clear Marketing Transactions	UC-023	Delete Marketing Transaction
FR-024	Delete Promo/Discount	UC-024	Delete Promo/Discount

Use Case Diagram

The system's overall behavior is modeled using a Use Case Diagram, which provides a high-level visual representation of the system functionalities and actors involved. The primary actors include: Admin, Warehouse Clerk, Marketing Staff, Finance Admin, Sales Staff. The use case diagram ensures that all user needs are covered and helps stakeholders verify the completeness of the system functions.

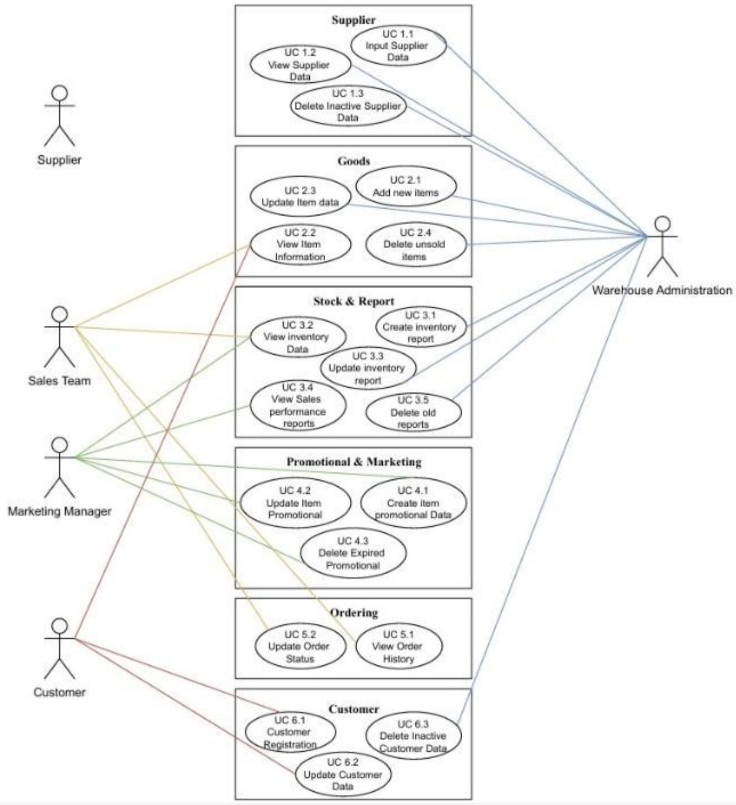

Figure 2. Use case diagram of MGMS

Sequence Diagrams

To provide detailed behavioral modeling, 24 sequence diagrams were developed, each corresponding to the system's core use cases. These diagrams illustrate the time-based interaction between objects in the system, including message flows, processes, and responses for each user activity. Examples of developed sequence diagrams include:

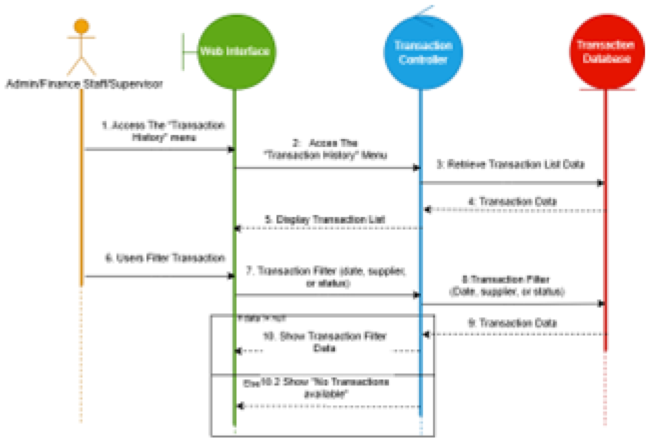

Figure 3. A sequence diagram displays transaction history

In this system, the process begins when the Admin, Finance Staff, or Supervisor accesses the "Transaction History" menu through the web interface. This request is forwarded by the Web Interface to the Transaction Controller, which handles the data flow logic. The Transaction Controller then sends a request to the Transaction Database to retrieve the stored transaction history data. Once the data is successfully retrieved, the Transaction Controller returns it to the Web Interface to be displayed to the user as a transaction list.

After the transaction list is displayed, the user has the option to filter the transactions based on specific criteria such as date, supplier, or transaction status. This filter request is sent again to the Transaction Controller, which then sends a filtered data request to the Transaction Database. The database responds with transaction data that matches the applied filter. If matching data is found, the system displays the filtered results to the user. However, if no data matches the filter criteria, the system displays a message saying "No Transactions Available" to inform the user that there is no data to show.

