Indian Journal of Software Engineering and Project Management (IJSEPM)
ISSN: 2582-8339 (Online), Volume-6 Issue-1, January 2026

Engineering a Topic Modeling System: Architecture, Process, and UML Modeling

Fitian Shafeeq Ajeel, Hasan Aqeel Abbood, Raad Mahmood Mohammed, Mohammed Ali Mohammed

Abstract: ***Software engineering is a discipline that uses Unified Modelling Language diagrams, which are among the most widely accepted standards for visualising object-oriented design models. Using UML diagrams, system and application requirements are specified by providing understandable models of the objects involved. Thus, the goal of this study is to describe how software engineering is applied to our topic modelling system. This study describes the software engineering process from story to system testing, and then supports it with a UML diagram for clarity. This paper demonstrated empirically how software engineering principles are integrated into a system using Latent Dirichlet Allocation (LDA). Several engineering operations were performed on data from a group of websites, yielding measurable, verifiable results that support the system's reliability.***

Keywords: ***Software Engineering, Topic Model, Unified Modelling Language (UML) Diagrams, Latent Dirichlet Allocation (LDA) Algorithm.***

Nomenclature:
LSI: Latent Semantic Indexing
LDA: Latent Dirichlet Allocation
UML: Unified Modelling Language
API: Application Programming Interface

I. INTRODUCTION

In recent years, companies have been able to integrate software into their systems as an essential part, thereby improving their services and expanding into diverse market segments. One of the most critical areas of software engineering is improving the speed and flexibility of administrative systems [1]. Focusing on different aspects of systems makes improving them a difficult task [2].

Manuscript received on 08 November 2025 | First Revised Manuscript received on 15 November 2025 | Second Revised Manuscript received on 17 December 2025 | Manuscript Accepted on 15 January 2026 | Manuscript published on 30 January 2026.

*Correspondence Author(s)
Fitian Shafeeq Ajeel, Department of College of Business Informatics, University of Information Technology and Communications (UOITC), Baghdad, Iraq. Email ID: Fitian.shafeeq@uoitc.edu.iq. ORCID ID: 0000-0002-0680-3619
Hasan Aqeel Abbood, Department of College of Business Informatics, University of Information Technology and Communications (UOITC), Baghdad, Iraq. Email ID: 992hassan@gmail.com ORCID ID: 0009-0005-4099-7989
Raad Mahmood Mohammed, Department of College of Business Informatics, University of Information Technology and Communications (UOITC), Baghdad, Iraq. Email ID: raad.mahmoud.alum@uoitc.edu.iq. ORCID ID: 0009-0002-6986-0386
Dr. Mohammed Ali Mohammed*, Department of College of Business Informatics, University of Information Technology and Communications (UOITC), Baghdad, Iraq. Email ID: mohammed.ali@uoitc.edu.iq ORCID ID: 0009-0009-7706-1905

Software engineering has improved and enabled the development of reliable, approved systems from inception to repair, with lower development costs [3].

It has become common to use graphical models, such as the Unified Modelling Language (UML), to improve systems across various fields, including software engineering. Graphical models provide a general and precise understanding of a system's structure and its detailed operation. Models specify the requirements for understanding how the system works. They also show the system's boundaries using graphical models [4]. Many research papers have been published in this field, demonstrating the importance of graphical models in improving software engineering systems. In our research, graphical models were used to identify commonly used diagrams, their objectives, and the most common areas of implementation [2].

Topic modelling is a text mining and concept extraction method that identifies topics (i.e., coherent word clusters) from large corpora of textual documents to discover hidden semantic structures [5]. An advantage of topic modelling over other techniques is that it helps analyse long texts [6] [5], creates clusters as “topics” (rather than individual words), and is unsupervised [5].

Topic modelling is widely popular in software engineering research [7][8]. For example, [7] found that topic modelling enhances source code comprehension, feature localisation, and fault prediction. Furthermore, [8] found that several studies used topic modelling on text data to extract data from repositories, such as source code comprehension and event logs, to suggest code structure improvements, [9] or to identify faults [10].

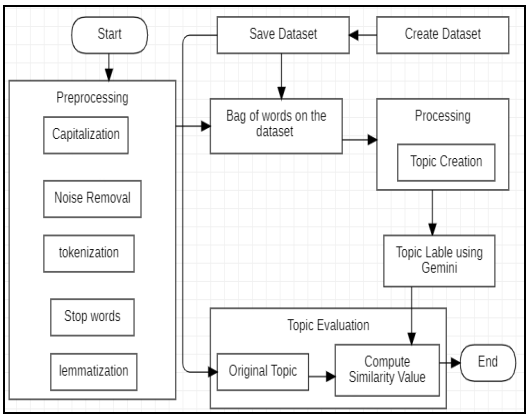
Topic modelling demonstrates capabilities such as [11] and latent Dirichlet allocation (LDA) [12], which identify topics in a corpus of text based on word frequencies and co-occurrence [13]. However, [14] cautions against systematic biases in the presentation of latent Dirichlet allocation models, which degrade topic accuracy [13]. It also produces low-level issues when traditional topic models are applied to short text documents.

The paper's goal is to describe how software engineering has been applied in topic modelling systems, covering the story, requirements, design, and implementation steps. Also, the paper is supported by UML diagrams, including a class diagram, a state machine diagram, and a component diagram.

The paper's structure is as follows: in Section 2, we describe our topic model system. In Section 3, we show in detail the software engineering process for our system. Section 4 presents the UNL diagrams for our topic model system. Section 5: Conclusion.

II. SYSTEM OVERVIEW

In [15], a software system for mining web content was developed and implemented using the proposed topic model [16]. A topic title is derived from the content of a web page. Our system consists of the following stages: word collection, identification of the most frequent words, and application of the LDA algorithm to obtain the document's weighted word distribution. To give the content a title, the Chat Bot Gemini [17] system was employed to derive a title from the extracted words using the LDA algorithm. Figure 1 shows the flowchart, and Algorithm 1 illustrates the system's algorithmic stages [15].



[Fig.1: Proposed System [15]]

Algorithm 1: Proposed Methodology [15]	
Input:	Document
Output:	Topic_Name
Steps:	
1.	Preprocessing Step (Tokenisation, Stop words, Lemmatisation, Data Cleaning, etc.).
2.	Compute the Bag of words.
3.	Processing Topic Creation using the LDA algorithm.
4.	Topic Labelling using Gemini with the top 5 words for a document.
5.	Compute the Similarity value between (generate topic, original topic).

III. SOFTWARE ENGINEERING PROCESS

This section describes the sequence of software engineering steps for our topic model system, as outlined in Section 2. At each step, we will focus on the details of how the system works and what is needed.

A. User Story

Mohammed is a second-year PhD student in computer science. He is a hardworking student who searches the Internet for academic content because of the abundance of resources, and he often finds himself confronted with a vast array of information, including articles, research papers, journals, magazines, forums, tutorials, and reviews. There may be a single web page or research paper that touches on many different topics, with no clear organization and a haphazard structure. Mohammed often spends more time looking at scientific material than on actual learning.

One day, while browsing the internet, he discovered an innovative learning environment newly developed specifically for science students. What sets this learning environment apart is its use of automated semantic analysis of page content. When moving between different sections - be it a research paper, a blog post, or a description of a dataset - a thematic representation or summarization of the content on the page is automatically generated by advanced linguistic algorithms.

It's worth noting that upon closer reading, a critical point emerges: the basic structure of data structures, such as links and algorithms. These are the main topics, followed by an explanatory summary and links to related studies.

We will cover the key areas that will benefit Mohammed:

- Ability to identify main headings and their content
- Smooth and comprehensive navigation between main and subheadings
- Interconnectedness between main topics reveals relevant headings and content
- By reducing time and effort, this will allow us to study and delve deeper.