Result and Discussion

Result

This study has resulted in the design of a web-based Marketing Goods Monitoring Information System, aimed at enhancing the efficiency and accuracy of managing marketing products, stock data, transactions, and supplier information. The results of this system design are detailed below.

Functional Size Estimation

To assess the functional complexity of the system, a use case-based evaluation was conducted using the Complexity Function Point (CFP) method. The table below outlines each use case, including its complexity level, functional range, and corresponding average CFP value.

Table 3: Functional Size Estimation

Sequence Diagram	E (CFP)	W (CFP)	R (CFP)	X (CFP)	V+EH (CFP)
Add category	1	1	0	0	2
Add supplier	1	1	0	0	2
Displays stock items	1	1	2	2	0
Display marketing report	1	2	2	1	3
Display supplier data	2	2	3	3	1
Display transaction history	1	1	2	2	1
Updating the list of items	2	2	2	1	1
Update stock item	1	1	1	1	3
Update Discount	1	1	0	0	3
Update Transactions	1	1	0	0	3
Delete Product	1	3	1	0	3
Delete Product Category	1	3	1	0	3
Subtotal	14	19	14	10	25
Total	82				

Alignment with High-Level Requirements

The system design aligns with the 29 High-Level Requirements (HLR) identified in the SRS, such as: Role-based access control, Real-time stock monitoring, Automated sales reporting, Integration with inventory systems, Multidevice (mobile-friendly) support, User-friendly interface, and security features.

Non-Functional Aspects

The system design also addresses five key non-functional requirements:

Table 4: Non-Functional Requirements of MGMS

ID	Parameters	Needs
NFR - 01	Data Security	The system must have user authentication and data encryption.
NFR - 02	Performance	The system must be able to process data in less than 3 seconds for each use.
NFR - 03	Accessibility	The system must be accessible for various devices with a responsive display.
NFR - 04	Scalability	The system must be able to handle the increasing number of users and data without degrading performance.
NFR - 05	User Friendly	The system should have an interface that is easy to use by users without the need for specialized training.

Security: Data encryption and user authentication, Performance: Each request is processed in under 3 seconds, Accessibility: Responsive design for mobile and desktop access, Scalability: Able to handle growth in user and data volume, Usability: Interface designed for ease of use. Provided by the drop-down menu to differentiate the head from the text. These non-functional goals were incorporated into the architectural and interaction models, ensuring the designed system is not only functionally complete but also practical for real-world deployment.

Cost Estimate

One of the fundamental assumptions in cost estimation using the COSMIC approach is the establishment of a productivity rate, also known as the Project Delivery Rate (PDR)—that is, the amount of functional output a developer can deliver within a specific unit of time. According to benchmark data from the International Software Benchmarking Standards Group (ISBSG, 2025), particularly for business application system development projects, the average productivity rate is estimated to be in the range of 7 to 10 COSMIC Function Points (CFP) per person per day. This benchmark is derived from an extensive analysis of thousands of software projects measured using the COSMIC method and serves as a reliable baseline for organizations and practitioners to plan resources and estimate costs objectively. For instance, in development environments utilizing Java and targeting multi-platform applications, the median PDR is approximately 20 hours per CFP, with adjustment factors applied for medium-scale applications—resulting in an estimated productivity rate of approximately 7 to 10 CFP per developer per day (Czarnacka Chrobot, 2013). With this assumption, project managers are equipped to convert the functional size of the system (expressed in CFP) into more accurate and realistic estimations of development time and cost, aligned with current industry best practices.
The next assumptions in cost estimation using the COSMIC are based on a recent study published in BMC Public Health (2024) reported that the average working hours per week in Indonesia amount to approximately 42 hours. Assuming a six-day workweek, this translates to an average of 7 hours per day, which aligns with the standard full-time employment model commonly adopted in the country (Siregar, 2024).

Another critical underlying assumption in cost estimation using the COSMIC approach is the determination of the average salary of business application system developers as an input variable to convert development time estimates into cost estimates. According to data from Glassdoor (2025), software developers in Jakarta earn an average total salary of IDR 22,800,000 per month. Assuming an average of 7 working hours per day and a COSMIC productivity rate of 7 CFP per day, the average hourly wage of a system developer is calculated to be approximately IDR 135,800. This hourly rate can then be used as the basis for estimating the cost per CFP, which also amounts to IDR 135,800 per CFP. Such a calculation enables project managers to estimate the functional unit cost more realistically and in alignment with local industry practices.