B. Scenario

Many researchers, including Mohammed, a PhD student, struggle to search and navigate online content due to its sheer volume. However, after Mohammed found a way to handle the vast number of topics and their details, he developed software to summarise website content and create a content map by identifying the primary and secondary issues and their connections.

This has allowed Mohammed to:

- Quickly grasp key ideas
- Reduce the time required to search for the required titles and clearly understand how to access them
- Discover relevant topics
- Reducing effort and saving time, enabling faster and more accurate academic progress

Facilitating searches across vast amounts of electronic data has led to a radical change in learning and the performance of various experiments.

C. User Requirements

By implementing the system, we can gain a clear understanding of the site's content and its pages. This allows us to work smoothly and quickly and understand the required content from various perspectives.

- Clear titles for each topic will help them understand the issues more quickly.
- The ability to quickly navigate to sections related to a specific topic.
- Providing access to topics similar to the topic the user is searching for.

D. System Requirements

Through a set of steps, the system ensures that user needs are met with certainty.

- The system uses natural language processing (LDA) algorithms, enabling it to generate 3-7 topics for each page of content.
- The system uses a preprocessing pipeline to preprocess text (encoding, stopword removal, and linguistic analysis) before modelling.
- The system uses an Application Programming Interface (API), providing a programming interface for REST applications (/api/topics?page_id={id}) to serve topic data.

Retrieval Number:100.1/ijsepm.A902406010126
DOI:10.54105/ijsepm.A9024.06010126
Journal Website: www.ijsepm.latticescipub.com

Published By:
Lattice Science Publication (LSP)

- The system uses a database to store and retrieve topic data associated with page IDs.
- To avoid blocking user requests, system tasks are processed and executed asynchronously.

E. Design and Architecture

Through topic modelling, structured topic metadata is used on web page content to support end-user browsing and understanding. Topics and abstracts are generated via topic modelling of text using LDA algorithms. Data can be easily retrieved using the page ID where it is stored. The data is displayed on the front-end API server, which significantly facilitates caching and secure access. Topics and keywords are dynamically displayed on the front end.

F. Implementation

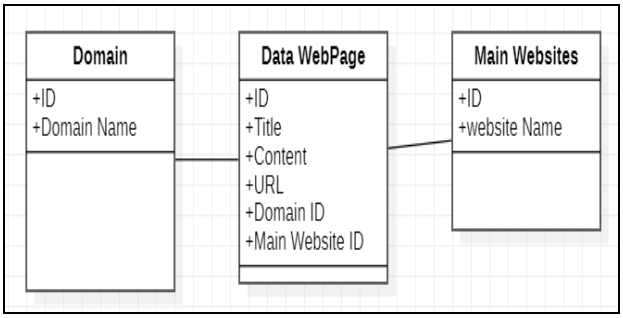
In this step, our system was implemented in Python. The system mentioned in section 2 includes frontend code, such as text input, and backend code, such as calling the Gemini API. The frontend will integrate with the backend topic modelling API to fetch and display topic data for each document.

G. Testing

The system [18] was tested on a large dataset to ensure accurate responses, assess its reliability and ability to handle APIs, and evaluate the effectiveness and capabilities of the front-end. Using extensive data, the system's technical capabilities were verified, and its scalability under load was demonstrated, enabling us to ensure that the software interfaces meet user requirements.