This cost estimation analyzes the development effort for MGMS based on functional tasks represented by sequence diagrams. Each function is mapped to a standard work hour and cost, offering a practical overview of the required budget using a rate of IDR 135,800 per hour. This estimation helps stakeholders make informed decisions in the early planning stages.

Table 5: Estimated Cost of MGMS

Sequence Diagram	Effort (Hours)	Cost (IDR)
Add category	4	543,200
Add supplier	4	543,200
Displays stock items	6	814,800
Display marketing report	9	1,222,200
Display supplier data	11	1,493,800
Display transaction history	7	950,600
Updating the list of items	8	1,086,400
Update stock item categories	7	950,600
Update Discount	5	679,000
Updating Transactions	5	679,000
Delete Product	8	1,086,400
Delete Product Category	8	1,086,400
Total	82	11.135.600

The cost estimation results indicate a total development effort of 82 working hours with a projected cost of IDR 11.135.600, assuming a rate of IDR 135,800 per hour. These figures highlight that the system can be developed within a reasonable budget, making it accessible and feasible for SMEs. The estimation method also ensures transparency and traceability, enabling stakeholders to plan their financial and human resources effectively. Overall, the system design demonstrates not only technical and functional feasibility but also economic viability for organizations aiming to improve their marketing goods monitoring operations without investing in high-end or complex technologies.

Discussion

This research presents a hybrid approach that combines UML-based modeling and the COSMIC functional size measurement method to design and evaluate a web-based Marketing Goods Monitoring System (MGMS), with a particular emphasis on the operational needs of small and medium-sized enterprises (SMEs). This methodological integration addresses the common challenges faced by SMEs in transitioning from manual to automated monitoring systems by offering a structured, measurable, and replicable system development process.

The functional design comprises 24 functional requirements, which have been systematically modeled using use case diagrams and sequence diagrams. These requirements support full CRUD operations across five user roles: Admin, Warehouse Staff, Sales Staff, Marketing Manager, and Finance Admin. In addition, five non-functional requirements—including security, performance, accessibility, scalability, and usability—were also defined to ensure that the system is secure, responsive, and scalable for real-world implementation. To objectively assess the system's functional complexity, the COSMIC method was applied, resulting in a total of 82 COSMIC Function Points (CFP). These CFPs are composed of 14 Entry, 19 Write, 14 Read, and 10 Exit data movement operations, as well as 25 validation and error-handling procedures. According to the COSMIC Practitioners' Guide (Abran & Symons, 2020), small, medium, and large functional processes typically correspond to around 5, 10, and 15 CFP, respectively. Based on these benchmarks, the total size of 82 CFP indicates that the system is composed of several medium-to-large functional processes, involving a relatively high level of interaction between components and data-handling operations. As such, the system can be classified as having a high level of complexity, which has direct implications for planning, risk management, and development effort estimation.

The cost estimation was carried out by assuming a productivity rate of 7 CFP per developer per day and an hourly wage of IDR 135,800. This resulted in an estimated development effort of 82 working hours, equivalent to a total cost of IDR 11,135,600 (approximately USD 750). When compared with regional benchmarks in the Asia-Pacific region, where software project costs typically range from USD 50,000 to 150,000 for small- to mid-scale implementations (Wheelhouse Software, 2025), the MGMS developed in this study clearly falls within the low-cost category. This reinforces the economic feasibility of the proposed system, making it a suitable option for SMEs and organizations with limited technological budgets.

In summary, the integration of UML modeling and COSMIC functional sizing has proven to be effective in delivering a system that is functionally complete, technically sound, and economically viable. The resulting MGMS design provides not only a practical solution for marketing goods monitoring but also contributes a generalizable development framework that can be replicated for other types of information systems across various sectors. The findings highlight how combining model-driven engineering with standardized functional sizing metrics can improve development accuracy, transparency, and cost-effectiveness, especially for small organizations operating in resource-constrained environments.

Conclusion

This study introduces and validates a hybrid approach that combines Unified Modeling Language (UML) for system design and the COSMIC method for functional size measurement and cost estimation in the development of a Marketing Goods Monitoring System (MGMS). The approach addresses key challenges in software development for small and medium-sized enterprises (SMEs), particularly those related to the early design phase and budget constraints in transitioning from manual to digital systems.

The research successfully demonstrates the applicability of UML modeling in capturing both functional and non-functional requirements through 24 use cases and sequence diagrams. These diagrams not only formalize the system's behavior but also enable a structured analysis of user roles and their interactions with system functions. The inclusion of five non-functional requirements—security, performance, accessibility, scalability, and usability—ensures the practical relevance of the system for real-world deployment.