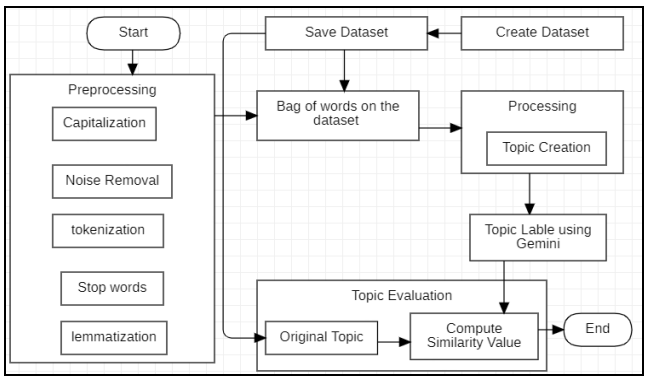
IV. UML DIAGRAMS

A. Class Diagrams: They represent a lower level of abstraction than component diagrams. Component diagrams are used to decompose a system, but they cannot be directly translated into code, unlike class diagrams, which can be translated into object-oriented languages. Designing a detailed or less detailed class diagram does not affect the diagram's meaning, as it describes the objects and their relationships to achieve the system's functionality. Translated into an object-oriented language, the diagram will reflect a class. design [19][20]. The Class Diagram of the proposed system shown in figure 2.



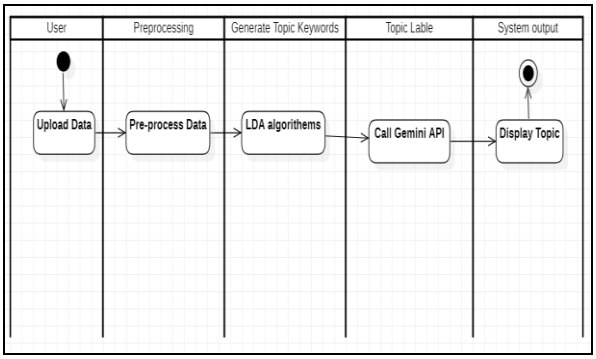
[Fig.2: Class Diagram for the Dataset]

B. Flowchart Diagram: A flowchart depicts the workflow for solving a specific problem. It can also be used to document and analyse problems, whether software-related or non-software. The flowchart is drawn using simple geometric shapes with agreed-upon meanings to explain process details and arrows to show relationships and the direction of process/data flow [19][21]. The Flowchart Diagram of the proposed system shown in figure 3.



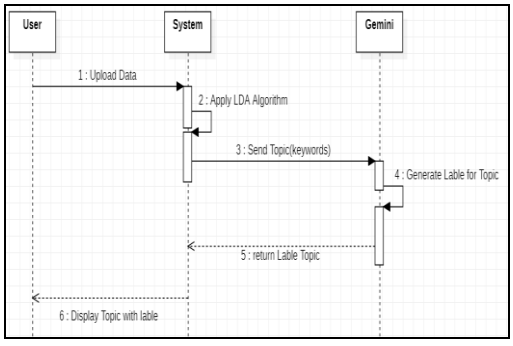
[Fig. 3: Flowchart Diagram of the Proposed System]

C. Activity Diagram: This diagram describes the outcomes of completed processes, provides deeper insight into those outcomes, and assesses the extent to which they align with the desired outcomes. Business management methods can be defined by describing the company's state. A careful look at Figure 4 below shows the workflow and data flow lines [22][23].



[Fig.4: Activity Diagram of the Proposed System]

D. Sequence Diagram: The diagram in Figure 5 shows how the system components interact with each other and with external components. By identifying these interactions, we can improve system performance by determining the system's operating time and requirements. This will allow us to conduct various tests to improve performance [21][23].



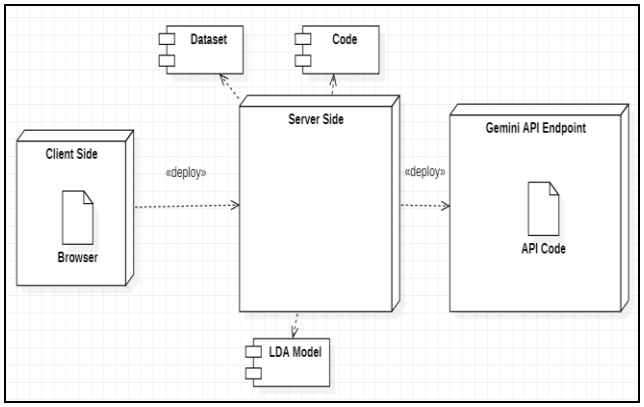
[Fig.5: Sequence Diagram of the Proposed System]