By applying the COSMIC functional size measurement, the system is quantified at 82 COSMIC Function Points (CFP), including detailed entries, writes, reads, exits, and validation operations. According to industry benchmarks, this functional size reflects a high-complexity software product, indicating that the MGMS features rich business logic and multiple interdependent components. This classification has significant implications for development planning and resource allocation.

Furthermore, the cost estimation, based on a productivity rate of 7 CFP per developer per day and an hourly rate of IDR 135,800, yielded a total development effort of 82 working hours, equating to an estimated budget of IDR 11,135,600 (approximately USD 750). When benchmarked against typical regional software development projects, which often exceed USD 50,000 for similar systems, the MGMS falls into the low-cost category. This affirms the economic feasibility and accessibility of the proposed system for SMEs, particularly in developing economies.

Overall, this study contributes a replicable and scalable development framework that integrates model-driven engineering with standardized size-based estimation techniques. It enhances the reliability, traceability, and transparency of early-stage system development processes. The hybrid UML-COSMIC methodology presented here not only bridges the gap between abstract system modeling and practical project estimation but also offers a valuable reference for future research and system design across diverse industrial domains.

Future studies are encouraged to extend this framework with empirical validation during the implementation phase, explore the integration of automated modeling tools, or apply the methodology to different types of information systems, including mobile-first or cloud-native applications. This would further strengthen the generalizability and adaptability of the proposed approach in various technological and organizational contexts.

References

Abran, A., & Symons, C. (2020). Early software sizing with COSMIC: Practitioners guide (Version 1.0) [PDF]. COSMIC Sizing.org. https://doi.org/10.13140/RG.2.1.4195.0567

Abran, A., Araki, M., Desharnais, J.-M., Dumke, R., Fagg, P., Meli, R., Morris, P., Oligny, S., Onvlee, J., Nevalainen, R., Rule, G., St Pierre, D., & Symons, C. (2000). Functional size measurement methods – COSMIC-FFP: Design and field trials. Proceedings of FESMA-AEMES Software Measurement Conference.

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). The Unified Modeling Language User Guide (2nd ed.). Addison-Wesley.

Cosmic Sizing. (2021). Knowledge base archive. https://cosmic-sizing.org/publications/

Czarnacka Chrobot, B. (2013). Analysis of software development and enhancement projects work effort per unit based on the COSMIC method with regard to technological factors: Case study. Journal of Software Engineering and Applications, 6(11), 597–609. https://doi.org/10.4236/jsea.2013.611072

Dennis, A., Wixom, B. H., & Tegarden, D. (2015). Systems analysis and design: An object-oriented approach with UML (5th ed.). Wiley.

Fowler, M. (2003). UML distilled: A brief guide to the standard object modeling language (3rd ed.). Addison-Wesley.

Kotler, P., & Keller, K. L. (2016). Marketing management (15th ed.). Pearson.

Object Management Group. (2017). Unified Modeling Language (UML) specification, version 2.5.1. https://www.omg.org/spec/UML/

OECD. (2021). The digital transformation of SMEs. https://doi.org/10.1787/bdb9256a-en

Pressman, R. S. (2015). Software engineering: A practitioner's approach (8th ed.). McGraw-Hill.

Ross, J. W., Beath, C. M., & Mocker, M. (2019). Designed for digital: How to architect your business for sustained success. MIT Press. https://doi.org/10.7551/mitpress/12188.001.0001

Siau, K., & Rossi, M. (2011). Evaluation techniques for systems analysis and design modeling methods: A review and comparative analysis. Information Systems Journal, 21(3), 261–287. https://doi.org/10.1111/j.1365-2575.2010.00363.x

Siregar, A. Y. M. (2024). Work hours and the risk of hypertension: The case of Indonesia. BMC Public Health, 24, Article 2480. https://doi.org/10.1186/s12889-024-20003-z

Sommerville, I. (2015). Software engineering (10th ed.). Pearson.

Wasik, Z., Nugroho, K. C., & Suhermawan, D. (2023). The performance of a national logistics company measured by dynamic capabilities and innovation performance: Literature review. Journal of Management and Social Sciences, 1(1).

Wheelhouse Software. (2025). Software development costs: Complete pricing guide 2025. https://wheelhouse.software/blog/software-development-costs-complete-pricing-guide-2025

World Bank. (2021). World development report 2021: Data for better lives. https://doi.org/10.1596/978-1-4648-1600-0