E. Deployment Diagram: Figure 6 illustrates the relationships among the system components, providing a bird's-eye view of the system and identifying the modelling approaches, programming methods, and

Retrieval Number:100.1/ijsepm.A902406010126
DOI:10.54105/ijsepm.A9024.06010126
Journal Website: www.ijsepm.latticescipub.com

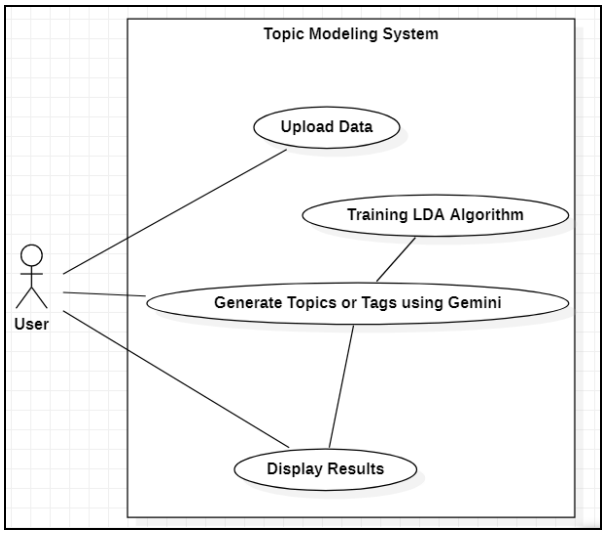
Published By:
Lattice Science Publication (LSP)

their interconnections [24][25].



[Fig. 6: Deployment Diagram of the Proposed System]

F. Use Case Diagram: Figure 7 identifies the system's interactors and provides a complete picture of its operation by simulating actual user behaviour. Through this simulation, we can understand the system's structure and internal and external relationships, determine the target audience and its suitability, and write a clear scenario for the system's workflow and interaction with potential users [20][24][25].



[Fig.7: Use Case Diagram for the Proposed System]

V. CONCLUSION

UML models provided a complete picture of the system's structure, the sequence of processes, the ways users interact with the system, and the documentation of its operation. This paper demonstrated empirically how software engineering principles are integrated into a system using Latent Dirichlet Allocation (LDA). Several engineering operations were performed on data from a group of websites, yielding measurable, verifiable results that support the system's reliability. The research paper demonstrated the potential for future discoveries in topic modelling to support the system and improve its accuracy and capabilities.

DECLARATION STATEMENT

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted objectively and without external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Author's Contributions:** The authorship of this article is contributed equally to all participating individuals

REFERENCES

1. G. Casale, C. Chesta, P. Deussen, E. Di Nitto, P. Gouvas, S. Koussouris, et al., "Current and future challenges of software engineering for services and applications," Procedia Computer Science, vol. 97, pp. 34–42, 2016. DOI: https://doi.org/10.1016/j.procs.2016.08.278
2. D. Akdur, V. Garousi, and O. Demirörs, "A survey on modelling and model-driven engineering practices in the embedded software industry," Journal of Systems Architecture, vol. 91, pp. 62–82, 2018. DOI: https://doi.org/10.1016/j.sysarc.2018.09.007
3. I. Sommerville, Software Engineering, 10th ed. Boston, MA: Pearson, 2016. ISBN 1-292-09613-6. https://www.pearson.com/en-us/subject-catalog/p/software-engineering/P200000003258/9780137503148?srsltid=AfmBOoqFRLIwoOrGUnrE0sKf7CyXooXVstFEIolQ4r2jK_yr_zCgfKqY
4. A. Tsertsvadze, Y. F. Chen, D. Moher, P. Sutcliffe, and N. McCarthy, "How to conduct systematic reviews more expeditiously?" Systematic Reviews, vol. 4, no. 1, p. 160, 2015. DOI: https://doi.org/10.1186/s13643-015-0147-7
5. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. "A brief survey of text mining: Classification, clustering and extraction techniques". arXiv preprint arXiv:1707.02919. 2017. DOI: https://doi.org/10.48550/arXiv.1707.02919
6. C. Treude and M. Wagner, "Predicting good configurations for GitHub and Stack Overflow topic models," in Proc. IEEE/ACM 16th Int. Conf. Mining Software Repositories (MSR), Montreal, QC, Canada, May 2019, pp. 84–95. DOI: https://doi.org/10.1109/MSR.2019.00022
7. X. Sun, X. Liu, B. Li, Y. Duan, H. Yang, and J. Hu, "Exploring topic models in software engineering data analysis: A survey," in Proc. 17th IEEE/ACIS Int. Conf. Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Shanghai, China, May 2016, pp. 357–362. DOI: https://doi.org/10.1109/SNPD.2016.7515925
8. T. H. Chen, S. W. Thomas, and A. E. Hassan, "A survey on the use of topic models when mining software repositories," Empirical Software Engineering, vol. 21, no. 5, pp. 1843–1919, 2016. DOI: https://doi.org/10.1007/s10664-015-9402-8
9. Kurbatova, Z., Veselov, I., Golubev, Y., & Bryksin, T. "Recommendation of move method refactoring using path-based representation of code". In Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops (pp. 315-322). 2020, June. DOI: https://doi.org/10.1145/3387940.3392191
10. Mehta, P., Aggarwal, S., & Tandon, A. "The Effect of Topic Modelling on Prediction of Criticality Levels of Software Vulnerabilities". Informatica, 47(6). 2023. DOI: https://doi.org/10.31449/inf.v47i6.3712
11. Zhang, M., Li, P., & Wang, W. "An index-based algorithm for fast online query processing of latent semantic analysis". Plos one, 12(5), e0177523. 2017. DOI: https://doi.org/10.1371/journal.pone.0177523
12. Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., & Zhao, L. "Latent Dirichlet allocation (LDA) and topic modelling: models, applications, a survey". Multimedia tools and applications, 78(11), 15169-15211. 2019.

DOI: https://doi.org/10.1007/s11042-018-6894-4
13. Feng, J., Zhang, Z., Ding, C., Rao, Y., Xie, H., & Wang, F. L. "Context reinforced neural Topic modelling over short texts". Information Sciences, 607, 79-91. 2022. DOI: https://doi.org/10.1016/j.ins.2022.05.098
14. A. Agrawal, W. Fu, and T. Menzies, "What is wrong with topic modelling? And how to fix it using search-based software engineering," Inf. Softw. Technol., vol. 98, pp. 74–88, 2018. DOI: https://doi.org/10.1016/j.infsof.2018.02.005
15. M. A. Mohammed, R. M. Mohammed, and H. A. Abbood, "Topic modelling for web page using LDA algorithm and web content mining," J. Educ. Pure Sci., vol. 15, no. 3, 2025. DOI: https://doi.org/10.32792/jeps.v15i3.686
16. M. A. Mohammed, H. A. Abbood, and R. M. Mohammed, "MRH: A large-scale text dataset for web content mining," J. Port Sci. Res., vol. 8, no. 4, pp. 321–326, 2025. DOI: https://doi.org/10.36371/port.2025.4.2
17. G. Team et al., "Gemini: A family of competent multimodal models," arXiv preprint arXiv:2312.11805, 2023. DOI: https://doi.org/10.48550/arXiv.2312.11805
18. N. M. Abbas, R. M. Mohammed, H. A. Abbood, and M. A. Mohammed, "Topic modelling for web page using LDA algorithm and web content mining: Testing and evaluation," Int. J. Comput. (IJC), vol. 55, no. 1, pp. 117–129, 2025. Available: https://ijcjournal.org/InternationalJournalOfComputer/article/view/2405
19. Bass, L., Clements, P., & Kazman, R. "Software architecture in practice (4th ed.)". Addison-Wesley Professional. 2021. ISBN 978-0136886020. https://www.oreilly.com/library/view/software-architecture-in/9780136885979/
20. B. Unhelkar, Software Engineering with UML. Boca Raton, FL: Auerbach Publications, 2017. DOI: https://doi.org/10.1201/9781351235181
21. A. Z. Umar, M. M. Gumel, and H. S. Tuge, "Comparing flowchart and swim lane activity diagram for aiding transitioning to object-oriented implementation," Amer. J. Educ. Technol., vol. 1, no. 2, pp. 99–106, 2022. DOI: https://doi.org/10.54536/ajet.v1i2.612
22. R. G. Tiwari, A. P. Srivastava, G. Bhardwaj, and V. Kumar, "Exploiting UML diagrams for test case generation: A review," in Proc. 2nd Int. Conf. Intelligent Engineering and Management (ICIEM), London, U.K., Apr. 2021, pp. 457–460. DOI: https://doi.org/10.1109/ICIEM51511.2021.9445383
23. A. Salim, J. C. A. Sujanto, F. W. Putra, L. Angelina, and C. O. Doaly, "UML modelling for web-based book lending library business systems," in AIP Conf. Proc., vol. 2680, no. 1, Dec. 2023. DOI: https://doi.org/10.1063/5.0176378
24. S. Sundaramoorthy, UML Diagramming: A Case Study Approach. Boca Raton, FL: Auerbach Publications, 2022. DOI: https://doi.org/10.1201/9781003287124
25. K. C. S. Murti, UML for Embedded Systems, Singapore: Springer, 2022, pp. 119–153. DOI: https://doi.org/10.1007/978-981-16-3293-8_5

AUTHOR'S PROFILE

Fitian Shafeeq Ajeel is a lecturer at the University of Information Technology and Communications, specializing in Software Engineering. He completed his Bachelor's and Master's degrees at the University of West London in the United Kingdom. He holds a CCNA certification from Cisco and has completed professional training in software and hardware maintenance in the UK. His academic and research interests focus on software development, integration of modern computing technologies in education, and geospatial data analysis. Fitian Shafeeq has published research on geospatial analysis using Google Earth Engine (GEE), reflecting his commitment to applying practical computing tools in research. He is actively involved in teaching Software Engineering, RStudio, and MongoDB courses and supervising undergraduate research projects. He is passionate about bridging theoretical knowledge with practical applications, helping students and researchers leverage modern technologies to develop innovative solutions in software and data analysis.

Hasan Aqeel Abbood received the Bachelor of Science degree in Information Systems Management in 2021. He currently works at the University of Information Technology and Communications (UoITC), where he contributes to the development and improvement of administrative, technical, and digital systems. His work focuses on enhancing information management processes, supporting digital transformation initiatives, and contributing to the development of effective software solutions for enterprise environments. Hassan is particularly interested in database design, software engineering, and integrating modern technologies to improve workflows within organisations. He continues to develop his skills through practical projects and academic activities within the university.

Raad Mahmood Mohammed received the B.Sc. Degree in Informatics System Management in 2021. He is currently working at the University of Information Technology and Communications (UoITC), where he contributes to the development and improvement of administrative, technical, and digital systems. His work focuses on enhancing information management processes, supporting digital transformation initiatives, and developing effective software solutions for enterprise environments. Raad is particularly interested in database design, software engineering, information security and integrating modern technologies to improve workflows within organisations. He continues to develop his skills through practical projects and academic activities within the university.

Dr. Mohammed Ali Mohammed, received the PhD Degree in computer science in 2023. He is an inventor at the Iraqi Centre of Innovation and Creativity. He authored a book on computer security. Also, he received a patent award in 2016, a gold medal from Egypt in 2017, and the Iraqi Science Day Award in 2023. Research interests focus on data science, Information Security, and Computer Science Applications. Currently working as a lecturer at the University of Information Technology and Communications (UoITC).